Fourier Optics and Spatial Light Modulator

Report for the Advanced Lab Course

by

Yoshua Hempel

and

William Wenig

supervised by Xinxin Hu

June 2022, Berlin
Humboldt-Universität zu Berlin
Institut für Physik

Contents

Abstract

By using Fourier transforms for filtering and manipulating image characteristics through the spatial domain, Fourier optics studies optical elements. With a monochromatic laser light source with wavelength $\lambda = 523$ nm we determined the pixel pitch $p = 35(2)$ µm of a spatial light modulator and analyzed it's amplitude modulation. Afterwards the used the SLM to achieve Fourier transforms and spatial filtering with different setups as well as sucessfully performing holography through phase retrieval by deploying a Gerchberg-Saxton algorithm.

Introduction

In harmonic analysis we are concerned about decomposing arbitrary signals into their »bulding blocks« (i.e. simple waves). As these signals can also be functions of space, we can represent a spatial signal as a superposition of spatial waves. Applying this concept to an arbitrary propagating wave of light, will decompose it into plane waves [LPH22].

We can use this in Fourier Optics to perform »spatial spectral analysis«. After a plane wave hit an optical element with a certain transmittance we can analyze the outgoing beam with a lens to learn more about features of the optical element. With the help of an additional aperture we can also perform spatial filtering in which we will modify the »building blocks« of the beam. This will change the appearance of images produced by the beam.

For analyzing different transmitting optical elements and using varying apertures one would have to interchange said elements after each test. We are using a spatial light modulator to prevent unnecessary operation on the setup.

1. Characterization of the SLM

The spatial light modulator[1] we used in this lab course is the LC 2012 Spatial Light Modulator [HOL] made by the HOLOEYE Photonics AG. Before using the SLM to perform experiments in Fourier Optics, we determine the average pixel size of it and compare it to the manufacturers specification and analyze its amplitude modulation.

1.1. Determination of the pixel size

To achieve the phase modulation via enclosed liquid crystals, electrodes of certain width are built into the display of the SLM. On these electrodes light will be diffracted analogical to a two dimensional grating. The manufacturer specifies this with a fill factor of 55 % (see [HOL]).

We achieve the determination of the pixel pitch p (i.e. characteristic distance between the grating elements) by using the general formula of a diffraction grating where the grating elements are assumed to have an infinitesimal width.

$$p \sin\theta_m = m\lambda \tag{1.1}$$

Where θ_m denotes the diffraction angle of the m-th intensity maximum. The laser we used in this and all upcoming setups produces a beam of wavelength $\lambda = 532$ nm.

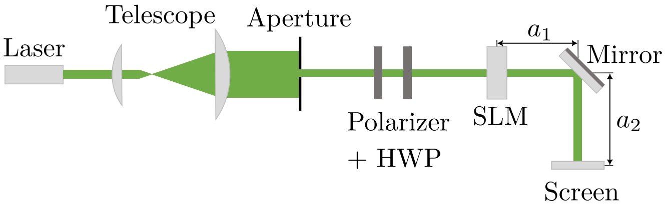


Figure 1: Experimental setup for determination of the pixel size. Derived from Figure 2 from [LPH22].

The setup we used follows the one depicted in [LPH22] with one mirror removed after the beam passes the SLM. In order to determine θ_1 we measured lengths a_1 and a_2 illustrated in Figure 1 as well as the distance $2d_m$ from the $-m$-th to the m-th maximum for $m = 1$. With these and a trigonometric argument we can modify expression (1.1) and get

$$p = m\lambda \sin^{-1}\left[\arctan\left(\frac{d_m}{a_1 + a_2}\right)\right]. \tag{1.2}$$

The lengths a_1 and a_2 were measured multiple times with a ruler of 30 cm length. The data of these lengths are scattered which, we suppose, is due to the limited accessibility of the setup that imposes human error. We are assuming the repeated measurements justify a Gaussian distribution and therefore estimate the random error to $\sigma = 2/3 \cdot s_{max}$, where s_{max} is the maximum absolute spread of the data from the mean (see [HH10] Chapter 2.3). As stated in Figure 12 of [Mül07] we then accounted for the uncertainty of half a division of the ruler e_r (see [HH10] Chapter 1.3.1) by combing σ and e_r via Pythagorean addition. For the measurement of d_1 we simply estimated the error to be $1/2 \cdot e_r$ where we omitted the horizontal measurement. The horizontal measurement did not coincide with the symmetry of the mirror and was therefore changed by the mirror.

Table 1: Measured quantities for the determination and the result for the pixel pitch p.

Quantity	a_1	a_2
Measurement	12.3(3) cm	22.5(4) cm
Quantity	d_1	θ_1
Measurement	5.3(3) mm	0.86(4)°

At last we computed the pixel pitch p via equation (1.2) and the following illustrated Gaussian error propagation.

$$u_p{}^2 = \Xi\left[\left(u_{a_1}{}^2 + u_{a_2}{}^2\right) + \frac{1}{d_1{}^2}\left(a_1 + a_2\right)^2 u_{d_1}{}^2\right] \tag{1.3}$$

$$\text{with} \quad \Xi = \left(\frac{\lambda}{d_1}\right)^2 \left(\frac{d_1{}^2}{(a_1 + a_2)^2} + 1\right)^{-1}.$$

This led us to the results shown in Table 2.

Table 2: Result of the pixel pitch p with $\bar{p}$ the one specified from the manufacturer [HOL].

Quantity	p	$\bar{p}$
Measurement	35(2) µm	36(1) µm

Due to the missing specification of the associated uncertainty in $\bar{p}$ we are going to assume an error equal to a single unit in the last figure quoted (see [HH10] Chapter 2.8.1).

---

[1] In this report further referred to as SLM

At first glance it is evident from Table 2 that our achieved result p is deficient in accuracy and in precision compared to $\bar{p}$. But nonetheless our best result for the pixel pitch lies withing the range of one σ of the pixel pitch specified by the manufacturer. That our result missed the expected value by roughly 3 % *should* be caused by systematic reasons (see [HH10] Chapter 1.4). But due to the single measurement in d_1 the result could be subject to a random offset by the estimation of the middle of the smudged maxima when measuring the distance $2d_1$. An offset in the measurement in $2d_1$ by as much as 0.5 % (which is approximately half a division on a ruler) in the neighborhood of $d_1 = 5.3(3)$ mm will result a change of p by roughly 1 %. Because our deviation from the real value is three times as big as this random influence we think that there must be systematic error that explains this deviation. In the setup in Figure 1, a_2 is depicted as the length of the path of the beam. In the experiment there was no fixture to ensure that the screen is centered relative to the beam.

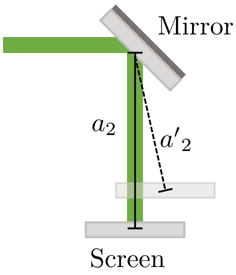


Figure 2: Illustration of not centering the screen pole to the beam. For visibility reasons the mirrors are not placed into each other

This would not be problem but due to the constraints in measuring all lengths with a ruler, instead of noting a_2 as depicted in Figure 2 we measured a'_2, the distance from the pole of the mirror to the pole of the screen. As $a'_2 > a_2$ the result for p will turn out smaller than it should be.

When thinking about how obtain a better precision we can definitely increase the amount of measurements taken to reduce the random error in all quantities. For this, one could try to do a linear fit of the form

$$d_1(s) = \mathbb{X} \cdot s \tag{1.4}$$

$$\text{with} \quad \mathbb{X} = \tan\left[\arcsin\left(\frac{\lambda}{p}\right)\right]$$

$$\text{and} \quad s = a_1 + a_2$$

in which the distance of the screen s to the SLM will be varied and subsequently $2d_1$ will be measured. For further improvement, the mirror could be removed and s can then be measured as a whole, which also makes the vertical measurement for $2d_1$ available.

1.2. AMPLITUDE MODULATION

One would think that for improved contrast and ease of deployment the amplitude modulation of the SLM is linear such that the gray value and light intensity behind the SLM are proportional to each other. But as it turns out the mapping of the applied electric field to the different gray values is not optimized in such a way and therefore has to be analyzed. For this task we added another mirror, a converging lens of focal length $f = 100$ mm and a polarizer to the setup from Figure 1. For quantifying the light intensity we used a photo diode that was connected to a DSO5054A 5000 Series oscilloscope made by Agilent Technologies, Inc.

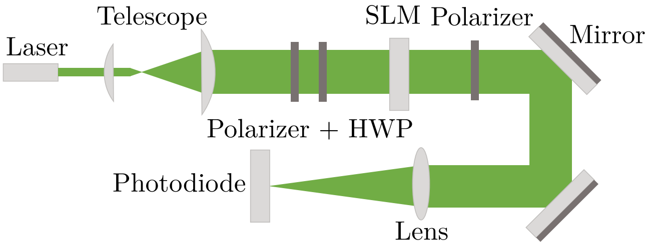


Figure 3: Experimental setup for analyzing the amplitude modulation. Derived from Figure 3 from [LPH22].

To analyze the amplitude modulation we set the SLM controller via the Lab GUI to change the gray values by looping from its minimum to its maximum. In the meantime we recorded the corresponding voltages U that are proportional to the outgoing light intensity I. We saved six series of measurements with a loop completed in each series. The gray values were set to loop at a rate of change of 60 FPS, which *should* lead to a loop duration of 4.25 s.

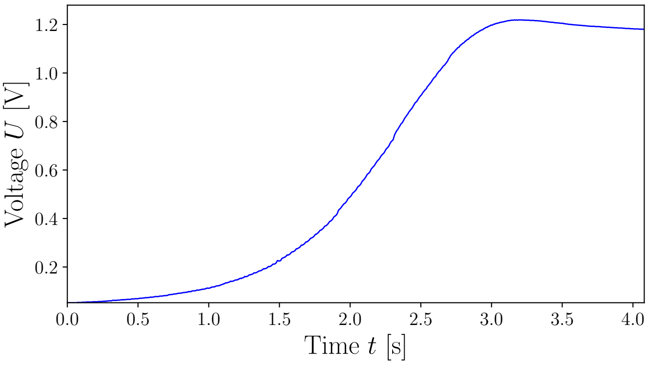


Figure 4: Voltage data acquired with the DSO5054A 5000 Series oscilloscope. Illustrated is the first measurement.

We illustrated the first of the measurement series we acquired from the oscilloscope. From Figure 4 it is visible that the loop duration $\Delta t_0 < 4.25$ s. For further investigation we took the mean of all loop durations and noted

$$\overline{\Delta t} = 4.1 \text{ s} < 4.25 \text{ s}. \tag{1.5}$$

Which implies that while looping from the minimum to the maximum gray value, roughly[2] $n = 244$ rather than $\bar{n} = 255$ steps where made by the SLM controller.

---

[2] We calculated the mean of the steps made for all measurement series.

Assuming that the missing gray value steps lie in the end of the loop[3], we mapped the voltages to the gray values by dividing the time axis into n steps.

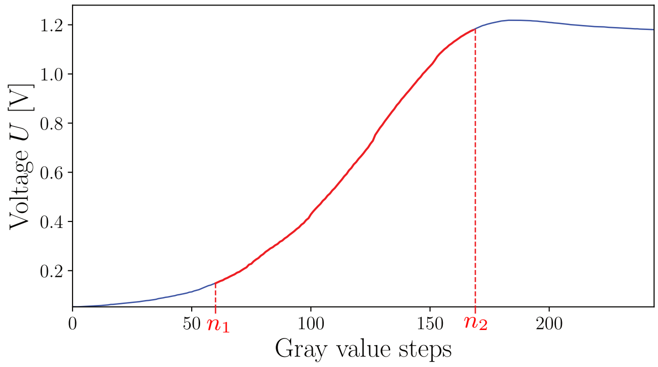


Figure 5: Voltage data mapped to gray values for data from Figure 4. $n_1 = 60$ and $n_2 = 169$ denote the points for the new gray scale on the SLM for this series.

When talking about contrast it is important to note that the rate of change of light intensity for subsequent gray values matters. We can therefore see from Figure 5 that gray values below n_1 and above n_2 will not contribute as much to the contrast as gray values that are between n_1 and n_2. Hence we should use n_1 and n_2 as new boundaries for the whole gray scale to achieve a better contrast, when displaying images with the SLM.

Because of the amplitude modulation profile we suspect, that the missing steps, caused by the shorter loop duration, in the end of the loop do not contribute to a significant loss in contrast of the picture.

The n_1 and n_2 from Figure 5 are gray scale values where the numeric derivative of U amounts to one third of its peak value. The mean of n_1 and n_2 were calculated for all measurement series.

Table 3: Results of the average new scale boundaries over all measurement series $\overline{n_i}$ and at experiment estimated $\widetilde{n}_i$

Quantity	$\overline{n_1}$	$\overline{n_2}$	$\widetilde{n}_1$	$\widetilde{n}_2$
Measurement	60.2	169.2	80	175

When performing the experiment we also noted new scale values $\widetilde{n}_i$ to improve the contrast in the pictures we used. These are also noted in Table 3. The estimated value $\widetilde{n}_2$ is coming much closer to the numerically calculated value $\overline{n_2}$ then the other pair does. We suspect that is due to the fact that for gray scale values bigger than n_2 the behavior of the light intensity approximates a constant much better than the behavior of U for steps smaller than n_1. This made it much harder for us to determine at which point $\widetilde{n}_1$ the intensity curve started to increase more rapidly.

1.3. THE BEHAVIOR OF THE OUTGOING LIGHT

For the explanation why the outgoing light behind the SLM is performing a change in its intensity we have to look at the setup in Figure 3. Because of the polarizer and the half wave plate in front of the SLM, the polarization of the beam is mostly linear when it hits the SLM. Figure 1 in [LPH22] depicts the design of the SLM we used in our experiment except that our SLM is missing the P_2 layer. In our lab we installed another linear polarizer behind the SLM which will act as the layer P_2.Inside the SLM when there is no electric field applied to the liquid crystal layer G by E_1 and E_2, the liquid crystal layer will form a helical structure and because of that, turn the polarization of the light beam by $\pi/2$. The changed polarization will now result in a complete absorption of the light in the following polarizer behind the SLM.

Applying more and more voltage to the layer G will result in a change of orientation of the molecules that formed the helical structure. Under the twisted nematic field effect, the molecules will align with the electric field produced by layers E_1 and E_2. Because the change in polarization fades, when getting closer to the saturated arrangement of the molecules, the polarizer behind the SLM will not absorb as much light as before. This results in a rising intensity of the outgoing beam.

2. SLM AS AMPLITUDE MODULATOR

As discussed in the previous section, we acquired $\widetilde{n}_1$ and $\widetilde{n}_2$ in the experiment, shown in Table 3, to improve the contrast performance of the SLM. In order to use the SLM for displaying images we entered the new boundary values into the Lab GUI and exchanged the photo diode in Figure 3 with a camera.

Placing the camera into the focal plane of the lens, we were successful in acquiring sharp images of pictures we loaded into the Lab GUI.

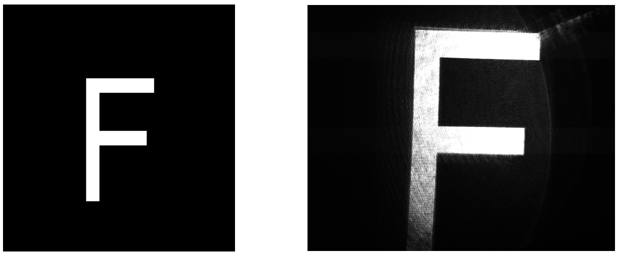


Figure 6: The letter **F** (left) and the acquired picture displayed in the image plane (right).

However as it is visible in Figure 6 the image is not displayed perfectly. Despite broadening up the beam (see Figure 3) with a telescope the beam was not of sufficient diameter to hit every pixel of the SLM. It rather hits a small region in the center which will be displayed in the image plane.

---

[3] We do so because the voltages at supposed gray value zero are close to vanishing.

Another good illustration of the apparent crop is the attempt to display the seal of the Humboldt University.

Figure 7: HU Seal (left) and the acquired picture displayed in the image plane (right).

On the right in Figure 7 we can see that the curved text around the seal disappeared.

Because the optical table was not long enough, the light beam had to be redirected by two mirrors. This design decision now shows when looking at the right image in Figure 7. Even though the image is as sharp as we could get it to be, it is washed-out to the right due to the impurities on the two mirrors.

2.1. WHY DO WE NEED A LENS?

Because our beam of light is collimated, the lens we used in Figure 3 is necessary to achieve a focused image. To demonstrate this we acquired a picture in the image plane of the letter **F** from Figure 6 without a lens.

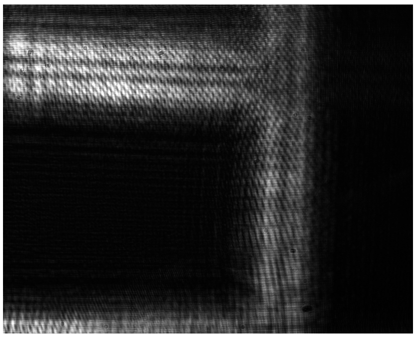


Figure 8: Letter **F** displayed in image plane without a lens.

Clearly the image we acquired in Figure 8 is shaped by the dominating effects of diffraction and impurities. Moreover we can see that the letter does not have the right orientation. Because the setup is designed with the lens in mind, the SLM will display **ꟻ** rather than **F**, so when the lens is deployed, the images captured by the camera have the right orientation.

We also want to illustrate how the image transforms when it is not exactly in focus. For which we captured images in front of and behind the image plane.

As we moved the camera away from the image plane, we noted that the image decomposes into the diffraction pattern that we noted when performing the measurements in Section 1.1. In Figure 9 we can see the **F** splitting up into the higher diffraction orders and consequently overlapping with each other.

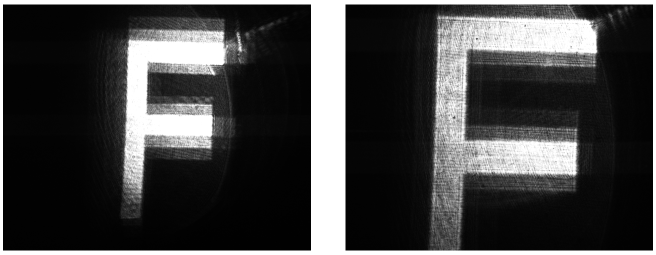


Figure 9: Letter **F** out of focus at a distance smaller (left) and bigger (right) than f.

2.2. FINDING THE IMAGE PLANE

As described in Section 4 in [LPH22] we used the thin lens equation

$$\frac{1}{f} = \frac{1}{o} + \frac{1}{i} \tag{2.1}$$

with the focal length of the used lens f and the distance o from the SLM to the lens, to determine the position i of the image plane.

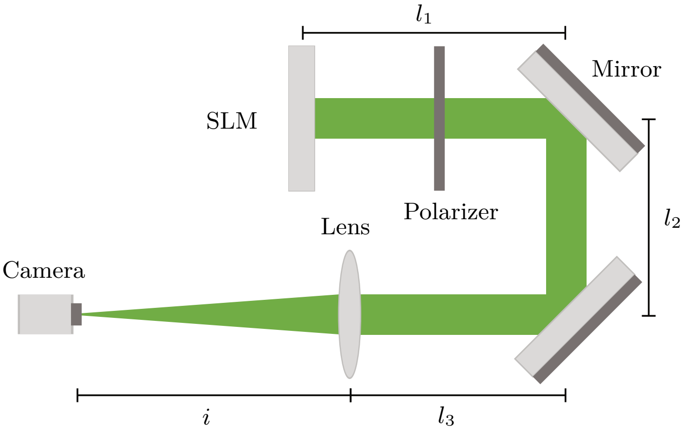


Figure 10: Cutout of the setup in Figure 3. Lengths l_i form the distance o from SLM to Lens.

To get the distance between SLM and lens, we measured the lengths l_i as depicted in Figure 10. We measured the l_i only once because the obtained image distance i should give us a rough orientation where to look for the focus.

Table 4: Measured quantities from Figure 10

Quantity	l_1	l_2	l_3
Measurement	11.8(1) cm	9.8(1) cm	9.4(1) cm

We determined to image distance to be $i \approx 14.8(2)$ cm. This coincides with the image distance $\widetilde{i} \approx 14.7(1)$ cm we measured after we adjusted the position of the camera for optimal focus.

2.3. IMPROVEMENTS ON THE SETUP

One thing to mention is that Figure 3 and Figure 10 depict an ideal setup for the amplitude modulation and imaging. In the real setup it was apparent that the beam did not take a path that was as orthogonal as illustrated in the Figures we used. This is due to the fact that the mirrors in this setup did not undergo a fine readjustment to ensure that the beam leaves the second mirror parallel to the incoming beam of the first mirror. We suspect that performing this readjustment can save time later on when positioning the camera. Additionally a guard for varying the distance between lens and camera will definitely save time because then only the distance rather than distance and angle of the camera has to be adjusted for a good focus.

Another possibility is performing the experiment without the mirrors and coming back to a linear setup. The impurities on the mirrors as well as the complicated adjustment cause the captured images to have a rather poor quality.

3. FOURIER-OPTICS

In the following we will explain how the SLM can be utilized to create an optical Fourier-transformation of an image. Firstly we'll be going into the mathematical and physical reasoning behind why this is possible and then presenting the results we achieved during our time in the lab and what we learned from them.

3.1. MATHEMATICAL FOUNDATION

The quintessential effect that makes an SLM so useful for Fourier transforms is the fact that transmission of light through an optical element with a harmonic transmittance necessitates a diffraction angle, that depends on the transmission function. In our case the transmission function maps the position of each pixel to the amplitude of the light behind each pixel. So in order to understand the SLM's role in the creation of Fourier-transforms, we must first understand why this happens and what the relationship between transmission function and diffraction angle is.

Let's assume that the complex amplitude transmittance $f : \mathbb{R}^2 \to \mathbb{C}$ of our optical element is given as depicted in Figure 6 of [LPH22] with $f(x, y) = \exp(-i2\pi\nu_x x)$. We want to find out what happens in the case of a plane wave $U(z) = U_0 \exp(i2\pi\frac{z}{\lambda})$ hitting an optical element with such a transmission function.

The light gets diffracted into plane waves of many different angles immediately behind the grating, however only countably many interfere with each other constructively. This constructive interference happens when two parallel waves, separated by one period of the grating, have the same phase relative to each other. Because the wavelength remains unchanged, the first order of constructive interference happens if the angle fulfills

$$l = \frac{\sin\theta_x}{\nu_x} \overset{!}{=} \lambda \quad \Leftrightarrow \quad \theta_x = \arcsin(\lambda\nu_x) \tag{3.1}$$

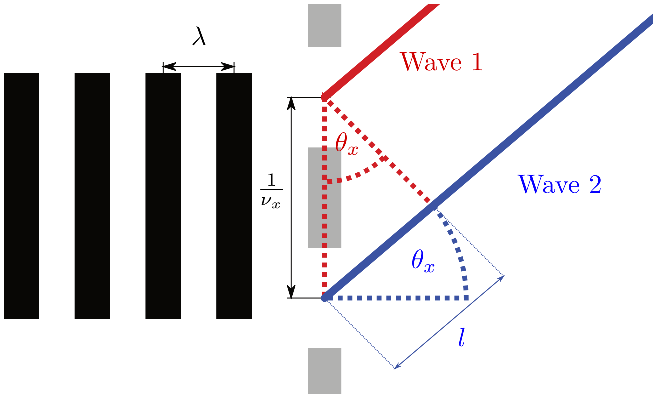


Figure 11: Illustration of diffraction angles

as seen geometrically with Figure 11. Of course the symmetry of this situation causes another first order diffraction with at mirrored angle $-\theta_x$.

The other effect we utilize to create optical Fourier transforms is the relationship between the angle of incoming light and the point it is focused onto in the focal plane of a lens.

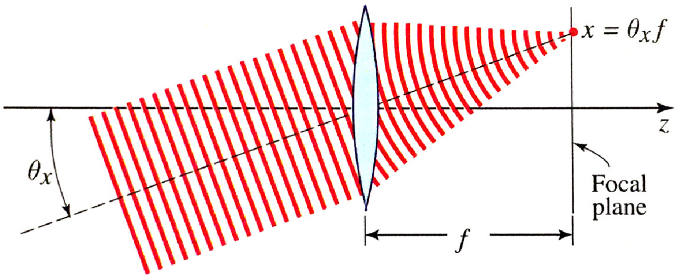


Figure 12: Graphic showing the focusing of plane waves using a lens from [LPH22].

As is shown in Figure 12, a lens with focal length f focuses a plane wave, incident at an angle θ_x, onto a single point x. This point has a distance to the lens' optical axis of $x = \theta_x f$ in paraxial approximation.

Combining both effects, if we modulate a plane wave of light with a wavelength λ and a complex transmission function $f(x, y) = \exp(-i2\pi(\nu_x x + \nu_y y))$, followed by a lens of focal length f, we get two bright spots on the focal plane. One at $P_1 = (\lambda\nu_x f, \lambda\nu_y f)$ and one at $P_2 = (-\lambda\nu_x f, -\lambda\nu_y f)$.

Now if our transmission function is no longer of this simple form, we know that the amplitude in the focal plane at (x, y) has to be proportional to its Fourier transform at the frequencies $(\nu_x, \nu_y) = (\frac{x}{\lambda f}, \frac{y}{\lambda f})$. This is because the Fourier transform of the image at these points describes the density and phase of plane waves with these frequencies that it consists of. If laser light now gets diffracted through this transmission function, we can view it as a superposition of diffractions for all the plane waves it consists of, scaled by the Fourier transform at those frequencies. For each of these diffractions the scaling carries over to the diffracted light and to the intensity in the focal plane.

3.2. Imaging the Fourier Plane

We will now go over some of the images we loaded onto the SLM, how we would expect the resulting data to look like and how it turned out in the actual experiment.

To capture the Fourier transforms of the loaded images we used the setup from Section 2 but now instead of placing the camera into the image plane (Figure 10), we put it into the *focal* plane of the converging lens.

Going by that alone didn't result in a satisfying image however, so going from the focal length we moved the camera backwards or forwards until the image was as focused as possible. One thing we observed by doing this, was that when moving away from the focal plane, one could see the original image emerging out of its Fourier transform. This results in images similar to the ones we encountered in Figure 9.

For the purpose of this experiment we generated a few images in `Python`, which we thought were well suited to showcase the functionality of our optical Fourier transformation. We created a series of images of a slit with varying thickness b. Since these are constant in the y-direction, we expected to only see a resulting image on the x-axis in the focal plane. Specifically we would expect the intensity to have the following form.

$$I(x) \propto b^2 \mathrm{sinc}^2\left(b\pi\nu_x\right) = b^2 \mathrm{sinc}^2\left(b\pi\frac{x}{\lambda f}\right) \tag{3.2}$$

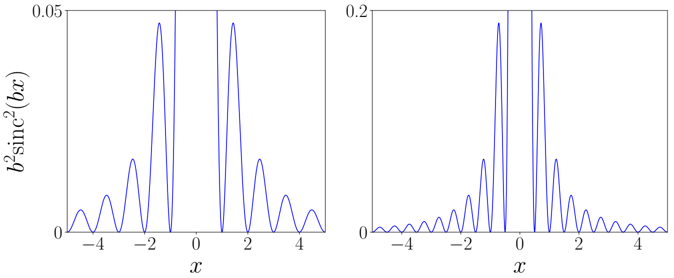


Figure 13: Expected shape of the intensity distribution with $b = 1$ (left) and $b = 2$ (right).

We would expect to see a comparatively very bright maximum in the center, followed by equally spaced maxima getting smaller at a rate of $\left(\frac{b\pi x}{\lambda f}\right)^{-2}$. By increasing the size of the slit, the maxima should move closer together and become more intense. This makes sense from the perspective of the Fourier transform, as smaller structures require waves of higher frequencies in the Fourier plane, since plane waves of higher frequencies themselves exhibit a finer structure. Of course the slits are not actually infinitely long on the y-axis, as equation 3.2 would suggest, which should cause the same relationship between intensity and the position on the y-axis as well.

$$I(x, y) \propto b^2 \mathrm{sinc}^2\left(b\pi\frac{x}{\lambda f}\right) \cdot h^2 \mathrm{sinc}^2\left(h\pi\frac{y}{\lambda f}\right) \tag{3.3}$$

Here h is the height of the image, which is 768 pixels without including the pixel pitch. Since the height of the image is way larger than the width of the slits though, this contribution will not be visible as the bright maxima are accumulated around the x-axis and get very small very quickly. This makes sense as in the limit of $h \to \infty$ the contribution of the y-axis should become a δ-distribution.

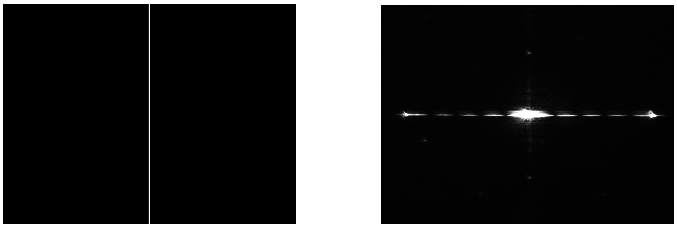


Figure 14: Input image of a slit with width $b = 5$ pixels (left) and its Fourier transform (right)

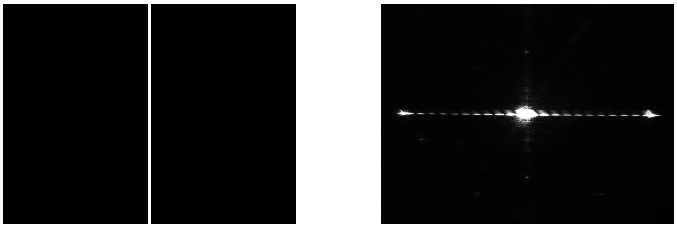


Figure 15: Input image of a slit with width $b = 11$ pixels (left) and its Fourier transform (right)

These two examples of relatively thin slits coincide nicely with our expectations. The resulting image shows a bright spot in the middle and roughly equally spaced maxima on the x-axis, losing intensity as they move further away from the middle. Close to the left and right borders one can see two further bright spots, which are most likely caused by the image being diffracted due to the space between the pixels.

As the slits get wider and the ratio between the height of the image and the width of the slit gets closer to one, the finer structure of the Fourier transform gets harder to see.

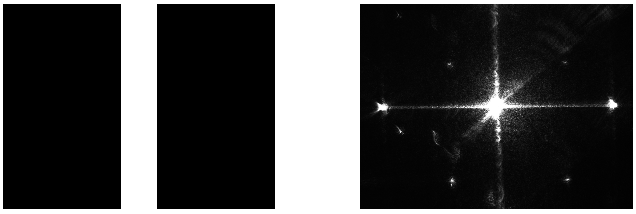


Figure 16: Input image of a slit with width $b = 135$ pixels (left) and its Fourier transform (right)

More light gets through while at the same time the maxima get closer together, causing the image to become blurred. At the same time the contribution of the y-axis gets more visible. With the higher overall intensity the imperfections of the instruments and experimental setup became more apparent as well, as there are many structures in the Fourier transform in Figure 16, which are theoretically impossible as they contradict the symmetries of mirroring around the x- and y-axis.

Now going to a slightly more complicated image, we generated disks of varying radii r at the center. We expect their Fourier transforms to be radially symmetric, because this is also true for the images themselves. We calculated their Fourier transforms numerically using the function `scipy.fft.fft2` from the Python package `scipy`. In order to increase the contrast and brightness of the numerically calculated Fourier transforms we applied an arctan function to these images. This was done to make its structures, which are further away from the center, more visible.


(a) original image

(b) absolute value of numerical `fft`


(c) absolute value of numerical `fft` after applying arctan

(d) spacial FT

Figure 17: FT of disks with radius $r = 20$ pixels.


(a) original image

(b) absolute value of numerical `fft`


(c) absolute value of numerical `fft` after applying arctan

(d) spacial FT

Figure 18: FT of disks with radius $r = 40$ pixels.

The Fourier transforms once again have a bright maximum at the center, with seemingly equally spaced maxima of decreasing intensity going out from there. As the radii of the circles become larger, the maxima get closer together and become thinner. This is quite similar to the FTs of the slits, except that these higher order maxima are not anymore seen as spots but rather as concentric circles around the center, which preserves the radial symmetry. However when looking more closely at the images, symmetry breaking structures can be observed. Interestingly some of these differ between the optically and the numerically obtained FT. One which is observable in both cases is that the width of the concentric circles appears to modulate with changing polar angle φ. This dependence on φ becomes stronger the farther away from the center of the image you go.

A likely explanation is the fact that the disks do not have perfect circular boundaries as their shape is limited by the resolution of the image. This is to say that the distance of the pixels on the boundary to the center of the disk is not constant, as it should be for an ideal disk.


Figure 19: Zoomed in pictures of the diks with radius of 20 pixels (left) and 40 pixels (right)

As this imperfection of the disks is a discrepancy in its finer structure, it is encoded in the higher frequencies of its FT, which could explain the breaking of radial symmetry becoming greater the farther away from the center to go. In other words the breaking of radial symmetry in the smaller details of the disk causes a breaking of radial symmetry in the higher frequencies of its FT. As the resolution of the disks becomes larger, this effect becomes smaller, as the disks approach the shape of an ideal one. As a result of this, it takes higher orders of concentric circles for the disk of radius 40 pixels until a dependence on φ of the FT becomes more pronounced than for the disk of radius 20 pixels.

In the optically obtained FTs one can also see unexpected contributions on the x- and y-axes, which are not present in the numerically calculated ones. A fundamental difference between both methods which could possibly explain this, is that while the SLM's pixels are of a square shape, the `scipy.fft.fft2` algorithm views the image's pixels as points. It then does a discrete FTbased on these points. Because of this, diagonals and horizontal or vertical lines in the input picture aren't fundamentally different using the numerical method, whereas in the SLM diagonals consist of steps of horizontal or vertical lines. As we've seen with the slits, such horizontal and vertical lines mainly have contributions on the x- and y-axes.

3.3. SPACIAL FILTERING

For the next part of the lab we now placed pre-manufactured, pictures in front of a converging lens in whose focal plane we placed the SLM. So the SLM was now modulating the light in the Fourier plane. This was followed by another lens to reverse Fourier transform the signal, which was then again lead into the camera.

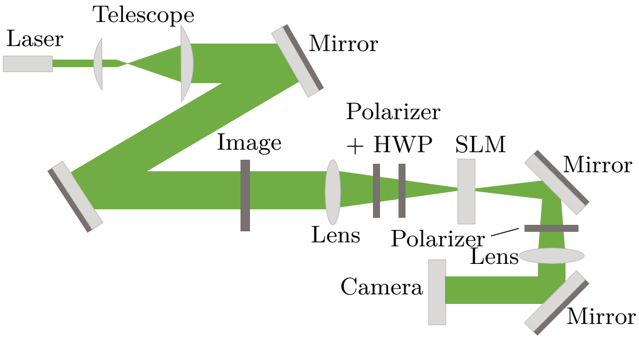
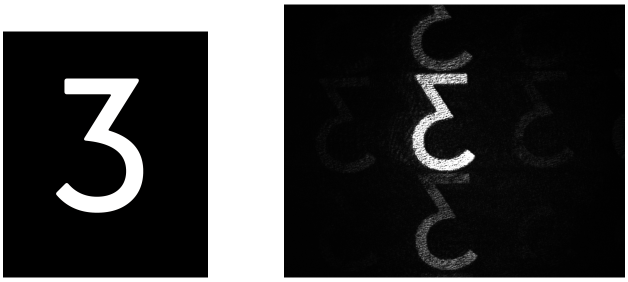


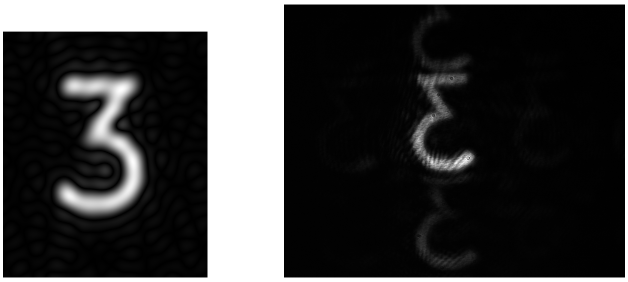
Figure 20: Experimental setup for spatial filtering derived from Figure 12 in [LPH22].

With this setup we were now able to filter out frequencies of the signal by displaying images in the SLM. Specifically we used images of vertical and horizontal slits, either to filter out or limit to frequencies near the ν_x- and ν_y-axes, as well as of disks to cut off or to limit frequencies up to a certain size. In order to compare the results we obtained, to what would be possible in an ideal scenario, we again also implemented this mask filtering numerically, by using `scipy.fft.fft2` applying the mask over the image and then transforming back using `scipy.fft.ifft`.



(a) Input Image

(b) Displayed image without filter



(c) Numerically applied circular low-pass filter

(d) Optically filtered image with circular low-pass filter of radius 5 pixels

Figure 21: Applying a circular low-pass filter to the image of the 3

(a) Numerically applied circular high-pass filter

(b) Optically filtered image with circular high-pass filter of radius 50 pixels

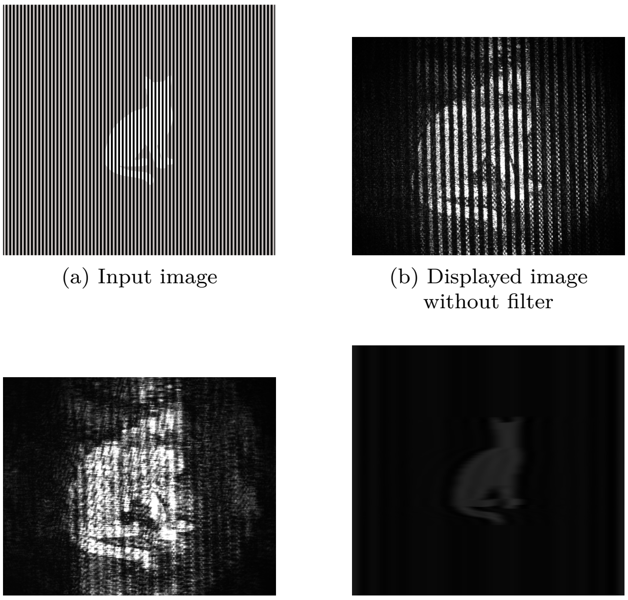
Figure 22: Applying a circular high-pass filter to the image of the 3

In order to showcase the effect of a circular filter, we used an image of the number 3. The optically obtained image, for which we used the low-pass filter falls in line with the expectations fairly well, as it became blurred since sharp edges require high frequencies. However that is all that can be observed, whereas in the numerically acquired image one can also see waves emanating from the shape of the 3, as higher frequencies are required to eliminate those contributions of the lower frequencies. It is possible that these waves were too dark to be picked up by the camera, so perhaps the shutter speed should have been lowered or a smaller radius could have been chosen, if we wanted to see them.

In the case of the high-pass filter the difference between numerical method and optical filtering becomes more apparent, as the filter's effect when applied numerically is very distinctive, whereas optically it isn't so clear. The image we generated using python shows how high frequencies are required to create sharp edges and clearly defined outlines for an image, as the filter removes everything inside the boundary of the 3.

This cannot be observed in the optically obtained image. Instead alot more detail inside of the three becomes more distinguishable when compared to the output without a filter. This detail prevents the filling of the 3 to be removed completely by the filter, as it is linked to high frequencies in the Fourier plane. However in an idealized environment, such as when loading the image directly into python, these details inside of the number don't exist and as such the filling gets removed.

One specific task we tried to do with this setup, was to load in an image of a cat inside of a cage and to then filter the frequencies in such a way, that the cage becomes obscured whereas the cat remains visible. During the lab it seems we had the right idea with applying a vertical low-pass filter, to cut out the cage, as it is given by waves with k-vector exclusively in the ν_x-direction. Doing this instead of, what one might think of first, using a horizontal high-pass filter to exclude those waves, has the advantage that the low frequencies which make up the general shape of the cat remain included.


(a) Input image

(b) Displayed image without filter

(c) Displayed image with vertical low-pass filter of width 13 pixels

(d) Numerically filtered image with a vertical low-pass filter

Figure 23: Mask filtering of the image of a cat using a vertical low-pass filter

We clearly didn't get as good results experimentally, as we did numerically, but it did tend into the right direction as the lines of the grating are blurred, connecting the parts of the cat, which are not obscured. One problem that might have made this more difficult is that the center of the SLM may not have been in alignment with the center of the Fourier plane. This could cause linear filters to have an offset to the axis one would expect them to go through and circular ones not to cut off frequencies uniformly around the center. For numerical uses of mask filtering this isn't a problem, making using them quite simple.

This example showcases the usefulness of Fourier transforms for filtering out periodic imperfections or otherwise omnipresent imperfections in an image, by analyzing their shape in the Fourier plane and selectively removing those components. However care must still be taken to ensure one doesn't also alter too much of the rest of the image. Using more complicated filters, one could probably have achieved a significantly better result.

3.4. CUTOFF FREQUENCY FOR A CIRCULAR APERTURE

Because the optical Fourier transform maps the spatial frequencies ν_x and ν_y onto coordinates x and y in our focal plane, we can filter out certain frequencies.

If we use circular apertures of diameter D for spatial filtering, what frequencies will be cut off?

To find the answer to this question we have to look closer at the frequency-space mapping that is happening. As described in [LPH22] and undermined by us in Section 3.1 the composites of the original wave are mapped onto the points $(x, y) = (\lambda f \nu_x, \lambda f \nu_y)$. A circular aperture with diameter D will now cut off the frequencies that satisfy

$$\lambda^2 f^2 \left({\nu_x}^2 + {\nu_y}^2\right) \geq \left(\frac{D}{2}\right)^2 . \tag{3.4}$$

Defining the cutoff frequency as ${\nu_c}^2 := {\nu_x}^2 + {\nu_y}^2$ therefore yields that

$$\nu_c = \frac{D}{2\lambda f} . \tag{3.5}$$

4. HOLOGRAPHY BY PHASE RETRIEVAL

In this part of the lab we went back to the setup in Figure 3 to perform holography with our SLM. Important to note is that we now removed the polarizer behind the SLM. We motivate holography by the ambition to synthesize the focal plane from Section 3.2. If we are able to do so, we can display images using only the phase modulation of our SLM rather than converting the modulated phase into an amplitude modulation.

The light from the synthesized focal plane holds information in form of amplitude and phase. As described in [LPH22] we will assume that all the important information for displaying the pictures lie within the phase information. Doing so we involve an error that we can illustrate by numerically performing an FT and IFT on the letter **F** from Figure 6.

As we can see in Figure 24 the letter **F** is visible but only because of very distinct and high frequency features like the corners and edges.

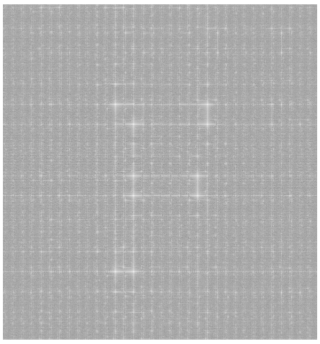


Figure 24: FT and IFT on the letter **F** but with removed amplitude before applying IFT.

4.1. THE GERCHBERG-SAXTON ALGORITHM

The prominent error in assuming the important information lies within the phase, can be iteratively corrected by the Gerchberg-Saxton algorithm. It is a phase retrieval algorithm that can be used to reconstruct the phase of an original image by knowing it's FT. It can be used in our case to construct a »phase mask« which, if transformed, will display the image we want.

The original image or also called *source* image we will use

Figure 25: 2D-Gaussian with $\sigma = w/4$ where w is the picture width.

is going to be a 2D-Gaussian, that will represent the amplitude profile of the incoming beam when it hits the SLM. The target image will be the image we want to display. So that is just the letter **F** from Figure 6.

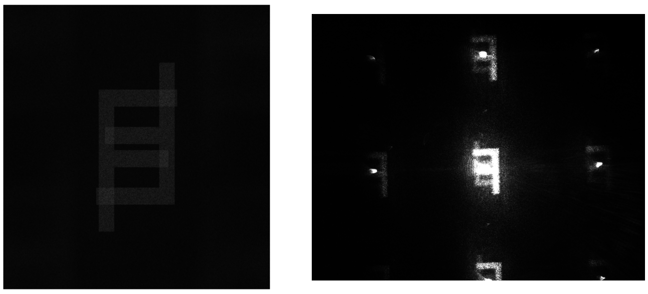


Figure 26: Numerically achieved FT of the phase mask (left) and acquired image in the lab after image reconstruction (left).

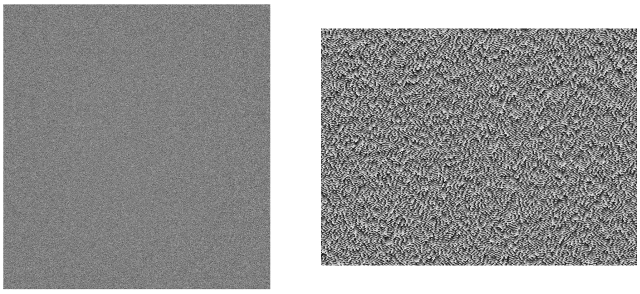


Figure 27: Numerically achieved phase mask (left) and phase mask by the Lab GUI (left).

In Figure 26 we can see that the image reconstruction by performing the FT on the numerically achieved phase mask in Figure 27 yields the result we expected. The only thing that is off, is the orientation. But since we used a converging lens to focus our collimated beam, the image will be horizontally and vertically flipped. Of course the numerically obtained image in Figure 26 is much cleaner than the one we captured in the lab. This is due to the zeroth order maximum that is not diffracted at an angle and therefore blows up the exposure in the center of the image.

4.2. Discussion the error reduction

For characterizing how good the algorithm approximates the original picture from Figure 6 we need to introduce a metric. Because we did not want to deploy similarity comparisons, we opted for a rather simple metric that will yield as a starting point for comparing the quality.

The arrays P_{OG} and P_{NUM} for the pictures will be normalized by their maximum value and then combined in the following way to form the Metric M_n where n are the iteration counts.

$$M_n = \left|\left| \left|\left|1 - \frac{P_{\mathrm{OG}}}{\max(P_{\mathrm{OG}})}\right|\right|_{\mathrm{F}} - \left|\left|1 - \frac{P_{\mathrm{NUM}}}{\max(P_{\mathrm{NUM}})}\right|\right|_{\mathrm{F}} \right|\right| \tag{4.1}$$

The picture for comparison is not the one contained in Figure 26. There we performed the FT on the phase mask ϕ_{A} only. What we have to do is perform the FT on

$$\phi'_{\mathrm{A}} = \exp\left(i\phi_{\mathrm{A}}\right) \tag{4.2}$$

Figure 28: Numerically achieved reconstruction by performing a FT on $exp(i \cdot \phi_{\mathrm{A}})$ where ϕ_{A} is the phase mask.

For each iteration we can plot the metric M_n which will give us

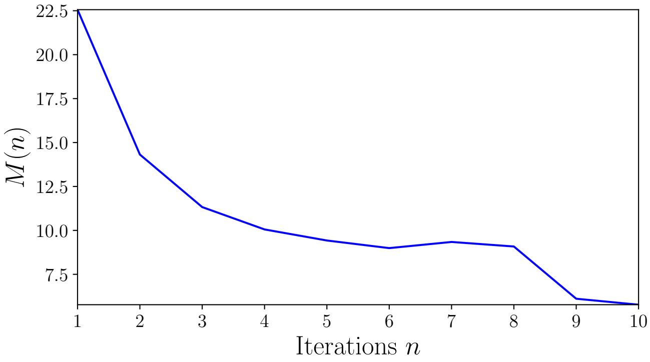


Figure 29: The metric M_n for the comparison of the picture accuracy when approximating with the Gerchberg-Saxton algorithm.

We can see from this illustration that this algorithm converges rather fast, and will not yield much more improvement when increasing the number iterations.

References

[HH10] I. Hughes and T. Hase. *Measurements and their Uncertainties: A Practical Guide to Modern Error Analysis.* Oxford University Press Inc., 2010. ISBN: 978-0-19-956632-7.

[HOL] HOLOEYE Photonics AG. *Brochure Spatial Light Modulators*, p. 6. URL: https://holoeye.com/lc-2012-spatial-light-modulator.

[LPH22] D. Lechner, L. Pache, and X. Hu. *Fourier Optics and Spatial Light Modulator Lab Course.* 2022.

[Mül07] U. Müller. *Physikalisches Grundpraktikum: Einführung in die Messung, Auswertung und Darstellung experimenteller Ergebnisse in der Physik.* 2007. URL: https://gpr.physik.hu-berlin.de/Skripten/Einfuehrung/PDF-Datei/Einfuehrung.pdf.

GerchbergSaxton

July 2, 2022

```
[2]: import numpy as np
     import matplotlib.pyplot as plt
     import matplotlib as mpl
     import scipy as sc
     from scipy.fft import fft2,ifft2,fftshift
     from PIL import Image
     from scipy import stats
```

1 Importing the F image

```
[9]: OG = plt.imread("./F.bmp")
     OG = OG[:,:,1]
```

2 Building a Gaussian dot

```
[10]: def gaus2d(shape0, shape1, mx=0, my=0, sx=40, sy=40):
          x = np.linspace(-100, 100,shape1)
          y = np.linspace(-100, 100,shape0)
          x, y = np.meshgrid(x, y) # get 2D variables instead of 1D

          return 1. / (2. * np.pi * sx * sy) * np.exp(-((x - mx)**2. / (2. * sx**2.)␣
        ↪+ (y - my)**2. / (2. * sy**2.)))

      z = gaus2d(1590, 1500)
      plt.figure(figsize=(5,5))
      plt.axis("off")
      plt.imshow(z,cmap="gist_gray")
```

```
[10]: <matplotlib.image.AxesImage at 0x7fd4dec6e620>
```

3 Bulding the algorithm

```
[16]: from numpy import linalg

def gerchsax(OG, iterations):
    #First we guess a phase for the first run
    phi0 = np.random.rand(OG.shape[0],OG.shape[1])

    #set Gaussian beam as source
    SC = gaus2d(OG.shape[0],OG.shape[1])

    metrics = np.zeros(iterations)
    #Now we begin the iterative process in which we optimize the phase to␣
 ↪contain amplitude information
    for i in range(iterations):
        #First build the new complex function
        E = SC * np.exp(1j * phi0)

        #Then perform FT on E
        FFTE = fft2(E)

        #extract phase of FFTE
```

```
        phi0 = np.angle(FFTE)

        #Build the new complex function
        E = OG * np.exp(1j * phi0)

        #Perform IFFT on E
        IFFTE = ifft2(E)

        #extract phase of IFFTE
        phi0 = np.angle(IFFTE)

        #Test the images
        newpic = np.abs(fft2(np.exp(1j*phi0)))
        newpic = 1 - newpic / np.max(newpic)

        #Metric
        metrics[i] = np.abs( linalg.norm( 1- OG/np.max(OG) ) - linalg.
 ↪norm(newpic) )


    return phi0,metrics


phiA,metrics = gerchsax(OG,10)

piclol = fft2(np.exp(1j*phiA))

plt.figure(figsize=(5,5))
plt.imshow(np.abs(piclol),cmap="gist_gray")
plt.axis('off')
```

```
[16]: (-0.5, 1499.5, 1596.5, -0.5)
```