Lab 4 - Fourier Optics

Ahmed Abdelrahman and Teagan Kilian

Instructor: Emmanuel Marin

November 11th, 2024

Faculty of Sciences and Technology, UJM

1. Introduction

This experiment was conducted in order to demonstrate the scalar diffraction theory that has been introduced in the course *Physical and Fourier Optics*. The primary aim was to see how different types of diffraction gratings affected the diffraction pattern of a monochromatic laser then analyze the resulting spatial frequencies. Sequentially, a Fresnel lens was to replace the diffraction gratings and the resulting diffraction patterns were to be observed.

The primary objective of scalar diffraction theory is to determine the resulting wave on a screen produced by the initial wave as it passes through an aperture and undergoes diffraction. This theory helped to explain the propagation of light, diffraction phenomena, and helped in many different applications (*figure 1)*.

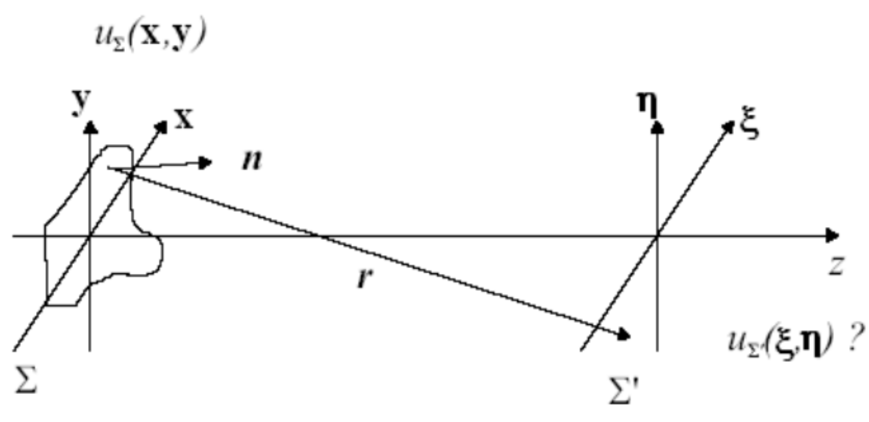


Figure 1: Schematic depicting scalar diffraction theory

$$u_{\Sigma'}(\xi,\eta)=\iint_{\Sigma} h(x,y,\xi,\eta)u_{\Sigma}(x,y)dxdy \quad , \quad h(x,y,\xi,\eta)=\frac{1}{j\lambda r}e^{jkr}\cos(n,r)$$

Equation 1: (a) Huygen Fresnel Integral. (b) Impulse Response.

Using *equations 1(a and b)* we can calculate Uz (at the screen) from Uo (from the source).

Fresnel proposed *equation 2*: an approximation to this equation, assuming that z>>size of the aperture, which results in R to be approximately equal to z.

$$u_{\Sigma'}(\xi,\eta)=\frac{1}{j\lambda z}e^{jk\left[z+\frac{\xi^2+\eta^2}{2z}\right]}\iint_{\Sigma} u_{\Sigma}(x,y)e^{jk\frac{x^2+y^2}{2z}}e^{-2j\pi\frac{(x\xi+y\eta)}{\lambda z}}dxdy$$

Equation 2: Fresnel Approximation

Fraunhofer completed Fresnel's work, and proposed an even simpler equation (*equation 3*) by assuming that z is almost infinite (in comparison to the size of the screen).

$$u_{\Sigma'}(\xi,\eta)=\frac{1}{j\lambda z}e^{jk\left[z+\frac{\xi^2+\eta^2}{2z}\right]}\iint_{\Sigma} u_{\Sigma}(x,y)e^{-2j\pi\frac{(x\xi+y\eta)}{\lambda z}}dxdy$$

Equation 3: Fraunhofer Approximation

---

2. Setup and Experiments

The materials used to complete this lab include a monochromatic laser, modulator, polarizer set up that was used to propagate the laser onto the image plane. Specifically, the laser is a red light laser with an approximate wavelength of *632.8 μm*. The modulator is a spatial light modulator (SLM) with a *832 x 624* pixel liquid crystal display with a period of *32 μm*. The previously mentioned equipment was part of the OpiXplorer - Optics Educational Kit by Holoeye. A sensor was installed just in front of the image plane to capture the diffraction pattern created by the system. The general setup can be seen in *figure 2*. A computer program was used to manipulate the SLM by applying voltage to the liquid crystals in order to generate the network that was given to the program. A second computer program was used to read the measurements of the distance between fringes as collected by the sensor. Finally, Python was used to generate the images used to manipulate the SLM and to interpret the experimental results.

Figure 2: Schematic of Experimental Setup (not drawn to scale)

We conducted three different experiments throughout this lab:

1. **Moiré's network**: In this experiment, we used three one-dimensional binary networks, each having a different grating pattern.
   - Λ_o = 2 pixels
   - $\Lambda = a\Lambda_o$ with a = 1.15,1.4

The following python code shows how these different grating images were created:

```
import matplotlib.pyplot as plt
T_0 = 2
T_1 = 1.15*T_0
T_2 = 1.4 *T_0
# modulator matrix is 832x624
x = 624
y = 832

#1D binary network with Λ=2 pixels
grating = [[1,0]*int(624/2)]*832
plt.imsave('1pixel.bmp', grating, format='bmp', cmap='gray')

#1D binary network with Λ=2*1.15 pixels
grating2=[[0]*624]*832
i=0
for n in range(832):
  while(i<624):
    grating2[n][int(i)]=1
    i+=T_1
plt.imsave('115pixel.bmp', grating2, format='bmp', cmap='gray')

#1D binary network with Λ=2*1.4 pixels
grating3=[[0]*624]*832
i=0
for n in range(832):
  while(i<624):
    grating3[n][int(i)]=1
    i+=T_2

plt.imsave('14pixel.bmp', grating3, format='bmp', cmap='gray')
```

The images generated by the previous Python code are shown in *figure 3*.

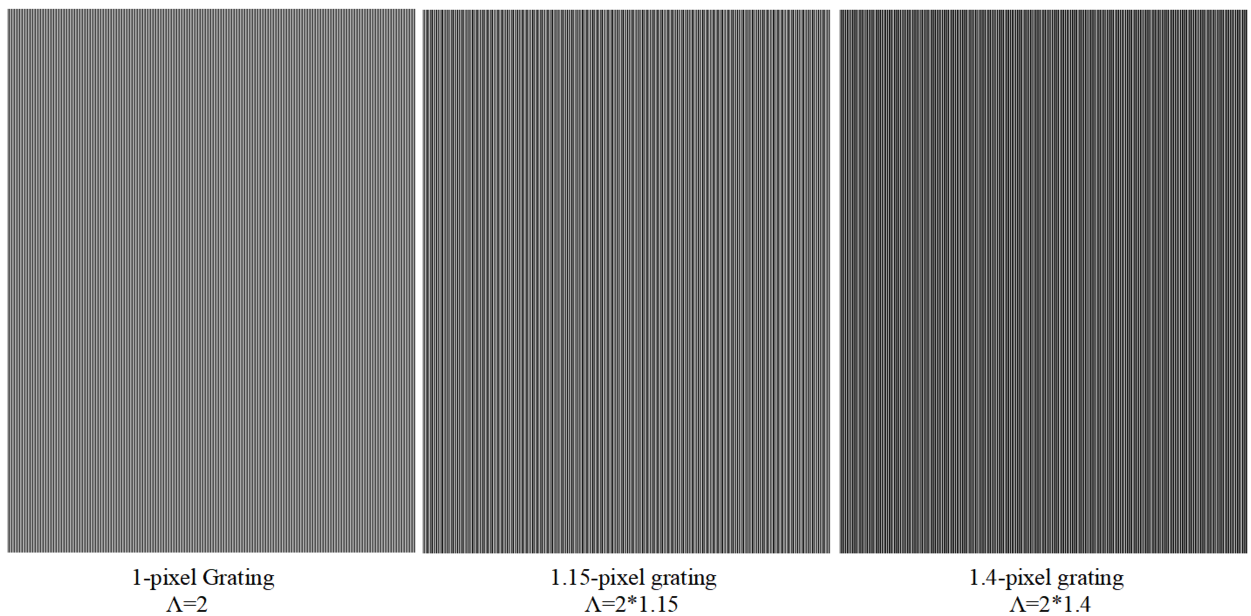


Figure 3: Diffraction gratings used in part 1 of the experiment.

The grating images were successively used to show the different diffraction patterns on the display screen. We then measured the distances between the peaks using the sensor and software available in the lab. The results were recorded.

2. **Symmetrical Fresnel lens**: In this experiment, a Fresnel lens grating was generated in Python and used in the SLM. This set up acted as a hologram formed by a spherical wave originating from a point source and a plane wave. The grating pattern is symmetrical (meaning only one focal length was provided. *Equation 4* were used to generate the grating of the Fresnel lens.

$$\mathrm{Lens}(r) = \mathrm{Argument}\left(e^{i\varphi(r)}\right) \qquad \text{with } \varphi(r) = \pi\frac{r^2}{\lambda f}$$

Equation 4: Formula of a symmetric Fresnel lens.

```
import numpy as np
import matplotlib.pyplot as plt
import cmath,math
#832x624
circularGrating=[]
for n in range(832):
  circularGrating.append([0]*624)
iMid=832/2
jMid=624/2
wavelength=632.8e-9
focal_length=0.5 #0.5 for symmetrical and 0.3,0.8 for asymmetrical
for i in range(832):
  for j in range(624):
    rSquared=((i-iMid)*32.e-6)**2 + ((j-jMid)*32.e-6)**2
    deltaR=((np.pi*rSquared)/(wavelength*focal_length))%(2*np.pi)
    arg=(deltaR)/(2*np.pi)
    circularGrating[i][j]+=int(arg*255)
plt.imshow(circularGrating)
plt.imsave('circular.bmp', circularGrating, format='bmp', cmap='gray'
```

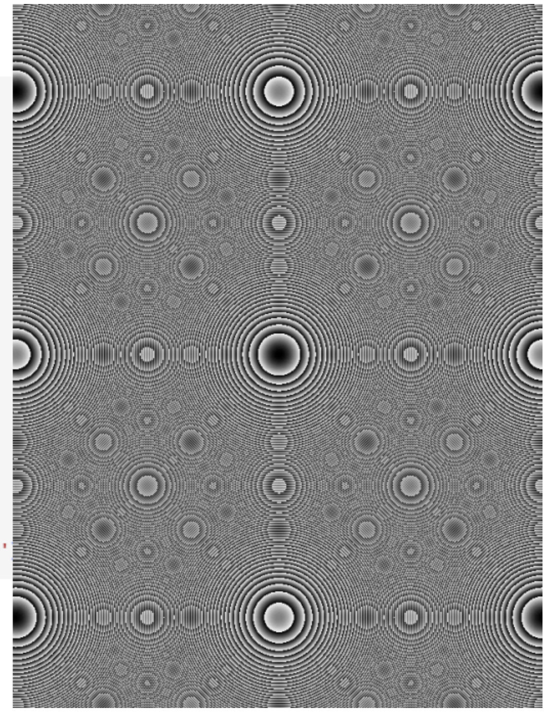


Figure 4: Symmetrical Fresnel lens with a focal length of 50 cm.

Figure 4 shows the python code used to create the symmetrical Fresnel lens as well as the generated grating pattern image used for our experiment. By setting the focal length(f) to 50 cm, and the wavelength to *632.8x10*$^{-9}$m, we were able to create this lens, and observe the convergence at the focal point. We changed the position of the screen to notice the changes of the image focus at these different positions, and recorded the results.

3. **Asymmetric Fresnel lens**: In this experiment, we modified *equation 4* to create an asymmetrical grating pattern of an elliptical shape. *Equation 5* was ultimately used to generate the asymmetrical lens. In order to get to this equation from the previous, r was decomposed into its x and y components so that a different focal length could be assigned to each direction.

$$\frac{\pi}{\lambda}\left(\frac{x^2}{f1}+\frac{y^2}{f2}\right)$$

Figure 5: Equation used to create an asymmetrical Fresnel lens.

```
import numpy as np
import matplotlib.pyplot as plt
import cmath,math
#832x624
circularGrating=[]
for n in range(832):
  circularGrating.append([0]*624)
iMid=832/2
jMid=624/2
wavelength=632.8e-9
for i in range(832):
  for j in range(624):
    xsq=((i-iMid)*32.e-6)**2
    ysq = ((j-jMid)*32.e-6)**2
    phi = ((np.pi/wavelength)*((xsq/.3)+(ysq/.8)))%(2*np.pi)
    arg=(phi)/(2*np.pi)
    circularGrating[i][j]+=int(arg*255)
plt.imshow(circularGrating)
plt.imsave('circular.bmp', circularGrating, format='bmp', cmap='gray')
```

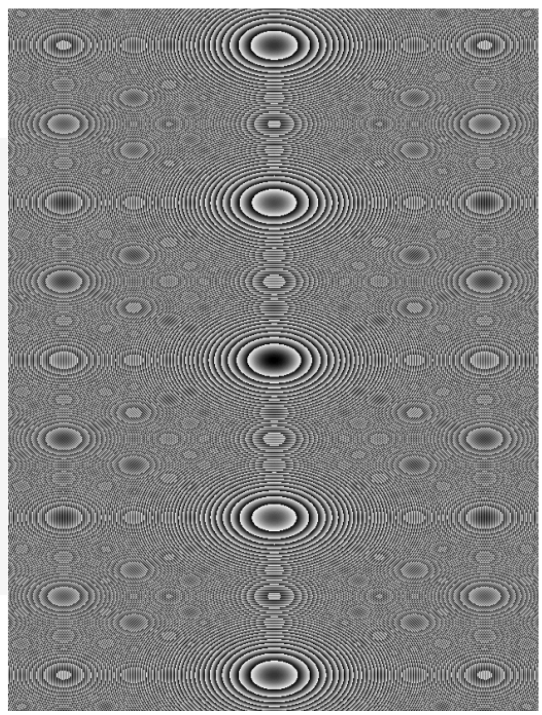


Figure 5: Asymmetrical Fresnel lens with a focal length of 30 cm in the x direction and a focal length of 80 cm in the y direction.

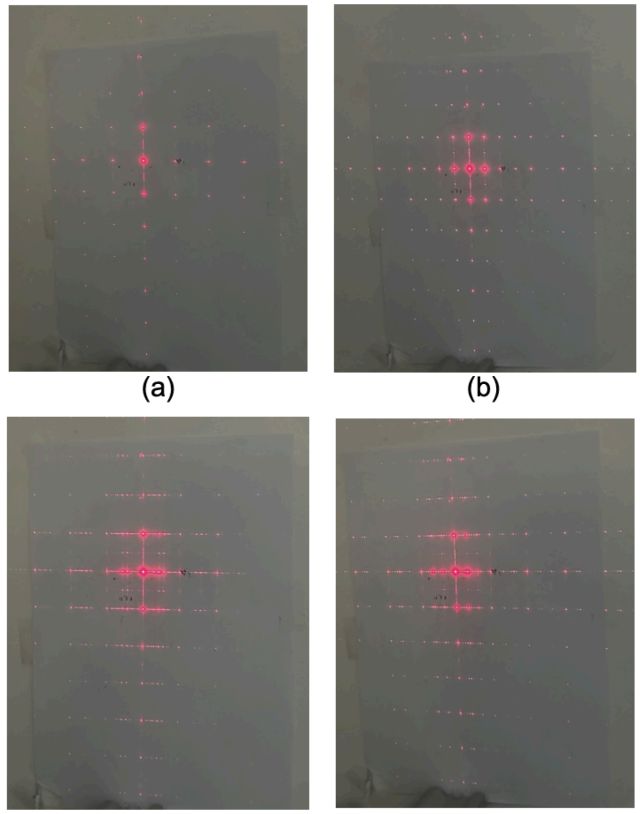
Figure 5 shows the python code used to create the asymmetrical Fresnel lens as well as the generated grating pattern used for our experiment. The focal length in the x direction was set to 30 cm and the focal length in the y direction was set to 80 cm, and the laser had a wavelength of $632.8x10^{-9}$ *m*. These values were plugged into *equation 5* which allowed for the generation of an asymmetrical lens which contained elliptical shapes rather than circular ones. The position of the screen was changed to notice the differences in the image at the different positions. The results were recorded.

---

3. Results

1. **Moiré's network**:

As seen in *figure 6*, the diffraction grating that the laser travels through has a significant impact on the diffraction pattern that is observed on the image plane. The images show a change in intensity as you move from the central fringe which is caused by the diffraction of the pixel due to its size. The period of the pattern produced with no grating added is *1/32 µm* which is induced by the period of the modulator. The distance between subsequent bright fringes is inversely proportional to the sampling period.
The set up with a diffraction grating having a 2 pixel period shows the Nyquist frequency of the experiment. The Nyquist frequency is the maximum frequency that is available with the specific equipment.



(a) (b) (c) (d)

Figure 6: (a) Diffraction pattern produced by the SLM with no diffraction grating

(b) Diffraction pattern produced by a diffraction grating with a period of 2 pixels

*(c) Diffraction pattern produced by a diffraction grating with a period of 2 * 1.15 pixels*

*(d) Diffraction pattern produced by a diffraction grating with a period of 2 * 1.4 pixels*

Table 1 depicts the fringe locations as collected from the sensor located at the image plane. The values show the distance of each fringe from the zeroth order fringe. The grating with a period of 2 only has one data point collected because there is only one additional fringe between the zeroth order fringe and the fringe associated with the Nyquist frequency. This fringe is located at the 0.5 - 1 pixel frequency.

Table 1: Location of Fringes Produced by Diffraction Gratings of Different Periods

	Distance from fringe # to zeroth order fringe (μm)			
Period (pixels)	# 1	# 2	# 3	# 4
2	14,056			
2 * 1.15	12,264	8,568	7,280	3,752
2 * 1.4	12,096	10,080	8,008	6,048

Based on the data collected regarding the fringe location, it is possible to perform a fast Fourier transform (FFT) calculation on each experimental setup to find the theoretical diffraction pattern. The location of the fringes found from the FFT can then be compared to what was observed experimentally.

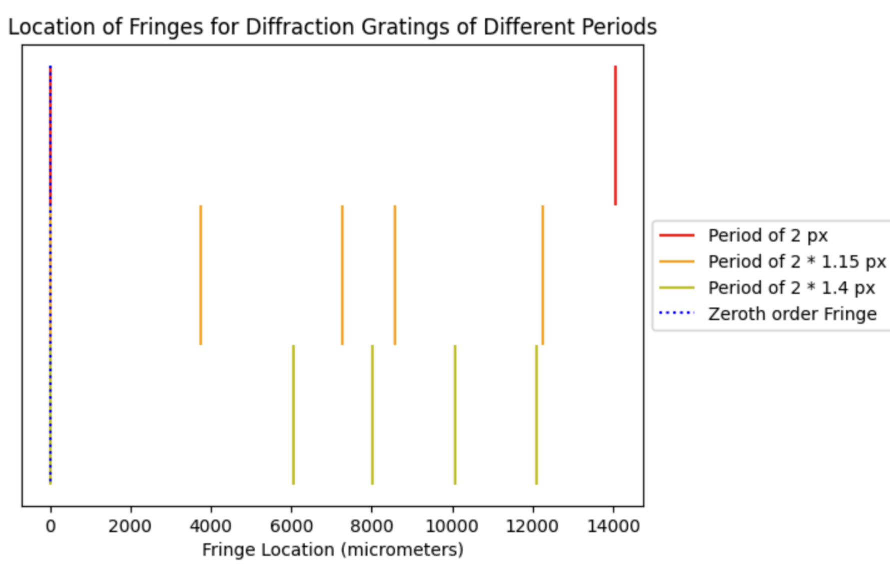


Figure 7: Location of Fringes for Diffraction Gratings of Different Periods

As seen in figure 7, the frequencies that are of interest are the ones that are between the zeroth order fringe and the first order fringe. It can be observed that the higher the period of the diffraction grating, the closer together the fringes are and the more fringes occur between the zeroth and first order fringes. Figure 8 shows the FFT of the diffraction grating. Compared to the data collected about the location of the fringes, these results are not very robust. They accurately show the zeroth order fringe in all of the cases but the higher order fringes are not in the expected positions. One reason for this may be that the sensor was able to be positioned exactly on the image plane and it therefore may have not collected an accurate reading.

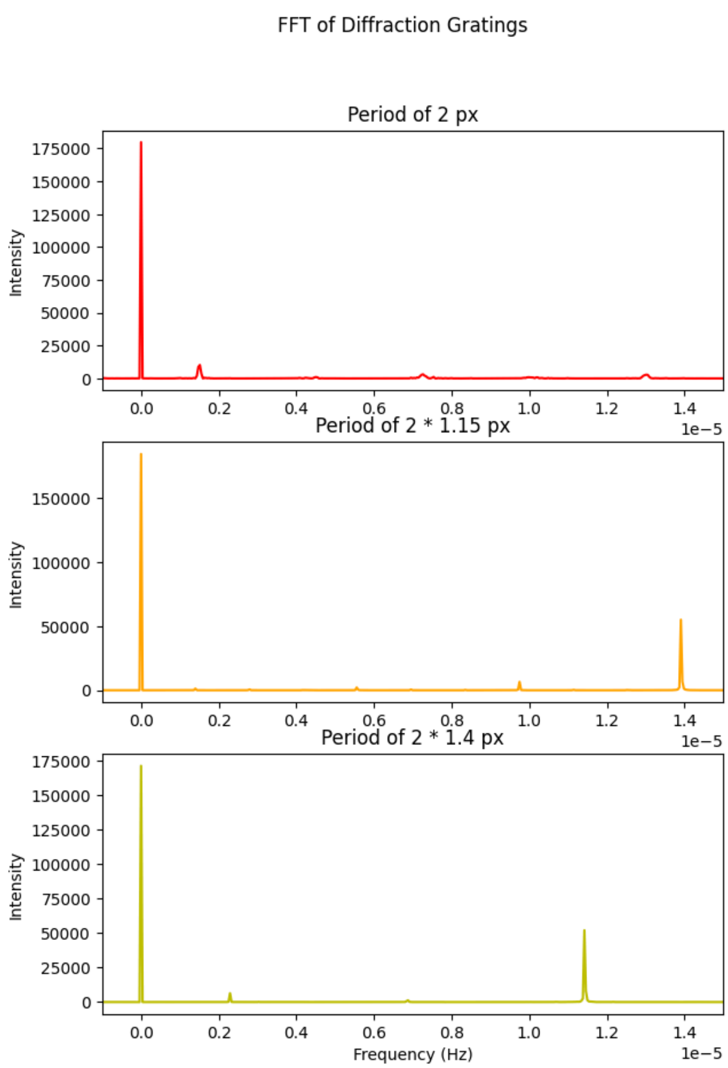
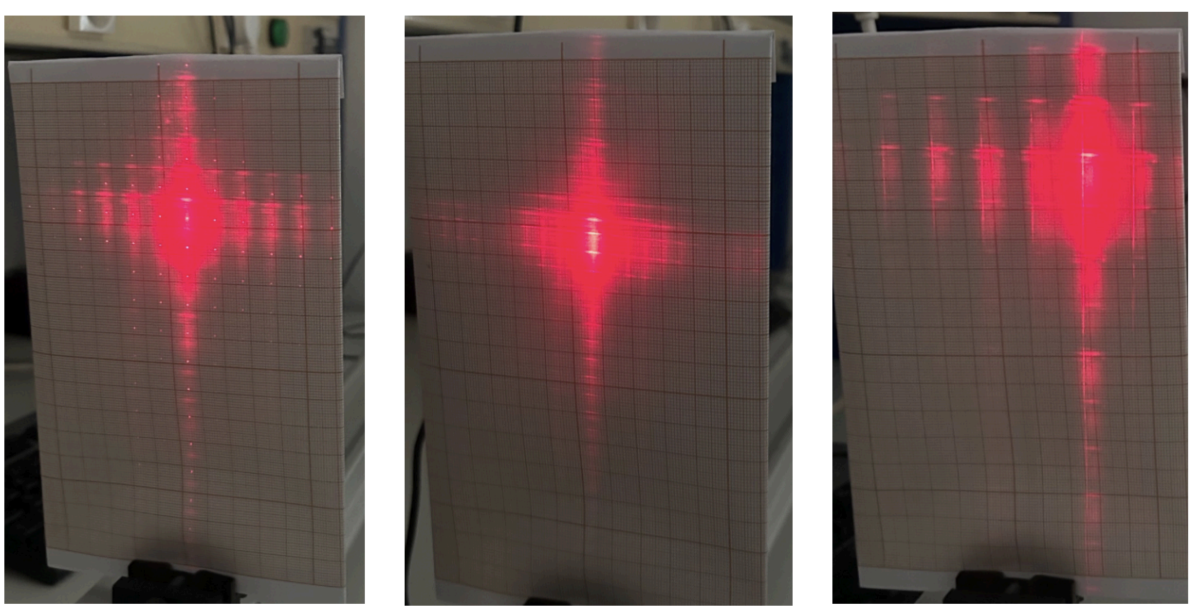


Figure 8: FFT of Diffraction Gratings

These results do, however, verify the theory associated with the splitting of diffraction orders. Although one laser beam was used to illuminate the diffraction grating, the interference between the resulting rays results in constructive and destructive interference which explains the light and dark fringes that appear on the image. This also explains the various fringes seen on the graphs in *figure 8*. The multiple fringes represent a decomposition of the frequencies present in the diffraction pattern.

2. Symmetrical Fresnel lens

Figure 9(a) shows the image captured from introducing a symmetrical Fresnel lens to the modulator. There are similar sequences of light and dark fringes in both the x and y directions with the brightest (zeroth order) fringe being in the center. Since this Fresnel lens is symmetric, it is only generated using one focal length. In this case, the focal length is equal to 50 cm. Our experimental results confirm this because there is only one distance from the optical setup that produces the best image on the image plane. It is important to note that both the x and y directions find this focal point at the same focal length.



(a) (b) (c)

Figure 9: (a) Symmetric hologram with focal length of 50 cm (b) Asymmetric hologram at its x focal length of 30 cm (c) Asymmetric hologram at its y focal length of 80 cm

3. Asymmetrical Fresnel Lens

The final results requiring discussion are those from the asymmetrical Fresnel lens. *Figure 9 (b and c)* both show the image results of the asymmetric Fresnel lens produced by the modulator. The two images are produced by the same lens but at different distances from the optical set up. The focal length given to the x direction was 30 cm therefore the clearest image will stay sharp along the x axis at this location. Similarly, the y direction focal length was set to be 80 cm. When the image plane was set to 80 cm away from the aperture, the fringes in the y direction of the image came into focus. The term for such behavior is astigmatism and it can be summarized by an optical system that has different focuses for different aspects of the image rather than a single focal point.

---

5. References

[1] Emmanuel Marin. 2021 - 2022. *Lab Work in Optics MASTER OIVM 1* (Unpublished Manual). Faculty of Sciences and Technology, University Jean Monnet. pp.18 - 20.

[2] Holoeye, Optixplorer Education Kit Manual

[3] Joseph W. Goodman. (1996). *Introduction to Fourier Optics* (Second Edition). McGraw-Hill.

6. Appendices

```
import matplotlib.pyplot as plt
T_0 = 2
T_1 = 1.15*T_0
T_2 = 1.4 *T_0
# modulator matrix is 832x624
x = 624
y = 832

#1D binary network with Λ=2 pixels
grating = [[1,0]*int(624/2)]*832
plt.imsave('1pixel.bmp', grating, format='bmp', cmap='gray')

#1D binary network with Λ=2*1.15 pixels
grating2=[[0]*624]*832
i=0
for n in range(832):
 while(i<624):
   grating2[n][int(i)]=1
   i+=T_1
plt.imsave('115pixel.bmp', grating2, format='bmp', cmap='gray')

#1D binary network with Λ=2*1.4 pixels
grating3=[[0]*624]*832
i=0
for n in range(832):
 while(i<624):
   grating3[n][int(i)]=1
   i+=T_2

plt.imsave('14pixel.bmp', grating3, format='bmp', cmap='gray')
```

```
import numpy as np
import matplotlib.pyplot as plt
import cmath,math
```

```
#832x624
circularGrating=[]
for n in range(832):
 circularGrating.append([0]*624)
iMid=832/2
jMid=624/2
wavelength=632.8e-9
focal_length=0.5 #0.5 for symmetrical and 0.3,0.8 for asymmetrical
for i in range(832):
 for j in range(624):
   rSquared=((i-iMid)*32.e-6)**2 + ((j-jMid)*32.e-6)**2
   deltaR=((np.pi*rSquared)/(wavelength*focal_length))%(2*np.pi)
   arg=(deltaR)/(2*np.pi)
   circularGrating[i][j]+=int(arg*255)
plt.imshow(circularGrating)
plt.imsave('circular.bmp', circularGrating, format='bmp', cmap='gray')
```

```
import numpy as np
import matplotlib.pyplot as plt
import cmath,math
#832x624
circularGrating=[]
for n in range(832):
 circularGrating.append([0]*624)
iMid=832/2
jMid=624/2
wavelength=632.8e-9
for i in range(832):
 for j in range(624):
   xsq=((i-iMid)*32.e-6)**2
   ysq = ((j-jMid)*32.e-6)**2
   phi = ((np.pi/wavelength)*((xsq/.3)+(ysq/.8)))%(2*np.pi)
   arg=(phi)/(2*np.pi)
   circularGrating[i][j]+=int(arg*255)
plt.imshow(circularGrating)
plt.imsave('circular.bmp', circularGrating, format='bmp', cmap='gray')
```

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.fft import fft, fftfreq
from scipy import signal
```

```
# plot location of fringes

# for period of 2 pixels

T_2 = [14056, 0]

plt.figure(1)
plt.vlines(T_2, 9, 6, color = 'r', label = 'Period of 2 px')


# for period of 2 * 1.15 pixels in micrometers
T_115 = [12264, 8568, 7280, 3752, 0]

plt.vlines(T_115, 6, 3, color = 'orange', label = 'Period of 2 * 1.15 px'
)

# for period of 2 * 1.4 pixels
T_14 = [12096, 10080, 8008, 6048, 0]
plt.vlines(T_14, 3, 0, color = 'y', label = 'Period of 2 * 1.4 px')
plt.vlines(0, 9, 0, color = 'blue', label = 'Zeroth order Fringe',
linestyle = 'dotted')
plt.yticks([])
plt.xlabel('Fringe Location (micrometers)')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.title('Location of Fringes for Diffraction Gratings of Different
Periods')
plt.show()

# Perform FFT on the grating

T_0 = 2
T_1 = 1.15*T_0
T_2 = 1.4 *T_0

samples = 824


# Gratings
grating = .5 * (signal.square(2 * np.pi * np.arange(samples) / T_0) +1)
```

```
grating2 = .5 * (signal.square(2 * np.pi * np.arange(samples) / T_1) + 1)
grating3 = .5 *(signal.square(2 * np.pi * np.arange(samples) / T_2) +1)


# sampling period
T = 1/(32 * 10 **(-6))


xf0 = fftfreq(samples, T)
y0 = fft(grating)
intensity0 = np.abs(y0)**2
shift0 = np.fft.fftshift(intensity0)
shiftf0 = np.fft.fftshift(xf0)

xf1 = fftfreq(samples, T)
y1 = fft(grating2)
intensity1 = np.abs(y1)**2
shift1 = np.fft.fftshift(intensity1)
shiftf1 = np.fft.fftshift(xf1)

xf2 = fftfreq(samples, T)
y2 = fft(grating3)
intensity2 = np.abs(y2)**2
shift2 = np.fft.fftshift(intensity2)
shiftf2 = np.fft.fftshift(xf2)

plt.figure(2, figsize=(7,10))
plt.subplot(3,1,1)
plt.plot(shiftf0, shift0, color = 'r')
plt.title('Period of 2 px')
plt.ylabel('Intensity')
plt.xlim(- .1 * 10 ** (-5), 1.5* 10 **(-5))

plt.subplot(3,1,2)
plt.plot(shiftf1, shift1, color = 'orange')
plt.title('Period of 2 * 1.15 px')
plt.ylabel('Intensity')
plt.xlim(- .1 * 10 ** (-5), 1.5* 10 **(-5))
```

```
plt.subplot(3,1,3)
plt.plot(shiftf2, shift2, color = 'y')
plt.title('Period of 2 * 1.4 px')
plt.ylabel('Intensity')
plt.xlim(- .1 * 10 ** (-5), 1.5* 10 **(-5))

plt.xlabel('Frequency (Hz)')
plt.suptitle('FFT of Diffraction Gratings')
plt.show()
```