

# DD2423 - Lab I

## Alexandros Filotheou

### 1 Section 1.3 - Basic Functions

#### 1.1 Question 1

What we see here is that:

- The further the non-zero point  $(p, q)$  from the origin  $O(0, 0)$ , the smaller the wavelength of the spatial image (more dense lines in the real and imaginary part of the spatial image),
- The amplitude of all the spatial images is the same,
- The direction of the waveforms in the spatial images is dictated by the position of the non-zero point  $(p, q)$  relative to the origin  $O(0, 0)$

The output of the `fftwave` function for  $(p, q) = (5, 9)$ ,  $(p, q) = (9, 5)$ ,  $(p, q) = (17, 9)$ ,  $(p, q) = (17, 121)$ ,  $(p, q) = (5, 1)$  and  $(p, q) = (125, 1)$  is illustrated in figures 1, 2, 3, 4, 5 and 6 respectively.

$\hat{\mathbf{F}}$ :  $(u, v) = (5, 9)$

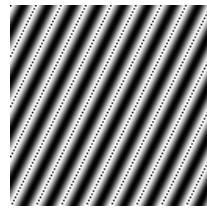


real( $\mathbf{F}$ )

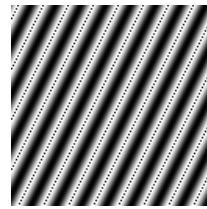
centered  $\hat{\mathbf{F}}$ :  $(uc, vc) = (4, 8)$



imag( $\mathbf{F}$ )



abs( $\mathbf{F}$ ) (amplitude 0.000061)



angle( $\mathbf{F}$ ) (wavelength 14.310835)

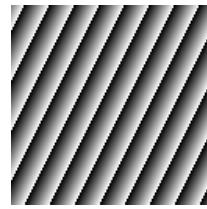


Figure 1:  $(p, q) = (5, 9)$

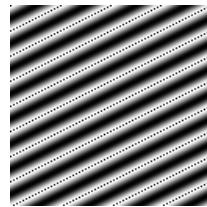
$\hat{\mathbf{F}}$ :  $(u, v) = (9, 5)$



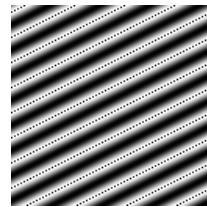
centered  $\hat{\mathbf{F}}$ :  $(uc, vc) = (8, 4)$



$\text{real}(\mathbf{F})$



$\text{imag}(\mathbf{F})$



$\text{abs}(\mathbf{F})$  (amplitude 0.000061)



$\text{angle}(\mathbf{F})$  (wavelength 14.310835)

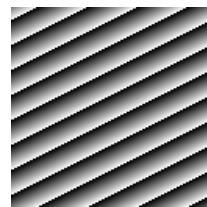


Figure 2:  $(p, q) = (9, 5)$

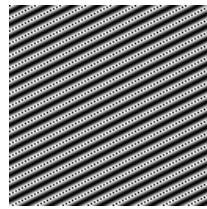
$\hat{\mathbf{F}}: (\mathbf{u}, \mathbf{v}) = (17, 9)$



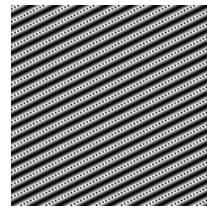
centered  $\hat{\mathbf{F}}: (\mathbf{u}_c, \mathbf{v}_c) = (16, 8)$



$\text{real}(\mathbf{F})$



$\text{imag}(\mathbf{F})$



$\text{abs}(\mathbf{F})$  (amplitude 0.000061)



$\text{angle}(\mathbf{F})$  (wavelength 7.155418)

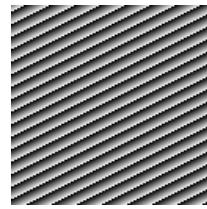


Figure 3:  $(p, q) = (17, 9)$

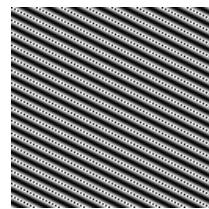
$\hat{F}$ :  $(u, v) = (17, 121)$



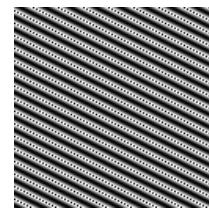
centered  $\hat{F}$ :  $(uc, vc) = (16, -8)$



$\text{real}(F)$



$\text{imag}(F)$



$\text{abs}(F)$  (amplitude 0.000061)



$\text{angle}(F)$  (wavelength 7.155418)

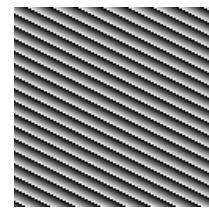


Figure 4:  $(p, q) = (17, 121)$

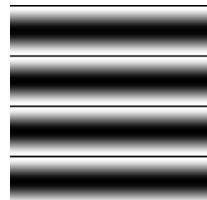
$\hat{\mathbf{F}}$ :  $(\mathbf{u}, \mathbf{v}) = (5, 1)$



centered  $\hat{\mathbf{F}}$ :  $(\mathbf{u}_c, \mathbf{v}_c) = (4, 0)$



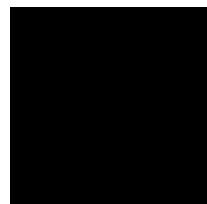
$\text{real}(\mathbf{F})$



$\text{imag}(\mathbf{F})$



$\text{abs}(\mathbf{F})$  (amplitude 0.000061)

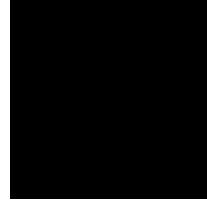


$\text{angle}(\mathbf{F})$  (wavelength 32.000000)



Figure 5:  $(p, q) = (5, 1)$

$\hat{\mathbf{F}}: (\mathbf{u}, \mathbf{v}) = (125, 1)$



centered  $\hat{\mathbf{F}}$ :  $(\mathbf{u}_c, \mathbf{v}_c) = (-4, 0)$



$\text{real}(\mathbf{F})$



$\text{imag}(\mathbf{F})$



$\text{abs}(\mathbf{F})$  (amplitude 0.000061)



$\text{angle}(\mathbf{F})$  (wavelength 32.000000)

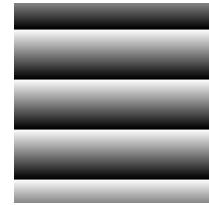


Figure 6:  $(p, q) = (125, 1)$

## 1.2 Question 2

We exploit equation 4.2 – 33 from [?]

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} s(x, y) \cdot A \delta(x - x_0, y - y_0) = A \cdot s(x_0, y_0) \quad (1)$$

knowing that the output Fourier transform is a delta function at  $(p, q)$ . Hence, for a quadratic image  $M = N$  and in the spatial domain:

$$f(x, y) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \delta(u - p, v - q) \cdot e^{\frac{2\pi i \cdot (xu + yv)}{N}} = \frac{1}{N^2} \cdot e^{\frac{2\pi i \cdot (px + qy)}{N}} \quad (2)$$

Hence,

$$f(x, y) = \frac{1}{N^2} \cdot \left( \cos\left(\frac{2\pi \cdot (px + qy)}{N}\right) + i \sin\left(\frac{2\pi \cdot (px + qy)}{N}\right) \right) \quad (3)$$

The spatial image for  $(p, q) = (5, 9)$  is plotted in figure 7. The real part of equation 3 is plotted in figure 8.

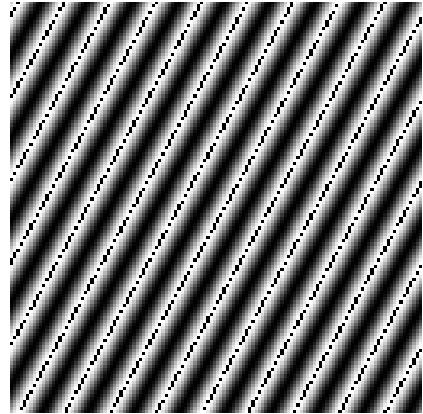


Figure 7: The inverse Fourier transform of an image for  $(p, q) = (5, 9)$ .

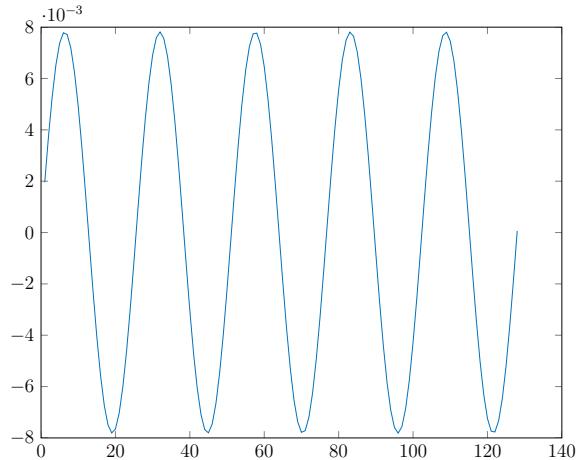


Figure 8: The real part of the spatial image above.

### 1.3 Question 3

As can be seen in equation 3, the amplitude of the waveform is

$$A = \frac{1}{N^2} \quad (4)$$

### 1.4 Question 4

As seen in the lecture notes,

$$\lambda = \frac{2\pi}{|\omega|} \quad (5)$$

and

$$\omega = \left[ \frac{2\pi u}{N} \quad \frac{2\pi v}{N} \right]^T \quad (6)$$

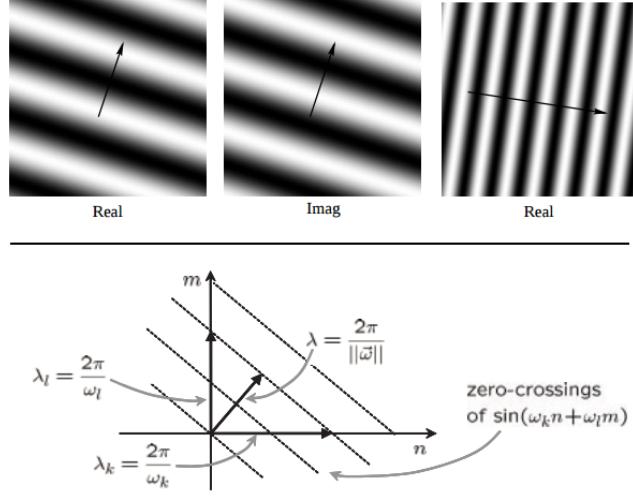


Figure 9: Image taken from Allan Jepson's notes ([www.cs.toronto.edu/~jepson/csc320/notes/linearFilters2.pdf](http://www.cs.toronto.edu/~jepson/csc320/notes/linearFilters2.pdf))

Hence, equation 5 for  $(u, v) = (p, q)$  becomes

$$\lambda = \frac{1}{\sqrt{p^2 + q^2}} \quad (7)$$

The direction of travel of the waveforms in the spatial images is dictated by the position of the non-zero point  $(p, q)$  relative to the origin  $O(0, 0)$ . Figure 9 illustrates the dependence of the direction of a waveform on  $p = \frac{\omega_l N}{2\pi}$  and  $q = \frac{\omega_k N}{2\pi}$ .

## 1.5 Question 5

For an quadratic image of size  $N$ , the highest number of cycles that can fit in it is  $N/2$ , that is, stripes of width of 1 pixel. The corresponding (maximum) frequency is  $\omega_{max} = 2\pi/2 = \pi$ . However,

$$\omega = \frac{2\pi u}{N} \leq \omega_{max} \quad (8)$$

Hence, when either  $p$  or  $q$  exceed the value of  $N/2$ , which in our case is  $N/2 = 64$ , the Nyquist frequency is exceeded and the corresponding waveform in the spatial domain is no longer a sinusoid. Figure 10 illustrates the waveform of the real part of a spatial image whose Fourier transform is an image with a non-zero pixel located at  $(p, q) = (69, 120)$ .

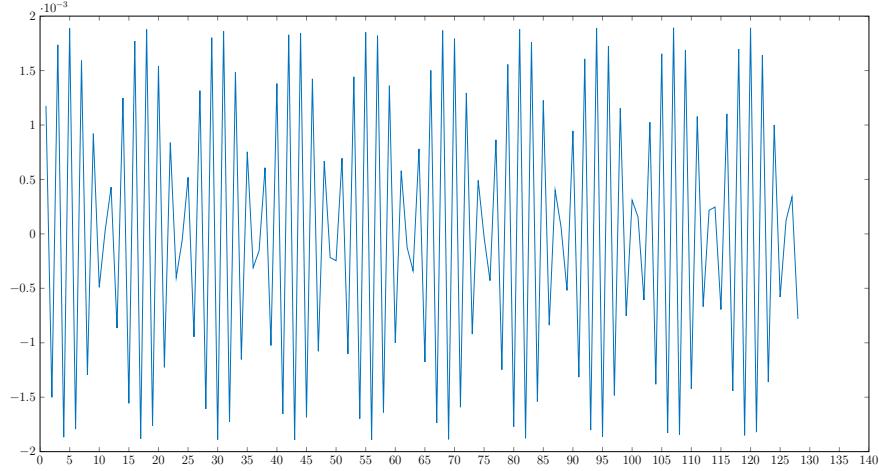


Figure 10: Example waveform in the spatial domain for  $(p, q) = (69, 120)$

## 1.6 Question 6

The purpose of these lines is to visualize the mapping of the angular frequency values  $\omega_x$  and  $\omega_y$  inside the interval

$$-\pi \leq \omega_x, \omega_y \leq \pi \quad (9)$$

The exact operation is performed by `fftshift()` function.

## 2 Section 1.4 - Linearity

Figures 11, 12 and 13 show the spatial  $F$ ,  $G$  and  $H$  images. Figures 14, 15 and 16 show their respective Fourier transforms, with the origin being in the upper left corner. Figures 17, 18 and 19 show the Fourier transforms of images  $F$ ,  $G$  and  $H$  with the origin in the middle of the image.

Here, the `fftshift()` method is used to shift the origin  $O(0, 0)$  from the upper left corner to the middle of each image. The `log` command is used to make details in the images in the frequency domain visible. Figures 20, 21 and 22 illustrate the Fourier transforms of images  $F$ ,  $G$  and  $H$  with the origin in the middle of the image without the use of the `log` command.

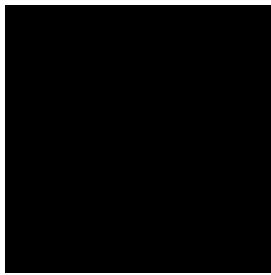


Figure 11: Image  $F$

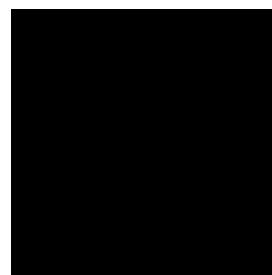


Figure 12: Image  $G = F'$



Figure 13: Image  $H = F + 2 * G$



Figure 14: Image  $\mathcal{F}(F)$ . Origin at the upper left corner.



Figure 15: Image  $\mathcal{F}(G)$ . Origin at the upper left corner.

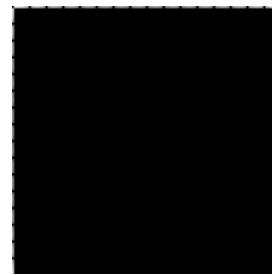


Figure 16: Image  $\mathcal{F}(H)$ . Origin at the upper left corner.

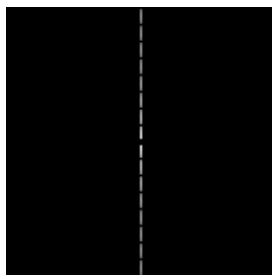


Figure 17: Image  $\mathcal{F}(F)$ . Origin at the middle.

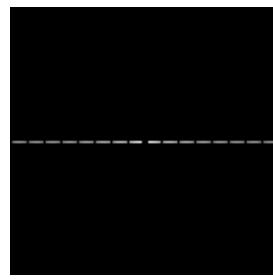


Figure 18: Image  $\mathcal{F}(G)$ . Origin at the middle.

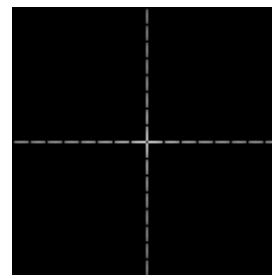


Figure 19: Image  $\mathcal{F}(H)$ . Origin at the middle.

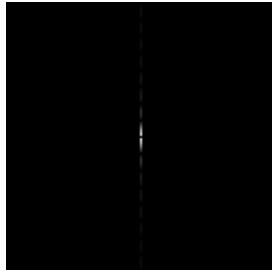


Figure 20: Image  $\mathcal{F}(F)$ . Origin at the middle. Illustration without the use of the log function.

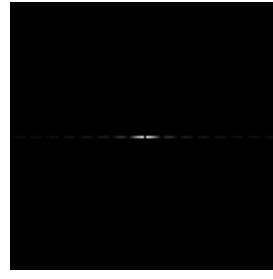


Figure 21: Image  $\mathcal{F}(G)$ . Origin at the middle. Illustration without the use of the log function.

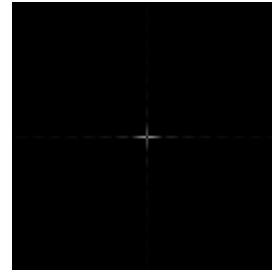


Figure 22: Image  $\mathcal{F}(H)$ . Origin at the middle. Illustration without the use of the log function.

## 2.1 Question 7

In essence, image  $F$  in the spatial domain is a two-dimensional box function:

$$F(x, y) = \begin{cases} 1, & x_1 \leq x \leq x_2 \\ 0, & \text{everywhere else} \end{cases} \quad (10)$$

Its Fourier transform is:

$$\begin{aligned}
\mathcal{F}(F(x, y)) &= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-\frac{2\pi i(xu+yv)}{N}} = \\
&\sum_{x=x_1}^{x_2} e^{-\frac{2\pi i xu}{N}} \sum_{y=0}^{N-1} e^{-\frac{2\pi i yv}{N}} = \sum_{x=x_1}^{x_2} e^{-\frac{2\pi i xu}{N}} \sum_{y=0}^{N-1} \mathbf{1} \cdot e^{-\frac{2\pi i yv}{N}} = \\
&\delta(v) \cdot \sum_{x=x_1}^{x_2} \cdot e^{-\frac{2\pi i xu}{N}} \quad (11)
\end{aligned}$$

where we exploited the identity

$$\mathcal{F}(1) = \delta(v) \quad (12)$$

Since  $\delta(v) = 1$  only where  $v = 0$ ,  $\mathcal{F}(F(x, y))$  will be non-zero only where  $v = 0$ . Hence, that is why  $F$ 's Fourier spectrum is concentrated in the left border. Similarly, the same can be derived for image  $G$ , but for transposed axes. As for  $H$ , since the Fourier transform possesses the property of linearity and  $H$  is a linear combination of  $F$  and  $G$ , its Fourier transform is the combination of those of  $F$  and  $G$ .

## 2.2 Question 8

As stated above, application of a logarithm on an image can reveal details in an image to the human eye. A logarithm transformation is an image enhancement technique used to compress the range of pixel values in an image so that bright regions are finely tuned, but darker ones are tuned coarsely so as for differences between pixels to be visible.

## 2.3 Question 9

As exhibited by image  $H$  above, its spectrum consists of the superposition of the spectra of images  $F, G$ . This is a direct result of the linearity of the Fourier transform.

In the general case, if  $N$  images are combined in a linear way, their collective spectrum will consist of the same combination of their individual spectra. In formal:

$$\mathcal{F}(a \cdot f + b \cdot g) = a \cdot \mathcal{F}(f) + b \cdot \mathcal{F}(g) \quad (13)$$

where  $a, b \in \mathbb{C}$  are constants and  $f, g$  functions of real variable(s).

# 3 Section 1.5 - Multiplication

## 3.1 Question 10

Images  $F \cdot G$  and  $\mathcal{F}(F \cdot G)$  are shown in figures 23 and 24 respectively.



Figure 23: Image  $F \ .*$   
 $G$ .

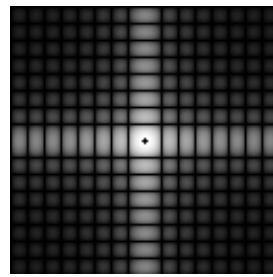


Figure 24: Image  
 $\mathcal{F}(F \ .* \ G)$ .

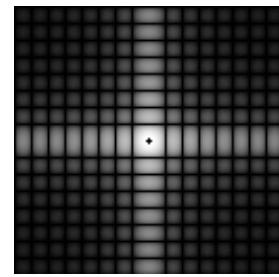


Figure 25: Image  
 $\mathcal{F}(F) * \mathcal{F}(G)$ .

Since the multiplication in either the spatial or the frequency domain is translated into a convolution in the other, the same result can be obtained by convolving the Fourier transforms of  $F$  and  $G$ . The result of this operation is illustrated in figure 25.

## 4 Section 1.6 - Scaling

### 4.1 Question 11

Image  $F$  and its Fourier transform are illustrated in figures 26 and 27 respectively.

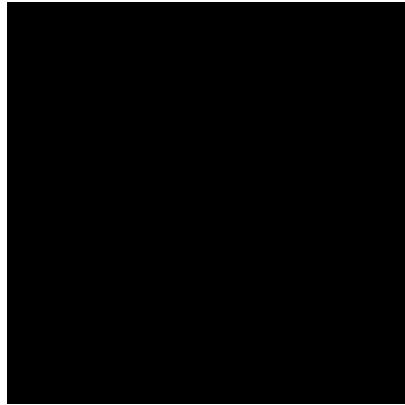


Figure 26: Image  $F$

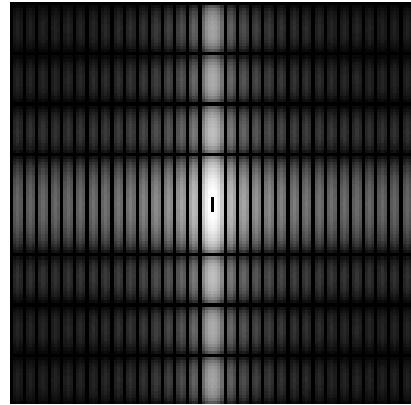


Figure 27: Image  $\mathcal{F}(F)$ .

Compared to figure 23, the height of the non-zero area in image 26 is cut in half, while its width is increased by a factor of 2. Comparing their respective Fourier transforms verifies the transform's scaling property: a compression in either the spatial or the frequency domain is expressed as an expansion in the other.

In formal manner, if  $f_1(x, y)$  is the function represented in figure 23 and  $f_2(x, y)$  the function represented in figure 26, then

$$f_2(x, y) = f_1\left(\frac{x}{2}, 2y\right)$$

hence due to the scaling property of the Fourier transform

$$f(ax, by) \iff \frac{1}{|ab|} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

$$F_2(u, v) = F_1(2u, \frac{v}{2})$$

which is why the number of spectrum zeros appear to be twice as many horizontally and only half vertically.

## 5 Section 1.7 - Rotation

### 5.1 Question 12

Figure 28 illustrates the effect of rotation of the original image on the spectra of the various images. The orientation of each spectrum follows the rotation of each image, i.e. it is rotated by the same angle and towards the same direction. However, because of the rotation, each image loses its original smoothness, due to the limited resolution and the nature of the shape of the pixels, with a degree of distortion depending on the angle of rotation. This has a direct effect on the Fourier transform of each image, as is clearly seen by distortion manifested in the wave-like patterns in the spectrum of the images rotated by 30 and 60 degrees.

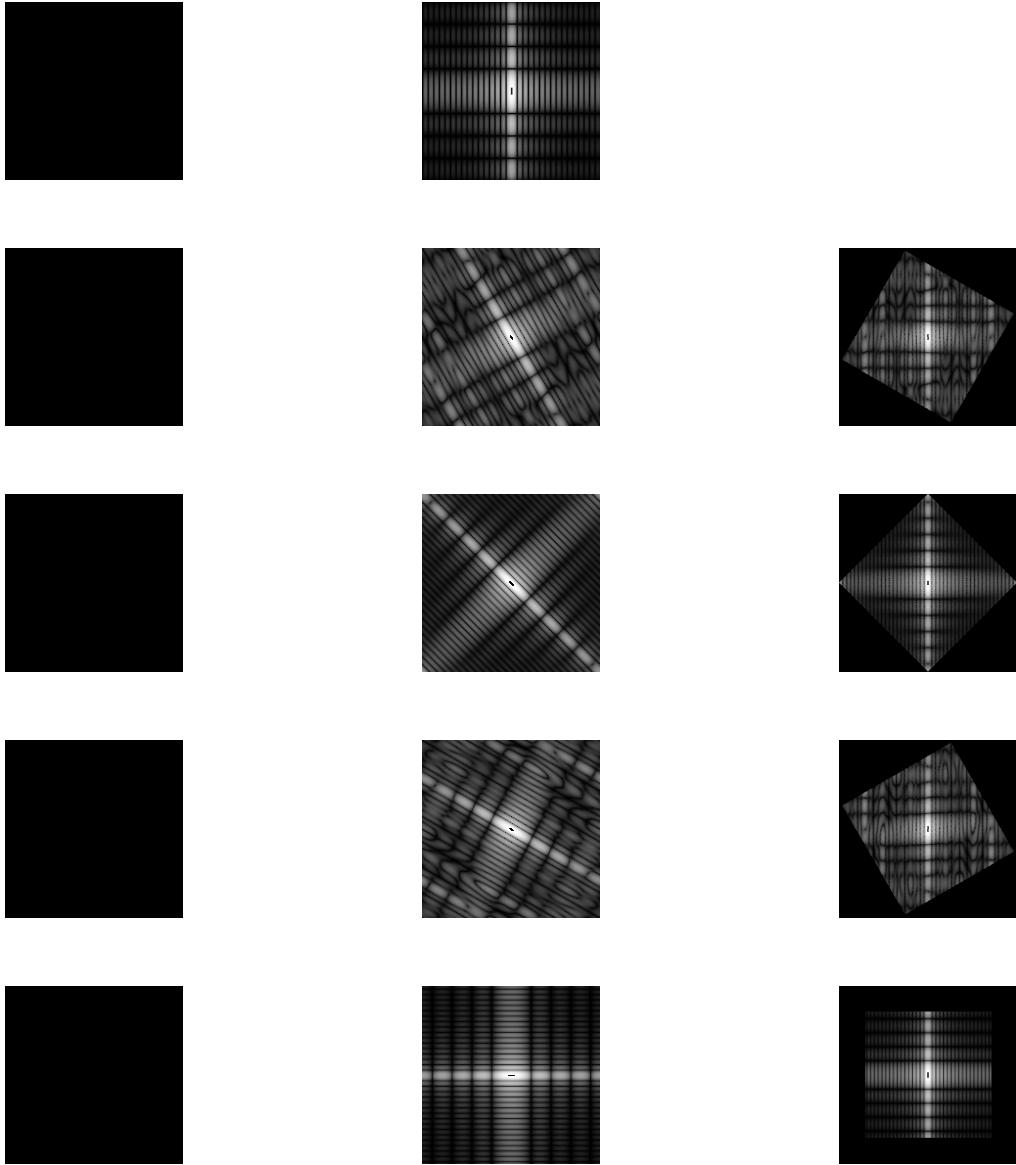


Figure 28: The first row depicts the original  $F$  image and its Fourier transform. The next images in the first column represent  $F$  rotated by a) 30, b) 45, c) 60 and d) 90 degrees respectively. The second column features their respective Fourier transforms. The third column features the Fourier spectra of the second column rotated, so as to match the orientation of the Fourier spectrum of image  $F$ .

The rotation of the image can be seen as the rotation of a sum of pixels. If a pixel with coordinates  $A(x, y)$  with respect to the center of the image is rotated by an angle  $\theta$ , then its new coordinates can be expressed as

$$A(x', y') = A(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$$

Hence,

$$x = x'\cos\theta - y'\sin\theta$$

and

$$y = x' \sin\theta + y' \cos\theta$$

If we now take the Fourier transform of the rotated image  $f(x', y')$ , then

$$\begin{aligned} \mathcal{F}(f') &= \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x', y') \cdot e^{-\frac{2\pi i(xu + yv)}{N}} = \\ &\quad \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x', y') \cdot e^{-\frac{2\pi i((x' \cos\theta - y' \sin\theta)u + (x' \sin\theta + y' \cos\theta)v)}{N}} = \\ &\quad \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x', y') \cdot e^{-\frac{2\pi i(x'(u \cos\theta + v \sin\theta) + y'(v \cos\theta - u \sin\theta))}{N}} = \\ &\quad \sum_{x'=0}^{N-1} \sum_{y'=0}^{N-1} f(x', y') \cdot e^{-\frac{2\pi i(x'(u \cos\theta + v \sin\theta) + y'(v \cos\theta - u \sin\theta))}{N}} \end{aligned} \quad (14)$$

which means that the rotation is propagated to the frequency domain:

$$u' = u \cos\theta + v \sin\theta$$

and

$$v' = v \cos\theta - u \sin\theta$$

## 6 Section 1.8 - Information in Fourier phase and magnitude

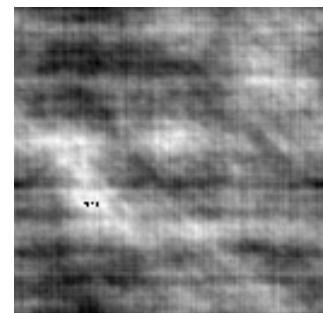
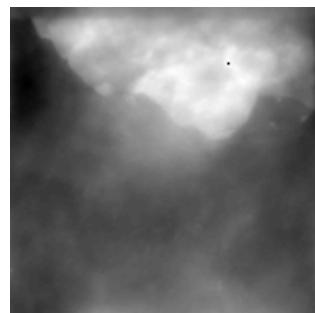
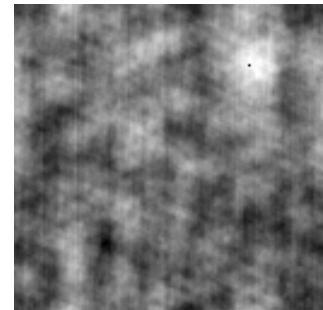
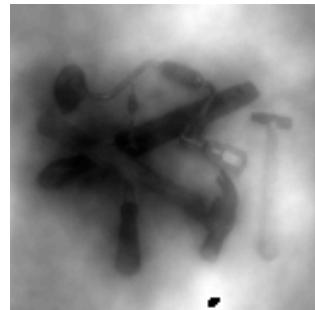
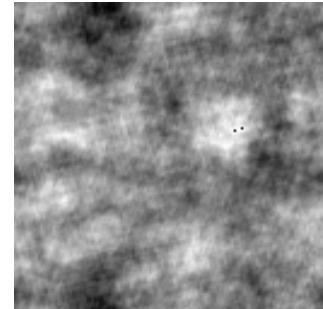
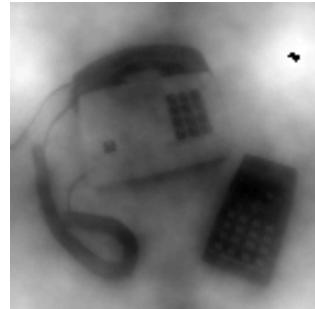


Figure 29: The first column features the original images `phone128`, `few128`, `nallo128`. Their respective power spectrum transformation `pow2image` and phase transformation `randphaseimage` are in the second and third columns respectively.

### 6.1 Question 13

Let us consider the one dimension, continuous case. A real signal can be expressed in the form of its inverse Fourier transform:

$$x(t) = \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega = \int_{-\infty}^{\infty} |X(\omega)| e^{j\phi} e^{j\omega t} d\omega = \int_{-\infty}^{\infty} |X(\omega)| e^{j(\omega t + \phi)} d\omega \quad (15)$$

where  $X(\omega) = |X(\omega)|e^{j\phi}$ , with  $|X(\omega)|$  being the amplitude of the signal in the frequency domain, and  $\phi$  its phase. Equation 15 describes that the amplitude  $|X(\omega)|$  is the weight for the contribution of the sinusoid  $e^{j\omega t}$  to  $x(t)$ , while the  $e^{j\phi}$  component determines the phase of this sinusoid with respect to other sinusoids in  $x(t)$ .

As seen in figure 29 the phase component is responsible for the information that humans can relate to, since the second column features images whose phase is the same as those of the original images. In contrast, the third column where the amplitude has been kept the same as that in the first column has no immediate visual information visible.

## 7 Section 2 - Gaussian convolution implemented via FFT

### 7.1 Question 14

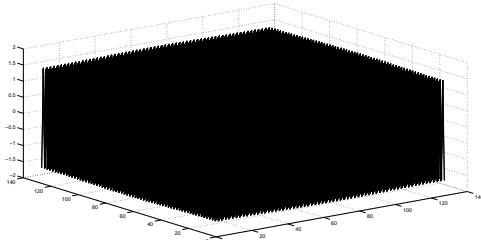


Figure 30: Impulse response of the discretized  $g(m, n; t)$  for  $t = 0.1$ .

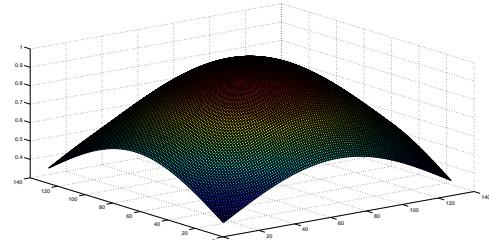


Figure 31: Impulse response of the continuous  $g(x, y; t)$  for  $t = 0.1$ .

The covariance matrices in this case are:

$$var_D = \begin{bmatrix} 0.0133 & 0 \\ 0 & 0.0133 \end{bmatrix}$$

$$var_C = \begin{bmatrix} 9.8124 & 0 \\ 0 & 9.8124 \end{bmatrix}$$

As we can see the variances appear to be approximately ten times less and ten times more than in the ideal case.

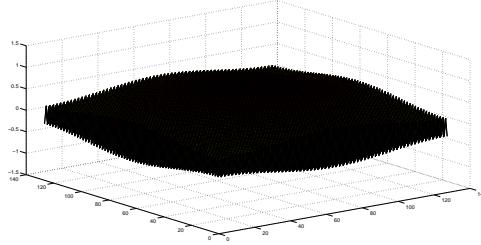


Figure 32: Impulse response of the discretized  $g(m, n; t)$  for  $t = 0.3$ .

The covariance matrices in this case are:

$$var_D = \begin{bmatrix} 0.2811 & 0 \\ 0 & 0.2811 \end{bmatrix}$$

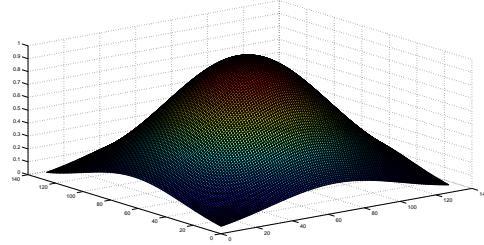


Figure 33: Impulse response of the continuous  $g(x, y; t)$  for  $t = 0.3$ .

$$var_C = \begin{bmatrix} 11.0845 & 0 \\ 0 & 11.0845124 \end{bmatrix}$$

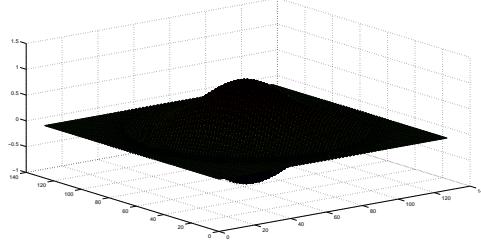


Figure 34: Impulse response of the discretized  $g(m, n; t)$  for  $t = 1$ .

The covariance matrices in this case are:

$$var_D = \begin{bmatrix} 1.0 & 0 \\ 0 & 1.0 \end{bmatrix}$$

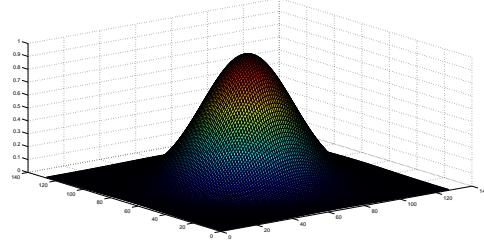


Figure 35: Impulse response of the continuous  $g(x, y; t)$  for  $t = 1$ .

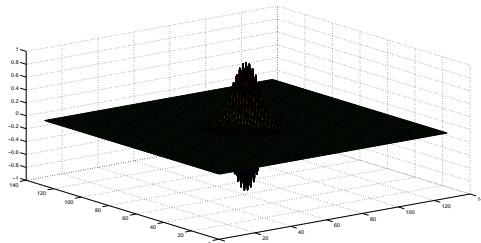


Figure 36: Impulse response of the discretized  $g(m, n; t)$  for  $t = 10$ .

The covariance matrices in this case are:

$$var_C = \begin{bmatrix} 2.1898 & 0 \\ 0 & 2.1898 \end{bmatrix}$$

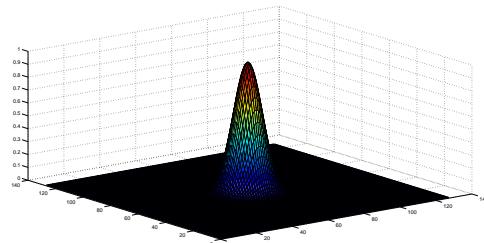


Figure 37: Impulse response of the continuous  $g(x, y; t)$  for  $t = 10$ .

$$var_D = \begin{bmatrix} 10.0 & 0 \\ 0 & 10.0 \end{bmatrix}$$

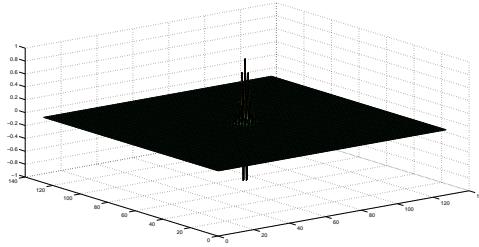


Figure 38: Impulse response of the discretized  $g(m, n; t)$  for  $t = 100$ .

$$var_C = \begin{bmatrix} 10.0 & 0 \\ 0 & 10.0 \end{bmatrix}$$

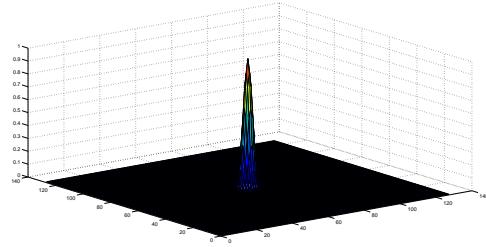


Figure 39: Impulse response of the continuous  $g(x, y; t)$  for  $t = 100$ .

The covariance matrices in this case are:

$$var_D = \begin{bmatrix} 100.0 & 0 \\ 0 & 100.0 \end{bmatrix}$$

In the case of the discrete Gaussian kernel, its impulse response for various values of  $t$  is shown in figures 40 - 44.

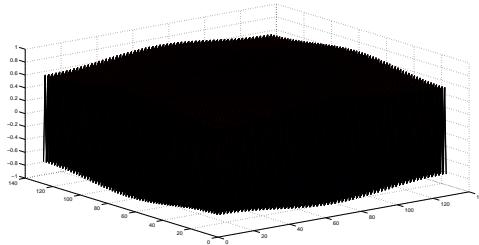


Figure 40: Impulse response of the discretized  $g(m, n; t)$  for  $t = 0.1$ , using the `discgaussfft` method.

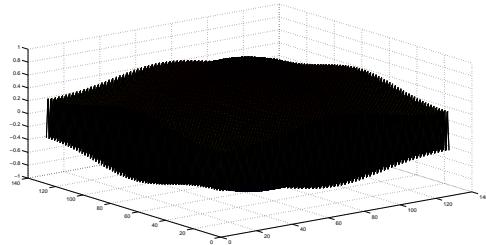


Figure 41: Impulse response of the discretized  $g(m, n; t)$  for  $t = 0.3$ , using the `discgaussfft` method.

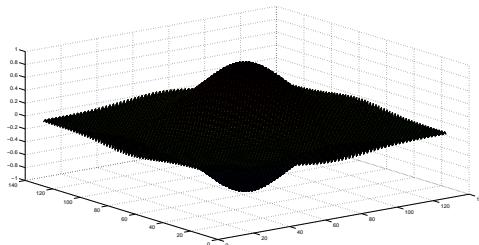


Figure 42: Impulse response of the discretized  $g(m, n; t)$  for  $t = 1$ , using the `discgaussfft` method.

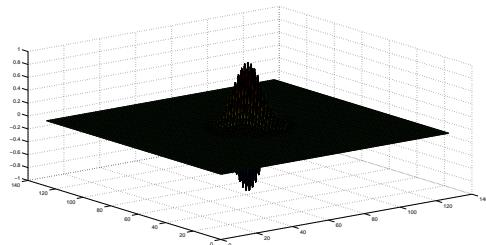


Figure 43: Impulse response of the discretized  $g(m, n; t)$  for  $t = 10$ , using the `discgaussfft` method.

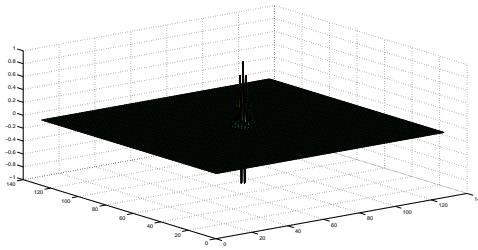


Figure 44: Impulse response of the discretized  $g(m, n; t)$  for  $t = 100$ , using the `discgaussfft` method.

As for their variance, it is indeed

$$var = t \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

## 7.2 Question 15

As seen in figures 30, 32, 34, 36 and 38, the first two surfaces do not represent a Gaussian distribution. In fact, only if the values for  $t$  are more than  $\pi/4$  they will be Gaussians. Only then can we compare their variances to the ones of the discretized Gaussian kernel.

## 7.3 Question 16

Figures 45, 46 and 47 show the original `phonecalc128`, `few128` and `nallo128` images. Figures 48 - 62 illustrate the effect of the application of a Gaussian filter for various values of  $t$ .



Figure 45: The origin `phonecalc128` image.



Figure 46: The origin `few128` image.



Figure 47: The origin `nallo128` image.



Figure 48: The result of the application of a gaussian function with  $t = 1$  to image phonecalc128.



Figure 49: The result of the application of a gaussian function with  $t = 1$  to image few128.



Figure 50: The result of the application of a gaussian function with  $t = 1$  to image nallo128.



Figure 51: The result of the application of a gaussian function with  $t = 4$  to image phonecalc128.

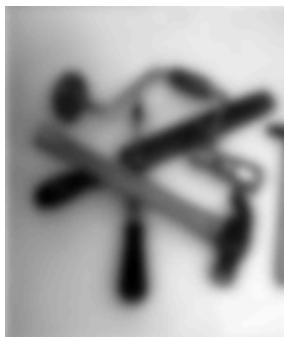


Figure 52: The result of the application of a gaussian function with  $t = 4$  to image few128.



Figure 53: The result of the application of a gaussian function with  $t = 4$  to image nallo128.

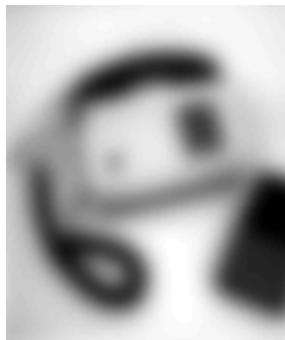


Figure 54: The result of the application of a gaussian function with  $t = 16$  to image phonecalc128.



Figure 55: The result of the application of a gaussian function with  $t = 16$  to image few128.

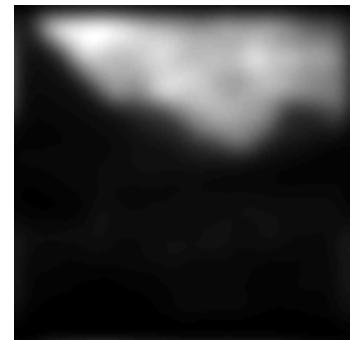


Figure 56: The result of the application of a gaussian function with  $t = 16$  to image nallo128.

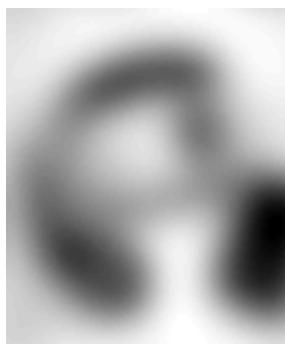


Figure 57: The result of the application of a gaussian function with  $t = 64$  to image phonecalc128.

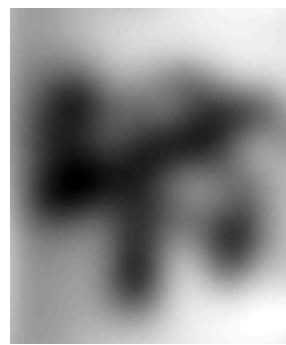


Figure 58: The result of the application of a gaussian function with  $t = 64$  to image few128.

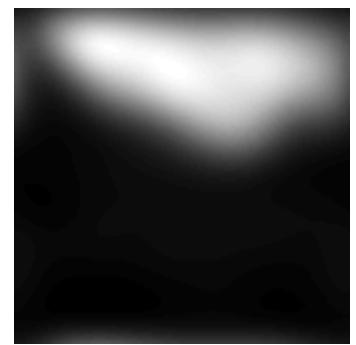


Figure 59: The result of the application of a gaussian function with  $t = 64$  to image nallo128.



Figure 60: The result of the application of a gaussian function with  $t = 256$  to image phonecalc128.

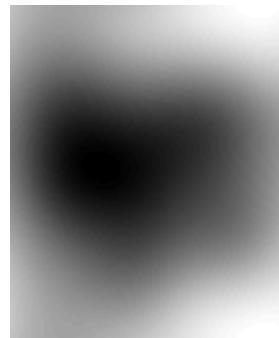


Figure 61: The result of the application of a gaussian function with  $t = 256$  to image few128.



Figure 62: The result of the application of a gaussian function with  $t = 256$  to image nallo128.

The thing to notice here is that the higher the value of  $t$ , the more blurry the output image is. This is reasonable since the higher the variance used, the smaller the cut-off frequency is as examined in the two previous questions, and the more restricted the band of frequencies is toward the lower frequencies. Hence, as  $t$  increases, the less details, that is, regions of higher frequencies, are preserved.

## 8 Section 3 - Smoothing



Figure 63: Image add, i.e. image office256 corrupted by white noise of  $\sigma = 16$ .



Figure 64: Image sap, i.e. office256 corrupted by salt-and-pepper noise.

## 8.1 Question 17

TODO

### 8.1.1 Gaussian smoothing



Figure 65: Smoothing of image add using a gaussian low pass filter of  $\sigma^2 = 0.1$ .



Figure 66: Smoothing of image add using a gaussian low pass filter of  $\sigma^2 = 1$ .



Figure 67: Smoothing of image add using a gaussian low pass filter of  $\sigma^2 = 4$ .



Figure 68: Smoothing of image add using a gaussian low pass filter of  $\sigma^2 = 10$ .

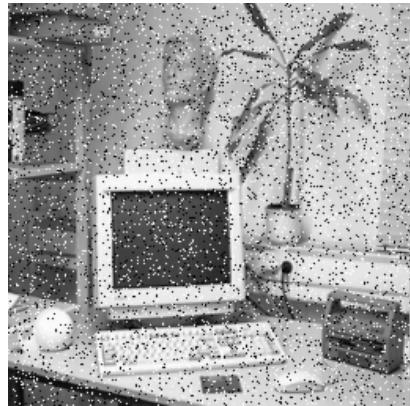


Figure 69: Smoothing of image `sap` using a gaussian low pass filter of  $\sigma^2 = 0.1$ .



Figure 70: Smoothing of image `sap` using a gaussian low pass filter of  $\sigma^2 = 1$ .



Figure 71: Smoothing of image `sap` using a gaussian low pass filter of  $\sigma^2 = 4$ .



Figure 72: Smoothing of image `sap` using a gaussian low pass filter of  $\sigma^2 = 10$ .

### 8.1.2 Median filtering



Figure 73: Smoothing of image `add` using a median filter of  $size = 1$ .



Figure 74: Smoothing of image `add` using a median filter of  $size = 4$ .



Figure 75: Smoothing of image `add` using a median filter of  $size = 10$ .



Figure 76: Smoothing of image `sap` using a median filter of  $size = 1$ .



Figure 77: Smoothing of image `sap` using a median filter of  $size = 4$ .



Figure 78: Smoothing of image `sap` using a median filter of  $size = 10$ .

### 8.1.3 Ideal low-pass filtering



Figure 79: Smoothing of image `add` using an ideal low-pass filter with cut-off frequency of 0.1 cycles per pixel.



Figure 80: Smoothing of image `add` using an ideal low-pass filter with cut-off frequency of 0.4 cycles per pixel.



Figure 81: Smoothing of image `add` using an ideal low-pass filter with cut-off frequency of 0.05 cycles per pixel.



Figure 82: Smoothing of image `sap` using an ideal low-pass filter with cut-off frequency of 0.11 cycles per pixel.



Figure 83: Smoothing of image `sap` using an ideal low-pass filter with cut-off frequency of 0.2 cycles per pixel.



Figure 84: Smoothing of image `sap` using an ideal low-pass filter with cut-off frequency of 0.05 cycles per pixel.

## 8.2 Question 18

TODO

## 9 Section 3.2 - Smoothing and subsampling

Figures 85, 86, 87 and 87 illustrate image `hand256` subsampled and smoothed using a Gaussian and an ideal low-pass filter.

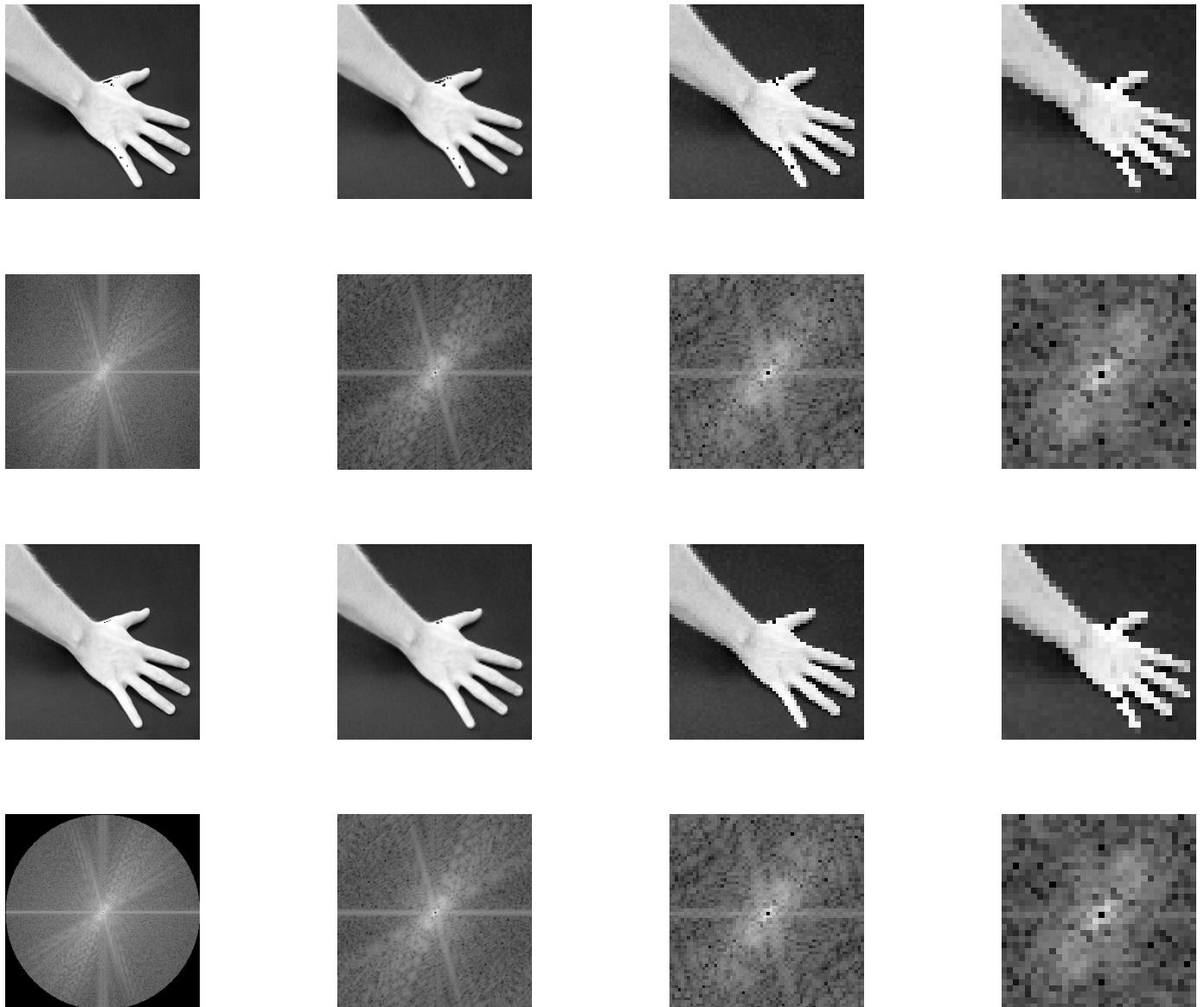


Figure 85: Image `hand256` subsampled 1, 2, 3 and 4 times are presented at row 1 respectively. Row 2 illustrates their respective Fourier transforms. Row 3 features the smoothed version of the images in row 1 using the `ideal` operator with  $CUTOFF = 0.5$ . Row 4 illustrates the Fourier transform of the smoothed images.

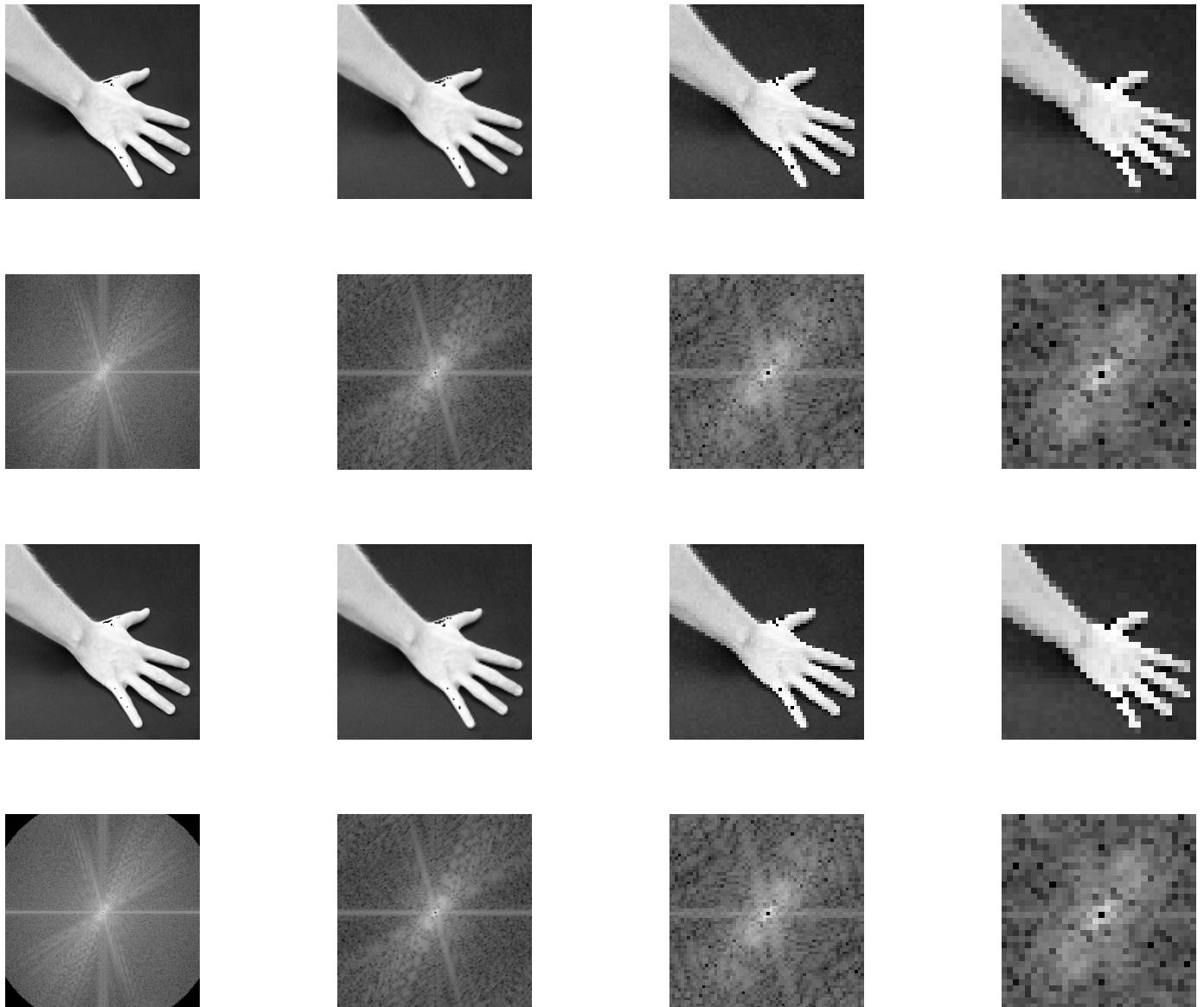


Figure 86: Image `hand256` subsampled 1, 2, 3 and 4 times are presented at row 1 respectively. Row 2 illustrates their respective Fourier transforms. Row 3 features the smoothed version of the images in row 1 using the `ideal` operator with  $CUTOFF = 0.6$ . Row 4 illustrates the Fourier transform of the smoothed images.

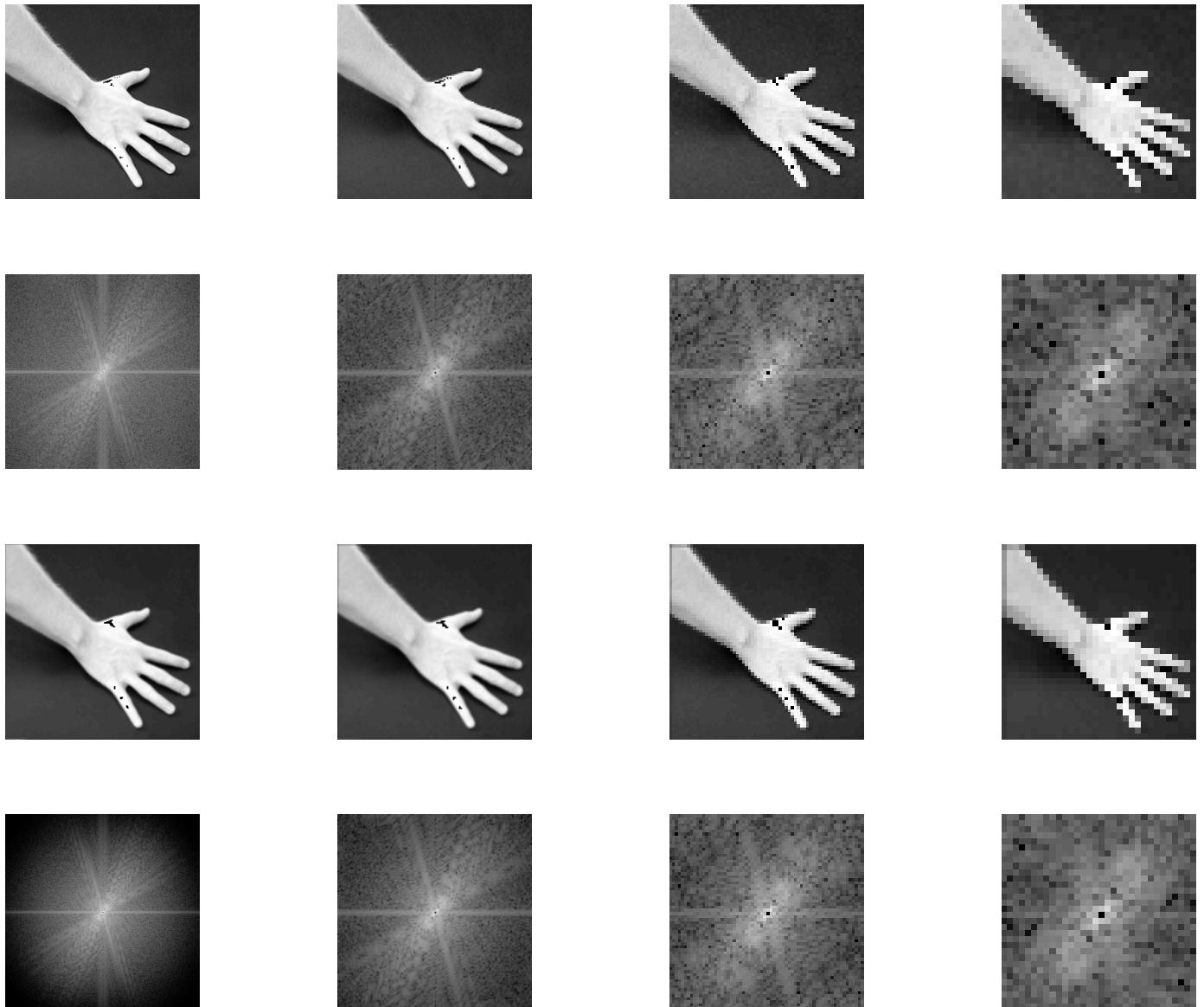


Figure 87: Image `hand256` subsampled 1, 2, 3 and 4 times are presented at row 1 respectively. Row 2 illustrates their respective Fourier transforms. Row 3 features the smoothed version of the images in row 1 using the `gaussfft` operator with  $t = 1$ . Row 4 illustrates the Fourier transform of the smoothed images.

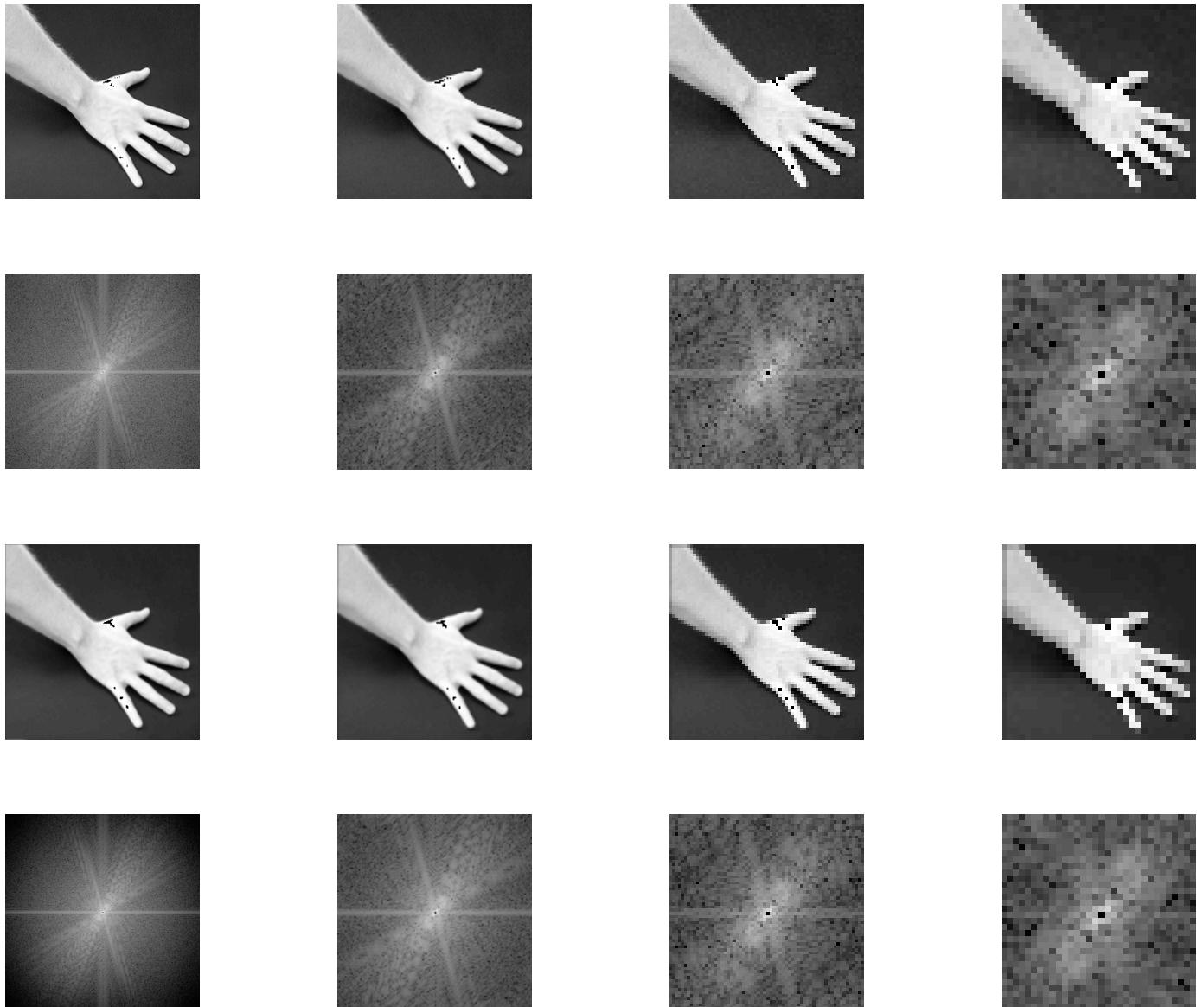


Figure 88: Image `hand256` subsampled 1, 2, 3 and 4 times are presented at row 1 respectively. Row 2 illustrates their respective Fourier transforms. Row 3 features the smoothed version of the images in row 1 using the `gaussfft` operator with  $t = 0.7$ . Row 4 illustrates the Fourier transform of the smoothed images.

## 9.1 Question 19

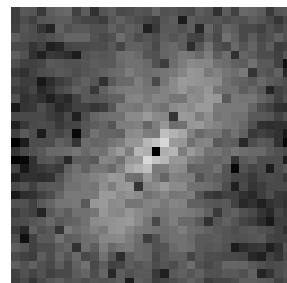
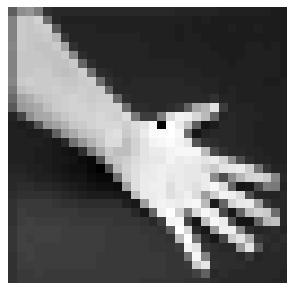
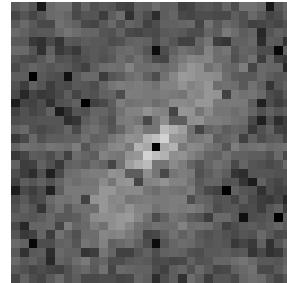
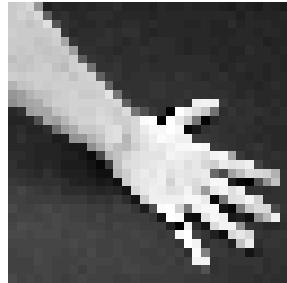


Figure 89: The upper left figure shows image hand256 subsampled 4 times. The lower left figure shows its smoothed version using a Gaussian filter with  $t = 5$ . The figures on the right-hand side illustrate their corresponding spectra.

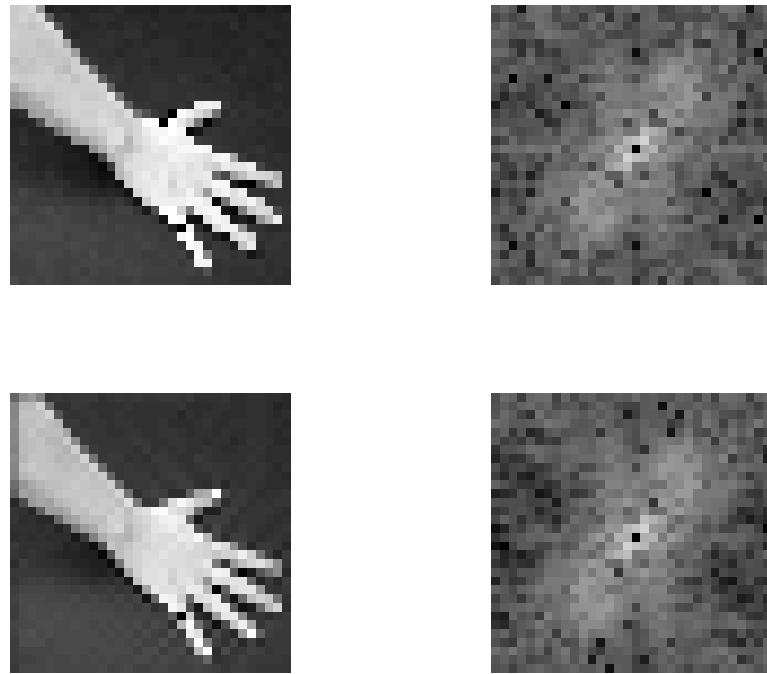


Figure 90: The upper left figure shows image `hand256` subsampled 4 times. The lower left figure shows its smoothed version using an ideal low-pass filter with  $CUTOFF = 0.1$ . The figures on the right-hand side illustrate their corresponding spectra.