SIM Image Simulation and Reconstruction in Python

Name: Shruti Mishra

Date of Submission: 29th August 2025

1. Introduction

Structured Illumination Microscopy (SIM) is an advanced imaging technique that surpasses the diffraction limit by introducing patterned illumination. The structured patterns shift high-frequency details into the passband of the microscope, enabling higher resolution reconstruction.

The goal of this project is to simulate SIM image formation and reconstruction using Python and analyze the improvements in resolution, both with and without noise.

2. Approach

The methodology followed:
1. Generate a synthetic Ground Truth (GT) image with a scale bar.
2. Apply low pass filtering to simulate microscope blurring.
3. Multiply the blurred image with a sinusoidal grating to simulate structured illumination.
4. Perform Fourier Transform analysis to compare frequency content.
5. Separate low- and high-frequency components of the modulated image.
6. Reconstruct the high-resolution image.
7. Introduce noise (Gaussian/Poisson) to evaluate robustness.

3. Implementation and Results

3.1 Ground Truth Image

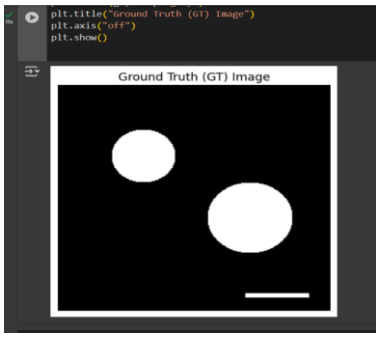
GT image with scale bar

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage import gaussian_filter
from skimage.draw import disk

size = 256
gt = np.zeros((size, size))
rr, cc = disk((80, 80), 30)
gt[rr, cc] = 1
rr, cc = disk((150, 180), 40)
gt[rr, cc] = 1


gt[-20:-15, -80:-20] = 1

plt.imshow(gt, cmap='gray')
plt.title("Ground Truth (GT) Image")
plt.axis("off")
plt.show()
```



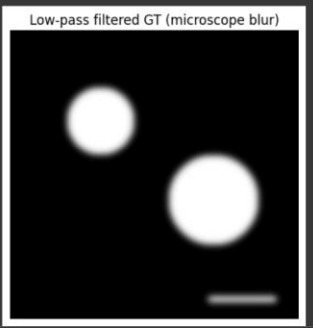
- Contains sharp objects with fine edges.

3.2 Low-pass Filtering

Blurred image after low-pass filter

```
low_passed = gaussian_filter(gt, sigma=3)

plt.imshow(low_passed, cmap='gray')
plt.title("Low-pass filtered GT (microscope blur)")
plt.axis("off")
plt.show()
```



- Mimics real microscopy blur (loss of high frequencies).

3.3 Structured Illumination

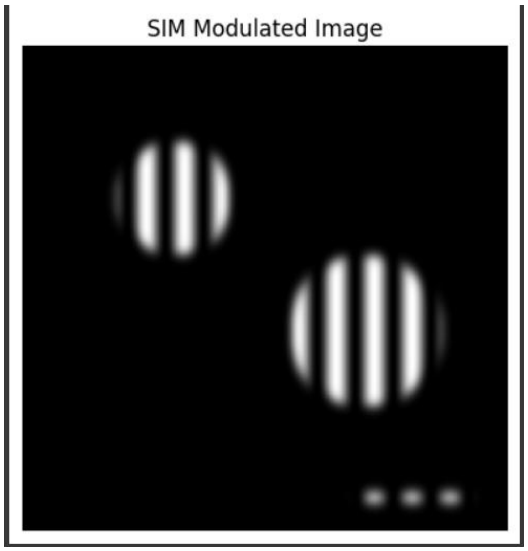
Grating pattern

Modulated image.

```
x = np.arange(size)
y = np.arange(size)
X, Y = np.meshgrid(x, y)
pattern = 0.5 * (1 + np.sin(2 * np.pi * X / 20))


modulated = low_passed * pattern

plt.imshow(modulated, cmap='gray')
plt.title("SIM Modulated Image")
plt.axis("off")
plt.show()
```



Structured illumination was simulated by multiplying the low-pass filtered Ground Truth (GT) image with a sinusoidal grating pattern. The grating is a periodic modulation of intensity, typically

represented as a sinusoidal function across one spatial dimension. This mimics the effect of illuminating the specimen with an interference pattern of light, as is done in actual SIM experiments.

In conventional microscopy, high-frequency details of the object lie outside the optical passband and are therefore lost due to diffraction. By projecting a structured (sinusoidal) illumination pattern onto the sample, these high-frequency object details are **heterodyned (shifted)** into lower frequency bands that the microscope can detect. In other words, the grating mixes the object's fine details with its own frequency, producing sidebands in the Fourier domain. These sidebands fall inside the observable region, making it possible to later extract and reconstruct the otherwise inaccessible high-frequency information.

Thus, the multiplication with a grating pattern is the key mechanism that allows SIM to overcome the diffraction limit.

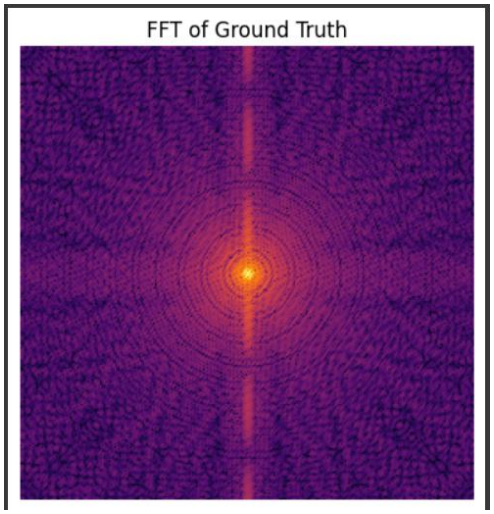
- Stripes shift high-frequency information into observable range.

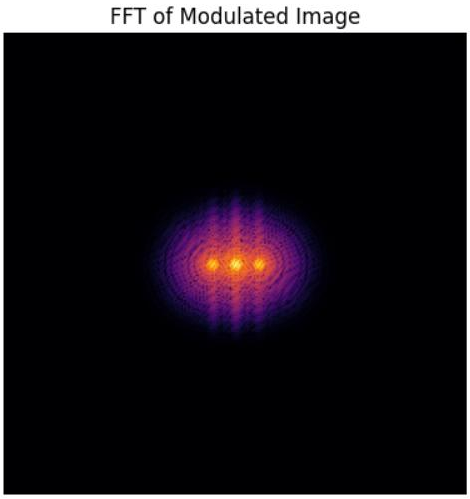
3.4 Fourier Analysis

FFT of GT
FFT of Modulated Image

```
def show_fft(img, title):
    f = np.fft.fftshift(np.fft.fft2(img))
    magnitude = np.log1p(np.abs(f))
    plt.imshow(magnitude, cmap='inferno')
    plt.title(title)
    plt.axis("off")
    plt.show()

show_fft(gt, "FFT of Ground Truth")
show_fft(modulated, "FFT of Modulated Image")
```





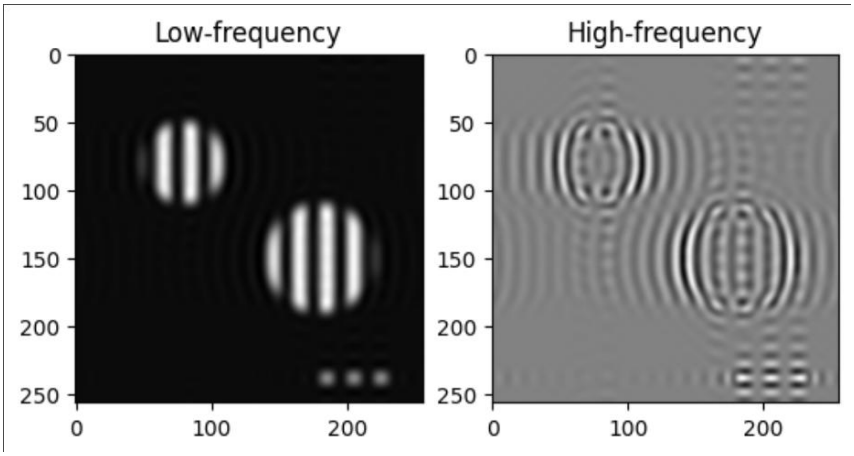
Compared to GT, the modulated image shows clear **frequency shifts (sidebands)** at $\pm k_i$ that are absent in the GT FFT. These sidebands encode otherwise inaccessible high-frequency object information within the observable region, enabling SIM's resolution gain.

-Sidebands are visible in modulated FFT, showing frequency shifts.

3.5 Frequency Separation

Low-frequency component & High-frequency component

To separate the low- and high-frequency components of the modulated image, a frequency-domain filtering approach was used. First, the modulated image was transformed into the frequency domain using the two-dimensional Fast Fourier Transform (FFT). In the Fourier spectrum, the central region represents the low-frequency content (smooth background), while the high-frequency components, including the shifted sidebands introduced by structured illumination, appear away from the center. A circular low-pass mask was applied around the central region to isolate the low-frequency part of the spectrum. The inverse FFT of this masked spectrum yielded the low-frequency image. The high-frequency component was then obtained by subtracting this low-frequency image from the modulated image. This method effectively separates smooth intensity variations from fine structural details, enabling reconstruction of higher-resolution images.




```
f_mod = np.fft.fftshift(np.fft.fft2(modulated))
rows, cols = f_mod.shape
crow, ccol = rows//2, cols//2


mask = np.zeros_like(f_mod)
mask[crow-20:crow+20, ccol-20:ccol+20] = 1
low_freq = np.fft.ifft2(np.fft.ifftshift(f_mod * mask)).real


high_freq = modulated - low_freq

plt.subplot(1,2,1)
plt.imshow(low_freq, cmap='gray')
plt.title("Low-frequency")

plt.subplot(1,2,2)
plt.imshow(high_freq, cmap='gray')
plt.title("High-frequency")
plt.show()
```

- Filtering allows isolation of useful high-frequency details.

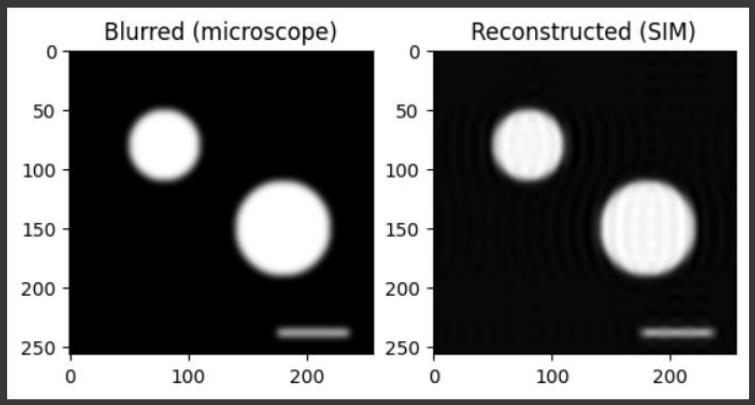
3.6 Reconstruction

Reconstructed image

```
reconstructed = low_passed + high_freq

plt.subplot(1,2,1)
plt.imshow(low_passed, cmap='gray')
plt.title("Blurred (microscope)")

plt.subplot(1,2,2)
plt.imshow(reconstructed, cmap='gray')
plt.title("Reconstructed (SIM)")
plt.show()
```



- Recovered sharper details compared to blurred image.

3.7 Noise Simulation

Reconstructed with Gaussian noise.

To simulate real microscopy conditions, random noise was added to the reconstructed images.

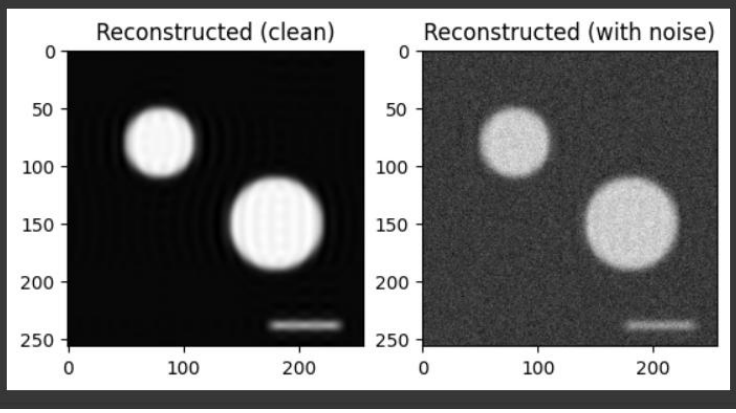
- **Gaussian noise**: represents electronic noise from detectors and background fluctuations. It was added using random samples from a normal distribution scaled by a noise factor.

After introducing noise, the workflow from Steps 2–5 (low-pass filtering, structured illumination, Fourier transform, and frequency separation) was repeated. The presence of noise significantly degraded the quality of the reconstructed images. Fine details became less distinct, and Fourier spectra appeared noisier, making frequency separation less effective. However, despite the degradation, the structured illumination approach still improved resolution compared to the purely blurred (low pass) image. This highlights that while SIM enhances resolution, it is highly sensitive to noise levels, which is a critical limitation in practical applications.

```
noisy = reconstructed + 0.1 * np.random.normal(size=reconstructed.shape)

plt.subplot(1,2,1)
plt.imshow(reconstructed, cmap='gray')
plt.title("Reconstructed (clean)")

plt.subplot(1,2,2)
plt.imshow(noisy, cmap='gray')
plt.title("Reconstructed (with noise)")
plt.show()
```



- Noise reduces clarity but SIM still improves resolution.

4. Results Summary

Step	Image	Observation
GT	Sharp objects	Full frequency content
Low pass	Blurred	High frequencies lost
Modulated	Patterned stripes	Frequency shifted
FFT	Sidebands visible	SIM shifts frequencies
Reconstruction	Sharper image	Higher resolution
With noise	Degraded	Noise sensitivity observed

5. Conclusion

- SIM effectively improves resolution beyond the microscope's diffraction limit.
- Fourier analysis confirms frequency shifting due to illumination patterns.
- Reconstruction is significantly sharper than the blurred image.

• Noise negatively impacts results, but improvements remain visible.
• The simulation validates SIM principles and highlights its sensitivity to experimental conditions.

6. References

1. Gustafsson, M. G. L. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. Journal of Microscopy.
2. Python Libraries: NumPy, SciPy, Matplotlib, scikit-image.