Answers to questions in Lab 1: Filtering operations

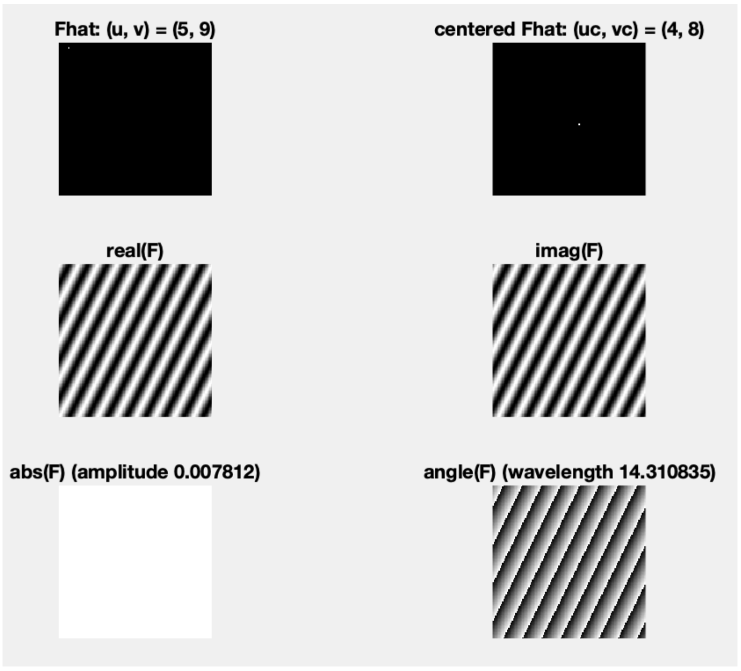
Name: Henrik Holm **Program:** CINEK4-DKOI / TIEMM1-MAIG

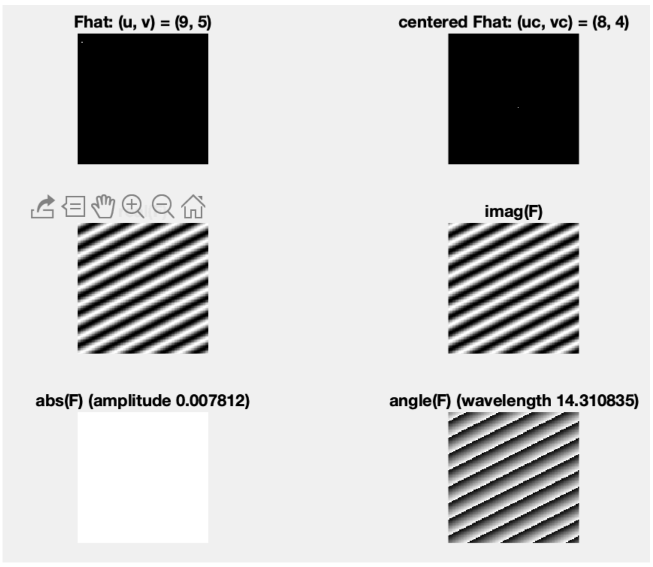
Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

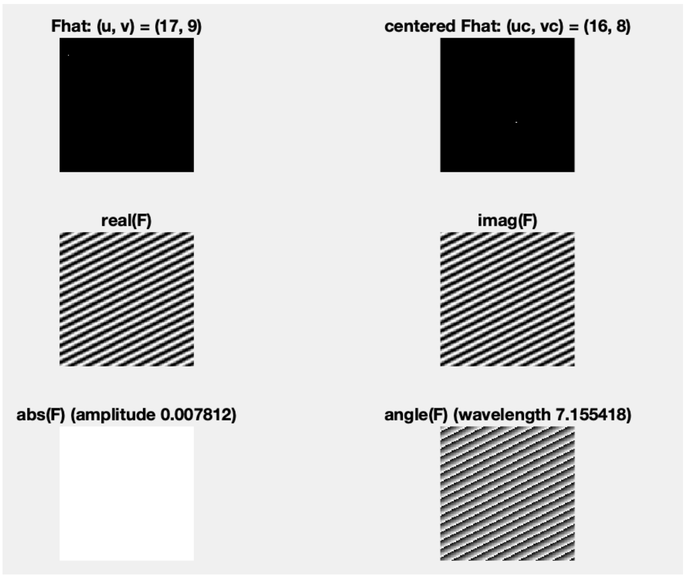
Good luck!

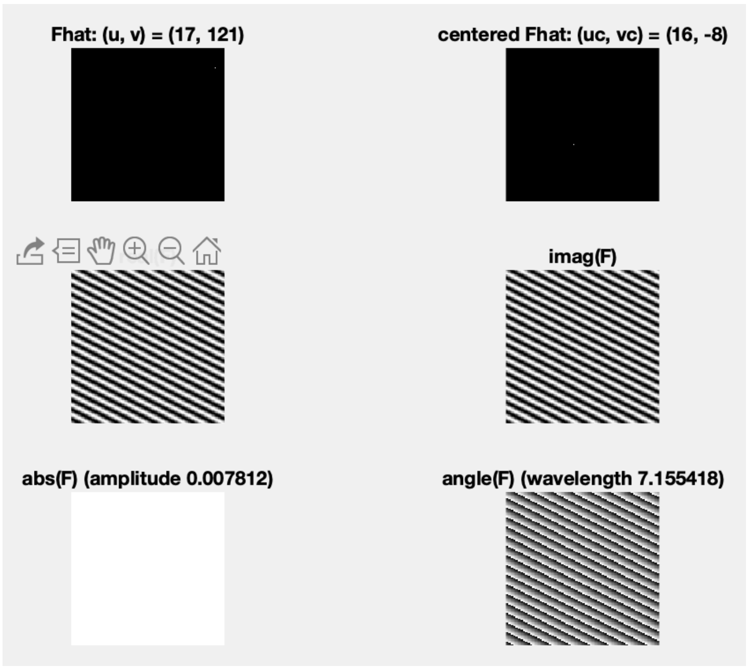
Question 1: Repeat this exercise with the coordinates p and q set to (5, 9), (9, 5), (17, 9), (17, 121), (5, 1) and (125, 1) respectively. What do you observe?

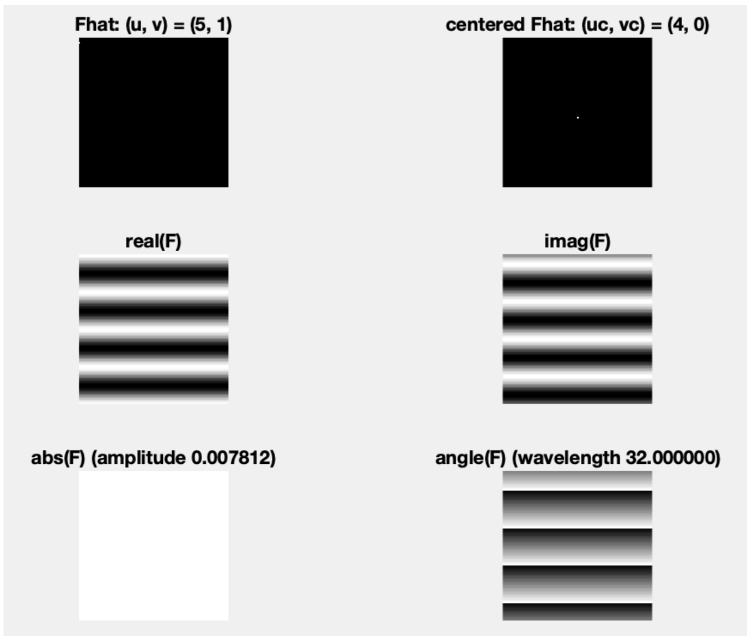
Answers:
- Placing a dot on the x-axis (y-axis) results in vertical (horizontal) lines.
- Other placements yield diagonal lines.
- The closer to the center we go, the higher the frequency, i.e. the more lines we get in the spatial domain. The smaller the absolute distance from the center, the higher the frequency.
- The plot of the imaginary part is typically the same as the real part, only that it is offset/shifted forward/backward (sin vs. cos).

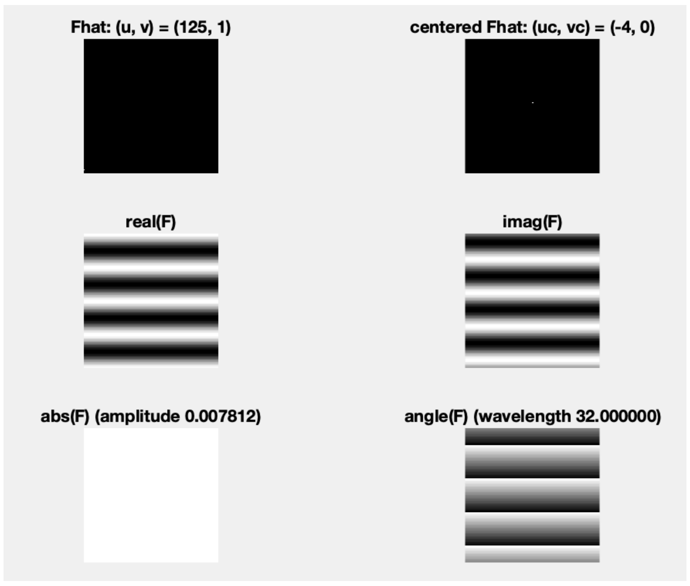



Fhat: (u, v) = (9, 5)
centered Fhat: (uc, vc) = (8, 4)
imag(F)
abs(F) (amplitude 0.007812)
angle(F) (wavelength 14.310835)


Fhat: (u, v) = (17, 9)
centered Fhat: (uc, vc) = (16, 8)
real(F)
imag(F)
abs(F) (amplitude 0.007812)
angle(F) (wavelength 7.155418)


Fhat: (u, v) = (17, 121)
centered Fhat: (uc, vc) = (16, -8)
imag(F)
abs(F) (amplitude 0.007812)
angle(F) (wavelength 7.155418)


Fhat: (u, v) = (5, 1)
centered Fhat: (uc, vc) = (4, 0)
real(F)
imag(F)
abs(F) (amplitude 0.007812)
angle(F) (wavelength 32.000000)

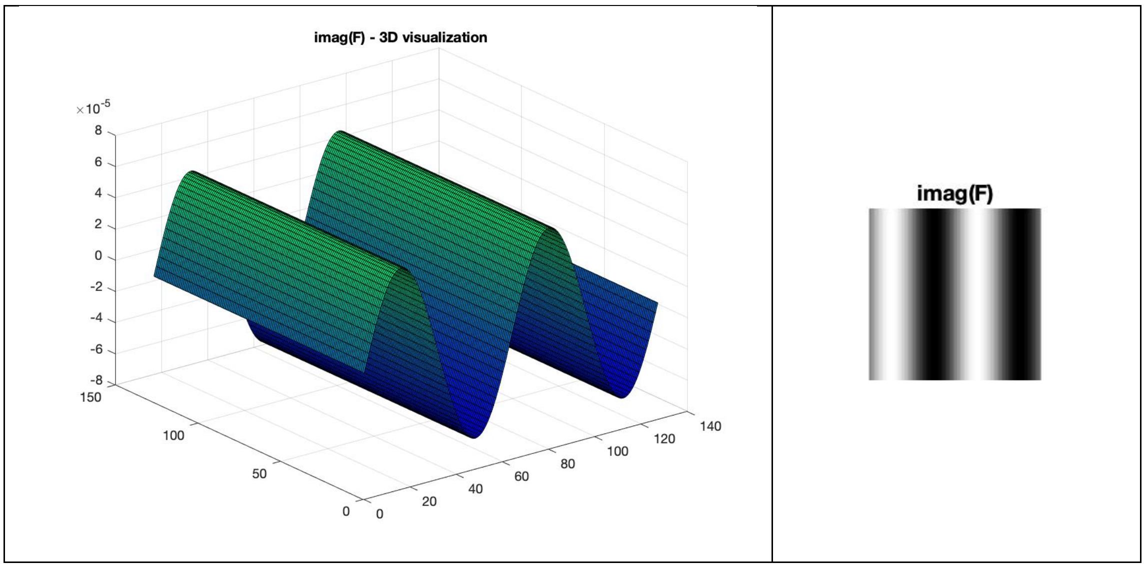

Fhat: (u, v) = (125, 1)
centered Fhat: (uc, vc) = (-4, 0)
real(F)
imag(F)
abs(F) (amplitude 0.007812)
angle(F) (wavelength 32.000000)

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:

- Discrete Fourier transform (3): $\hat{F}(u) = \mathcal{F}_D(F)(u) = \frac{1}{N}\sum_{x\in[0..N-1]^2} F(x)\, e^{\frac{-2\pi i u^T x}{N}}$
- Discrete Fourier inversion (4): $F(x) = \mathcal{F}_D{}^{-1}(\hat{F})(x) = \frac{1}{N}\sum_{u\in[0..N-1]^2} \hat{F}(u) e^{\frac{-2\pi i u^T x}{N}}$
- Euler's formula (5): $e^{i\omega^T x} = e^{i(\omega_1 x_1 + \omega_2 x_2)} = \cos(\omega^T x) + isin(\omega^T x)$

F(m, n) = {Euler's formula} $= \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \left[\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \left[\hat{F}(u,v) cos(\frac{2\pi nv}{N} + \frac{2\pi mu}{M}) + isin(\frac{2\pi nv}{N} + \frac{2\pi mu}{M}) \right] \right] = \{illustrate\ with\ u = p = 1\ and\ v = q = 3\} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} \left[\hat{F}(1,3) cos(\frac{2\pi n*3}{N} + \frac{2\pi m*1}{M}) + isin(\frac{2\pi n*3}{N} + \frac{2\pi m*1}{M}) \right] \right]$



2D and 3D visualisations of the imaginary part of the projection of the Fourier (frequency) domain position (1, 3) onto the spatial 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:

- Discrete Fourier Inversion (4): $F(x) = \mathcal{F}_D^{-1}(\hat{F})(x) = \frac{1}{N} \sum_{u \in [0..N-1]^2} \hat{F}(u) e^{\frac{-2\pi i u^T x}{N}}$
- Euler's Formula (5): $e^{i\omega^T x} = e^{i(\omega_1 x_1 + \omega_2 x_2)} = \cos(\omega^T x) + isin(\omega^T x)$
- Fourier Spectrum (6): $|F(u,v)| = \sqrt{Re^2(u,v) + Im^2(u,v)}$

From (4) we have:

$$F(m,n) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \left[\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \left[\hat{F}(u,v) e^{\frac{2\pi inv}{N}} \right] e^{\frac{2\pi imu}{M}} \right]$$

Using (5):

$$= \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \left[\frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \left[\hat{F}(u,v) cos(\frac{2\pi nv}{N} + \frac{2\pi mu}{M}) + isin(\frac{2\pi nv}{N} + \frac{2\pi mu}{M}) \right] \right]$$

Together, this yields:

$$real\ part\ Re(z) = \Re(z) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} \left[\hat{F}(u,v) cos(\frac{2\pi nv}{N} + \frac{2\pi mu}{M}) \right] \right]$$

$$imaginary\ part\ Im(z) = \Im(z) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} \left[\hat{F}(u,v) sin(\frac{2\pi nv}{N} + \frac{2\pi mu}{M}) \right] \right]$$

From the Fourier Spectrum (6), we get the amplitude A in the following way:

$$A = max(\sqrt{Re^2(u,v) + Im^2(u,v)})$$

Since the maxima of our trigonometric functions *cosine* and *sine* both equal 1 and they together at maximum equal one ($\sin^2 x + \cos^2 x = 1$), we can simplify to:

$$A = \frac{1}{\sqrt{MN}} max \left(\sum_{m=0}^{M-1} \left[\sum_{n=0}^{N-1} \left[\hat{F}(u,v) \right] \right] \right)$$

In our MATLAB script, this computation can be achieved the following way:

```
amplitude = max(abs(Fhat(:))) / sz;
```

Alternatively, with known values of the dimensions M and N as well as the max of the Fourier transform, we can compute the amplitude A ourselves directly:

$$M = N = 128$$

(our quadratic image has side M = N = 128 pixels)

$\max(\hat{F}(u,v)) = 1$ (we manually set the only non-zero point to equal 1)

which yields:

$$A = \frac{1}{\sqrt{128^2}} * 1 = \frac{1}{128} \approx 0.0078$$

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:

- Wavelength (7): $\lambda = \frac{2\pi}{|\omega|} = \frac{2\pi}{\sqrt{\omega_1^2+\omega_2^2}}$ (lecture 3)
- Phase / angle (8) of F. transf.: $\phi(\omega_1,\, \omega_2) = \tan^{-1}\frac{Im(\omega_1, \omega_2)}{Re(\omega_1, \omega_2)}$ (lecture 3)

  Or: $\phi(\omega\ \ ) = \tan^{-1}\frac{Im(\omega\ \ )}{Re(\omega\ \ )}$

- Relation between continuous angular frequency variable ω and discrete frequency variable u (9):

$$\omega_D = \frac{2 * \pi * \mathrm{u}}{N}$$

From lab description:

- "*The discrete Fourier transform may similarly be treated as periodic with a period of 2π, and we may change the definition domain to the interval [−π, π]^2 without losing in generality. This corresponds to a centering operation in the discrete Fourier transform.*"

  We are interested in calculating an angle. Therefore, it is advantageous to use centered coordinates. In the given MATLAB code, we find expressions for

$$u_c \text{ and } v_c.$$

- Along with N = sz, we get the following expression for λ using (7) and (9):

$$\lambda = \frac{2\pi}{\sqrt{\left(\frac{2\pi u_c}{sz}\right)^2 + \left(\frac{2\pi v_c}{sz}\right)^2}}$$

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:

From **help fftshift**:
Shift zero-frequency component to center of spectrum.

For vectors, fftshift(X) swaps the left and right halves of X. **For matrices, fftshift(X) swaps the first and third quadrants and the second and fourth quadrants.** For N-D arrays, fftshift(X) swaps "half-spaces" of X along each dimension.

fftshift is useful for visualizing the Fourier transform with the zero-frequency component in the middle of the spectrum.

Quadrants before swap:

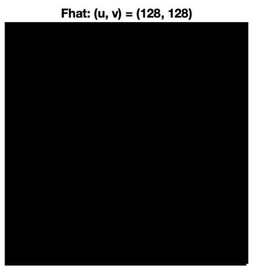
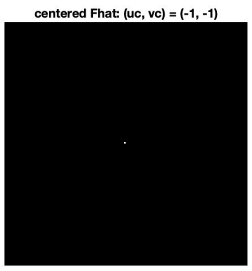
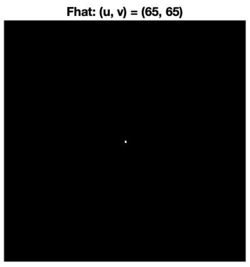
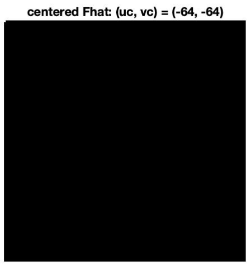
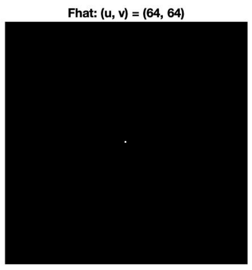
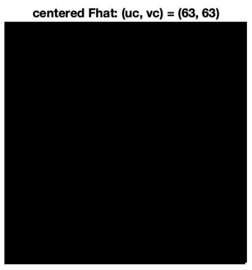
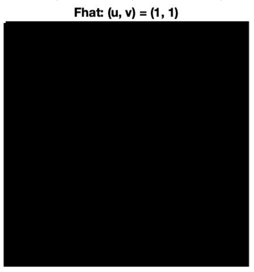
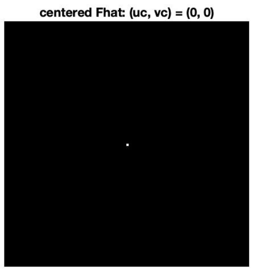
2	1
3	4

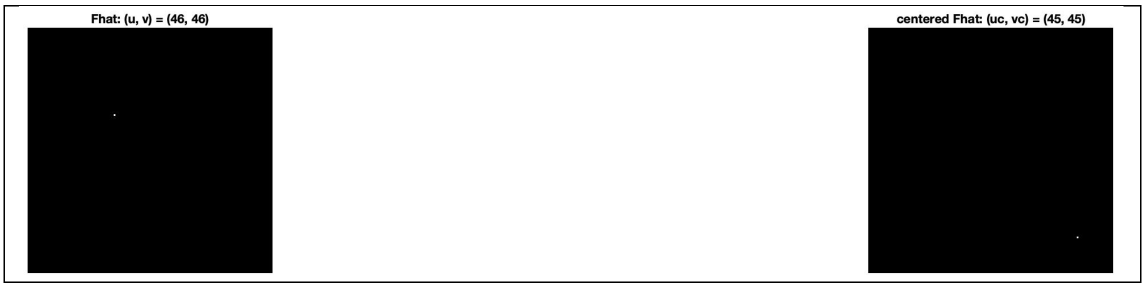
Quadrants after swap:

4	3
1	2

The default origin is located in the upper-left corner. Through fftshift, we move the origin to the center instead. Because of this, we need to calculate new coordinates corresponding to the new, shifted coordinate system. The new coordinates are calculated relative to the new, centered origin. This is why we check if the point exceeds the center or not.

Some examples (note that some points are located in corners):

u = 128, v = 128, uc = -1, vc = -1  
u = 65, v = 65, uc = -64, vc = -64  
u = 64, v = 64, uc = 63, vc = 63  
u = 1, v = 1, uc = 0, vc = 0  
u = 46, v = 46, uc = 45, vc = 45



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

Answers:
Fftshift, which is used in the second subplot, centers the plot by setting origo in the middle of the image.

The purpose of the code is to get uc and vc values which correspond to the positions of u and v in this centered plot given by fftshift. To summarize, it's just to translate the input u and v values to be displayed at their correct spots in the new, centered_Fhat plot, in which quadrants 1 and 3 have swapped places, as have quadrants 2 and 4.

The first if-else block in the code checks whether the point is in the upper side of the image (vertically). The second checks whether the point is in the left half of the image (horizontally).

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:
From lecture 3:

- $\hat{f}(u,v) = \frac{1}{\sqrt{MN}}\sum_{m=0}^{M-1}\left[\sum_{n=0}^{N-1}\left[f(m,n)e^{-2\pi i\left(\frac{mu}{M}+\frac{nv}{N}\right)}\right]\right]$ (1)

- $f(m,n) = \frac{1}{\sqrt{MN}}\sum_{u=0}^{M-1}\left[\sum_{v=0}^{N-1}\left[\hat{f}(u,v)e^{+2\pi i\left(\frac{mu}{M}+\frac{nv}{N}\right)}\right]\right]$ (2)

From the given code, we also get G equal to:

- $G(m,n) = \begin{cases}1\ \forall\ 57 \leq n \leq 72\\ 0\ otherwise\end{cases}$

The separability property mentioned in Lecture 4 says that "a 2D DFT can be implemented as a series of 1D DFTs along each column, followed by 1D DFTs along each row".

Applying the separability property to (1) and using our values for G yields:

- $\hat{f}(u,v) = \frac{1}{\sqrt{MN}}\sum_{n=57}^{72}\left[e^{-2\pi i\left(\frac{nv}{N}\right)}\sum_{m=0}^{M-1}\left[e^{-2\pi i\left(\frac{mu}{M}\right)}\right]\right]$ (3)

The Dirac distribution for discrete values is called the Kronecker delta function. It is mentioned in lecture 4. We use the Kronecker delta function to simplify our expression.

- $\delta(x,y) = \begin{cases}0\ \forall\ x,y \neq 0\\ 1\ otherwise\end{cases}$ $and\ \iint_{\infty\infty}^{\infty\infty}\delta(x,y)dxdy = 1$ (4)

- $or\ \delta(n) = \begin{cases}0\ \forall\ n \neq 0\\ 1\ otherwise\end{cases}$ $and\ \int_{\infty}^{\infty}\delta(x,y)dxdy = 1$

Note that in our case, this means the following to hold true:

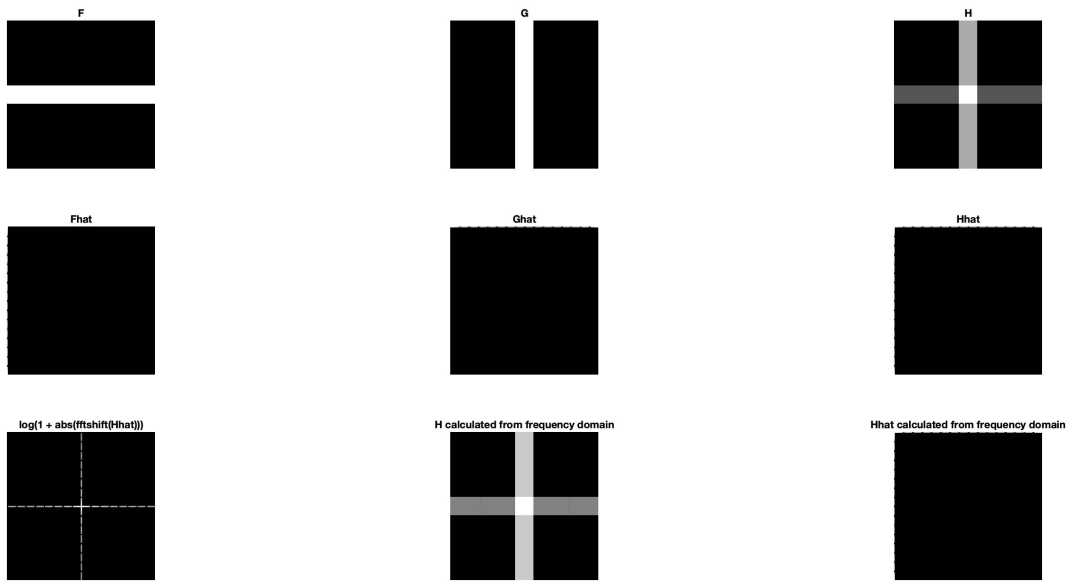
- $\sum_{m=0}^{M-1}\left[e^{-2\pi i\left(\frac{mu}{M}\right)}\right] = \delta(m)$ (5)

Put (5) in (3) and we get a final expression:

- $\hat{f}(u,v) = \frac{1}{\sqrt{MN}}\sum_{n=57}^{72}\left[e^{-2\pi i\left(\frac{nv}{N}\right)}\right] * \delta(m)$ (6)

Since the Dirac distribution is 1 only for $m = 0$, our $\hat{f}(u,v)$ expression takes on non-zero values only when $m = 0$. This is the sinc function times the Dirac function. In other words, the Fourier spectra for G will be concentrated to the upper border (where n = 0). Analogously, the Fourier spectra for F will be concentrated to the left border, since G is just the transpose of F ($G = F^T$). The sinc function gives the wave form.

Visualization:

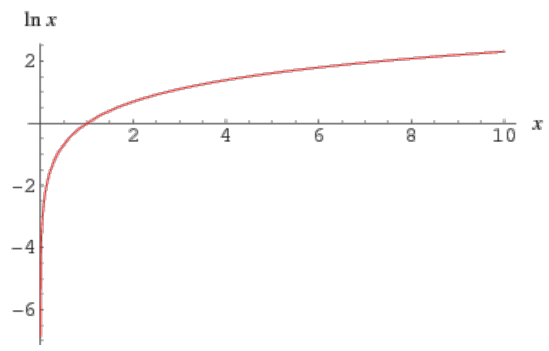

F
G
H
Fhat
Ghat
Hhat
log(1 + abs(fftshift(Hhat)))
H calculated from frequency domain
Hhat calculated from frequency domain

Question 8: Why is the logarithm function applied?

Answers:
From lecture 5 slide 7 (and also from the book), we note the following:

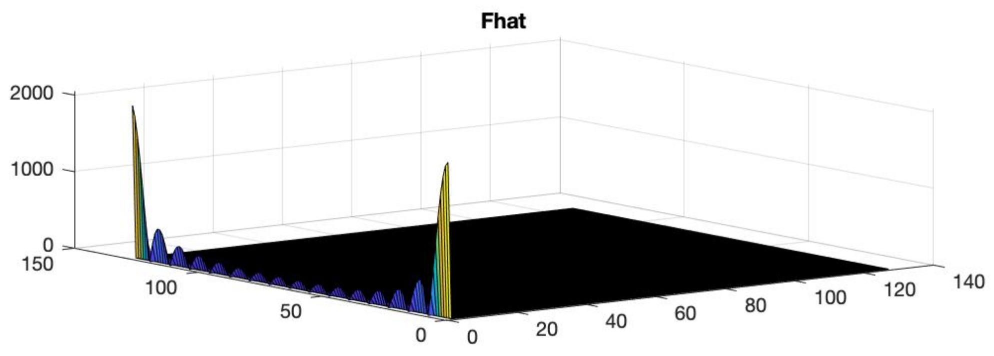
- "*log transformations are useful for compressing large dynamic ranges and making details visible*".

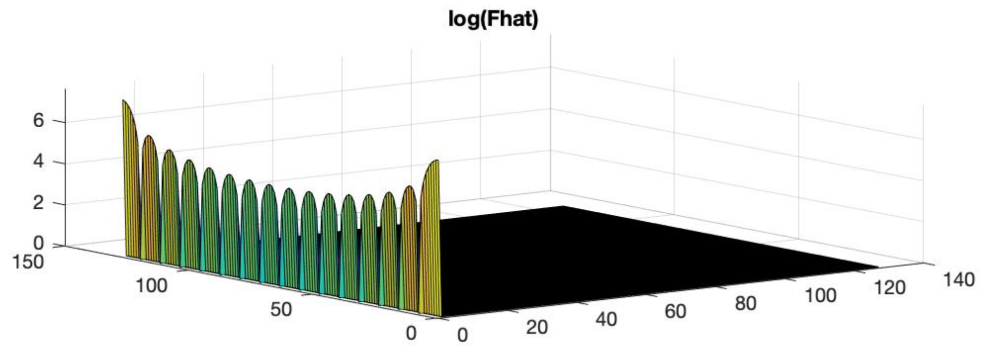
Referring to the shape of a logarithmic function, we are able to visualize why this is true:



In other words, a logarithm function applied to the transform will reduce the difference between the lowest and highest frequencies, resulting in a more balanced and visually digestible image. It "evens out" the distribution of the pixels, allowing those ranges to become visible who previously were not. Because higher frequencies will be compressed, there is however a risk of losing information.

Visualization:

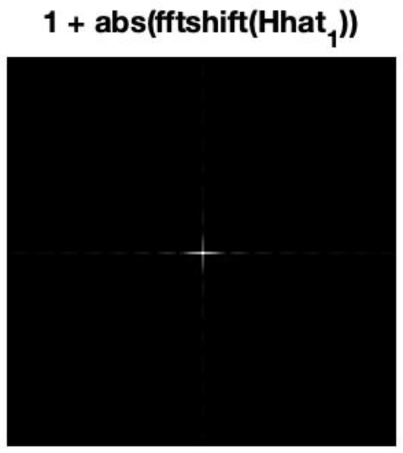


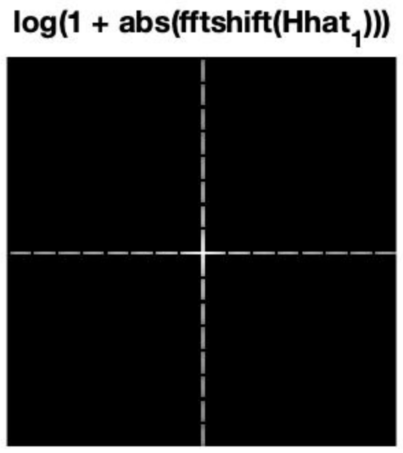


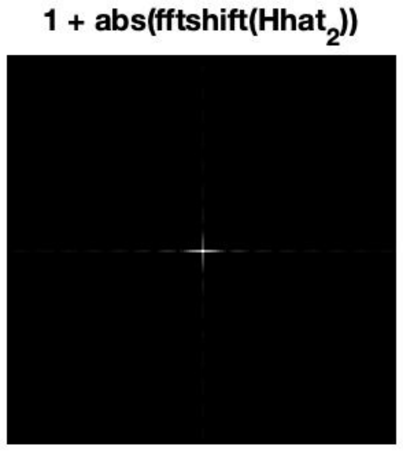
Question 9: What conclusions can be drawn regarding linearity? From your observations can you derive a mathematical expression in the general case?

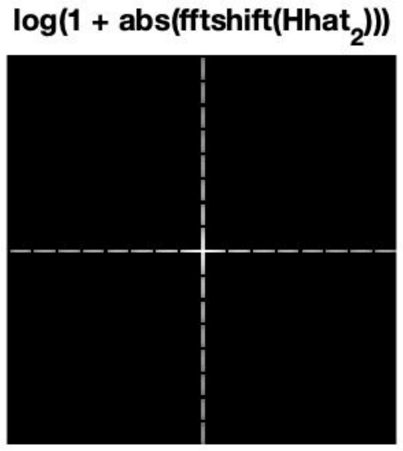
Answers:
Note that the relation between H and its two constituent parts F and G is mirrored in the relation between Hhat on the one side and Fhat and Ghat on the other. That is, we can get H

- EITHER by applying the H = F + 2 * G formula
- OR by first transforming F and G to the frequency domain (thus obtaining Fhat and Ghat respectively), computing Hhat by applying Hhat = Fhat + 2 * Ghat and then finally transforming the result back to the spatial domain (through ifft2(Hhat2) in the case of MATLAB).









Mathematical expression for the general case (the linearity property mentioned in lecture 4 slide 10):

- $\mathcal{F}[af_1(m,n) + bf_2(m,n)] = a\hat{f}_1(u,v) + b\hat{f}_2(u,v)$
- $af_1(m,n) + bf_2(m,n) = \mathcal{F}^{-1}\left[a\hat{f}_1(u,v) + b\hat{f}_2(u,v)\right]$

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.

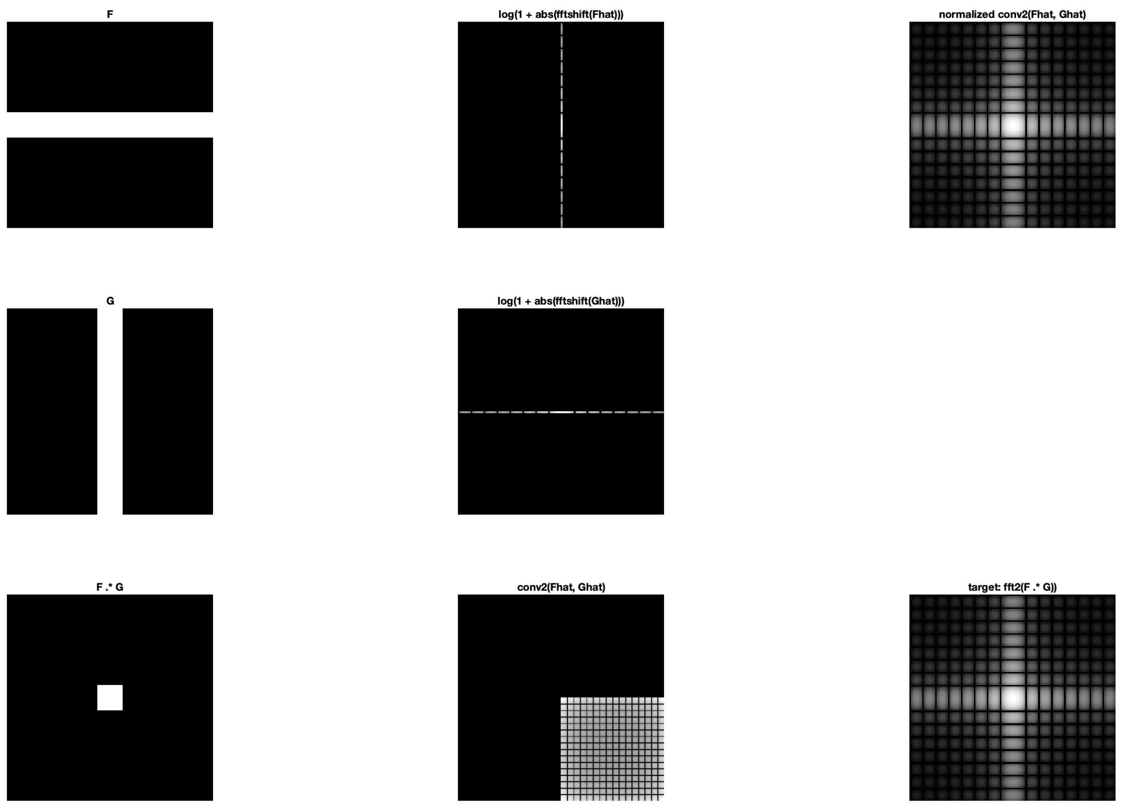
Answers:
Yes. Multiplication in the Fourier domain equals convolution in the spatial domain.

From lecture 3:
- "*Convolution in the spatial domain is the same as multiplication in the Fourier (frequency) domain*"
- $\mathcal{F}(h * f) = \mathcal{F}(h)\mathcal{F}(f)$

From Digital Image Processing (Gonzales and Woods), page 210:
- "*Convolution in the frequency domain is analogous to multiplication in the spatial domain.*"

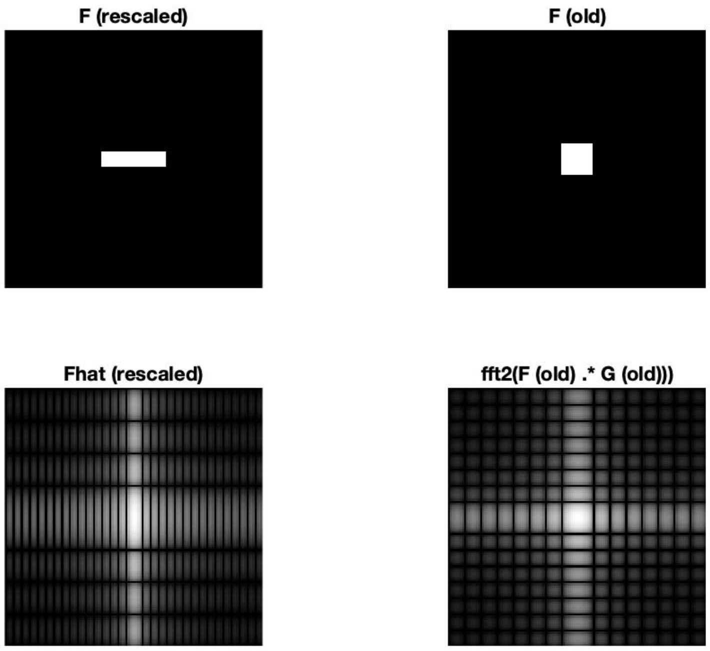
To prove the theorem in practice:
1. We create images F, G and F.*G. They are displayed in the left-most column in the image below.
2. We also graph the target image we wish to achieve, i.e. the Fourier transform of F.*G. It is shown in the bottom-right corner of the image below.
3. Our first step in getting there through convolution in the Fourier / frequency domain (as opposed to multiplication in the spatial domain followed by Fourier transforming the result) is to compute Fhat and Ghat respectively. They are showed in the upper and middle section of the mid-column.
4. We can now convolute Fhat with Ghat. The result is shown in the bottom part of the mid-column. Note that the graph has a dimension of 255 x 255 pixels, as opposed to 128 x 128.
   "*C = conv2(A, B) performs the 2-D convolution of matrices A and B.*
   If [ma,na] = size(A), [mb,nb] = size(B), and [mc,nc] = size(C), then
   mc = max([ma+mb-1,ma,mb]) and nc = max([na+nb-1,na,nb])."
5. The final step is now to reduce the convolution to 128 x 128 pixels and to normalize it by dividing it by a factor 128^2.
6. We are able to see that the resulting graph (shown in the upper-right corner of the image below) is the same as the target image.

E.g. F: no change in the y-direction for any x-value, therefore, no frequencies in the y-direction in Fhat. Vice versa for G.



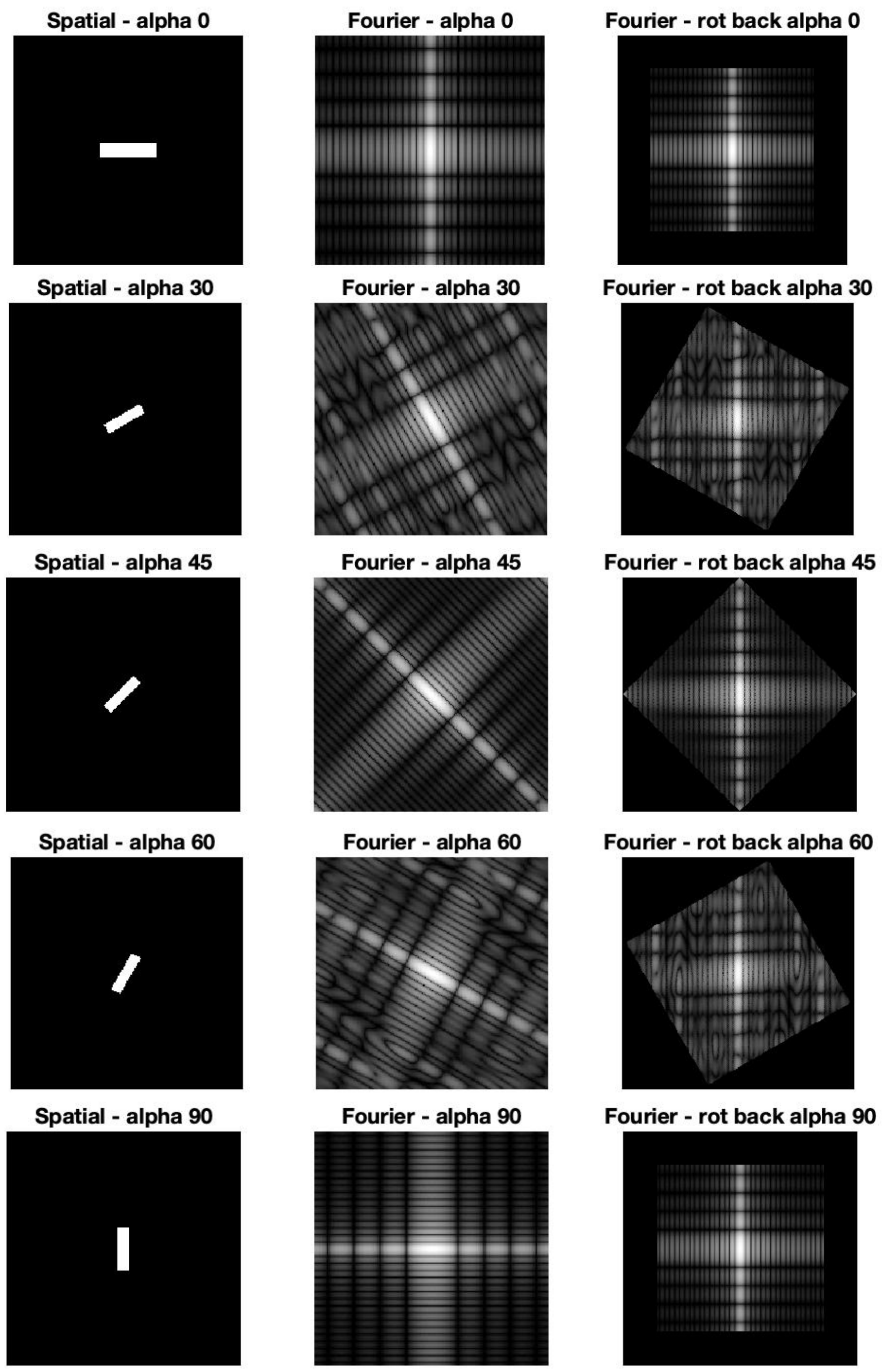
Question 11: What conclusions can be drawn from comparing the results with those in the previous exercise? See how the source images have changed and analyze the effects of scaling.

Answers:
Scaling in the spatial domain corresponds to scaling in the Fourier domain but in the opposite "direction". In other words, expansion in the spatial domain corresponds to compression in the Fourier domain (and vice versa).



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:



From lecture 4, we know that a rotation in one domain becomes a rotation in the other domain. Property V of the Fourier transforms (also found in the lecture 4 slides) further states that a rotation of the original image rotates fhat by the same angle. From our graphs above, we note that the distances from the edges to the origin stays the same in the Fourier domain, which we can interpret to mean that the frequencies will not change in value when rotated. However, it is clear that the direction of the waves changes; it changes corresponding to the angular change in the spatial domain.What's more is that distortion largely affects the rotated image of the Fourier transform greatly.

Why distortion?
Due to the implementation of rot:

```
   C = conv2(A, B) performs the 2-D convolution of matrices A and B.
   If [ma,na] = size(A), [mb,nb] = size(B), and [mc,nc] = size(C), then
   mc = max([ma+mb-1,ma,mb]) and nc = max([na+nb-1,na,nb]).
```

From "help rot":
"This is not a very interesting rotation, but it will, like all other values of ANGLE, create an output image ROTIM of size DxD, where D = NORM(SIZE(IMAGE)), with the original image in the center."

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

Answers:
Phase:

- Contains information about the position of edges. Distorting the phase makes the image completely unintelligible, even when keeping the magnitude unchanged (and therefore also the overall gray-levels contained in the image). The result can be compared to moving the pixels around in the image, while keeping their individual intensities unchanged.

Magnitude:

- Contains information about the intensities in the image. After distortion of the intensities through manipulation of the magnitude, one is still able to discern what is depicted in the image, since edges are left in their original locations.

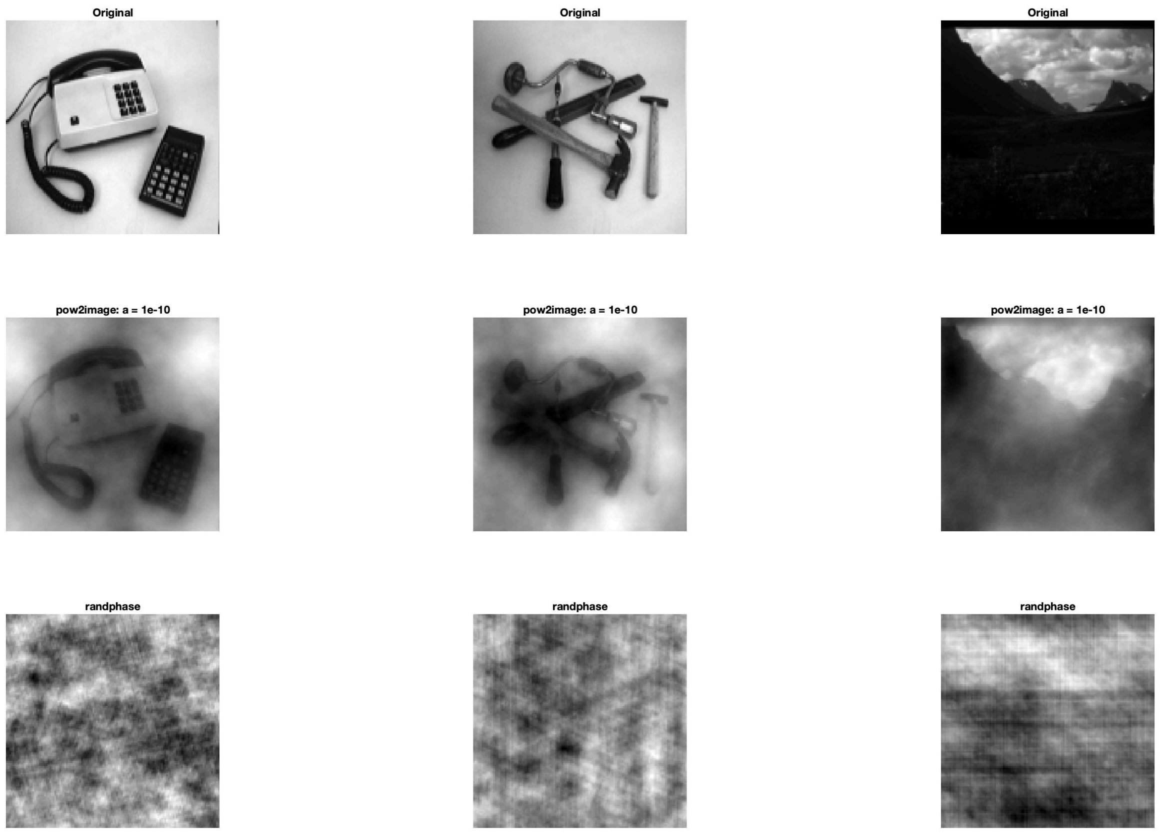
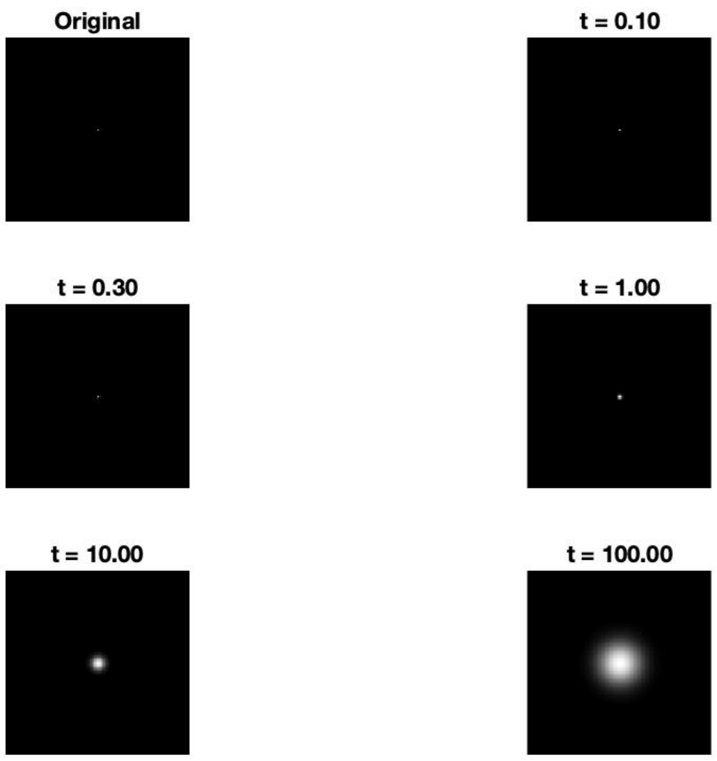


Figure: The first row contains the original images. The second row contains the same images but with changed magnitudes. The third row contains the same images but with their phases shifted.

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:
Using the impulse as described in the lab description (deltafcn(128, 128)) yields the following responses for the 5 different t values:



t	**covariance matrix**	
0.1	0.0133	0.0000
	0.0000	0.0133
0.3	0.2811	0.0000
	0.0000	0.2811
1.0	1	0
	0	1
10.0	10	0
	0	10
100.0	100	0
	0	100

High t-values correspond to higher variances. The higher the variance, the blurrier the response, and the higher its frequencies. By the nature of the Gaussian we know it to also be rotationally symmetric. In addition to being circularly symmetric, Gaussian blur can also be applied to a two-dimensional image as two independent one-dimensional calculations (it is a separable filter).

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:
For our comparison with the ideal continuous case, we used the identity matrix multiplied by the current t-value. For the discrete case, we used the discgaussfft given in the lab description.

t		**difference to ideal (discrete)**			**difference to ideal (continuous)**	
0.1		0.0867	0		0.0867	0
		0	0.0867		0	0.0867
0.3		0.0189	0		0.0189	0
		0	0.0189		0	0.0189
1.0		$0.2112*10^{-6}$	0		$0.2112*10^{-6}$	0
		0	$0.2112*10^{-6}$		0	$0.2112*10^{-6}$
10.0		$0.4526*10^{-11}$	0		$0.1283*10^{-11}$	0
		0	$0.4309*10^{-11}$		0	$0.1380*10^{-11}$
100.0		$0.5608*10^{-6}$	0		$0.6718*10^{-6}$	0
		0	$0.5608*10^{-6}$		0	$0.6718*10^{-6}$

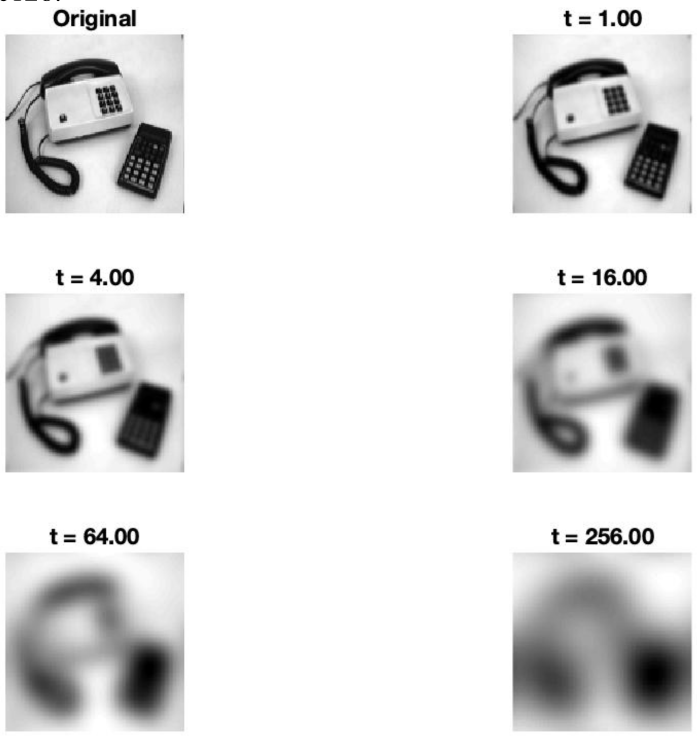
Large values for t generate smaller differences. Small values for t result in larger differences.

One can see that for t-values < 1, results different somewhat compared to the ideal cases (differences of roughly 0.0867 and 0.0189 for $t = 0.1$ and $t = 0.3$ respectively). For $t \geq 1$, our covariance matrises are less different from the ideal covariance matrises. All in all we see that there are smaller differences in the variance compared to the ideal cases for higher values of t.

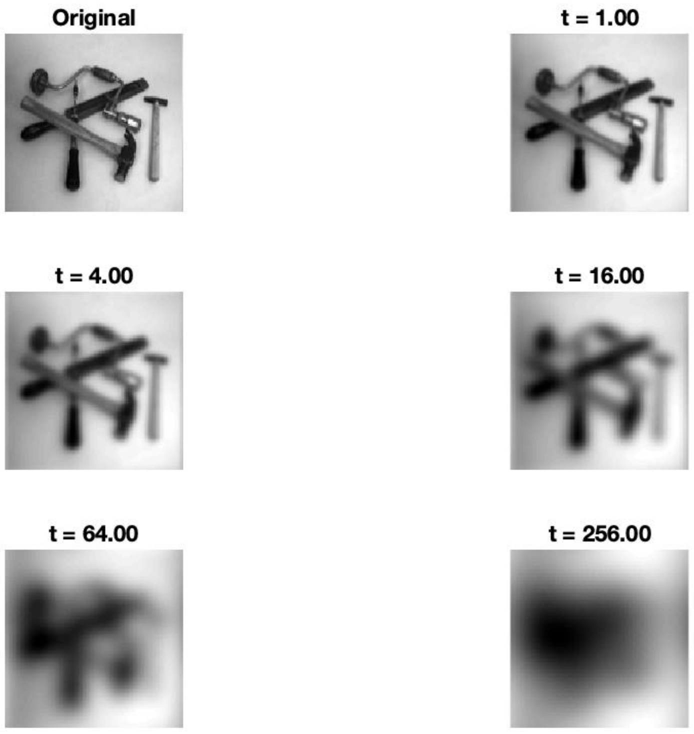
Question 16: Convolve a couple of images with Gaussian functions of different variances (like t = 1.0, 4.0, 16.0, 64.0 and 256.0) and present your results. What effects can you observe?

Answers:
First, let us display the results of using *phonecalc128*, *few128* and *nallo128* as input images.

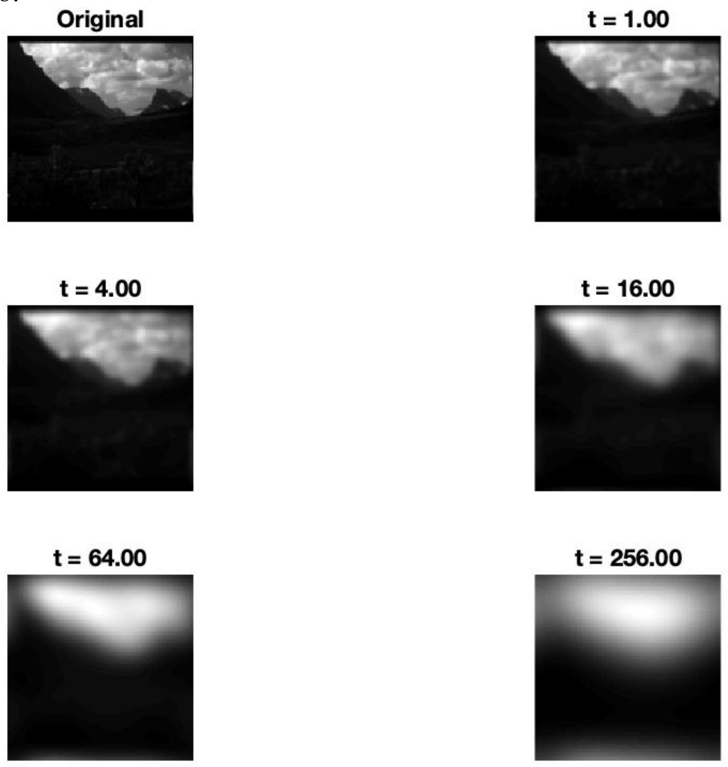
- phonecalc128:



- few128:



- nallo128:



As mentioned previously, we notice that the higher the variance of the Gaussian function used for convoluting, the blurrier the end result. In other words, more and more of the higher frequencies of the image are being supressed. Corners and edges are rendered less distinct (the original border we had in nallo128 for example disappears completely with high values for t).

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:

	Advantage	**Disadvantage**
Gaussian filtering	+ Smoothes/blurs the image (lowpass filter) + Good at handling Gaussian noise + Gaussian in both the spatial and the frequency domain + Can be applied separately for x and y directions (convolve first with 1-D Gaussian in x direcion, then with another 1-D Gaussian in y direction)	- Smoothes/blurs the image (lowpass filter) - Does not handle S&P noise well - Not good at preserving edges
Median filtering	+ Edge-preserving + Capable of handling both Gaussian and S&P noise (eliminates local extreme values) + Preserves shading	- Resulting image looks painted - Smoothes out non-distinct features
Ideal low-pass filtering	+ Good at handling white noise (one half of S&P noise) + Simple to use	- Performance not particularly good on neither Gaussian nor S&P noise - Distorts image with further noise by cutting off certain frequencies - Rippling / blurry effect

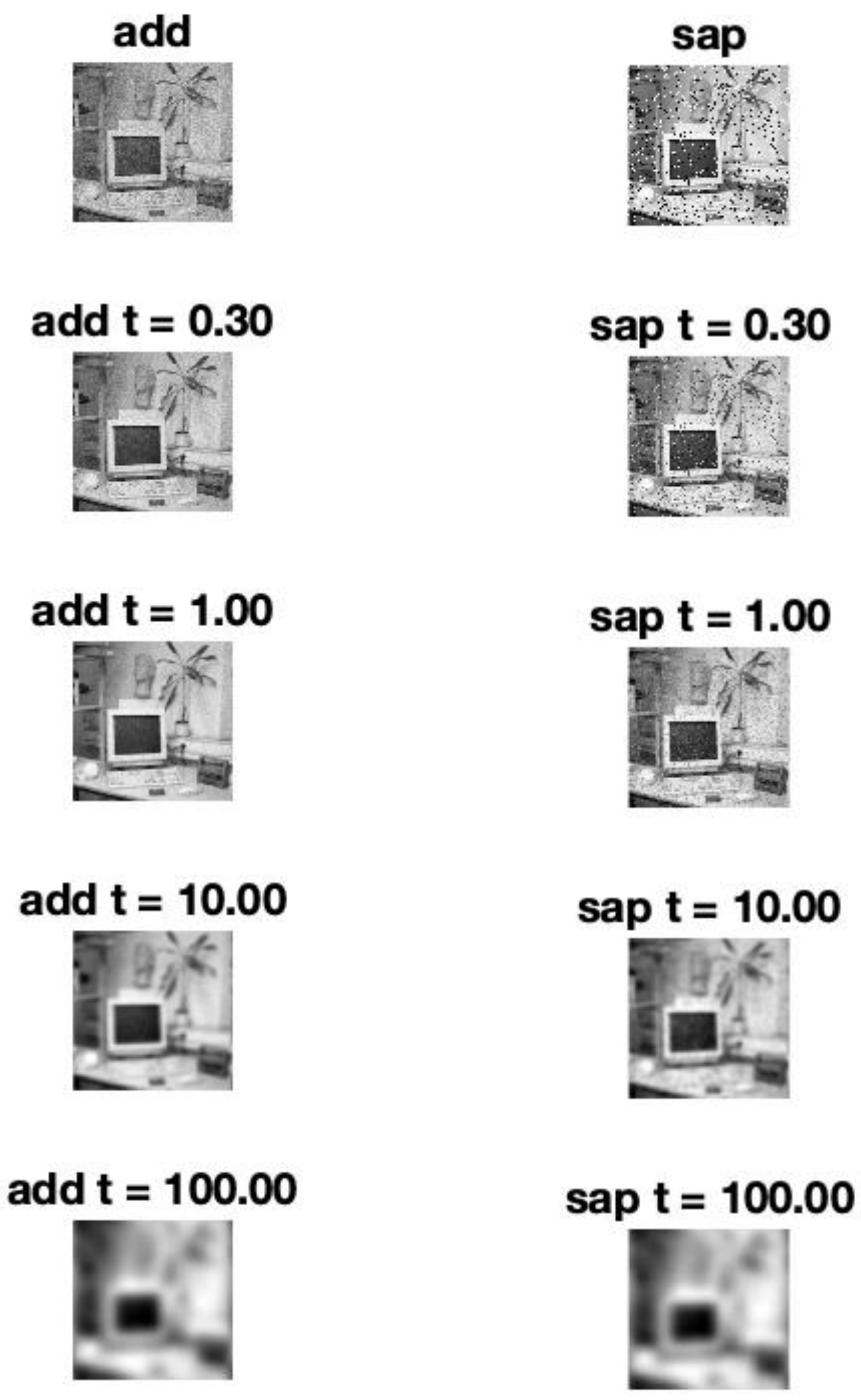


Figure: Gaussian filtering applied to the "office" image with different values for the variance (displayed here as "t"). Top-left corner: Gaussian noise added to the original image. Top-right corner: salt and pepper noise added to the original image. Results of median filtering displayed below (rows 2 to 5).

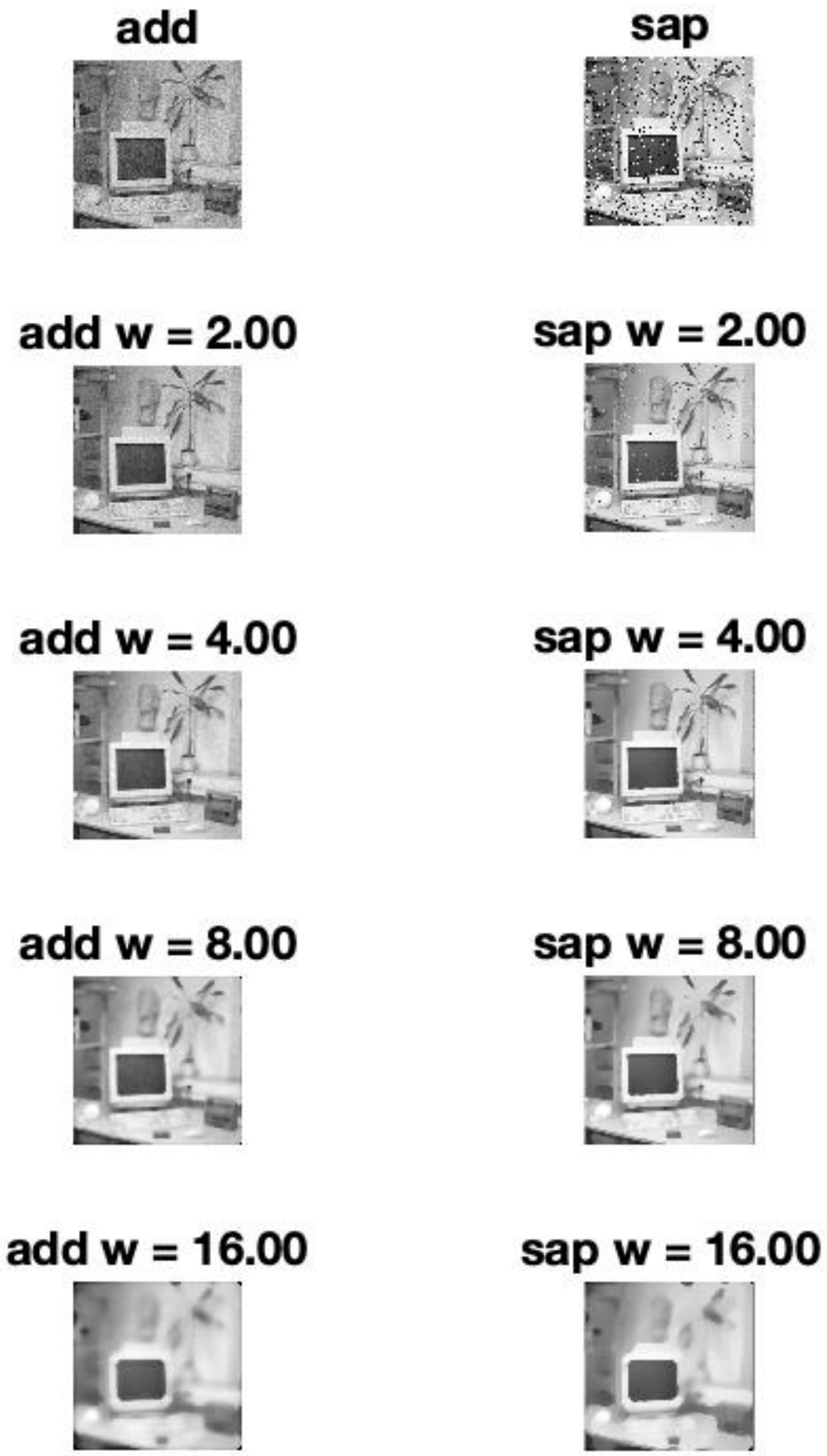


Figure: Median filtering with different values for the window width (displayed here as “w”). Top-left corner: Gaussian noise added to the original image. Top-right corner: salt and pepper noise added to the original image. Results of median filtering displayed below (rows 2 to 5).

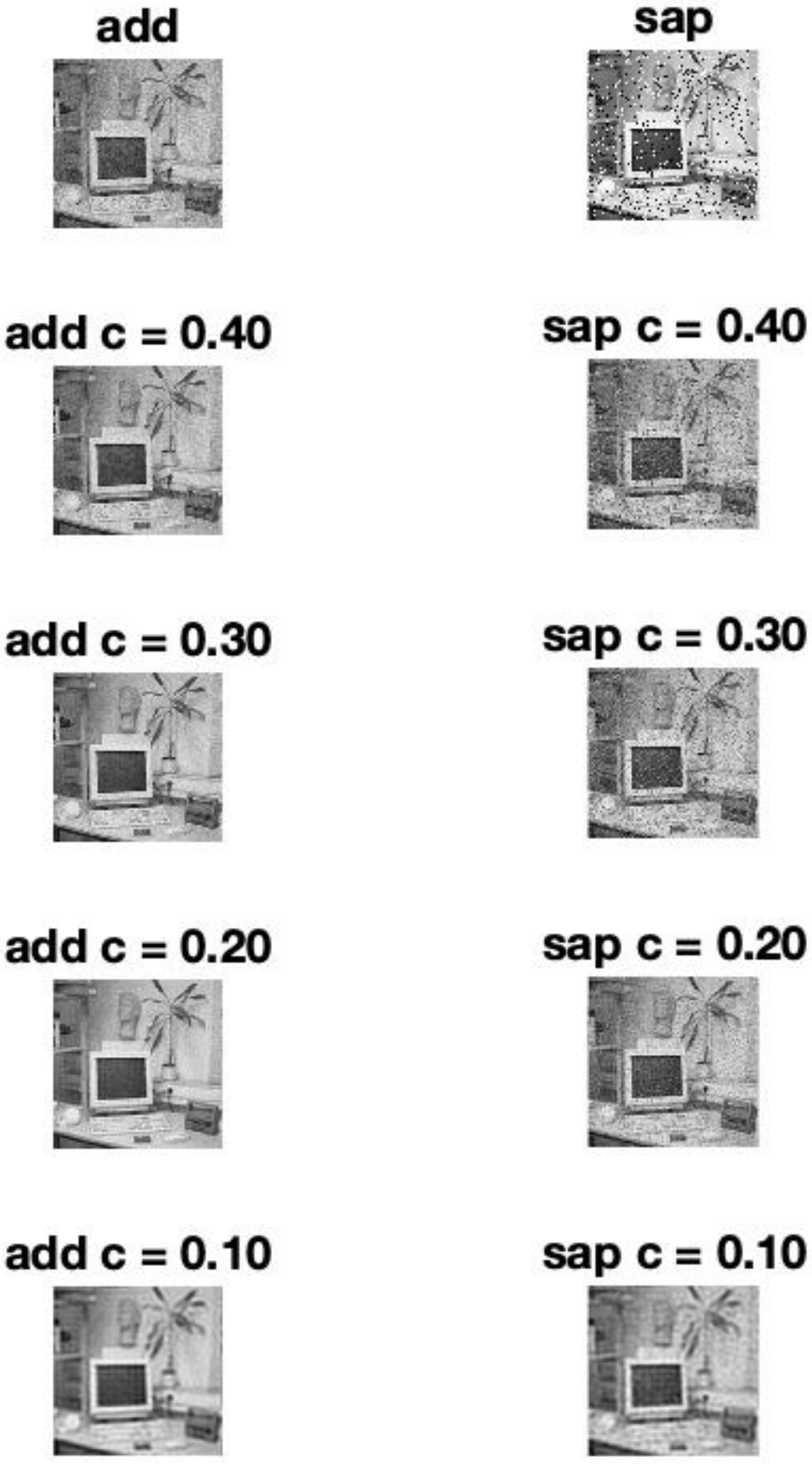


Figure: Ideal low-pass filtering with different values for the cut-off frequency (displayed here as “c”). Top-left corner: Gaussian noise added to the original image. Top-right corner: salt and pepper noise added to the original image. Results of ideal low-pass filtering displayed below (rows 2 to 5).

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

Answers:
Referring to our table of pros and cons of the respective filtering techniques, we conclude that median filtering is adept at removing both kinds of noise. It also preserves edges well. A risk is that with large(r) window sizes, the resulting image gets a painted look.

The Gaussian can also reduce noise, but at a higher cost. The more noise it removes, the more blur it adds to the resulting image (thus introducing another kind of noise, if one wishes to put it that way).

The ideal low-pass filter does not perform well in this particular setting. It reduces noise only by removing frequencies higher than the defined cut-off frequency. This results in this filter not being very effective at handling e.g. S&P noise, due to its sensitivity to outliers. White dots tend to be removed prior to black spots (the two halves of S&P noise).

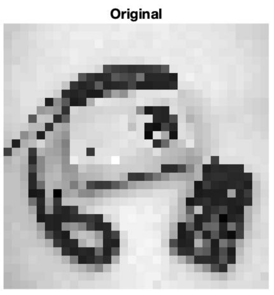
Question 19: What effects do you observe when subsampling the original image and the smoothed variants? Illustrate both filters with the best results found for iteration i = 4.

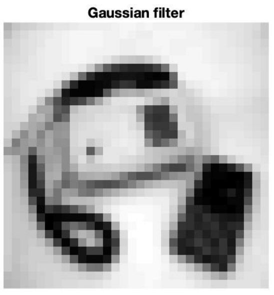
Answers:

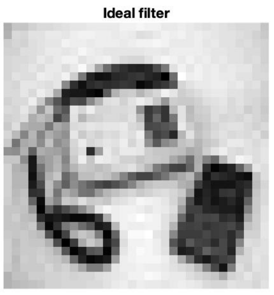
Using t = 0.6 for the Gaussian and cut-off frequency = 0.28 for the Ideal we get results as shown in the figure to the right.

Smooth before subsampling, in order to spread out the

First off, subsampling of the images results in pixelation, as can be clearly seen in the original image displayed here to the right. Furthermore, one can observe the rippling effect of the ideal filter. One can also observe the general smoothing/blurring capabilities of the Gaussian filter.



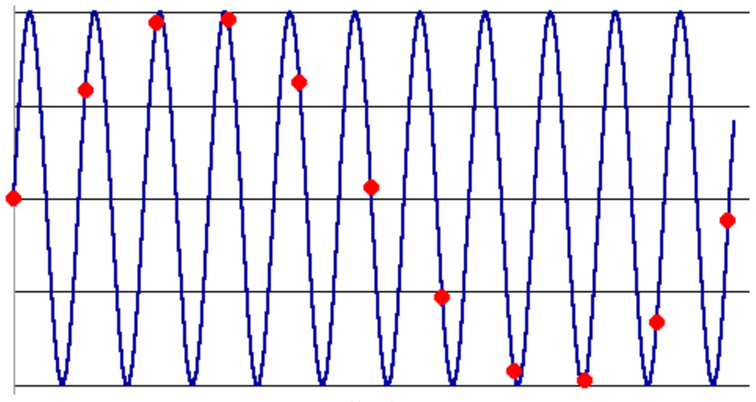




Question 20: What conclusions can you draw regarding the effects of smoothing when combined with subsampling? Hint: think in terms of frequencies and side effects.

Answers:

- Smoothing before subsampling, more data is spread out and therefore preserved after subsampling.
- When the image is blurred, the frequencies are reduced.
- Subsampling is when we sample at a rate lower than the Nyquist rate. It is usually done *after* having filtered the image through the use of e.g. a low-pass or bandpass filter, which would have reduced the information content of the original signal, thus reducing the Nyquist rate to a level closer to the new sampling rate.
- If we smooth the image before subsampling, we can prevent information from being lost in the image, due to the information having been spread out across larger areas (thus reducing the Nyquist rate).
- By blurring the image, we reduce its frequencies, thereby reducing the Nyquist rate resulting in better matching with the new subsampled image.
- Subsampling is a common approach to counteract the effects of aliasing.


Aliasing

Gaussian blurring is commonly used when reducing the size of an image. When downsampling an image, it is common to apply a low-pass filter to the image prior to resampling. This is to ensure that spurious high-frequency information does not appear in the downsampled image (aliasing). Gaussian blurs have nice properties, such as having no sharp edges, and thus do not introduce ringing into the filtered image.