Answers to questions in Lab 1: Filtering operations

Name: ________Alice Karnsund________ Program: ___TMAIM-17__________

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:
(5, 1) and (125, 1) represents the points (4, 0) and (-4, 0) in the centered Fhat image respectively. Real parts, imaginary parts and phases that are each other's **transposes**, just like the points themselves are both before and after the centering operation. We see that the vc-value of the frequency is 0, thus giving rise to a wave in the **vertical** direction. Translation affects the phase but not the magnitude, thus the magnitude is the same and the phase is shifted. With 4 and -4 as uc parts the difference in phase is π.

(17, 9) and (17, 21) represents (16, 8) and (16, -8) in the centered Fhat image respectively. They are each other's reflections in the vc=0 plane. Which is to say that the real parts, imaginary parts and phases that are each other's **horizontal flips**. A rotation in one domain is a rotation in the other, thus the spatial image is rotated accordingly and the magnitude is kept the same since both points are the same distance from the center.

(5, 9) and (9, 5) represents (4, 8) and (8, 4) in the centered Fhat image respectively. In this case the uc and vc values are flipped. Real parts, imaginary parts and phases that are each other's **vertical flips**. This is another case of rotation, thus the image is rotated the same amount and the magnitude is kept the same.

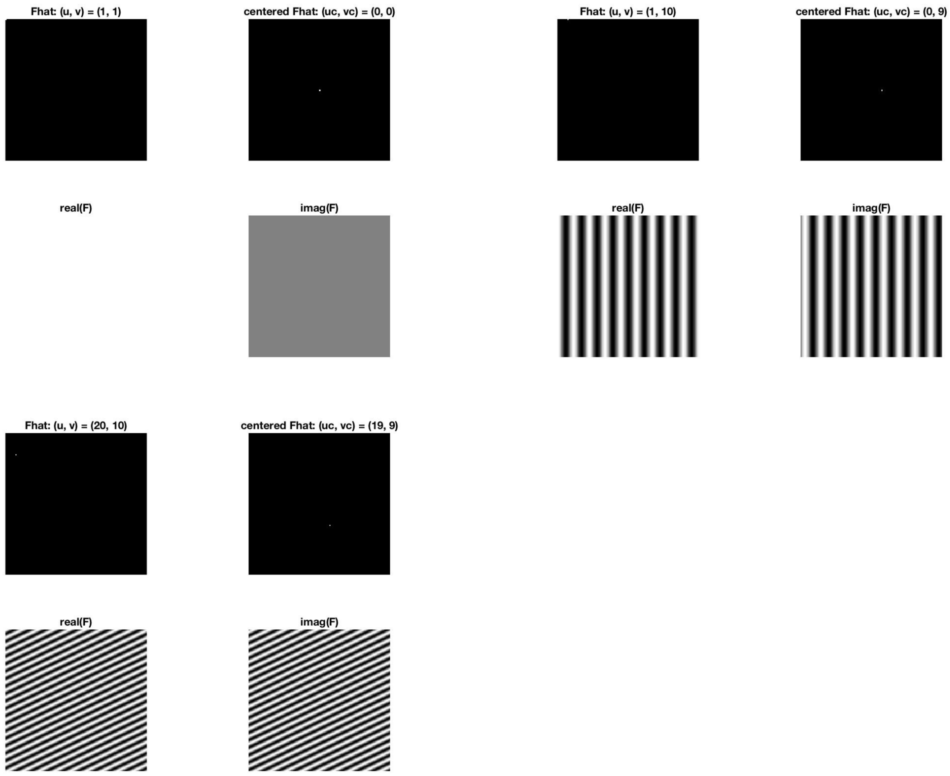
Question 2: Explain how a position (p, q) in the Fourier domain will be projected as a sine wave in the spatial domain. Illustrate with a Matlab figure.

Answers:
The discrete inverse Fourier transform expresses the image F as a sum of exponentials, i.e. a complex sum of cos and sin functions.

$$F(x,y) = \frac{1}{N}\sum_{u,v} \hat{F}(u,v)\ e^{i\frac{2\pi}{N}(ux+vy)}$$
$$= \frac{1}{N}\sum_{u,v} \hat{F}(u,v) \left(\cos\left(\frac{2\pi}{N}(ux+vy)\right) + i\sin\left(\frac{2\pi}{N}(ux+vy)\right)\right) \qquad (1)$$

In our case, there is only one non-zero $\hat{F}(u, v)$. Thus, the imaginary part of F corresponds to exactly one 2-dimensional sin function. The wavelength and orientation of this sin function depends on the specific location of the non-zero coordinates (u, v), or more specifically the centered coordinates (uc, vc). This is discussed further in question 4-6. The further away (uc, vc) is from the origin, the smaller wavelength. Since we are working in Matlab, the origin coordinates are simply (1,1), which become (0,0) after the centering operation. This generates a sin function with infinite wavelength, i.e. a constant function.

When the centered values (uc, vc) takes on values where uc=0 the real and imaginary parts of F will be waves travelling in the **horizontal** direction. But when vc=0 these will be waves travelling in the **vertical** direction. When uc=vc≠0 we get waves travelling in diagonal directions. This is simply due to the fact that the Fourier transform is a linear operation, a rotation in in one domain becomes a rotation in the other domain. The further we go from (uc,vc)=(0,0) the higher the frequencies. If (uc,vc)=(0,0) then the real and imaginary parts will simply show the means of F. Translations will only affect the phase as can be seen from comparing (uc,vc) with (-uc,-vc).



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:
From equation 1 above, we see that the amplitude of the imaginary sin function (and the real cos function) is $\frac{1}{N}|\hat{F}(u, v)|$. Since we construct $\hat{F}$ ourselves, we know that $|\hat{F}(u, v)| = 1$. Thus, the amplitude of the sin curve is $\frac{1}{N} = \frac{1}{128} \approx 0.0078$.

However, when studying the resulting $abs(F) = |F|$ matrix, the values are $\frac{1}{N^2} = \frac{1}{128^2} \approx 0.000061$. This is because in Matlab the fast Fourier transform FFT is implemented using a factor 1 (instead of $\frac{1}{N}$) and the inverse FFT using a factor $\frac{1}{N^2}$ (instead of $\frac{1}{N}$).

Further, $abs(F)$ is constant (shown as an all white plot because of the axis scaling). This is because F is simply one real cos curve and one imaginary sin curve of equal amplitudes and periods, and from trigonometric relations, we know that $|F(i,j)| = \sqrt{Re\big(F(i,j)\big)^2 + Im\big(F(i,j)\big)^2} = \sqrt{(\text{A}\cos\alpha)^2 + (\text{A}\sin\alpha)^2} = A\sqrt{\cos\alpha^2 + \sin\alpha^2} = A$.

(Note all the pixels are white in the amplitude image, since the ZMIN and ZMAX are used to shift the positions of black(0) and white(1). In our case ZMAX = 6.1035e-05 and ZMIN = -ZMAX).

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:
Again, note that (p, q) becomes (uc, vc) after the centering operation.

In the lecture notes, the wavelength is defined as $\lambda = \frac{2\pi}{\sqrt{\omega_x^2 + \omega_y^2}}$.
However, p and q are not continuous angular frequencies, but discrete frequencies, for which the relation $\omega_D = 2\pi\frac{u}{N}$ holds. Thus, the wavelength becomes $\lambda = \frac{N}{\sqrt{uc^2 + vc^2}}$. In other words, the further away (uc, vc) is from the origin, the smaller wavelength.

Similarly, the direction of the sin wave is defined by its phase $\phi(uc, vc)$, which in our case is defined as $\frac{2\pi}{N}(uc\ x + vc\ y)$, derived from equation 1 above.

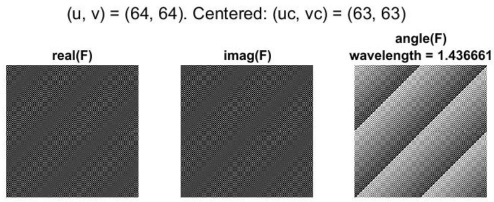
The set of points (x, y) that fulfills $uc\ x + vc\ y = 0$ all corresponds to $\sin 0 = 0$. Thus, the line $uc\ x + vc\ y = 0$, which can also be written as $y = -\frac{uc}{vc}x$, is perpendicular to the direction of the sin wave.

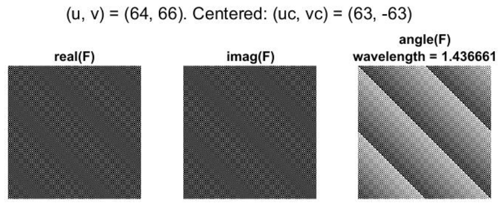
The line $y = \frac{vc}{uc}x$ is perpendicular to $y = -\frac{uc}{vc}x$, and is therefore exactly the direction of the sin wave. Thus, we can conclude that the positions (uc, vc) and $(-uc, -vc)$ result in the exact same sin wave (same wavelength and same direction).

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:
Since we have an even number of discrete x and y positions, there are four center positions: (64, 64), (64, 65), (65, 64) and (65, 65). After the centering operation, they become (63, 63), (63, -64), (-64, 63) and (-64,-64).

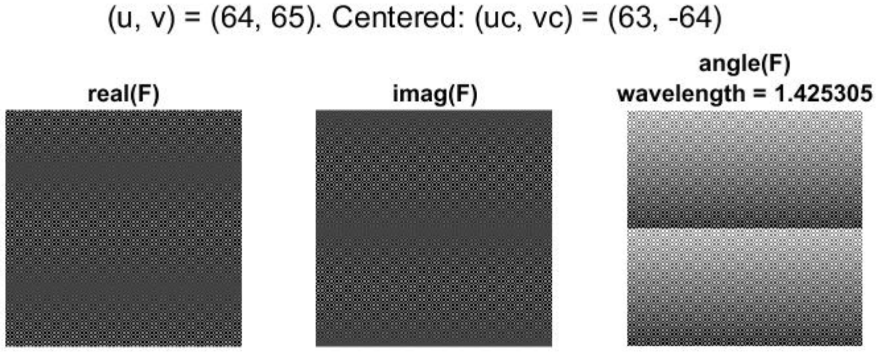
The first figures show that the positions (64, 64) and (64, 66) are each other's vertical flips.



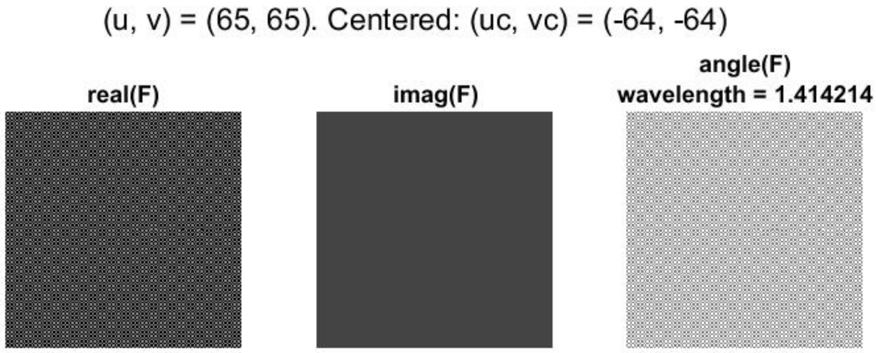


So, what happens in between them? The position (64, 65) result in horizontal waves, just like points on the uc-axis do. We return to the phase expression $\phi(uc, vc) = \frac{2\pi}{N}(uc\ x + vc\ y) = \frac{2\pi\ uc}{N}x + \frac{2\pi\ vc}{N}y$, and set $vc = -64 = -\frac{N}{2}$.

Then $\phi\left(uc, -\frac{N}{2}\right) = \frac{2\pi\ uc}{N}x - \pi y$, which is simply the argument of the cos and sin functions. Since they have a period of 2π, changing y one step up or down only results in a change of sign of the cos and sin functions, and thus does not affect $abs(F)$ at all.



Studying the case of $uc = vc = -\frac{N}{2}$. Then $\phi\left(-\frac{N}{2}, -\frac{N}{2}\right) = \pi x - \pi y$. Regardless of the values of x and y, $\sin(\pi(x - y)) = 0$, as shown in the homogeneous imag(F) below. $\cos(\pi x - \pi y)$ can be either 1 or -1.



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

Answers:
These lines of code perform a centering operation. Instead of having $u, v \in [0, \dots, N-1]$, we now have $uc, vc \in [-\frac{N}{2}, \dots, \frac{N}{2} - 1]$. This corresponds to changing the definition domain of the

frequency variable from $[0, 2\pi)$ to $[-\pi, \pi)$. To display this change, we simply need to shift the first and third quadrants and the second and fourth quadrants, respectively.

Please note that this also includes a transformation from Matlab indices to standard coordinates, i.e. the point (1,1) is in fact the origin (0,0). This means all coordinates are subtracted with 1, regardless of being shifted in the centering operation or not.

From these centered discrete variables uc, vc, interesting properties of the projected sin and cos waves can be concluded, as discussed in previous questions.

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 images are constructed from box-functions in two-dimensions, i.e for the image F we can write:

$$F(x,y) = \begin{cases} 1, \; x_1 \leq x \leq x_2 \\ 0, otherwise \end{cases}$$

x is pointing downwards due to tilted coordinate system.

Again, note that that the showgrey function has an inverted x-axis relative to the standard Matlab plot function.

To understand the behavior in the Fourier domain we calculate the Fourier transform of this function (image):

$$\mathcal{F}(F(x,y)) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} F(x,y)e^{-2\pi i\left(\frac{xu+yv}{N}\right)} = \frac{1}{N}\left(\sum_{x=x_1}^{x_2} e^{-\frac{2\pi ixu}{N}} \sum_{y=0}^{N-1} e^{-\frac{2\pi iyv}{N}}\right) =$$

$$= \{BETA\; p326\} = d_0(v)\frac{1}{\sqrt{N}}\sum_{x=x_1}^{x_2} e^{-\frac{2\pi ixu}{N}}$$

Here $d_k(v) = \begin{cases} 1, v = k + mN, m \in \mathbb{Z} \\ 0, otherwise \end{cases}$, and since we are only considering one period we get $d_0(v) = \delta(v)$. Thus,

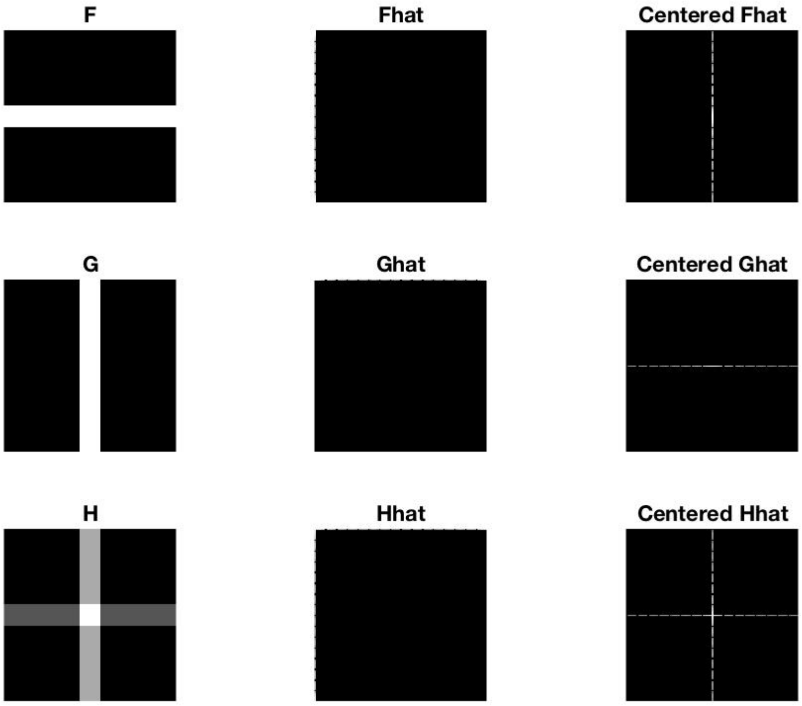
$$\mathcal{F}\big(F(x,y)\big) = \delta(v)\frac{1}{\sqrt{N}}\sum_{x=x_1}^{x_2} e^{-\frac{2\pi ixu}{N}}$$

Since $\delta(v) = 1$ only when $v = 0$, we get a spectrum that is concentrated to the left boarder. Similarly, for the second image G, the spectrum is concentrated to the upper boarder, where $u = 0$.

$$\mathcal{F}\big(G(x,y)\big) = \delta(u)\frac{1}{\sqrt{N}}\sum_{y=y_1}^{y_2} e^{-i\frac{2\pi}{N}vy}$$

Lastly, the image H is a linear combination of F and G. Since the Fourier transform fulfills homogeneity and additivity, $\hat{H}$ is concentrated at both the left and upper boarder.

$$\hat{H}(u,v) = \mathcal{F}\big(F(x,y) + 2\,G(u,v)\big) = \hat{F}(u,v) + 2\,\hat{G}(u,v)$$



Question 8: Why is the logarithm function applied?

Answers:
The dynamic range of the intensities can be compressed by replacing each pixel value with its logarithm. This has the effect that low intensity pixel values are enhanced relative high pixel values. In other words, it is an image enhancement method, specifically a grey-level transformation.

Since our images can have 0 as a pixel value, and $\log(0)$ is not defined, we add 1 to the Fourier spectrum before applying the logarithm.

The number 1 is appropriate since it guarantees that we will always end up with non-negative values $\log\left(1 + |\hat{F}|\right) \geq 0$ for non-negative pixel values. Specifically, an input pixel value of 0 will be mapped to an output pixel value of 0. Thus, the property $T(r_{min}) = r_{min}$ is fulfilled.

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

Answers:
As discussed in question 7, the image H is a linear combination of F and G and the Fourier spectra $\hat{H}$ is a linear combination of $\hat{F}$ and $\hat{G}$.

$$\hat{H}(u, v) = \mathcal{F}\big(F(x, y) + 2\, G(u, v)\big) = \hat{F}(u, v) + 2\, \hat{G}(u, v)$$

This is a consequence of the Fourier transform and its inverse fulfilling homogeneity and additivity, which generally can be written:

$$\mathcal{F}[af_1(m,n) + bf_2(m,n)] = a\widehat{f_1}(u,v) + b\widehat{f_2}(u,v)$$

$$af_1(m,n) + bf_2(m,n) = \mathcal{F}^{-1}\left[a\widehat{f_1}(u,v) + b\widehat{f_2}(u,v)\right]$$

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:
(Lecture 4) Multiplication in the spatial domain is the same as convolution in the Fourier domain, $\mathcal{F}(hf) = \mathcal{F}(h) * \mathcal{F}(f)$. And also (Lecture 3), a convolution in the spatial domain is the same as multiplication in the Fourier (frequency) domain. Thus in this case the last picture $\mathcal{F}(FG)$ can be computed as $\mathcal{F}(F) * \mathcal{F}(G)$.
In this case FG is a box-function which will be represented by a sinc function in the Fourier domain. The sinc function has an infinite number of side lobes and can thus be seen as an ideal low-pass filter.

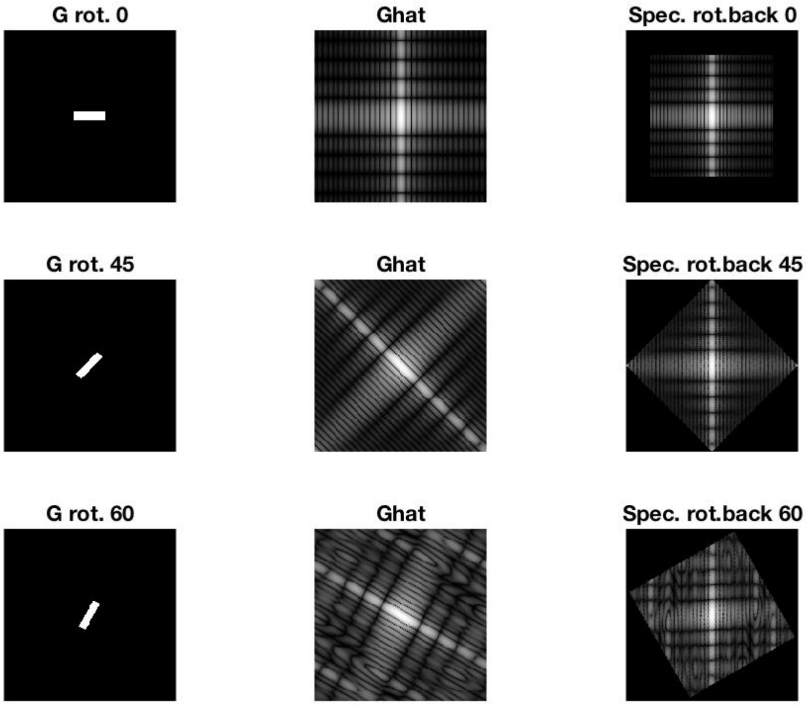
Question 11: What conclusions can be drawn from comparing the results with those in the previous exercise? See how the source images have changed and analyze the effects of scaling.

Answers:
(Lecture 4) Compression in the spatial domain is the same as expansion in the Fourier domain and vise versa. In our case the Figure of FG from the previous exercise has been compressed vertically and expanded horizontally. More exactly, FG is a white square in the middle of size 16x16 and F in this exercise is a rectangle with height 8 and width 32, 8x32, i.e. $\frac{1}{2}height(FG) \times 2width(FG)$. It is clear from the Fourier spectra of F and FG that a compression in the horizontal direction has taken place, as well as an expansion in the vertical direction. Which corresponds to the theory. In general we can express this as (Szeliski p 135),

$$f(ax, by) \leftrightarrow \frac{1}{ab}F(\frac{u}{a}, \frac{v}{b})$$

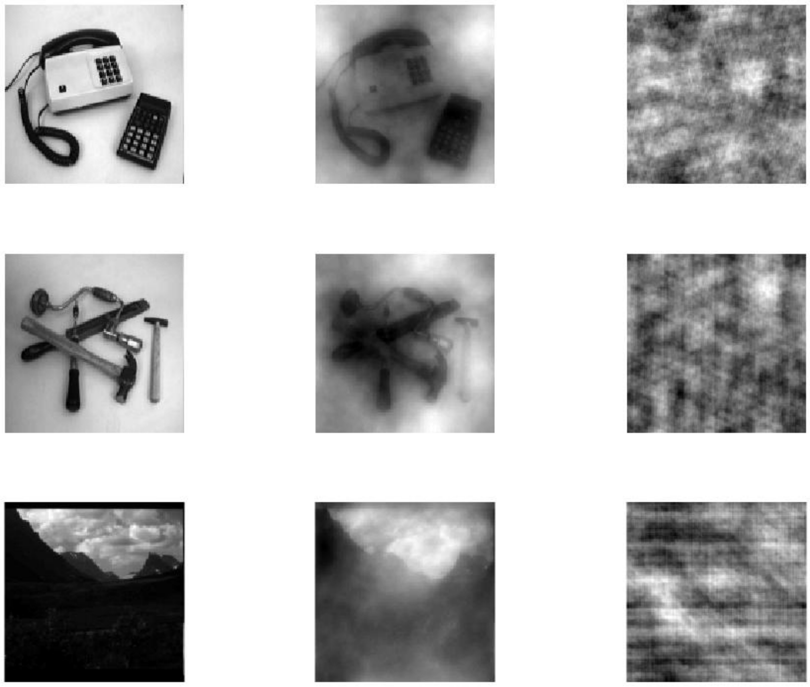
Question 12: What can be said about possible similarities and differences? Hint: think of the frequencies and how they are affected by the rotation.

Answers:
(Lecture 4) A rotation in one domain becomes a rotation in the other domain. Which is clearly seen from the pictures. This rotation causes the waves to travel in different directions, but the magnitude and phase are kept the same. The Fourier spectra rotates the same amount and in the same direction as the picture. But if we investigate the pictures more closely, we see that the rotated pictures are disturbed, i.e the edges are not smooth. This has to do with our fix resolution and the orientation of the pixels. In the picture below, this is especially highlighted for the image rotated 60 degrees. This picture also has the most disturbed Fourier spectra, thus a disturbed picture has a direct effect on the spectra. This distortion may be seen as noise.



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

Answers:
The function pow2image, takes the image and computes its modified transform and then take the inverse of this to output a modified picture.
When using a power spectrum of the form $\left|\widehat{f(\omega)}\right|^2 = \frac{1}{a+|\omega|^2}$, $(a \approx 10^{-10})$, we see that in the figure below, the pictures we obtain after applying the inverse (column 2) are kind of dimmed compare to the original ones (column 1). But we can still distinguish the main characteristics such as dark and bright areas and corners, thus the intuition of what the images are showing are kept. This has to do with the fact that in this case we still have the same phase as in the original pictures.
If we instead manipulate the phase of the Fourier transform, by using a random distribution, and keep the magnitude, we get what is shown in column three below. In this case we have no intuition of what the images are showing. Thus from this we can draw the conclusion that the phase holds a lot more information than the power spectra.

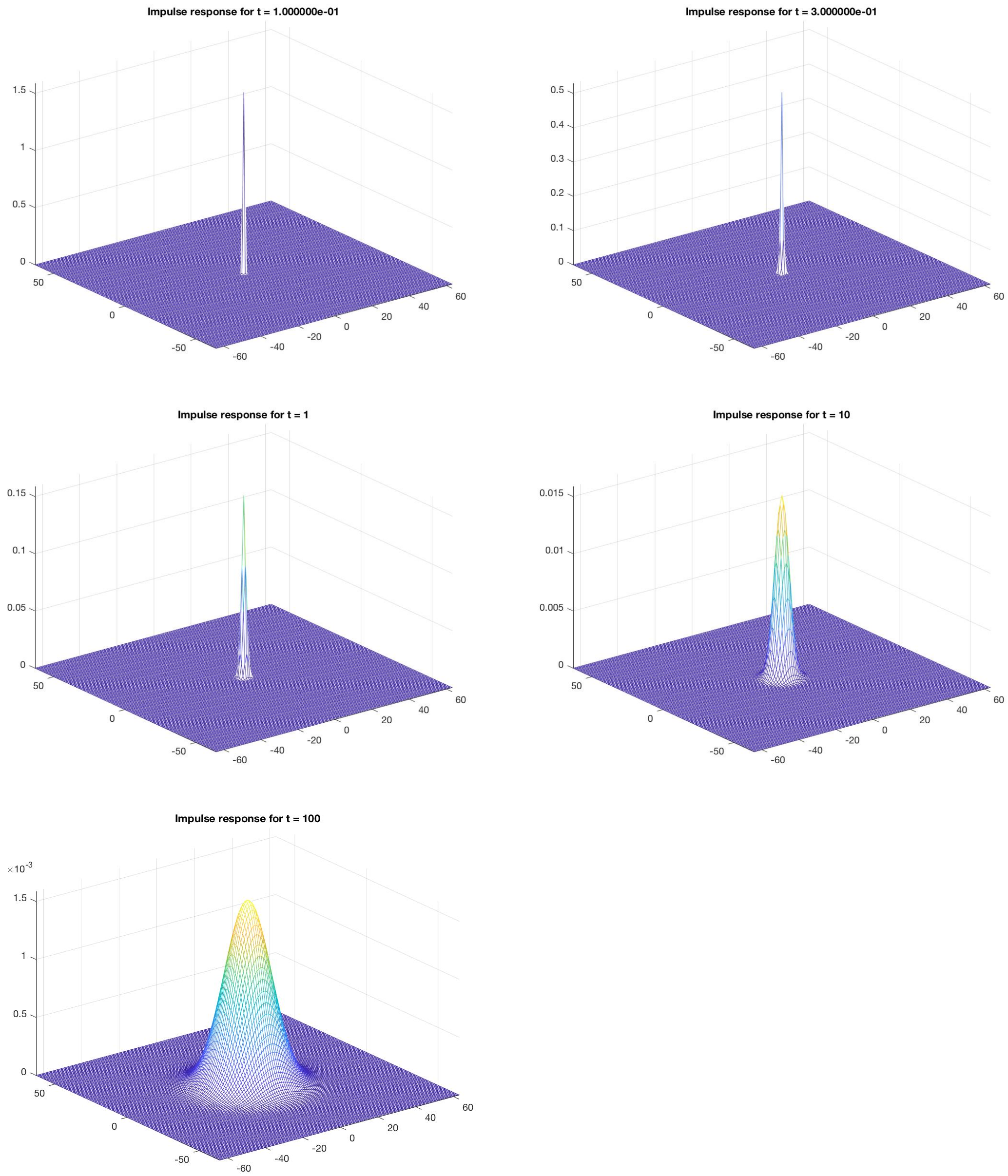


Question 14: Show the impulse response and variance for the above-mentioned t-values. What are the variances of your discretized Gaussian kernel for t = 0.1, 0.3, 1.0, 10.0 and 100.0?

Answers:
Below are the five different impulse responses for the different t values. Their corresponding Covariance matrices C are,

$t = 0.1$: $\quad C = 0.0133\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$t = 0.3$: $\quad C = 0.2811\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$t = 1.0$: $\quad C = 1.0\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$t = 10.0$: $\quad C = 10.0\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$t = 100.0$: $\quad C = 100.0\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Thus our responses are almost as in the ideal continuous case, $C = t\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

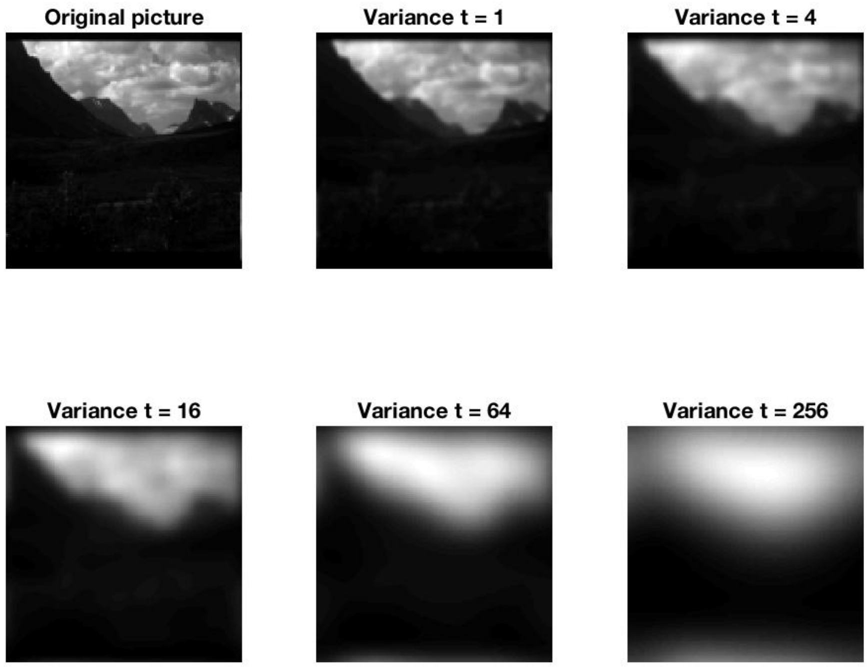


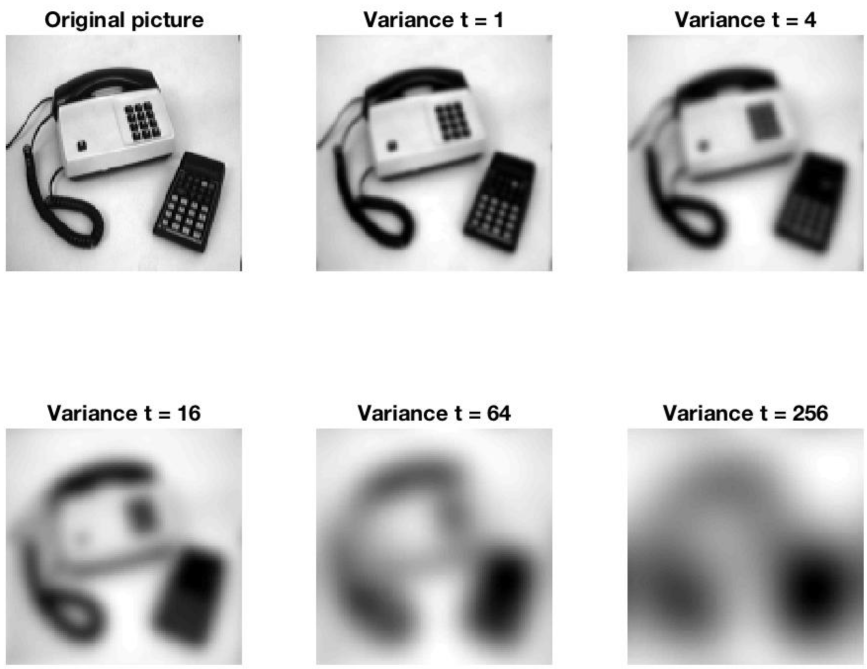
Question 15: Are the results different from or similar to the estimated variance? How does the result correspond to the ideal continuous case? Lead: think of the relation between spatial and Fourier domains for different values of t.

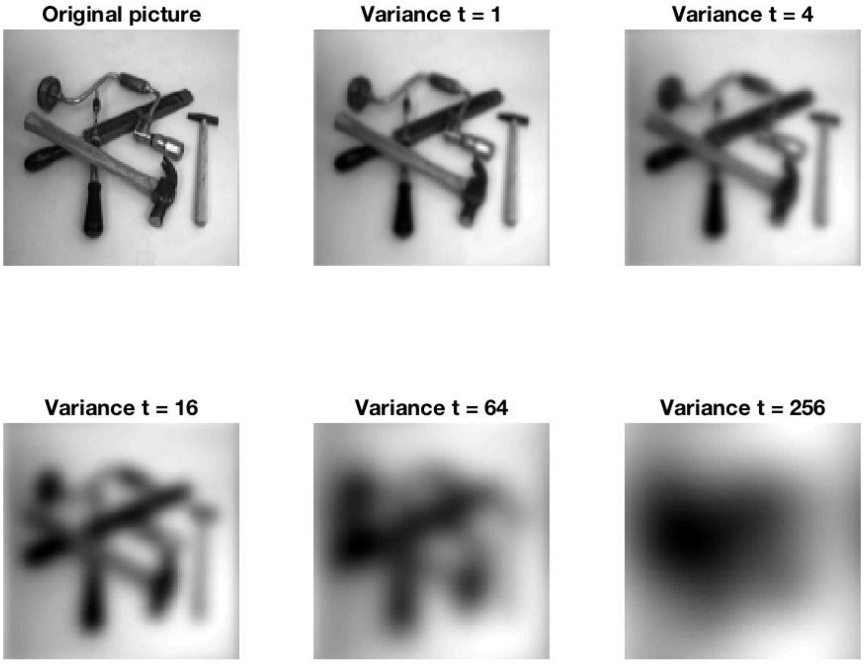
Answers:
In the first two cases the results are different from the estimated variance. But as t takes on higher values we reach values of the variance that corresponds to the ideal continuous case. Why we do not reach this when t<1 has to do with the discretization of the pixels. Thus when the t is very small it is difficult to model the distribution in the discrete case since we only have a fixed set of pixels. This discretization will thus introduce errors in the Fourier domain, that will decrease as the variance increases, as is also clear from the figure above.

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:
For this part, the same three pictures as in Question 13 are used. The convolution for each picture with the Gaussian kernel for different variances are shown below. As the variance increases the pictures get more and more blurred. This makes sense, since the wider the variance the more higher frequencies are allowed and thus the picture holds more noise. Details, like edges, that lives in the middle frequencies will not get as much attention and the weights between the frequencies will be more spread out.


Original picture
Variance t = 1
Variance t = 4
Variance t = 16
Variance t = 64
Variance t = 256


Original picture
Variance t = 1
Variance t = 4
Variance t = 16
Variance t = 64
Variance t = 256



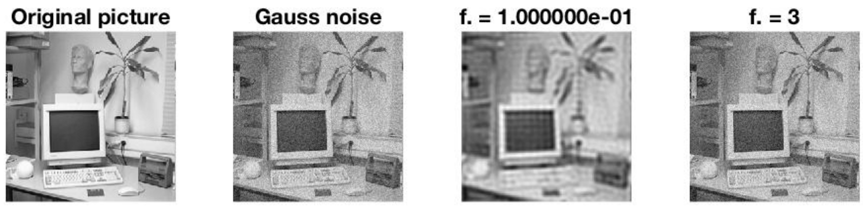
Question 17: What are the positive and negative effects for each type of filter? Describe what you observe and name the effects that you recognize. How do the results depend on the filter parameters? Illustrate with Matlab figure(s).

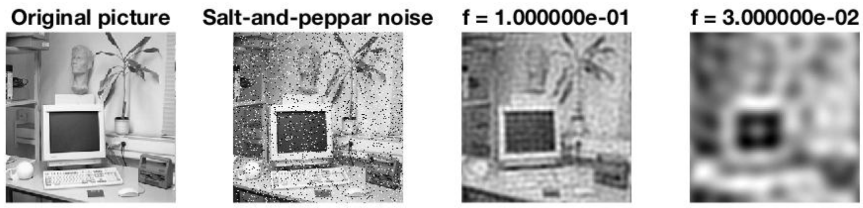
Answers:
Below a 3 sets of pictures, each containing 8 figures, showing the results of: Ideal low-pass filtering, Gaussian smoothing and Median filtering, respectively. In each set the first row represents the “add” figure, which is the original figure with added gaussian noise, and the second row represents the “sap” figure.

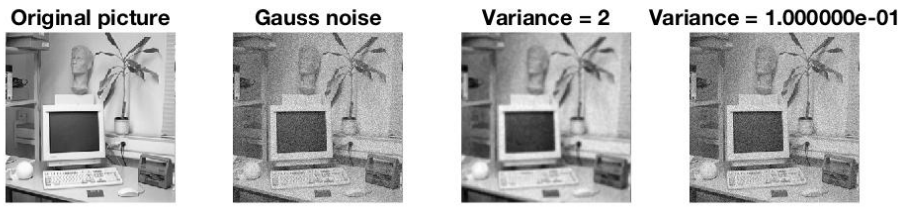
Ideal low-pass filtering, shown in the first set with eight figures below, filters the disturbed images, add and sap, with an ideal low-pass filter with a given cut-off frequency (cycles per pixel). In this case we noted that a good value of the cut-off frequency has a very short range, somewhere around 0.1. Greater values will accept more details in the picture and will not affect the original picture much, as can be seen in the figure with f = 3. But if we instead let the cut-off frequency, take on lower values we get a more blurred picture, as the one with f = 0.03. Thus a high cut frequency gives more details, which includes noise, and a low cut-off frequency gives a blurred image. However, we see that both the figures with f = 0.1 are far from optimal. This type of filter seem to introduce a new noise, that has nothing to do with either the gaussian noise or the salt-and-pepper noise. The impulse response on an ideal low-pass filter is the sinc function, and this results in ringing artifacts via the Gibbs phenomenon. These artifacts appear as spurious signals near sharp transitions in a signal (Wikipedia). This explains why the resulting images after filtering appears to have some kind of hidden grid. This is clearly seen in the figures with f = 0.1. The amount of artifacts seems to increase with a decreasing cut-off frequency, seen when f = 0.03.

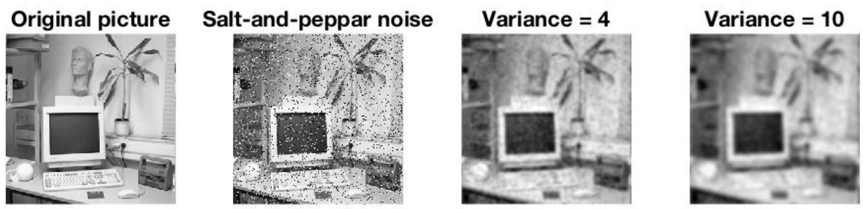
Gaussian smoothing, shown in the second set with eight figures below, definitely does a better job than the first filter. The best results in our case, where with variance 2 for the “add” picture and variance 4 for the “sap” figure. In these 2 figures we can see that the noise has been smoothen out, and the greater the variance gets the smoother the picture gets, as seen when the variance is set to 10. This is expected since the Gaussian filter reduces high-frequencies, and is thus a low pass filter. When this filter is applied, each pixel's new value is set to the weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receives smaller weights as their distance to the original pixel increases. This results in a blur that preserves boundaries and edges better that other, such as the other two discussed here (Wikipedia). A low value of the variance means that the neighborhood size will be smaller and thus less smoothing, as in the case with variance = 0.1. But with greater variance, the bigger the neighborhood and thus the greater smoothing, as when variance =10.

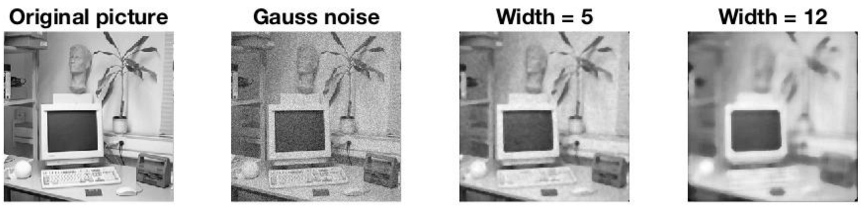
Median filtering, shown in the third set with eight figures below, does the best job (especially for the "sap" case). The main idea of the median filter is to run through the signal (image) entry by entry, replacing each entry with the median of the neighboring entries. The pattern of neighbors is called the "window", which slides entry by entry, over the entire figure. Just as for the previous case, if the neighborhood is too small, there will not be much of a smoothing effect. And if it is two great the image will be very smooth but with sharp edges where the pixel values change abruptly. The strength of using the median instead of mean is that the effect of outliers, i.e. noise, will directly be ignored. This is clearly seen in the case of the "sap" picture. However, this filtering is still not perfect. The resulting images gives an intuition of paintings, which increase with the window size. A reason for this could be that the resulting image lack the amount of variation in pixel values as the original picture has.

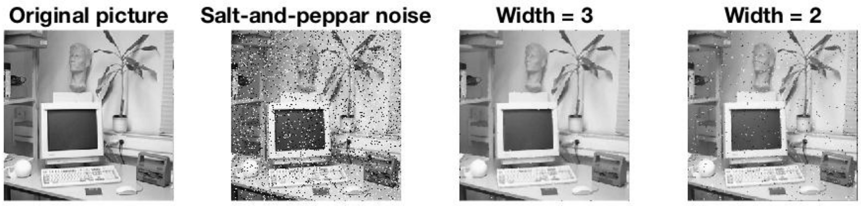

Original picture
Gauss noise
f. = 1.000000e-01
f. = 3


Original picture
Salt-and-peppar noise
f = 1.000000e-01
f = 3.000000e-02


Original picture
Gauss noise
Variance = 2
Variance = 1.000000e-01


Original picture
Salt-and-peppar noise
Variance = 4
Variance = 10



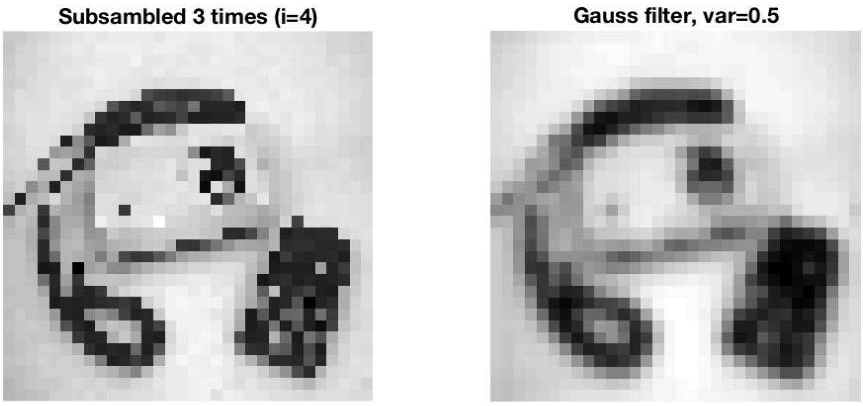


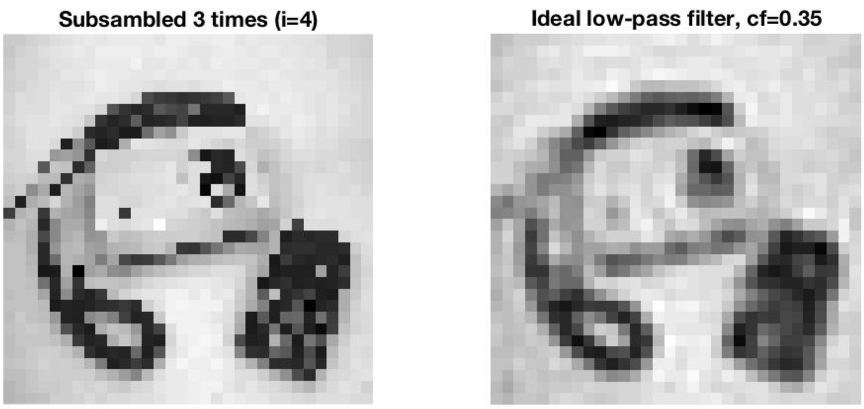
Question 18: What conclusions can you draw from comparing the results of the respective methods?

Answers:
Answered in Question 17.

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 subsample, we pick out every other pixel in each dimension in the original picture. This results in a loss of information and a different neighborhood in the subsampled picture. This is clearly shown in the pictures below, only coarse structures can be seen. Since both of these are lowpass filters, they will reduce high frequencies, even more information will get lost. Also, which can be more clearly seen when looking at the plots for i=3, the smoothing operation has introduced an common type of error, called spatial aliasing (Lecture 4). This type of error occurs if the picture is under-sampler or if the reconstruction is poor (discussed further in the next question). This often is seen waves in the smoothed picture. But this is not present if we prefilter the signal.





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:

First of all a subsampling decreases the information content, the process will thus get rid of details. Details are found in the higher frequencies, and thus together with a low-pass filter these will be even more discarded. As mentioned above, aliasing is a common error when subsampling a signal. This occurs if the signal is under-sampled.
(Lecture 4) A signal is band limited if its highest frequency is bounded. This frequency is called the bandwidth. The sin/cosine component of the highest frequency determines the highest “frequency content” of the signal. If the signal is sampled at a rate equal or greater than twice its highest frequency, the original signal can be completely recovered from it samples (Shannon). The minimum sampling rate for band limited function is called Nyquist rate. In our case, the signal is sampled at a lower rate than twice its highest frequency, thus aliasing is present. But this can be solved by sampling at a higher rate or prefiltering.
