DIP Assignment-3 Report

(Kedarnath P - 20902)

(1) Radial Sinusoid and its Frequency Response

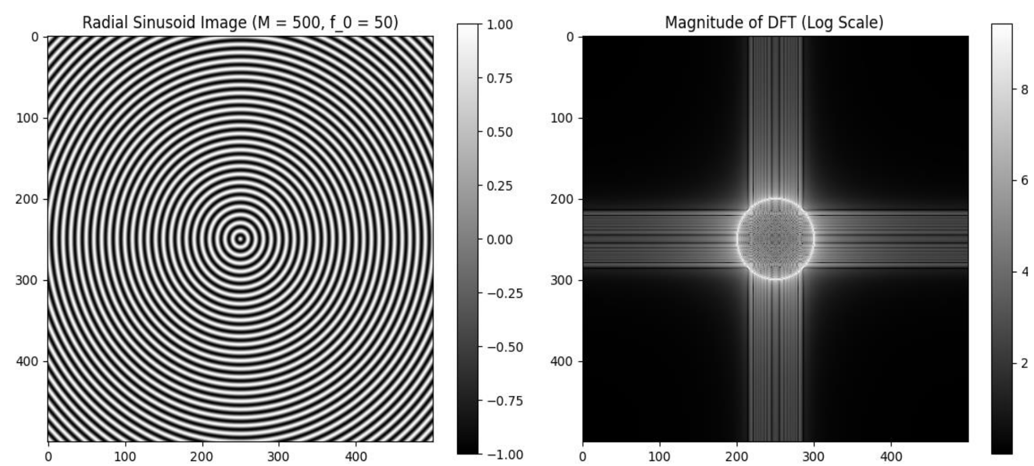
Implementation Details:

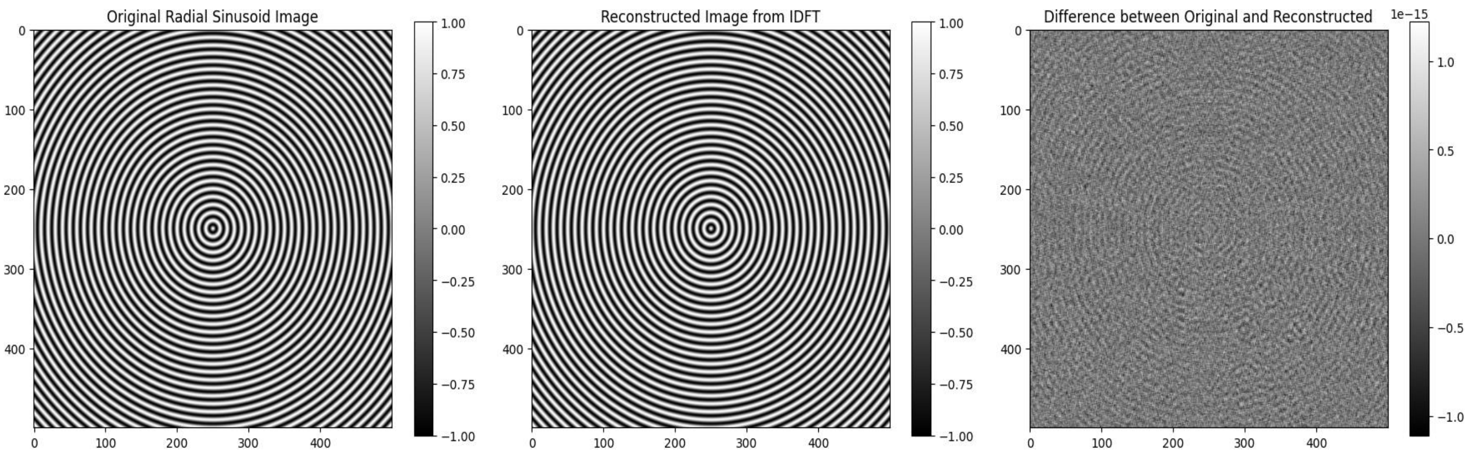
- **Radial Sinusoid Image Generation:** Utilized a meshgrid for u and v and calculated the distance `D(u,v)` from the center for each pixel in the image to generate the radial sinusoid image.
- **DFT Computation:** The 2D Fast Fourier Transform (FFT) was computed using numpy's `fft2` method.
- **Visualization of DFT:** The DFT was cyclically shifted to the center using `fftshift`, and its magnitude was visualized both with and without a logarithmic scale.
- **IDFT Computation:** The Inverse 2D Fast Fourier Transform (IFFT) was computed to get back the reconstructed image.
- **Comparison:** Visualized the original image, the reconstructed image from IDFT, and their difference.
- **Frequency Variation:** Changed the frequency f_0 and visualized the effect on the radial sinusoid and its DFT magnitude.

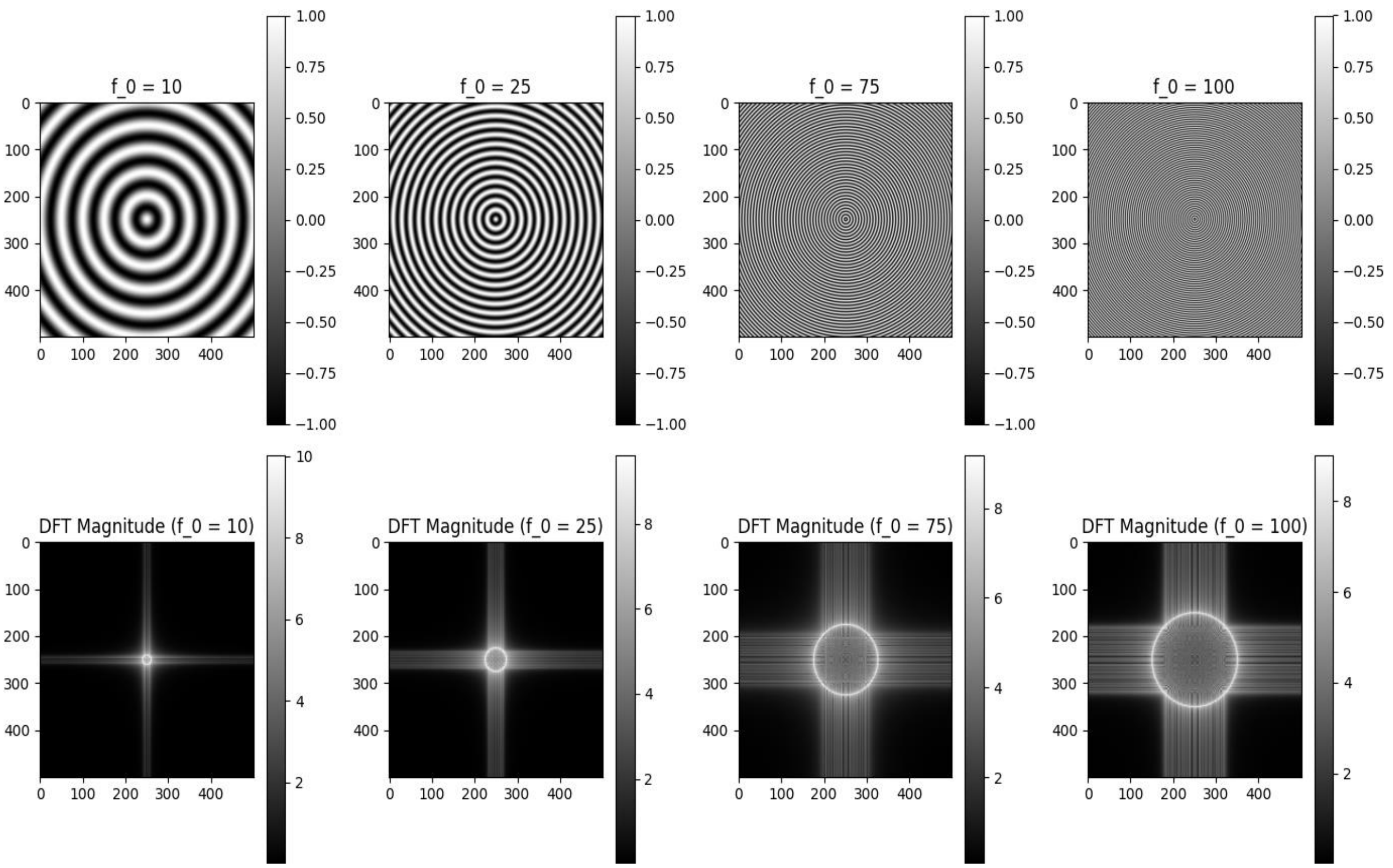
Observations and Analysis:

1. **Radial Sinusoid Image**:
    - The generated image displayed a clear radial pattern originating from its center, which is a manifestation of the cosine function modulated by distance from the center.
    - The density of the oscillations in the radial pattern was directly proportional to f_0 . Higher frequencies resulted in more oscillations within the same space, leading to a tighter radial pattern.
2. **DFT Magnitude**:
    - The DFT response of the radial sinusoid exhibited a central peak, representing the DC component or the average value of the image. This is a consistent feature irrespective of the sinusoidal frequency.
    - The radial patterns (bright rings) in the DFT magnitude spread out further from the center as f_0 increased. This is because higher frequencies in the spatial domain correspond to wider spacing in the frequency domain.
3. **Image Reconstruction**:
    - The reconstructed image from the IDFT was nearly identical to the original image, with a maximum difference on the order of 10^{-15} (`1.2212453270876722e-15`). This negligible difference reaffirms the lossless transformation property of DFT and its inverse.

Visualization:




Original Radial Sinusoid Image
Reconstructed Image from IDFT
Difference between Original and Reconstructed
1e−15


f_0 = 10
f_0 = 25
f_0 = 75
f_0 = 100
DFT Magnitude (f_0 = 10)
DFT Magnitude (f_0 = 25)
DFT Magnitude (f_0 = 75)
DFT Magnitude (f_0 = 100)

(2) **Frequency Domain Filtering**

Implementation Details:

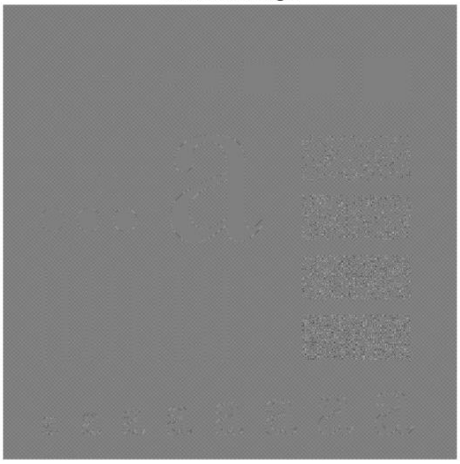
- **Image Loading and Normalization**: The image 'characters.tif' was loaded and was normalized to lie within the intensity range [0, 1].
- **Image Centering for DFT**: The image was preprocessed by centering, i.e., multiplying pixel intensities by $(-1)^{x+y}$. This step aids in shifting the zero-frequency component to the center of the spectrum.
- **Frequency Domain Transformation**: The 2D Fast Fourier Transform (FFT) of the centered image was computed.
- **Magnitude Spectrum Visualization**: The magnitude spectrum of the DFT was displayed, providing a visual representation of the frequency content of the image. This showcases the spread of different frequency components across the image.
- **Filter Design**: Two types of low-pass filters, ILPF (Ideal Low Pass Filter) and GLPF (Gaussian Low Pass Filter), were designed with a specified cutoff frequency $D_0 = 100$.
- **Frequency Domain Filtering**: The image's frequency domain representation (DFT) was multiplied by each of the two designed filters. This step effectively removes (or attenuates) the frequencies outside the filter's passband.
- **Spatial Domain Recovery**: The inverse DFT (Inverse FFT) was computed on the filtered frequency domain representations to obtain the spatial domain images post-filtering.
- **Visualizations**: Various stages of the process were visualized, which includes the original image, the centered image, the magnitude spectrum of the DFT, the designed ILPF and GLPF and Images obtained after filtering with both ILPF and GLPF.

Observations and Analysis:

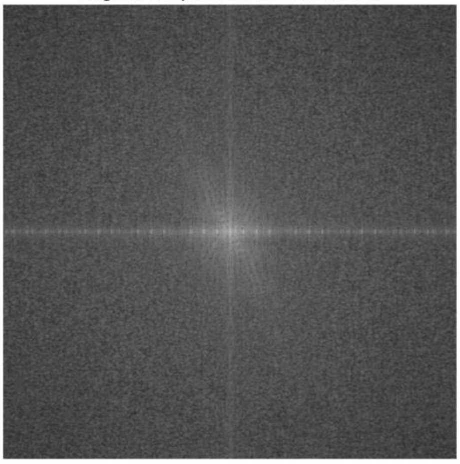
- **Image Centering**: Centering the image before computing the DFT ensures that the main frequency components are centered in the frequency domain representation. This is particularly beneficial when designing and applying filters.
- **Magnitude Spectrum**: Visualizing the magnitude spectrum helps in understanding the distribution of frequencies in the image. Bright regions in the spectrum represent dominant frequency components, while darker areas represent less dominant ones.
- **ILPF Filtering**: Using an Ideal Low Pass Filter tends to produce sharp cutoffs in the frequency domain. This can result in ringing artifacts (Gibbs phenomena) in the spatial domain due to the abrupt transition.
- **GLPF Filtering**: Gaussian Low Pass Filters have a smooth transition in the frequency domain, leading to smoother images in the spatial domain, typically with fewer artifacts than the ILPF.
- **Cutoff Frequency Impact**: A cutoff frequency $D_0 = 100$ was used. If you vary this value, the amount of detail retained in the filtered image will also vary. Lower values of D_0 (low frequencies i-e, central part of the Fourier-transformed image) will retain only the basic structure of the image, leading to a blurrier appearance. In contrast, higher values of D_0 (High-frequency components which correspond to fine details, sharp edges, and noise) will retain more of the original image's details and edges.

Visualization:

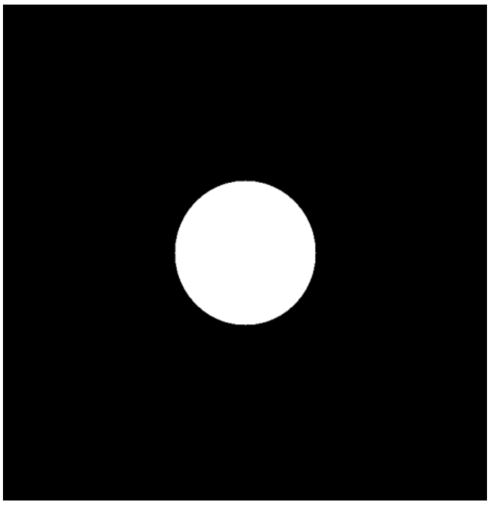
Centered Image



Magnitude Spectrum (Centered DFT)



Ideal Low Pass Filter (D0 = 100)



Gaussian Low Pass Filter (D0 = 100)

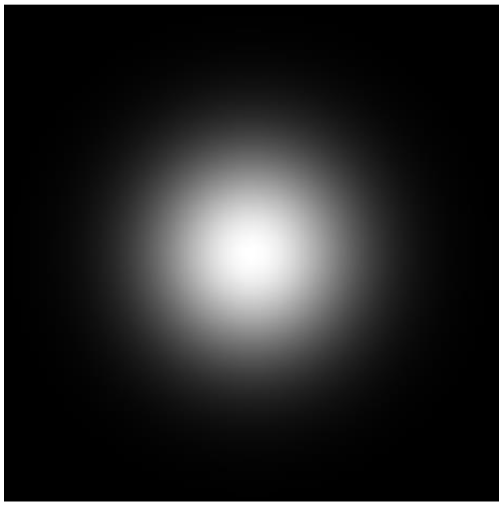


Image Filtered with ILPF

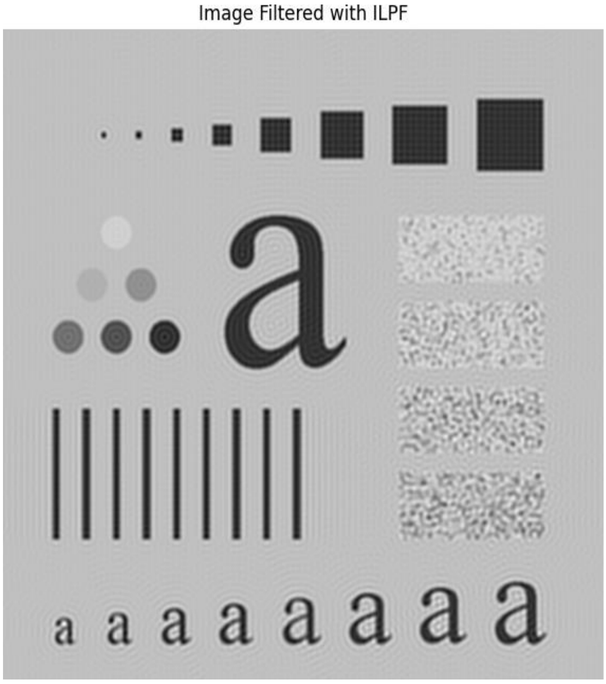
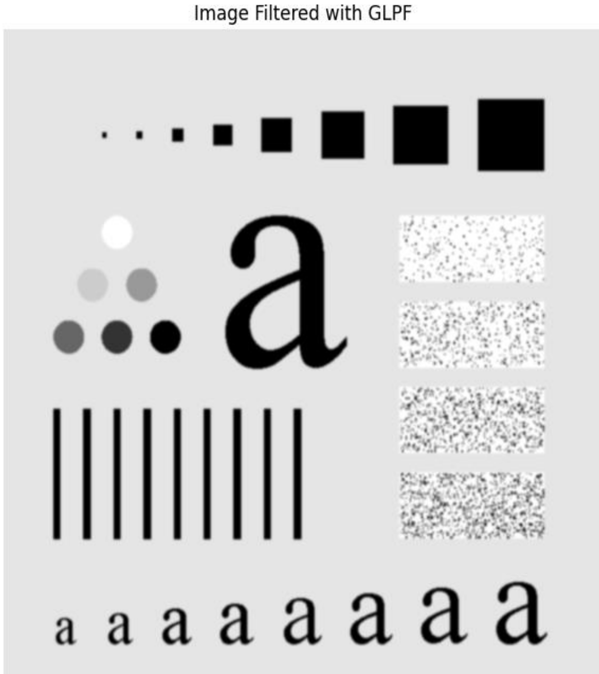


Image Filtered with GLPF



(3) **Image Deblurring**

Methodology, Observations and Analysis:

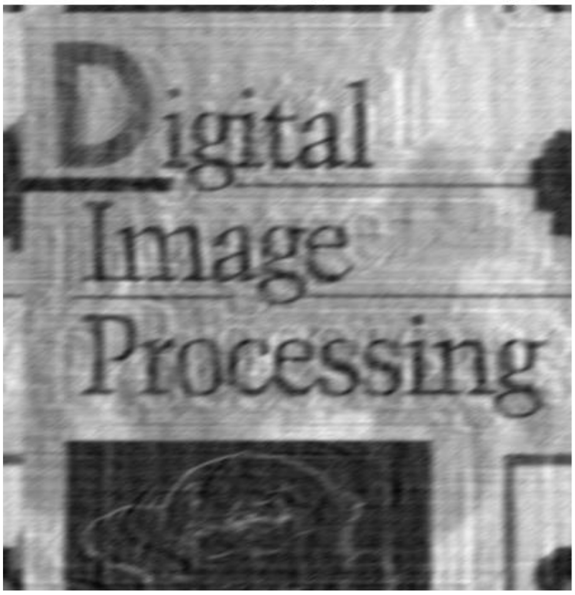
- **Image Loading and Initial Examination**:
  - Two images were provided for deblurring: `Blurred_LowNoise.png` (with white noise having a standard deviation $\sigma = 1$) and `Blurred_HighNoise.png` (with white noise having a standard deviation $\sigma = 10$).
  - Both images appeared visibly blurred, and the high noise image displayed more pronounced noise artifacts.
- **Kernel Examination**:
  - The provided blur kernel was centered at the coordinate (9, 24).
  - Before deblurring operations, this kernel was shifted so its center moved to the coordinate (0, 0), as specified.
- **Inverse Filtering**:
  - Inverse filtering was used to attempt to reverse the blurring effect.
  - Direct inverse filtering can amplify noise, especially at frequencies where the blur kernel has values close to zero. To mitigate this, a thresholding approach was applied. If the FFT value of the blur kernel was below a threshold of 0.1, the inverse filter FFT value was set to 0.
  - For the low noise image, the inverse filtering showed reasonable deblurring. However, for the high noise image, the inverse filtering was less effective due to the amplified noise.
- **Wiener Filtering**:
  - Wiener filtering was used as a more robust deblurring method, especially in the presence of noise.
  - The PSD of the original signal was computed using a power law, and the PSD of the noise was assumed to be equal to the noise variance (consistent for white noise).
  - Wiener filtering balances between inverse filtering and noise smoothing. For the low noise image, it provided a clearer result compared to inverse filtering. For the high noise image, the result was noticeably better than the one obtained using inverse filtering, showcasing the filter's ability to suppress noise while deblurring.

In conclusion,

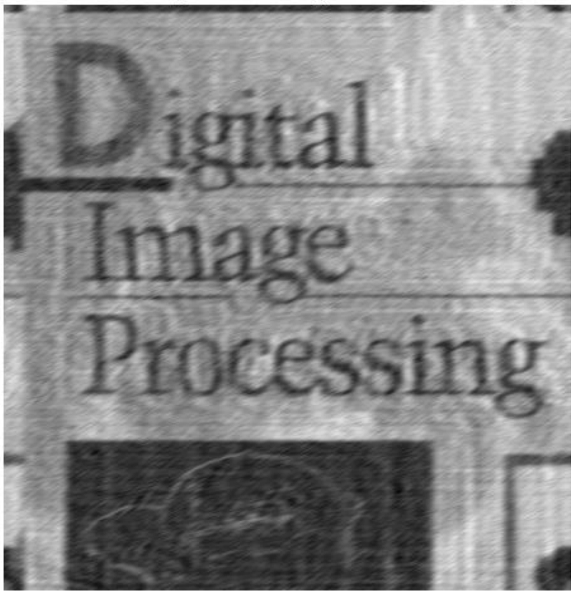
while inverse filtering can be effective for images with low noise levels, its vulnerability to noise amplification makes it less ideal for high noise images. Wiener filtering, on the other hand, provides more consistent deblurring results for both low and high noise scenarios.

Visualization:

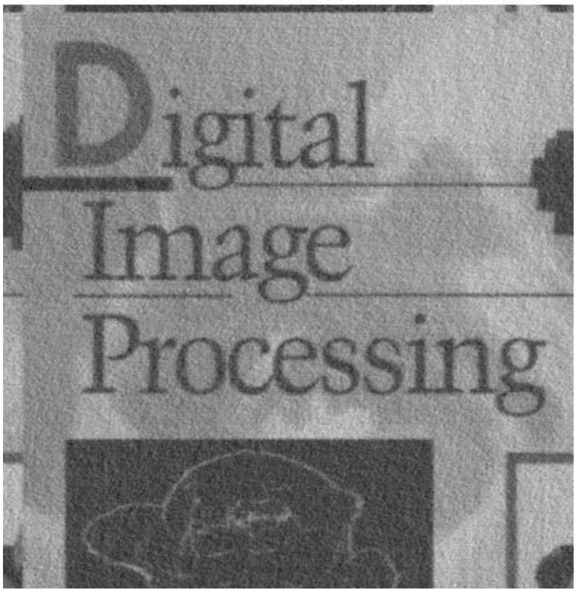
Deblurred Low Noise Image (Inverse Filtering)



Deblurred High Noise Image (Inverse Filtering)



Deblurred Low Noise Image (Wiener Filtering)



Deblurred High Noise Image (Wiener Filtering)