Answers to questions in Lab 1: Filtering operations

Name: Panteleimon Myriokefalitakis Program: TMAIM

Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

Good luck!

Question 1: Repeat this exercise with the coordinates p and q set to (5, 9), (9, 5), (17, 9), (17, 121), (5, 1) and (125, 1) respectively. What do you observe?

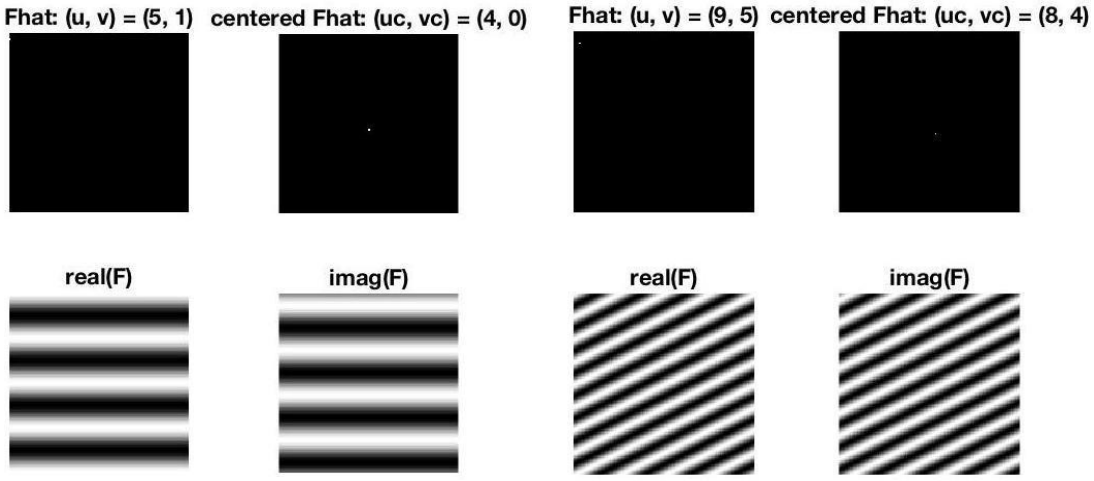
Answers:

Creation of a sine wave in the spatial domain.
In the case of (5,9) and (9,5): The frequency components of the x and y direction are interchanged. As we can see the amplitude and the wavelength is the same for both waves. The phase is rotated.
In the case of (17, 9) and (17, 121): The frequency components on the y direction are inversed. As we can see the amplitude and the wavelength is the same for both waves. The phase is rotated.
In the case of (5, 1) and (125, 1): Sine wave only on the y- direction. Makes sense since the y-frequency component is 0 (contains only the DC component). As we can see the amplitude and the wavelength is the same for both waves. The difference is the phase of the wave (the 4/-4 relation dictates $\Delta\varphi=\pi$)

Question 2: Explain how a position (p, q) in the Fourier domain will be projected as a sine wave in the spatial domain. Illustrate with a Matlab figure.

Answers:

A position (p, q) in the Fourier domain will give the frequency components for creating a sine wave in the x-y directions. If the point is along the horizontal or vertical axes this will give 0 frequency on this direction (DC component) and thus on the spatial domain will have a wave only on the x or the y direction. If the point is somewhere else this will cause a rotation on the wave and the frequency will be controlled by the frequency component of the Fourier domain.



Question 3: How large is the amplitude? Write down the expression derived from Equation (4) in the notes. Complement the code (variable amplitude) accordingly.

Answers:

The amplitude is given by the equation $|f| = \sqrt{Re(f)^2 + Im(f)^2}$ where Re(f) and Im(f) are real and imaginary parts respectively (In other words it's the norm of the function). Note that in the case of real number that will be the absolute value of it.

Question 4: How does the direction and length of the sine wave depend on p and q? Write down the explicit expression that can be found in the lecture notes. Complement the code (variable wavelength) accordingly.

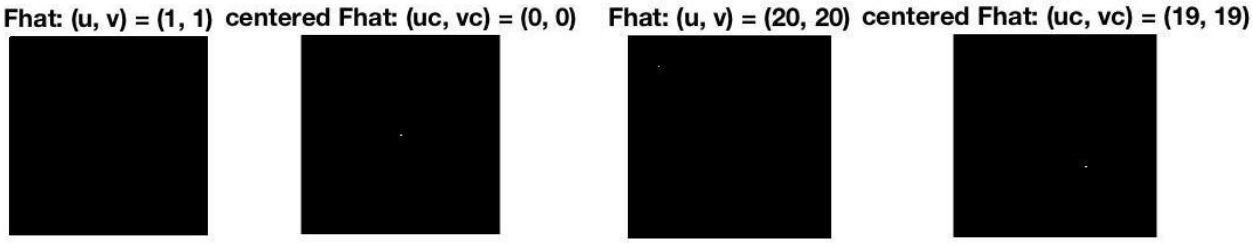
Answers:

Let $\omega=(\ \omega_1,\ \omega_2)^T$ where, ω_1 is the angular frequency in x direction and ω_2 is the angular frequency in y direction. Then, the phase of the sine wave is given by the equation $\varphi(\omega) = tan^{-1}(\frac{Im(\omega)}{Re(\omega)})$ and the wavelength by the equation $\lambda = \frac{2\pi}{||\omega||}$.

Question 5: What happens when we pass the point in the center and either p or q exceeds half the image size? Explain and illustrate graphically with Matlab!

Answers:

When we pass the point in the center and either p or q exceeds half the image size, the origin of the Fourier transform is moved to the center. Note that this is equivalent to multiplying the original function by $(-1)^{m+n}$, where m and n are image coordinates. This is commonly known as modulation. The result will be that the 1st quarter of the transform will move to the 4th. This can be seen as "capturing" the transform in different area by shifting the frequency rectangle which contains a full period to a new point. Note that happens because the 2-D Fourier transform (and its inverse) is infinitely periodic.



Question 6: What is the purpose of the instructions following the question *What is done by these instructions?* in the code?

Answers:

If the nonzero pixel is in the 1st quarter of the image, then the uc and vc are the same (-1 to go to the center). Otherwise, it also subtracts the length of horizontal and/or the vertical dimension from it. This happens because to do the centering operation this function uses the circshift function which shifts positions of elements circularly. From a more mathematical point of view, this centering operation will move the center by pi (or -pi). Note though that the Fourier and the spatial domain are continuous periodical and we only capture one full period. Thus, if we split the area in 4 quarters with alignment ABCD (clockwise) the centering operation (shift) will cause alignment DCBA. Thus, we need to take that into consideration when printing the new center.

Question 7: Why are these Fourier spectra concentrated to the borders of the images? Can you give a mathematical interpretation? Hint: think of the frequencies in the source image and consider the resulting image as a Fourier transform applied to a 2D function. It might be easier to analyze each dimension separately!

Answers:

The spectra are concentrated to the boarders of the images as the zero-point F(0,0) is in the top left corner. That's the reason that we use the "fftshift" function in Matlab i.e. to shift zero-frequency component to center of spectrum. As expected, the area around the origin of the transform contains the highest values. In the frequency domain the outcome appears as a dashed line. This makes perfect sense as the thickness of the line will be inversely proportional to the length of the object in the spatial domain and perpendicular to the direction of the biggest changes. Another way to see that is if we consider a 1-pixel slice of the image cut vertically to the axis. Then, we have 1-dimensional DFT of a square function in the spatial domain which yields a sinc function in the frequency domain. Note that if the thickness of the shapes H and G was not the same then, the zero crossings would be closer to the direction of the thickest one and the dashed lines would not be similar in that sense.

Question 8: Why is the logarithm function applied?

Answers:

We take the logarithm function (+ 1 to start from the 0 point) as the dynamic range of some intensities are overpowered by the DC component and in this way, we bring out the fine details. Note that this happens due to the frequency dependent exponential decay of the energy because of the multiple reflections etc. until the capturing moment.

Question 9: What conclusions can be drawn regarding linearity? From your observations can you derive a mathematical expression in the general case?

Answers:

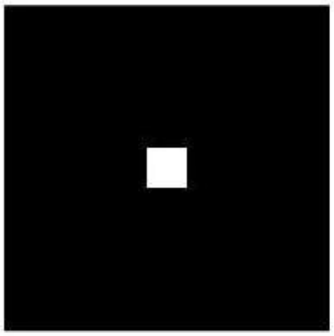
The Fourier transform respects the linearity of the original images in the sense of maintaining the linearity of operations in the spatial space to the frequency domain. In simpler words (ref: lecture slides 4) "You can add two functions (images) or rescale a function, either
before or after computing the Fourier transform." That is.

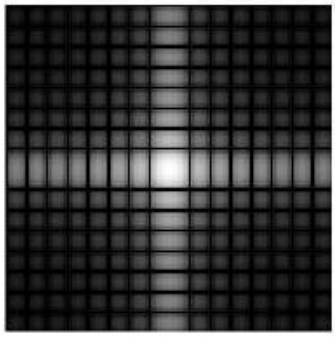
$$\mathcal{F}[a \cdot f_1 + b \cdot f_2] = a \cdot F_1 + b \cdot F_2$$

Question 10: Are there any other ways to compute the last image? Remember what multiplication in Fourier domain equals to in the spatial domain! Perform these alternative computations in practice.

Answers:

According to the Convolution Theorems; A convolution in the spatial domain is same as multiplication in the Fourier domain $f \circledast h = F * H$ and a convolution in the frequency domain is analogous to multiplication in the spatial domain, $f * h = F \circledast H$.
In our case we have an element-wise multiplication on the spatial domain and then we take the Fourier transform of it. This is analogous to applying the Fourier transform on f and g and then take their convolution.

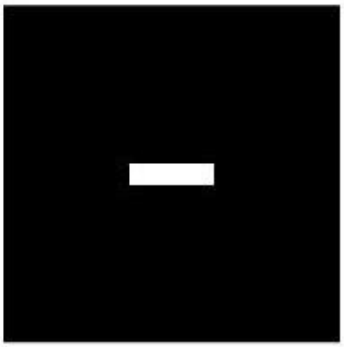


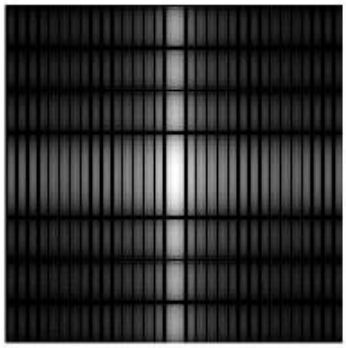


Question 11: What conclusions can be drawn from comparing the results with those in the previous exercise? See how the source images have changed and analyze the effects of scaling.

Answers:

In the previous example we saw the pattern analogous to the 2D sinc function. This was expected as if we cut a slice in each direction we will have a rectangular function which we know that will create a sinc "response" in the frequency domain. The zero crossings appeared symmetrical due to the symmetry in the spatial domain (in the x and y direction). In the second case though, the white shape is not symmetrical (in the x and y direction). This causes the side with the biggest length (x-axis) to appear on the frequency domain smaller distances between the zero crossings. In other words (quoting Lec 4) "compression in spatial domain is same as expansion in Fourier domain".





Question 12: What can be said about possible similarities and differences? Hint: think of the frequencies and how they are affected by the rotation.

Answers:

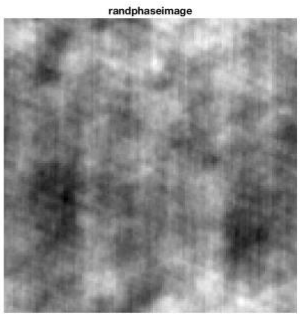
The rotation of the original images by some angle alpha will result in a rotation of the Fourier spectrum by the same angle. This is reasonable, since the shape is the same, but the orientation is different. Thus, the frequency components (of the differences) will be the same but the waves will "travel" in a different orientation. Nevertheless, when we examine the results we see some differences, like distortion or noise in the spectrum. This is due to the discretization of the pixels which result in anomalies in the borders of the shape which as a return will cause noise in the spectrum. As we can see, the bigger the anomalies the bigger the

introducing error. That's why in the 45º rotation the distortion is by far less than in the case of 30º or 60º.

Question 13: What information is contained in the phase and in the magnitude of the Fourier transform?

Answers:

When we replace the power spectrum with a non-linear version in the form of $|F(\omega)|^2 = \frac{1}{a+|\omega|^2}$ and keep the phase unchanged, the resulting images are like the original ones, but it seems like a smoothing filter has been applied on some areas with some spurious "clouds" floating around. Nevertheless, we can say that the "basic" information about the images is preserved. On the other hand, when we keep the power spectra unchanged, but we replace the phase by some random distribution, the resulting images have nothing to do with the original. As we can see, although we usually visualize the spectra of the transform, the phase contains crucial information about the image. This makes partially sense though, as we can do some basic inference and relate information from the spectra and the image in the spatial domain whereas, in the case on the visualization of the phase, no inference can be done what so ever.
To sum up, the phase visualization doesn't say much but it contains crucial information about the edges (sudden increase/drop in the amplitude). This can be seen if we reconstruct an image using only the phase. On the other hand, the visualization of the spectrum, gives information about the directions of the biggest changes and the size of them but if we use only the spectra to reconstruct the image the result will have nothing to do with the original one. In a more general sense it can be said that, the amplitude contains the one-wave information whereas the phase dictates how the delay on the superposition of them should be to reconstruct the image.



Question 14: Show the impulse response and variance for the above-mentioned t-values. What are the variances of your discretized Gaussian kernel for t = 0.1, 0.3, 1.0, 10.0 and 100.0?

Answers:

The variances of the discretized Gaussian kernel are:

For t = 0.1: $C = 0.0133 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

For t = 0.3: $C = 0.2811 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

For t = 1.0: $C = 1.0\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$

For t = 10.0: $C = 10.0\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$

For t = 100.0: $C = 100.0\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$

Note that the test C was in all cases $C_{test} = t\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$

Question 15: Are the results different from or similar to the estimated variance? How does the result correspond to the ideal continuous case? Lead: think of the relation between spatial and Fourier domains for different values of t.

Answers:

As a starting point we should note that when we generate a Gaussian kernel of the same size as the original image to be filtered, if the pixels of the image are even number then the Gaussian distribution cannot be sampled exactly symmetrically. Also, with our way of sampling, the sample is taken in the edges of the pixel and not in the center. This will have an impact on the estimated variance. Furthermore, as we discussed in Question 12, the non-continuity on the spatial domain due to pixel discretization (sampling) will introduce errors in the frequency domain. These errors, should be smaller as the "spreading" on the spatial domain is bigger i.e. bigger t, due to smoother shapes (according to the discretization). Also, the value of t will be inversely proportional to the 'spreading' on the frequency domain (as we saw before).

Question 16: Convolve a couple of images with Gaussian functions of different variances (like t = 1.0, 4.0, 16.0, 64.0 and 256.0) and present your results. What effects can you observe?

Answers:

When changing the magnitude of the variance, we observe more smoothing on the image with the similar pixels starting to form groups/clouds by suppressing the fine details of the image and enhance the fundamental structures of it. Note that this is the Scale-Space Theory framework, which is a multi-scale signal representation of an image as a one-parameter family of smoothed images, parametrized by the size of the smoothing kernel used for suppressing fine-scale structures. In our case, we use a Gaussian kernel and the size of it is controlled by the magnitude of the variance.

***Question 17**: What are the positive and negative effects for each type of filter? Describe what you observe and name the effects that you recognize. How do the results depend on the filter parameters? Illustrate with Matlab figure(s).
***Question 18**: What conclusions can you draw from comparing the results of the respective methods? Can you give a mathematical interpretation to explain the effects of each filter?

* I combined the answers to Questions 17 and 18 for giving a better inference.

Answers:

Ideal Low Pass Filter:
The Ideal LPF, "passes without attenuation all frequencies within a circle of radius from the origin and "cuts off" all frequencies outside this circle". It's "ideal" as it completely filters the frequencies outside the cutoff and completely let the wanted frequencies to pass. This is, in a sense, an ideal step function in the boundaries which cannot be exact when comes to physical systems. This filter could be useful only in case of having almost all information contained in low frequency area with sudden drop of information around the cutoff region. The Ideal LPF (as happens with all LPFs), has the net effect of image blurring. Another side-effect of LPFs, is the (strong in the case of ideal LPF) "ringing" which is a rippling artifact near sharp edges caused by the removal of the high frequency information. Generally, the Ideal LPF is not very practical and it is mostly used for pedagogical purposes.

Median Filter:
The Median filter is a special instance of rank filtering. It resembles the Mean filter but instead of computing the mean of the convolution, it ranks all the neighbors and afterwards it computes the outcome as its median. Thus, is a non-linear smoothing method, which is well suited for removing Shot noise from the images as is not affected by individual noise spikes. It has the advantage of not shifting the boundaries (shading) and the minimal degradation to edges allows its iteratively application to the image, which allows fine detail to be erased and large regions to take the approximately the same DN value. The main disadvantage of median filtering is that when a rectangular neighborhood is used (which is the most common case), it damages the thin lines and sharp corners and creates painting-like images. Nevertheless, the Median filter provides efficient denoising in images which containing binary noise but performs poorly when the noise is Gaussian or when the number of noise pixels in the boxcar is greater than half the number of pixels in it.

Gaussian Filter:
The Gaussian filter computes a weighted average of neighborhood pixel values through a convolution mask that its coefficients gradually decrease close to zero at the edge of the boxcar. This technique results minimization of the spurious oscillations (like a random emission noise e.g. Gaussian). The action of the Gaussian convolution is independent of the image content. The influence that a pixel has on another depends only on their distance and not on the actual image values. Thus, image edges are blurred because pixels across discontinuities are averaged together.

Salt & Pepper Noise:
Median LPF: With boxcar = 3, some noise is still present, but the image is not distorted. with boxcar = 5, the noise has been removed but the image distortion is present (blurring and the typical rectangle groups start to appear although in a small scale). Therefore, we used a window length of 5, but 3 is also suitable (depending on the application). Note that when we use median filtering only odd numbers are suitable for the boxcar. The Median LPF is commonly used for removing salt & pepper (speckle) noise due to its non-linearity and the less-dependency from the outliers as in the case of mean or gaussian filter.

Gaussian LPF: The salt and pepper noise should not be removed with a Gaussian filter (or any linear operator in general) since the result will be just a spreading of the noise in the neighbor pixels. Thus, when using a small variance for the filter, the blurring will be less, but the noise will be spread resulting "mountains" and "valleys" of intensity. On the other hand, with bigger variance in the LPF, the distortion due to the smoothing effects will be huge. Nevertheless, with t ~ 6 we got a fair balance of the aforementioned effects.
Ideal LPF: The salt and pepper noise should not be removed with an Ideal LPF as it will cut completely the high frequencies, causing the typical blurring and "ringing" effect which becomes finer in texture as the amount of high frequency content removed decreases. Nevertheless, with a cutoff of ~0.125 cycles per pixel, we got a fair balance of the aforementioned effects, although, the noise will mostly be present (even if it spreads across the image).

Gaussian Noise:
Gaussian LPF: With t ~ 3-4, we got pretty decent results in terms of side-effects to noise removal ratio (we wanted to preserve the edges). This makes sense though as we added white noise (Gaussian) and we used a similar filter type to remove it. We believe that we would have got even better results if we had used the provided function "discgaussfft" instead of ours for the reasons mentioned in the filter design section. As we increased the magnitude of t we noticed the typical blurring effects and the smoothing of the sharp edges although, the noise was almost absent in these cases.
Ideal LPF: When we applied the Ideal LPF to the image, with a cutoff of ~0.2 cycles per pixel, we got a fair balance of noise removal and ringing and blurring effect. When we lowered the cutoff, the blurring effects were increased along with the typical grouping in small areas.
Median LPF: This filter does not perform that well in the case of Gaussian noise due to its aforementioned properties. It's not that it will not remove the noise, but that the distortion of the image (painting-like effects) will not compensate for the noise removal. Nevertheless, with a boxcar of 5 pixels, we got decent results as we contaminate the side-effects and we removed some of the noise.

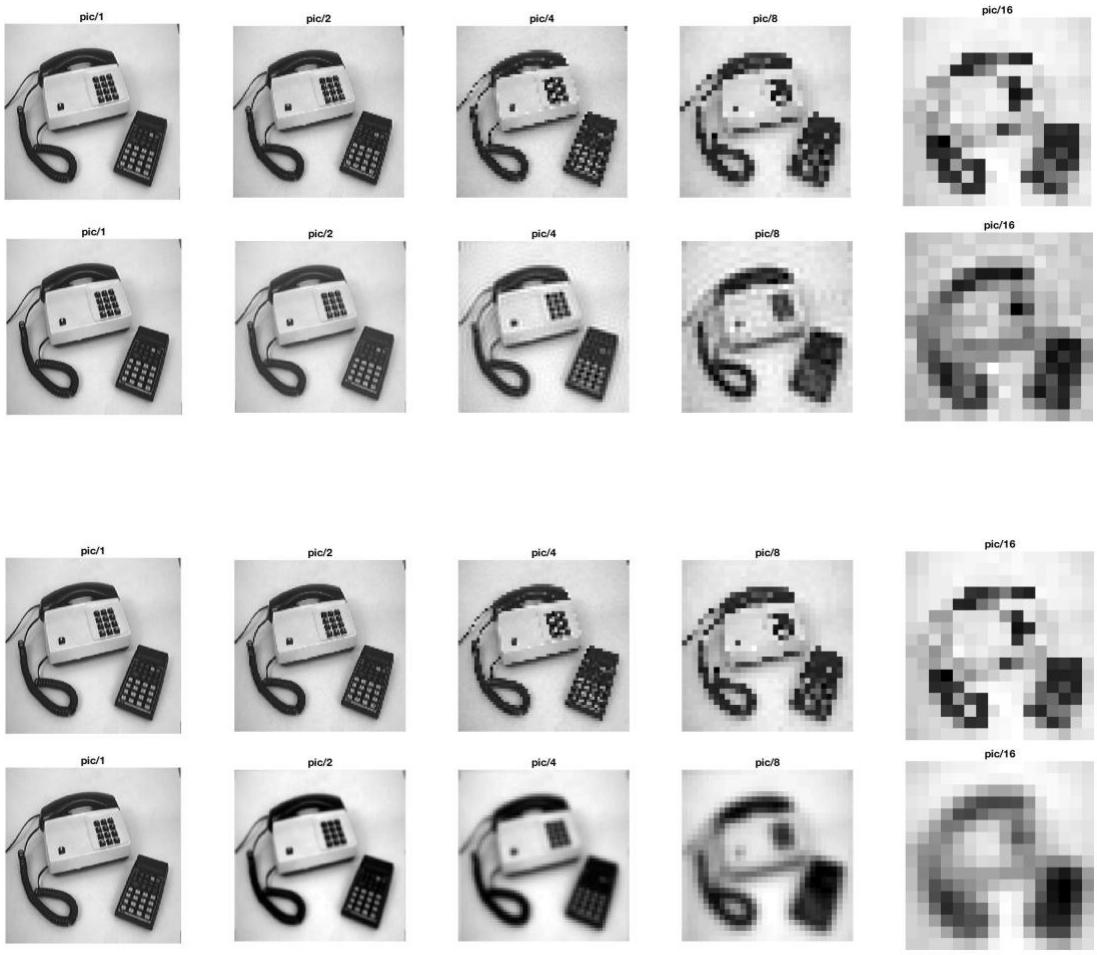
Question 19: What effects do you observe when subsampling the original image and the smoothed variants? Illustrate both filters with the best results found for iteration i = 4.

Answers:

When we observe the images after subsampling without preprocessing, the image distortion is present. The lines and the edges are appearing like steps and spurious DN values appears on pixels that shouldn't. These side-effects of subsampling are called aliasing and we give more information in the next question. When we preprocess the image before subsampling with a

LPF (Ideal and Gaussian) those effects are suppressed or eliminated but in return we get more blurring effects which result degradation of the fine details. We used a variance of t ~ 2 for the Gaussian LPF and a cutoff ~ 0.25 cycles per pixel for the Ideal LPF. With these values we mostly eliminated the aliasing effects and we keep a reasonable balance between smoothing and anti-aliasing.

Ideal LPF:



Question 20: What conclusions can you draw regarding the effects of smoothing when combined with subsampling? Hint: think in terms of frequencies and side effects.

Answers:

The aforementioned side-effect of subsampling is called “aliasing” which is, high frequency components of the original signal “masquerade” as lower frequencies in the sampled function. Aliasing is caused by the low sampling ratio. To be more accurate; Shannon rule, dictates that a band-limited function (signal which its highest frequency is bounded) can be completely recovered if and only if the sampling rate is exceeding the Nyquist rate, which equals twice the highest frequency content of the function. Therefore, when the sampling rate is less than the Nyquist rate, the signal is under-sampled, and aliasing occurs. In this part of the exercise, we sub-sampled a signal (image). This signal is not band-limited (as this is what happens in real data) and even if it was, we are sub-sampling it in a rate less than twice the highest frequency content of the image. Therefore, aliasing effects are present. When we use LPFs though, we suppress the higher frequency components and therefore, we lower the Nyquist

rate. Thus, we reduce the aliasing effects as now, we can sample at lower rates. Note that in the case of the Ideal LPF, we completely, eliminate the frequencies bigger than the cutoff and therefore we make the function band-limited.
