

Министерство науки и высшего образования Российской Федерации

Лабораторная работа № 4

«Влияние формы зрачка (cat's vision)»

Выполнил: Леко А.А..

Группа: Q4110

Проверила:

Иванова Т. В.

Санкт-Петербург 2024

Задание:

Моделирование влияния формы зрачка на изображение для эллиптического зрачка с разным соотношением сторон.

Формула эллипса:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

При моделировании возьмем $b = 1$, для изменения соотношения сторон возьмем $a = 0.1, 0.3, 0.5, 0.7, 1$

Возьмем сложное изображение:



Рисунок 1 – Исходное изображение

При моделировании получено:

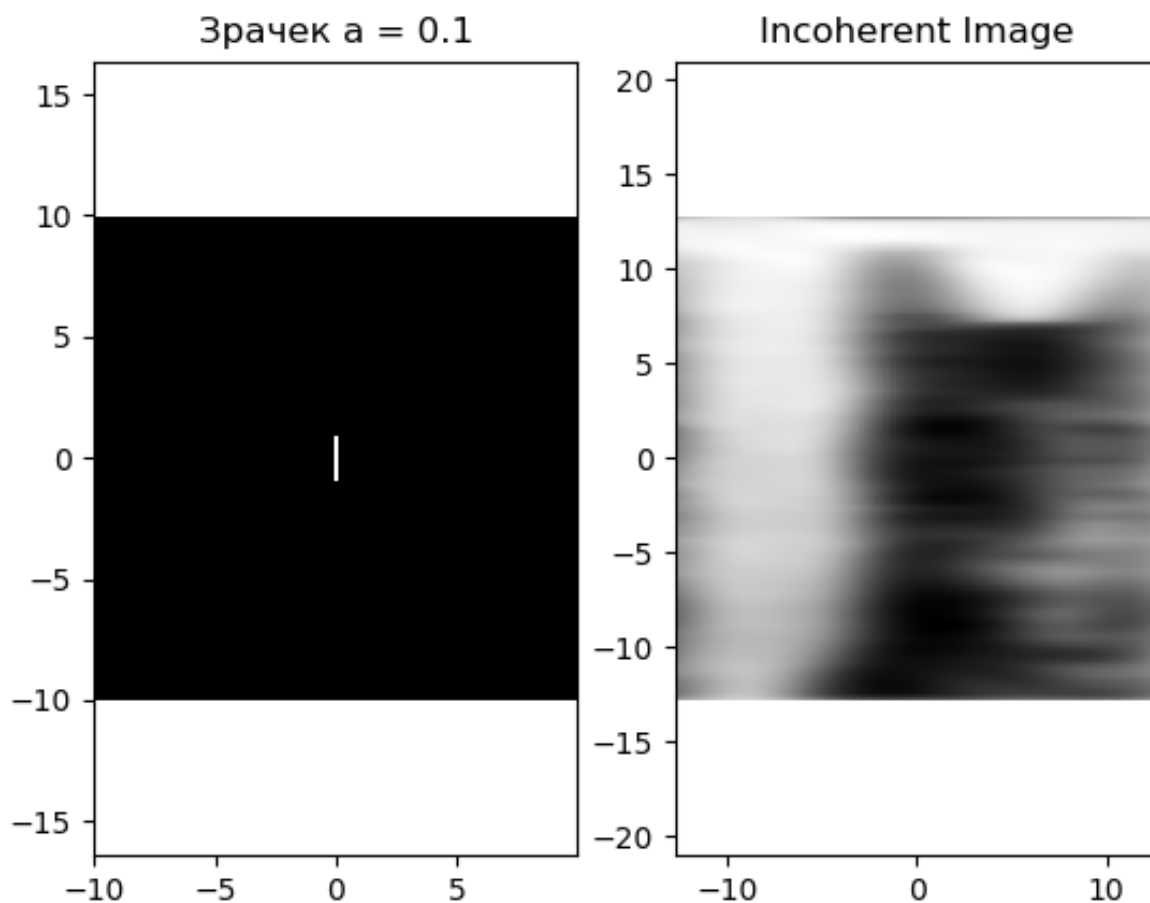


Рисунок 2 – Обработанное изображение при $a = 0.1$

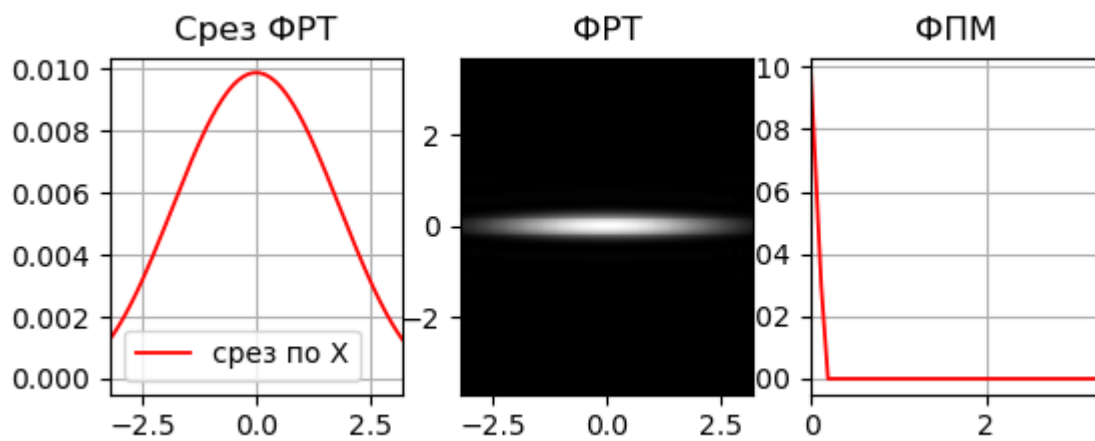


Рисунок 3 – ФРТ и ФПМ при $a = 0.1$

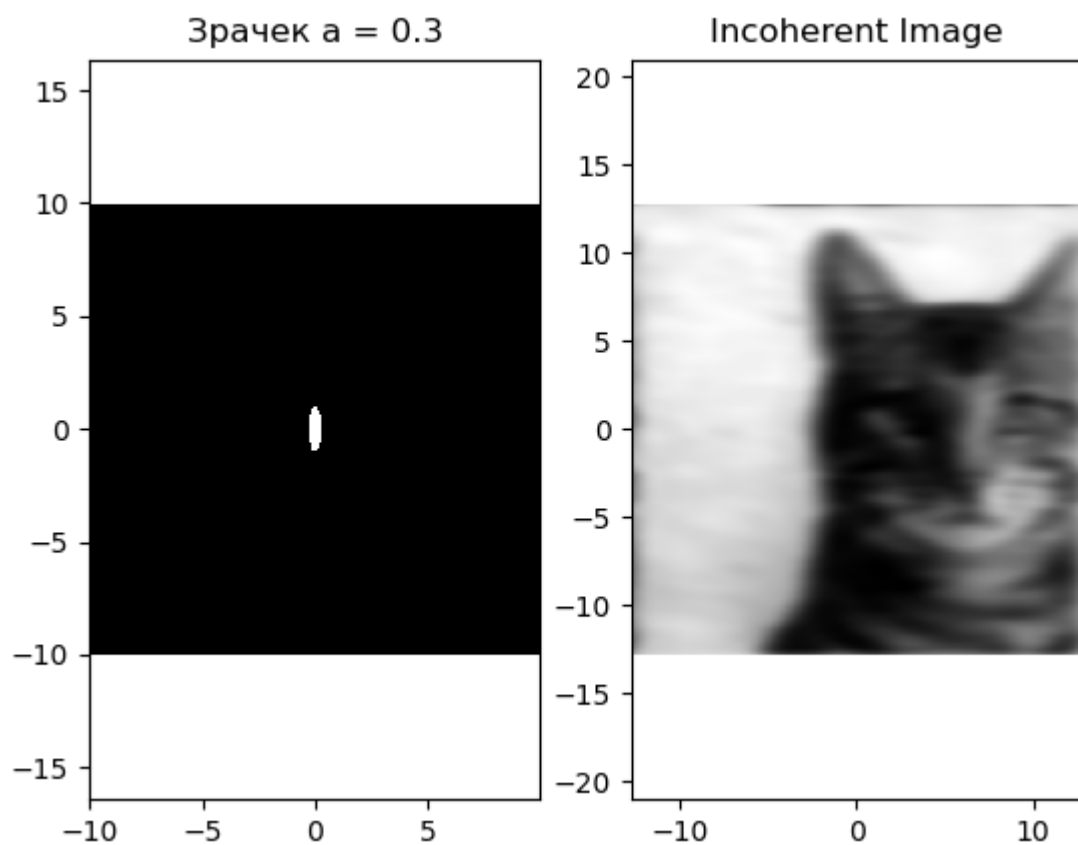


Рисунок 4 – Обработанное изображение при $a = 0.3$

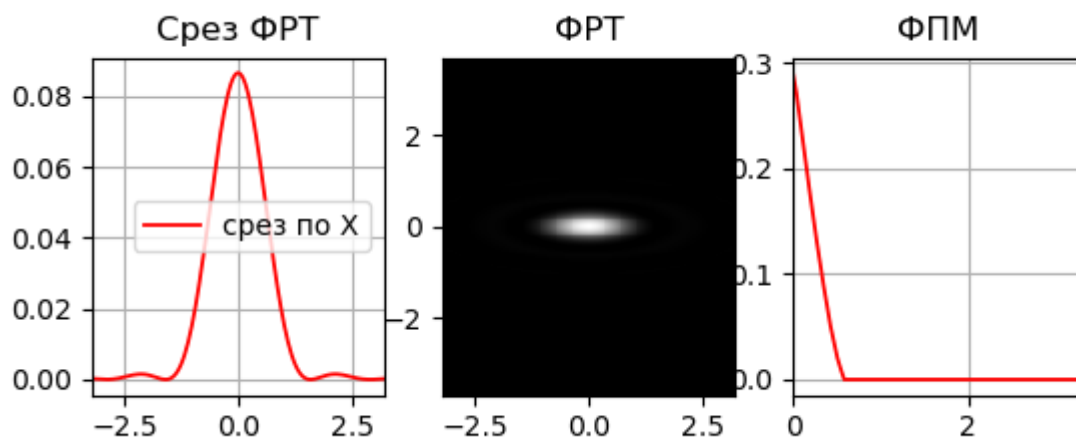


Рисунок 5 – ФРТ и ФПМ при $a = 0.3$

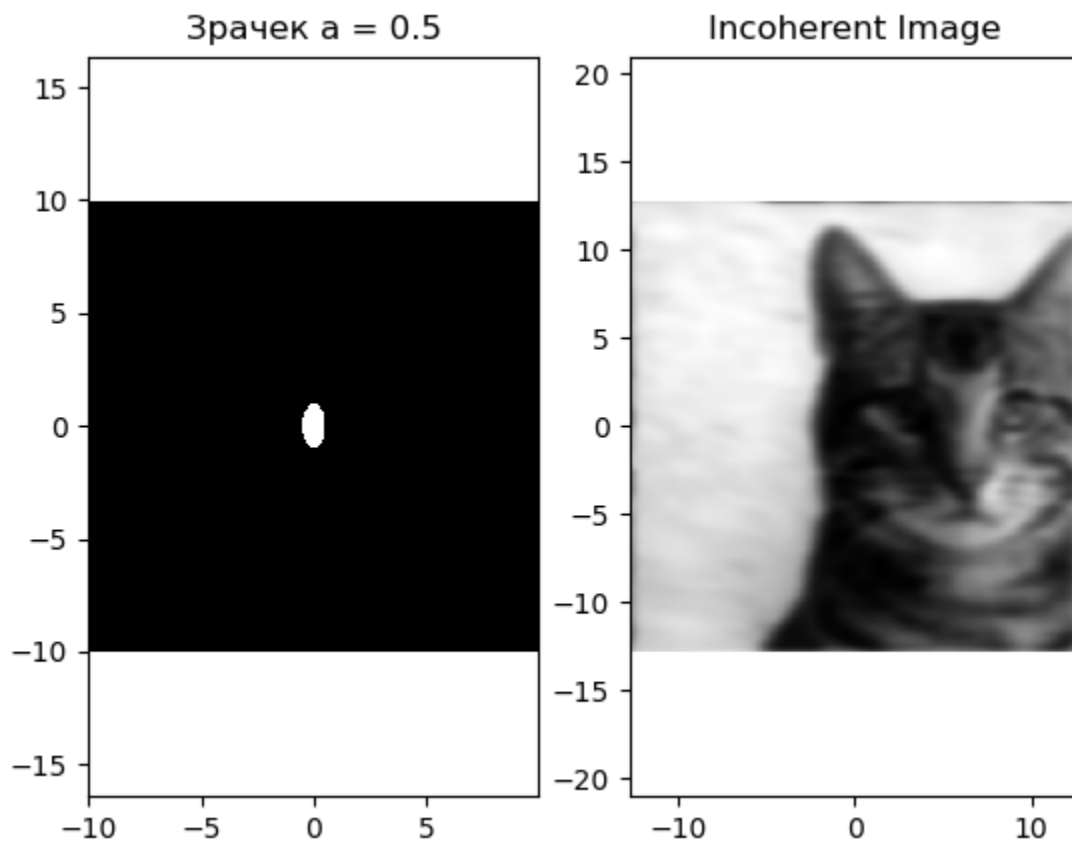


Рисунок 6 – Обработанное изображение при $a = 0.5$

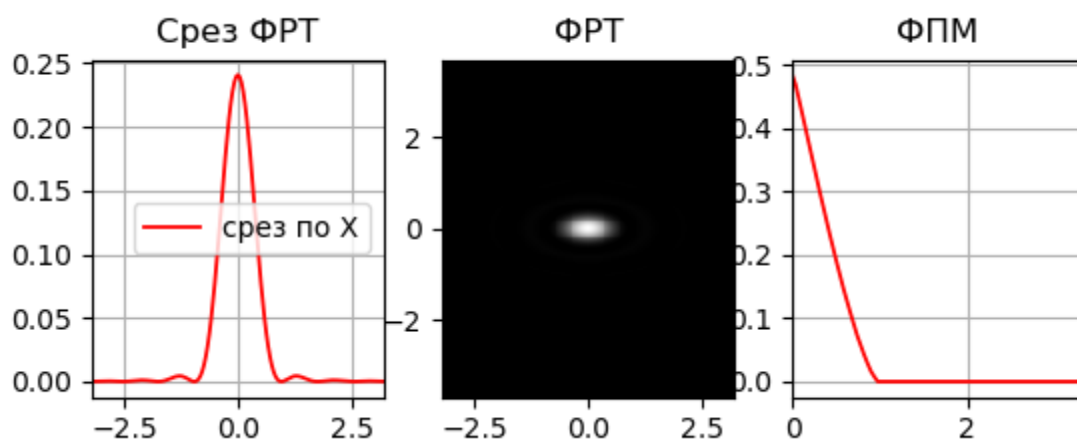


Рисунок 7 – ФРТ и ФПМ при $a = 0.5$

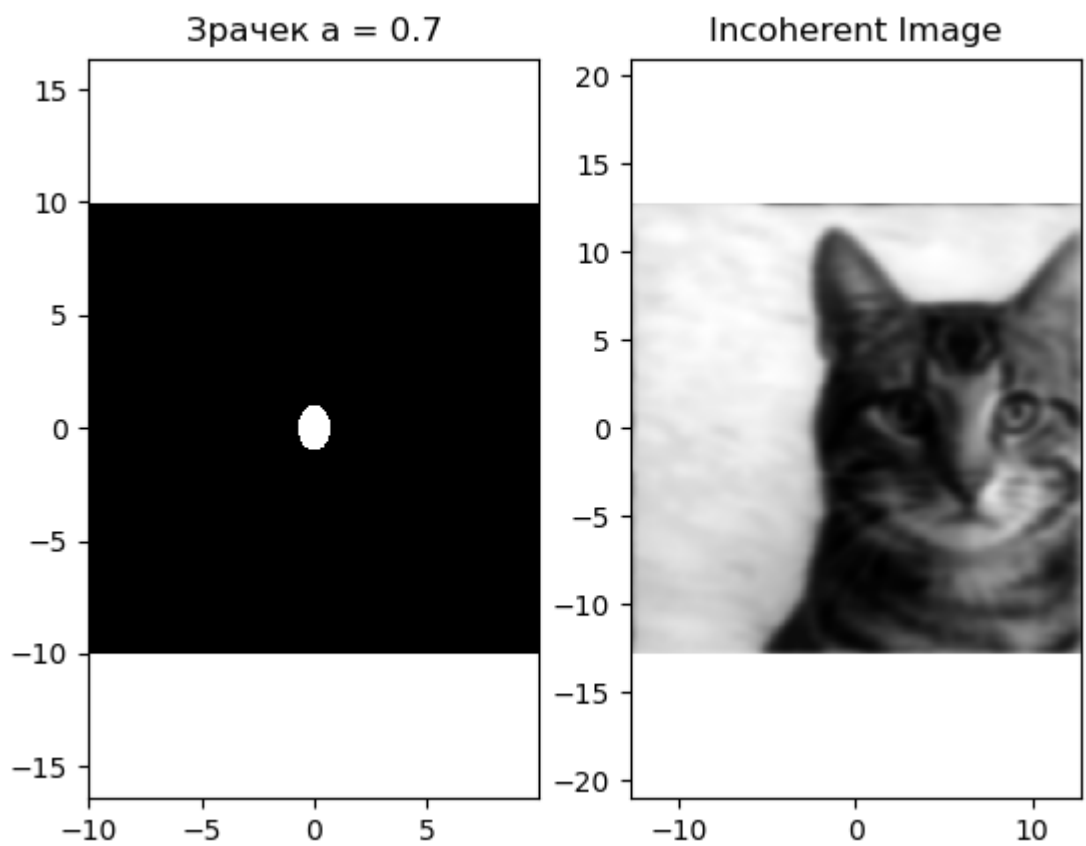


Рисунок 8 – Обработанное изображение при $a = 0.7$

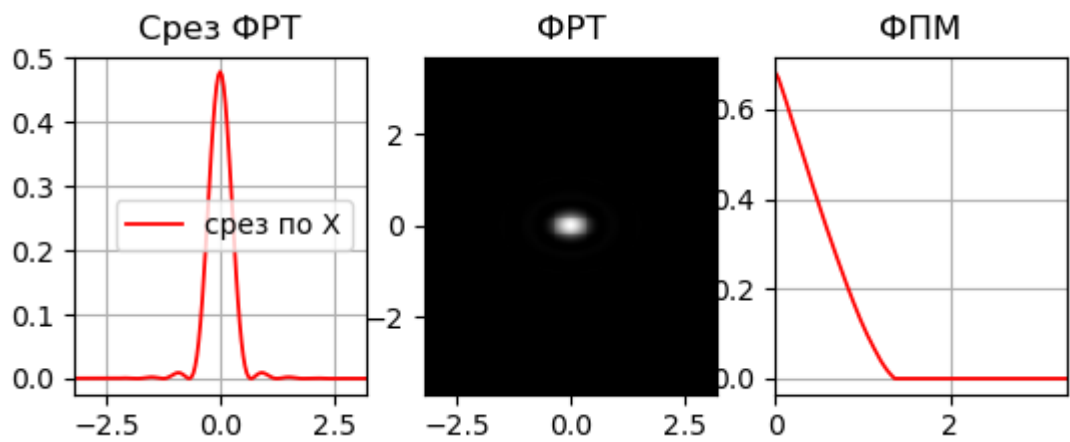


Рисунок 9 – ФРТ и ФПМ при $a = 0.7$

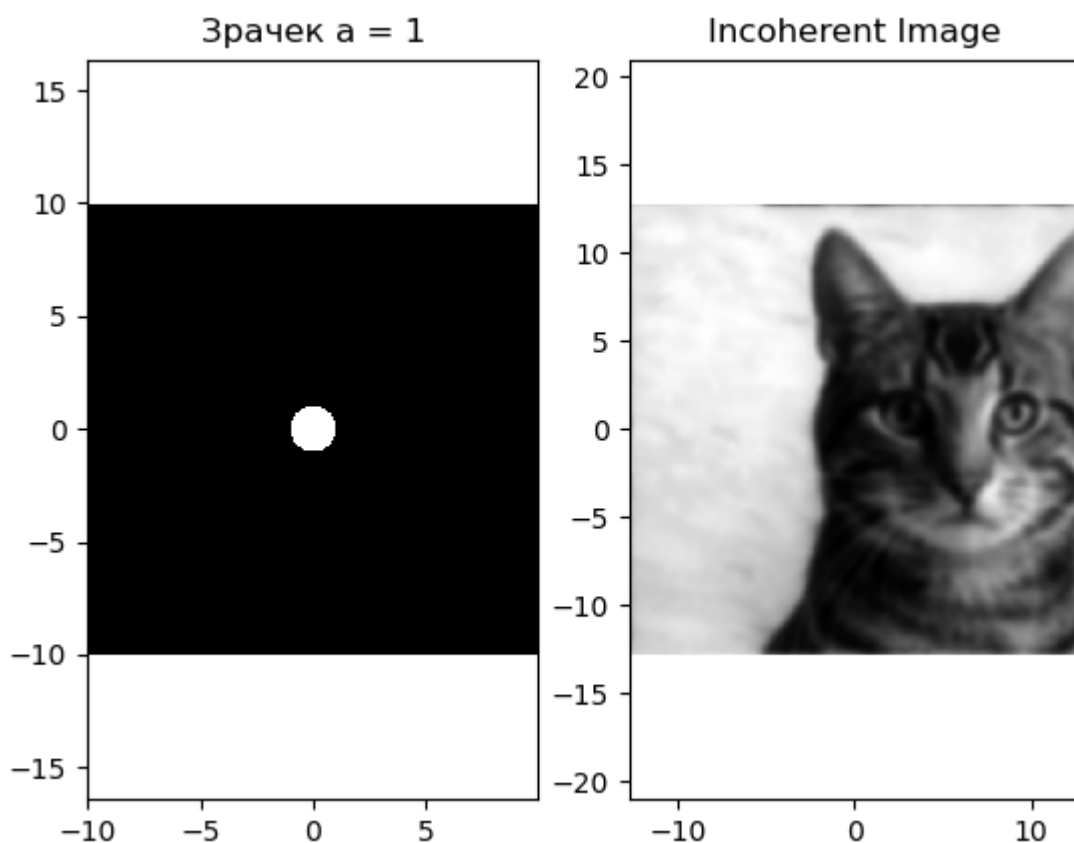


Рисунок 10 – Обработанное изображение при $a = 1$

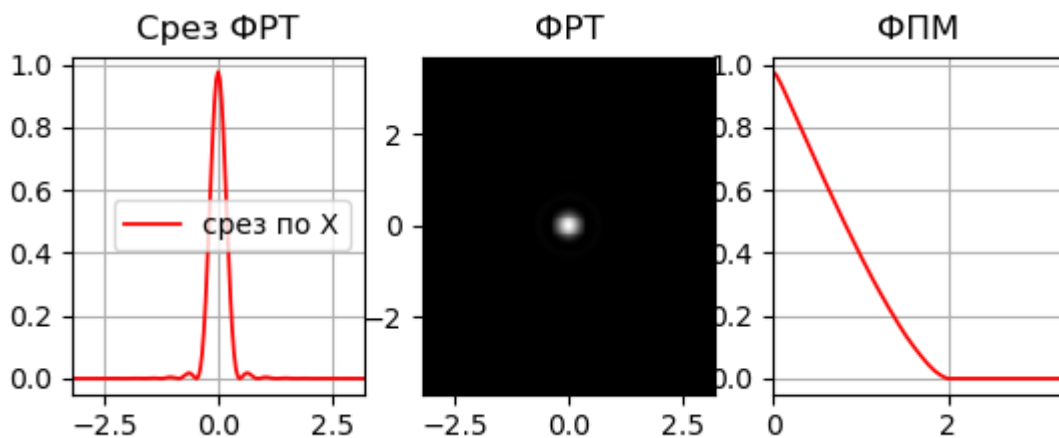


Рисунок 11 – ФРТ и ФПМ при $a = 1$

Вывод:

В ходе работы разработана программа для моделирования изображения для разных форм эллиптического зрачка. Как видно из графиков и изображений, чем больше эллиптичность зрачка, тем больше горизонтальный смаз изображения.

Текст программы:

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

# Load the image
image = plt.imread('cat.jpeg')

# Determine the image dimensions
height, width, _ = image.shape

# Determine the minimum size of the image
minSize = min(height, width)

# Crop the image to a square size
croppedImage = image[:minSize-1, :minSize-1, :]

# Resize the image to [512, 512]
resizedImage = np.array(Image.fromarray(croppedImage).resize((512, 512)))
item = np.mean(resizedImage, axis=-1) # Convert to grayscale

N = 512
A = 0.5
lambda_val = 0.5
D_zr = 20
step_zr = D_zr / N
step_it = 1 / (N * step_zr)
step_im = step_it * lambda_val / A
[im_axis_X, im_axis_Y] = np.meshgrid(
    np.arange(-(N/2) * step_im, (N/2) * step_im, step_im),
    np.arange(-(N/2) * step_im, (N/2) * step_im, step_im)
)

pupil = np.zeros((N, N))
x, y = np.meshgrid(
    np.arange(-(N/2) * step_zr, (N/2) * step_zr, step_zr),
    np.arange(-(N/2) * step_zr, (N/2) * step_zr, step_zr)
)
a = 1
for i in range(N-1):
    for j in range(N-1):
        if (x[i, j]**2)/(a**2) + y[i, j]**2 < 1:
            pupil[i, j] = 1

# Incoherent image
fft_intensity = 1 / N *
np.fft.fftshift(np.fft.fft2(np.fft.fftshift(np.abs(item))))
fft_func_rasp = 0.25 * N *
np.fft.fftshift(np.fft.fft2(np.fft.fftshift(np.abs(np.fft.fftshift
(np.fft.ifft2(np.fft.fftshift(pupil)))) ** 2)))

func_rasp_img = fft_intensity * fft_func_rasp
intensity_rasp_img = N *
```



```

np.fft.fftshift(np.fft.ifft2(np.fft.fftshift(func_rasp_img)))
intensity_rasp_img = np.abs(intensity_rasp_img) ** 2
intensity_rasp_img = intensity_rasp_img[::-1, :]

# Plot results
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.pcolormesh(x, y, pupil, cmap='gray')
plt.axis('equal')
plt.title('Зрачек a = 1')

plt.subplot(1, 2, 2)
plt.pcolormesh(im_axis_X, im_axis_Y, intensity_rasp_img,
cmap='gray')
plt.axis('equal')
plt.title("Incoherent Image")

plt.show()

[p_x, p_y] = np.meshgrid(
    np.arange(-(N / 2) * step_zr, (N / 2) * step_zr, step_zr),
    np.arange(-(N / 2) * step_zr, (N / 2) * step_zr, step_zr)
)

n_max = step_it * N / 2
x_max = step_im * N / 2
p_max = step_zr * N / 2

FRT_ = (step_zr / step_it) *
(np.fft.fftshift(np.fft.ifft2(np.fft.fftshift(pupil))) * N)
FRT_abs = (np.abs(FRT_) * np.abs(FRT_)) / (np.pi ** 2) # функция
рассеяния точки
D = (step_it / step_zr) *
(np.fft.fftshift(np.fft.ifft2(np.fft.fftshift(FRT_abs)))) / N #
ОПФ
D_norm = D * np.pi
D_abs = np.abs(D_norm)
# ФПТ
FRT = (np.abs(FRT_) * np.abs(FRT_)) / (np.pi ** 2)
print(len(p_x))
print(len(FRT))
# % Срез ФПТ, ФПТ
plt.figure(5)
plt.subplot(2, 3, 1)
plt.plot(p_x[N // 2 + 1, :], FRT[N // 2 + 1, :], color='r',
linewidth=1.3)
plt.xlim([-x_max / 4, x_max / 4])
plt.grid(True)
plt.legend(['срез по X'])
plt.title('Срез ФПТ')

plt.subplot(2, 3, 2)
plt.pcolormesh(p_x, p_y, FRT, cmap='gray')
plt.axis('equal')
plt.axis([-x_max / 4, x_max / 4, -x_max / 4, x_max / 4])
plt.title('ФПТ')

```

```
#  $\Phi\Pi\Pi$ 
plt.subplot(2, 3, 3)
plt.plot(p_x[N // 2 + 1, :], D_abs[N // 2 + 1, :], color='r',
linewidth=1.3)
plt.xlim([0, p_max / 3])
plt.grid(True)
plt.title('  $\Phi\Pi\Pi$  ')
plt.show()
```