DIP Assignment 3

Pritam Kumar
Sr No - 24021

October 21, 2025

1 Directional Filtering

Directional filtering improves an image by keeping or removing details in certain directions. It helps highlight features like edges, lines, or textures with specific orientations. This method can also reduce noise, clarify important patterns, and improve feature detection in object recognition, medical imaging, and texture analysis tasks.

$$\theta(u,v) = \tan^{-1}\left(\frac{v - \frac{N}{2}}{u - \frac{M}{2}}\right)$$

$$H(u,v;\theta_{\min},\theta_{\max}) = \begin{cases} 1, & \text{if } \theta_{\min} \le \theta(u,v) \le \theta_{\max} \\ 0, & \text{otherwise} \end{cases}$$

a. Generate an image x of size $M \times M$

$$x_1(m,n) = \sin\left(\frac{2\pi \cdot 12\, m}{M}\right), \qquad m,n = 0,1,\ldots,M-1$$

$$X_1(u,v) = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} x_1(m,n)\, e^{-j\frac{2\pi}{M}(mu+nv)}$$

Since x_1 depends only on m,

$$\sum_{n=0}^{M-1} e^{-j\frac{2\pi}{M}\ell n} = M\,\delta(0)$$

so

$$X_1(u,v) = M\,\delta(0)\sum_{m=0}^{M-1} \sin\left(\frac{2\pi \cdot 12\, m}{M}\right) e^{-j\frac{2\pi}{M}mu}.$$

Expressing the sine term as complex exponentials:

$$\sin\left(\frac{2\pi \cdot 12\, m}{M}\right) = \frac{1}{2j}\left(e^{j\frac{2\pi}{M}12m} - e^{-j\frac{2\pi}{M}12m}\right)$$

Substitute and simplify:

$$\sum_{m=0}^{M-1} \sin\left(\frac{2\pi \cdot 12\, m}{M}\right) e^{-j\frac{2\pi}{M}mu} = \frac{1}{2j}\left(\sum_{m} e^{-j\frac{2\pi}{M}(u-12)m} - \sum_{m} e^{-j\frac{2\pi}{M}(u+12)m}\right)$$

Each exponential sum equals M if its frequency index is zero (mod M), otherwise zero:

$$\sum_{m=0}^{M-1} e^{-j\frac{2\pi}{M}rm} = M\, \delta_{r \bmod M, 0}$$

Hence,

$$\sum_{m=0}^{M-1} \sin\left(\frac{2\pi \cdot 12\, m}{M}\right) e^{-j\frac{2\pi}{M}mu} = \frac{M}{2j}\big(\delta(u-12) - \delta(u+12)\big)$$

Therefore, the 2D DFT of $x_1(m,n)$ is:

$$X_1(u,v) = \frac{M^2}{2j}\, \delta(v)\, \big(\delta(u-12) - \delta(u+12)\big)$$

Or, we can write as

$$\boxed{\boldsymbol{X_1(u,v) = \frac{M^2}{2j}\, \big[\delta(u-12, v) - \delta(u+12, v)\big]}}$$

To center the DFT, we can do

$$\hat{X}_1(u,v) = X_1\big(u - \tfrac{M}{2},\, v - \tfrac{N}{2}\big)$$

So, the final centered DFT is given by

$$\mathbf{X}_1(u,v) = \frac{M^2}{2j}\left[\delta\big(u - \tfrac{M}{2} - 12,\, v - \tfrac{M}{2}\big) - \delta\left(u - \tfrac{M}{2} + 12,\, v - \frac{M}{2}\right)\right]$$

From the above equation, we can see that the magnitude of the 2D centered DFT is nonzero at two locations:

$$(i)\ \big(\tfrac{M}{2} + 12,\, \tfrac{M}{2}\big) \quad \text{and} \quad (ii)\ \big(\tfrac{M}{2} - 12,\, \tfrac{M}{2}\big)$$

The DFT is purely imaginary, and we can plot the magnitude spectrum by taking its absolute value. Figure 1 shows all the plots.

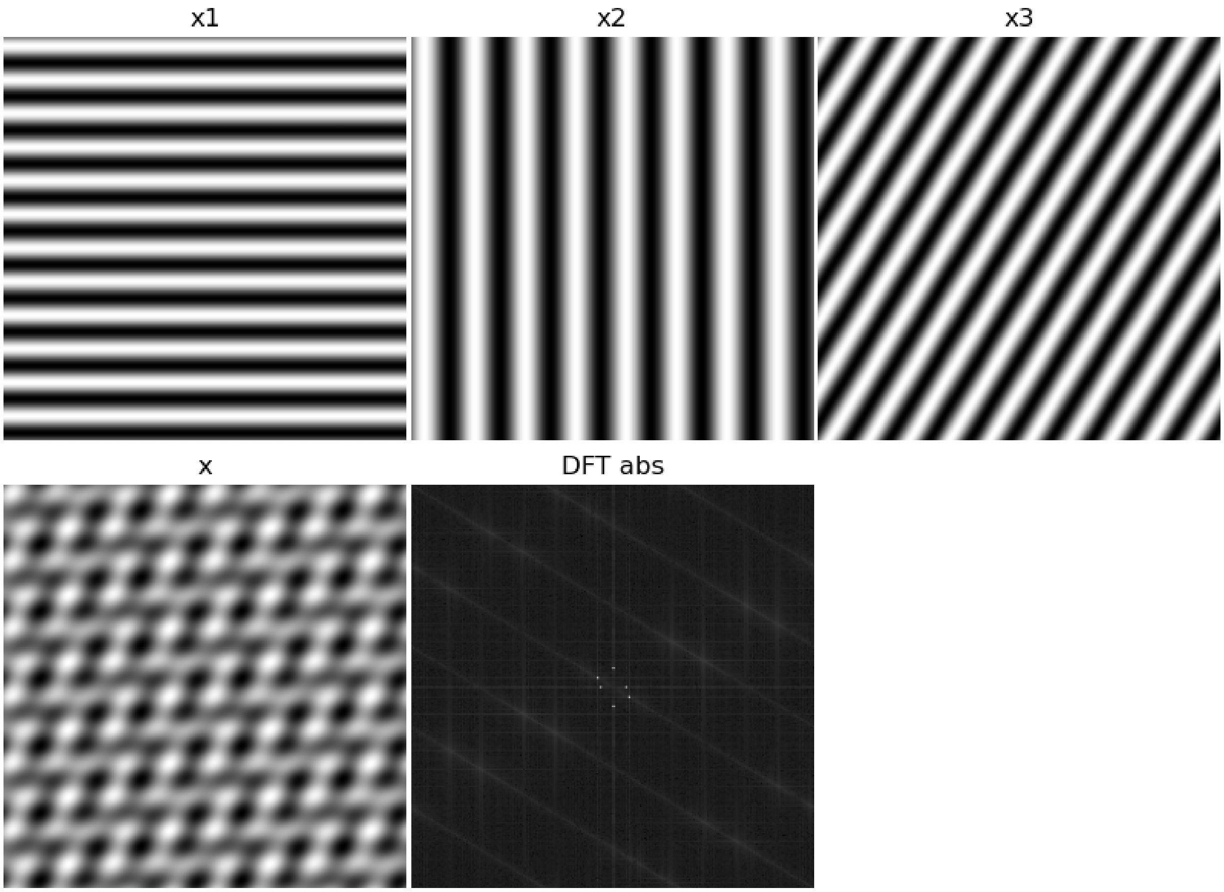

Figure 1: The top-left image is the plot of sinusoidal component $x1(m,n)$, top-center, sinusoidal component $x2(m,n)$, top-right, sinusoidal component $x3(m,n)$, bottom-left combined image, and the last one is the magnitude of the 2D DFT of the combined image

1.1 Individual Sinusoids Analysis:

The Discrete Fourier Transform (DFT) is linear, meaning that for any two signals $x1(m,n)$ and $x2(m,n)$ and constants a and b, we can write

$$DFT\big[ax_1(m,n) + bx_2(m,n)\big] = aDFT(x_1) + bDFT(x_2(m,n)$$

I will explain the individual contribution of DFT. Since the DFT is a `linear transform`, the overall spectrum of a signal composed of multiple components is the **superposition (sum)** of the spectra of its individual components.

x_1 **Horizontal Sinusoid**

- Pattern: Horizontal stripes (varies only along rows/m-direction)
- Frequency Domain: Creates two bright spots on the vertical axis of the magnitude spectrum
- Location: At frequencies (0, ± 12) in the frequency domain

- Contribution: Vertical line of energy in the DFT, representing horizontal periodicity

x_2 **Vertical Sinusoid**

- Pattern: Vertical stripes (varies only along columns/n-direction)
- Frequency Domain: Creates two bright spots on the horizontal axis of the magnitude spectrum
- Location: At frequencies (±8, 0) in the frequency domain
- Contribution: Horizontal line of energy in the DFT, representing vertical periodicity

x_3 **Diagonal Sinusoid**

- Pattern: Diagonal stripes at angle $\theta = \tan^{-1}\left(\frac{10}{6}\right) \approx 59^\circ$
- Frequency Domain: Creates two bright spots along a diagonal line
- Location: At frequencies (±6, ±10) - note the 6:10 ratio matches the spatial frequencies
- Contribution: Diagonal energy pattern representing the slanted periodicity

x **Combined Spectral Analysis**

- Six distinct bright spots (2 from each sinusoid)
- Symmetric pattern around the center (DC component)
- Energy distribution reflecting the spatial orientations:
  - Spots along the **vertical axis** ⇒ **Horizontal stripes** in the image.
  - Spots along the **horizontal axis** ⇒ **Vertical stripes** in the image.
  - **Diagonal spots** ⇒ **Diagonal stripes** oriented along the corresponding direction.

`The direction of frequency spots in the DFT is always perpendicular to the orientation of the stripes in the spatial image.`

b. **Design Directional Filters of size $M \times M$**
I have designed four directional filters named H_1, H_2, H_3, and H_4 Figure 2 shows all four filters. For each directional filter, I have shown the plots. See Figure 3 4 5 6
The frequency of each signal is 24/M, 16/M, (6/M,10/M) (M=256) respectively for signals x_1, x_2, and x_3. $x(m, n)$ will have all the frequencies present in the constituent signal components.

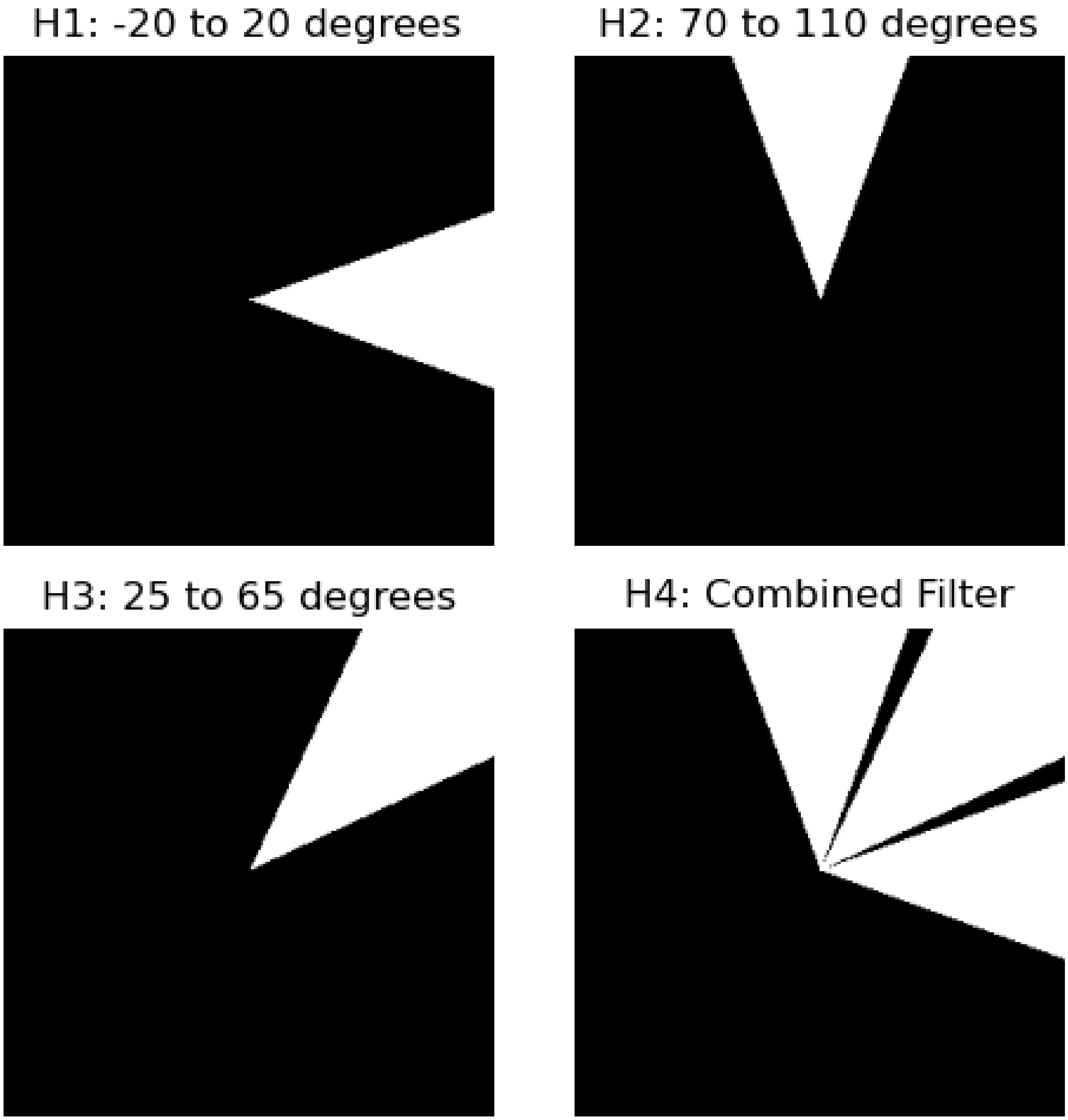

Figure 2: The binary representation of the four filters shows the frequency regions each filter allows or suppresses. The white regions indicate the frequencies that are passed (allowed), while the black regions correspond to frequencies that are blocked (suppressed).

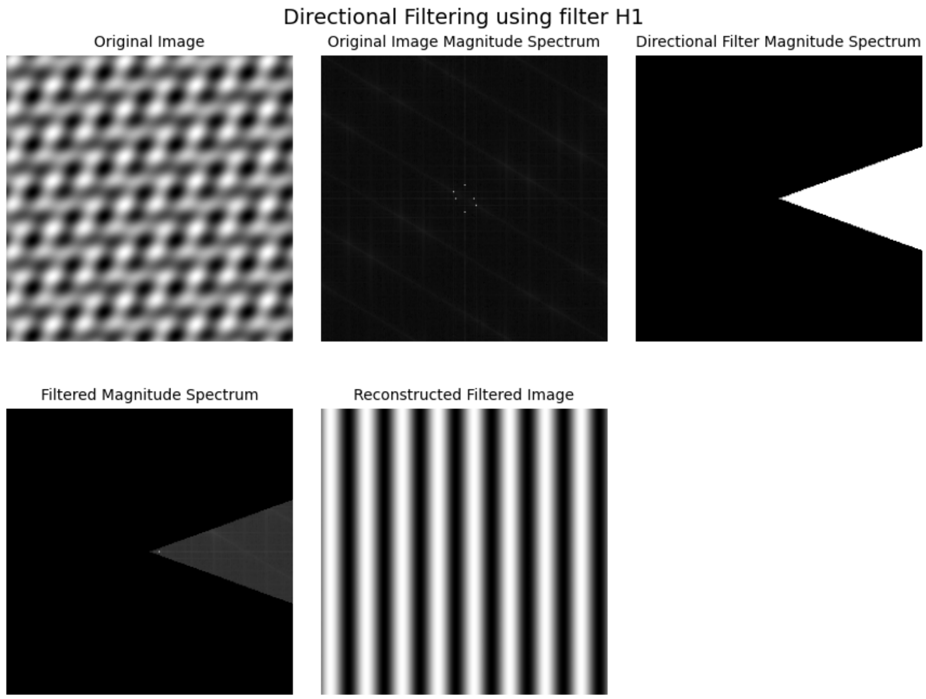

Figure 3: Directional Filter using H1. We can see the frequency passing along the horizontal axis but blocked along the vertical axis.

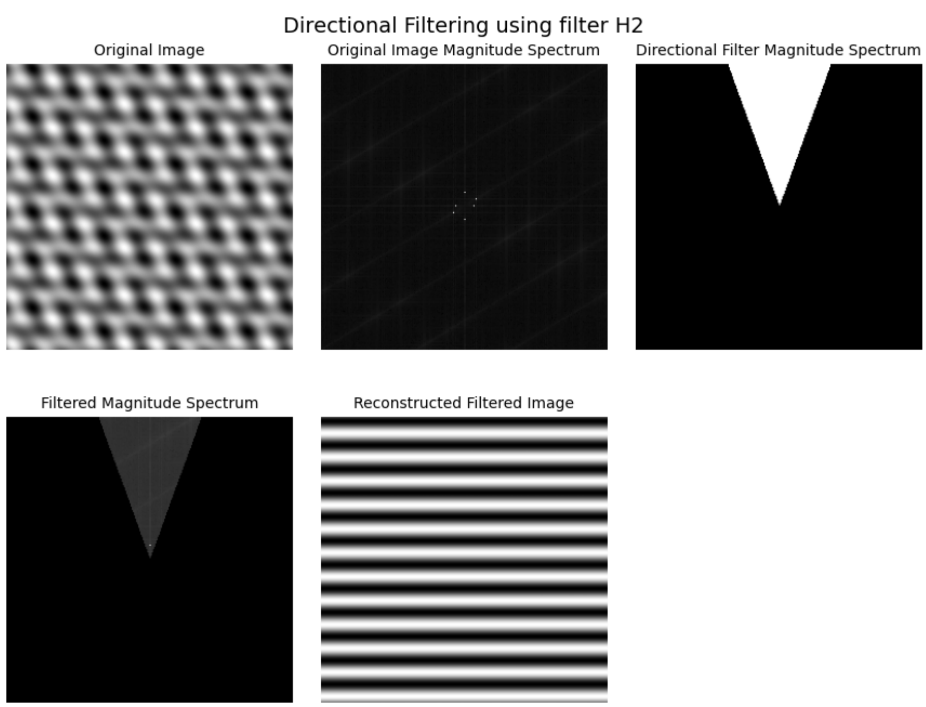

Figure 4: Directional Filter using H2. We can see the frequency passing along the vertical axis, but blocked along the horizontal axis.

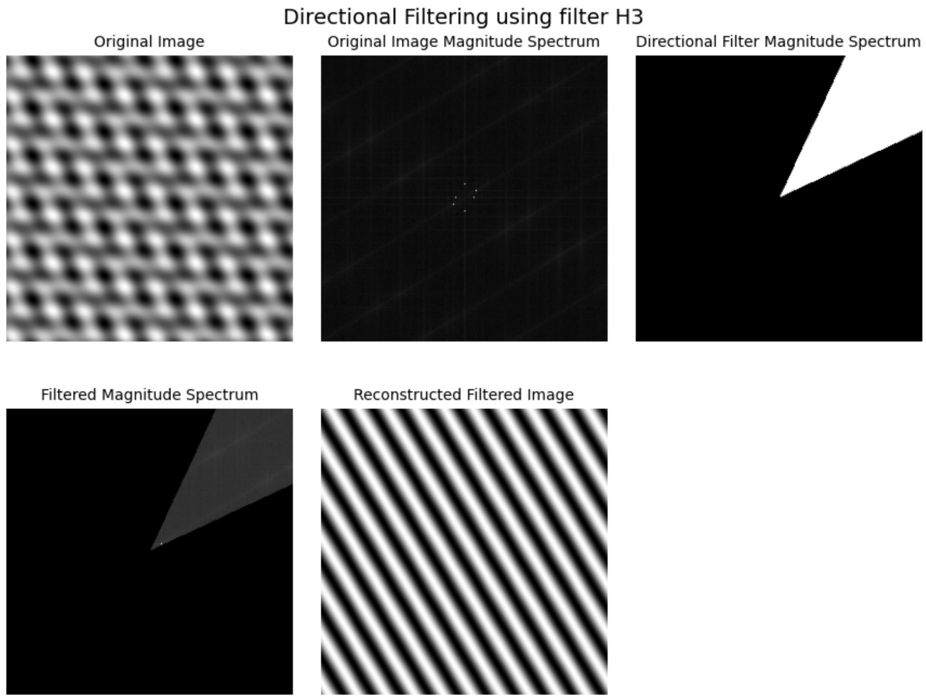

Figure 5: Directional Filter using H3. We can see the frequency passing along the diagonal $(y = x)$ but being blocked along the other diagonal $(y = -x)$.

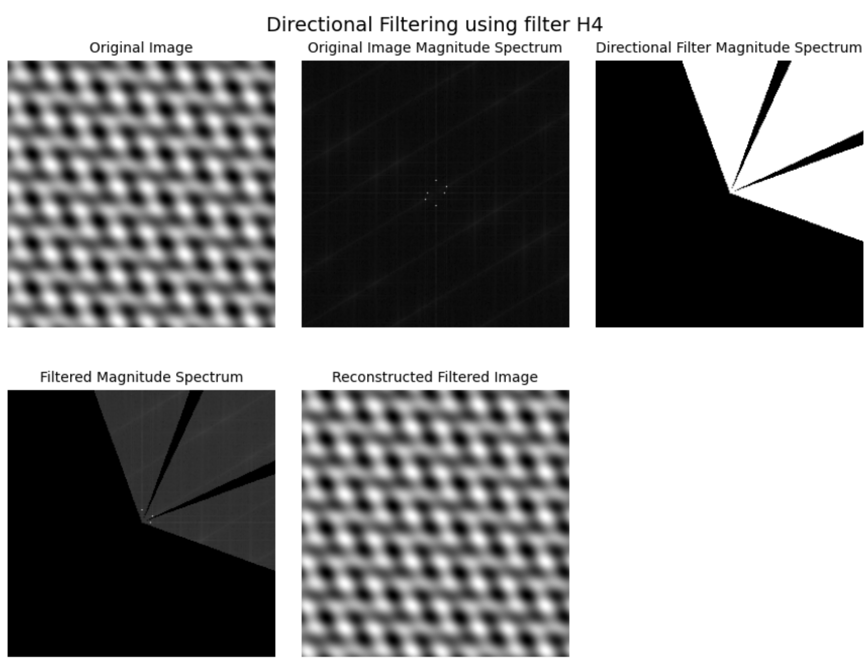

Figure 6: Directional Filter using H4. H4 is the combination or max(H1, H2, H3). These three filters do not overlap, and hence they allow frequency to pass in all directions. So, we are able to reconstruct the original image from the Fourier Transform.

1.2 Observations

- H1: Captures angles -20° to 20° (horizontal directions)
- H2: Captures angles 70° to 110° (vertical directions)
- H3: Captures angles 25° to 65° (diagonal directions)
- H4: Combination of H1 + H2 + H3 (covers all three directional ranges)

H4 reconstructs the image best because it combines all the significant frequency components present in the synthetic image. In the frequency domain, sinusoidal patterns produce sharp peaks along specific directions, which are horizontal, vertical, and diagonal. Filters H1, H2, and H3 each capture only one of these directional components, resulting in partial reconstructions and higher mean square error. On the other hand, H4 is formed by combining H1, H2, and H3, thereby preserving the entire range of frequency information. This comprehensive coverage enables accurate reconstruction since no frequency component is lost. The synthetic image's sparse frequency distribution aligns with the passbands of H4, making it the most effective filter for reconstructing the image with minimal distortion and maximum fidelity.

c. **Mean Square Error:** The Mean Squared Error (MSE) measures reconstruction quality by computing the average squared differences between original and filtered images. H4 achieves the lowest MSE because it combines all three directional filters, preserving complete frequency information from the synthetic sinusoidal image. Individual filters H1, H2, and H3 have higher MSE due to partial frequency loss.

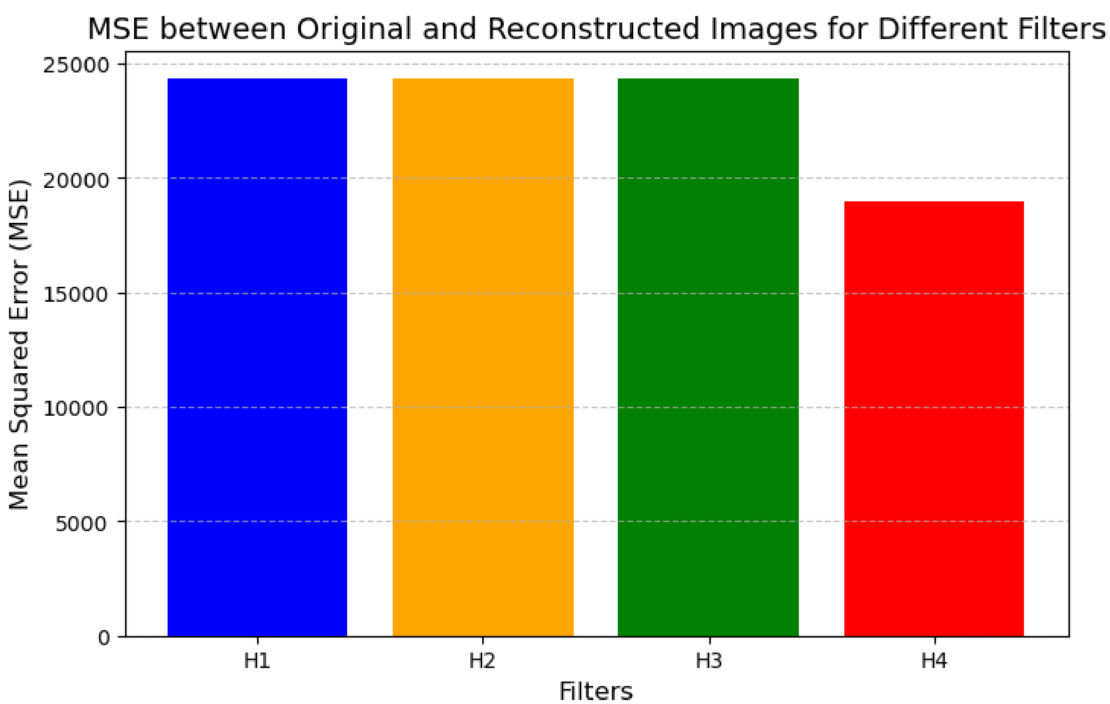

Figure 7: Mean Square Errors

2 Gaussian Blurring and Inverse Filtering

For this question, we have designed a Gaussian Filter of size 13×13. The figure 8 shows the grayscale plot of the filter.

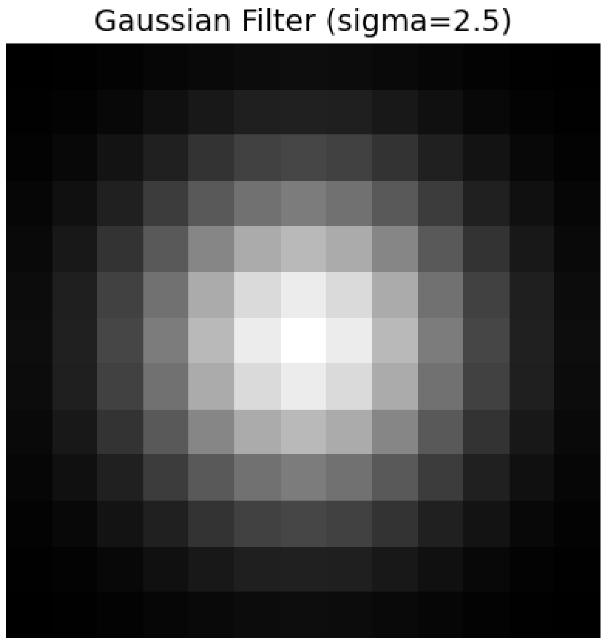

Figure 8: Gaussian Filter. The sum of all the values of the filter is 1.

a. **Frequency Domain Blur** The blurred image is shown in the figure 9

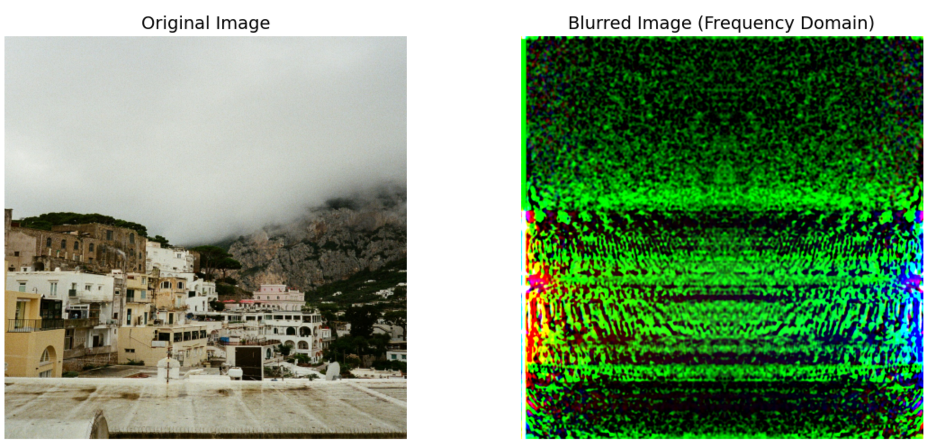

Figure 9: Original and reconstructed Gaussian blurred in the frequency domain

b. **Gaussian Kernel Plot Gaussian Kernel Magnitude Spectrum** The Gaussian kernel's frequency response exhibits 10 a smooth distribution centered at the origin (DC component). This demonstrates the low-pass filtering characteristic - it attenuates high frequencies while preserving low frequencies. The 13×13 kernel shows concentrated energy in the center with rapid decay toward edges, while the 1036×1036 padded version maintains the same shape but with finer frequency resolution due to zero-padding.

Inverse Filter Characteristics The inverse filter displays dramatically different behavior with extremely high values at the periphery, where the original Gaussian response approaches zero. The bright ring pattern in the inverse filter indicates severe amplification of high frequencies. The epsilon term (10^{-3}) prevents division by zero but creates artificial noise amplification in regions where the Gaussian response is naturally weak.

Size Comparison Effects The smaller 13×13 filter shows more aliasing artifacts and coarser frequency representation, while the larger 1036×1036 version provides smoother transitions and better frequency localization. However, both inverse filters exhibit the inherent instability problem; they amplify noise and high-frequency components that were naturally suppressed by the Gaussian blur, making direct inverse filtering impractical for real-world image restoration without regularization techniques.

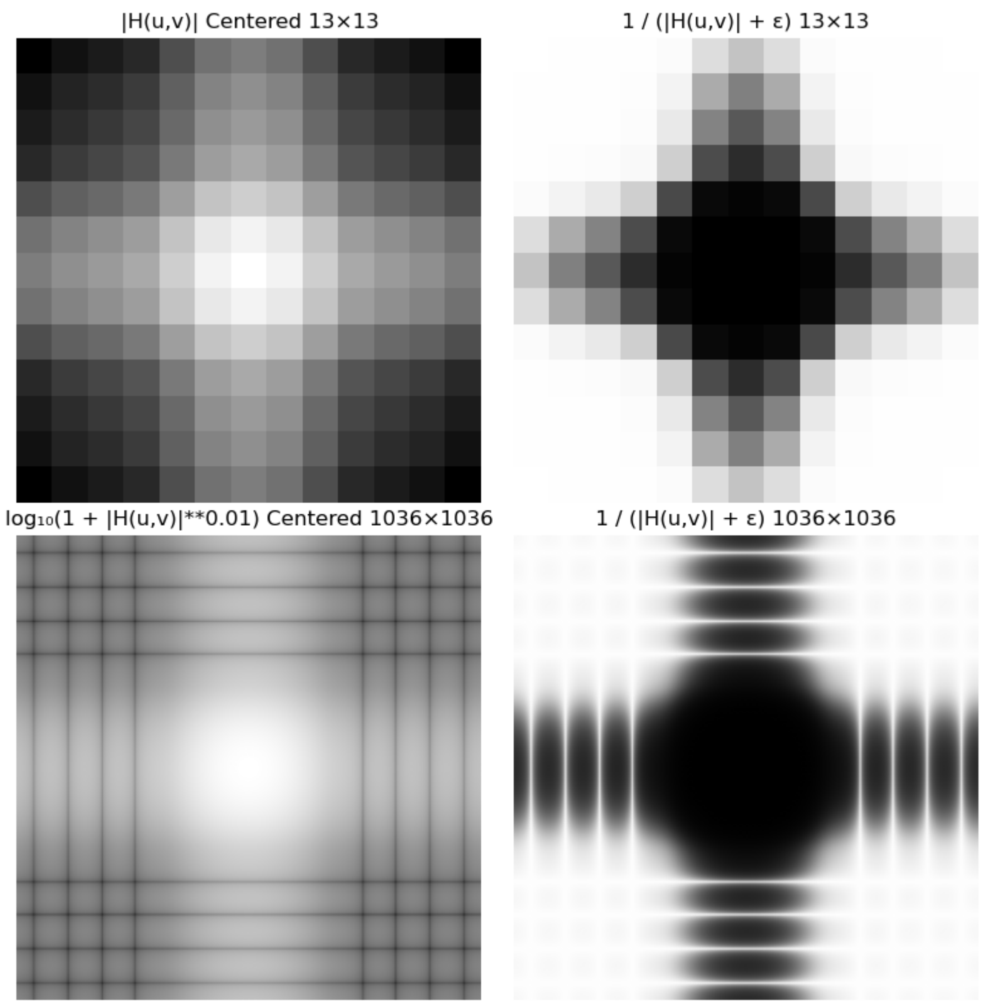

Figure 10: Enter Caption

c. **Gaussian Response Fit:** After modeling the Gaussian Kernel using the given formula, I have obtained the errors that I have plotted in Figure 11. The optimal k_{opt} is $\mathbf{1.13 \times 10^{-4}}$ and the corresponding minimum error is **6.943127**

$$H_{cont}(u, v) = \exp\big(-k(U^2 + V^2)\big), U = u - \frac{M-1}{2}, V = v - \frac{N-1}{2}$$

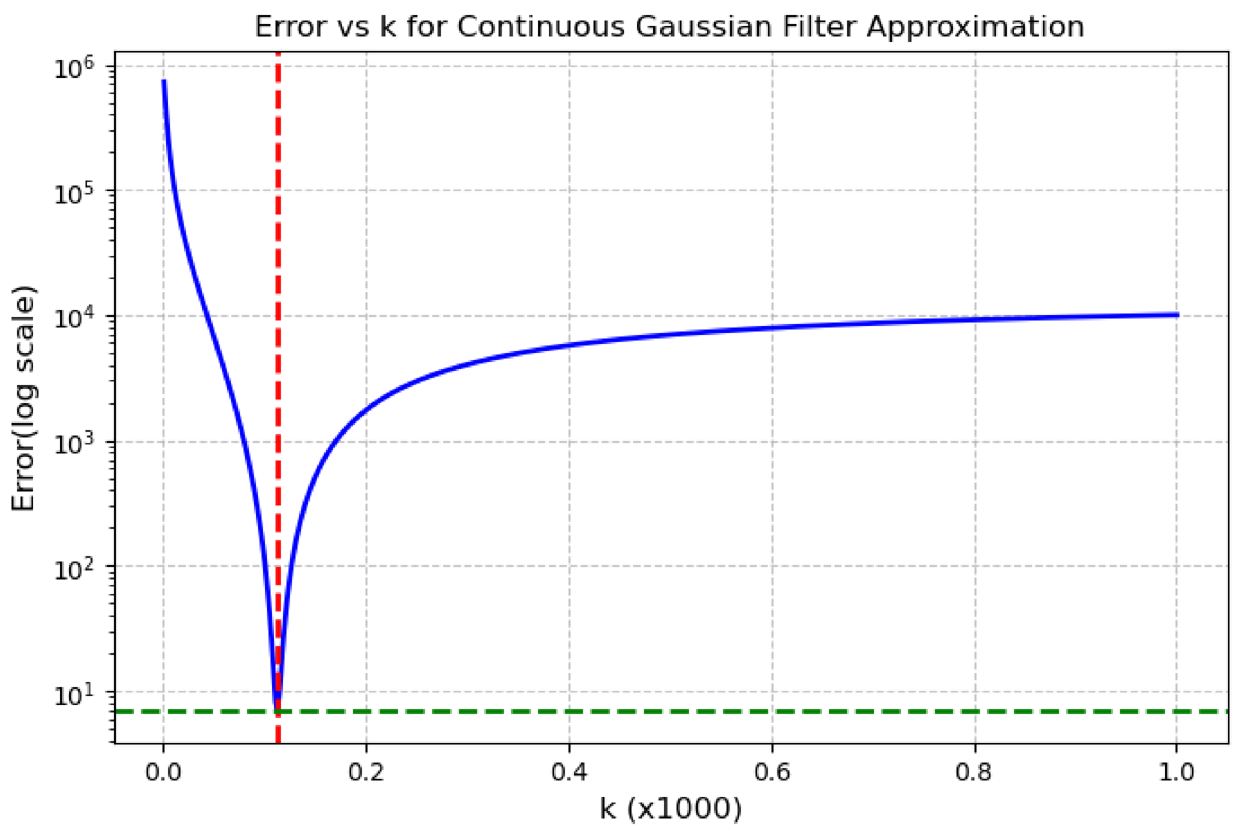

Figure 11: Plot of Errors for different k

Figure 12 shows the combined plot of the magnitude spectrum of the Gaussian fit $H_{cont}(u, v)$ and its inverse. The continuous Gaussian approximation $H_{cont}(u, v)$ shows a smooth, radially symmetric pattern with maximum intensity at the center (DC component) and exponential decay toward the edges. This creates the classic low-pass filtering characteristic where low frequencies are preserved while high frequencies are progressively attenuated. The optimal k value determines the bandwidth. Smaller k values create wider filters that preserve more frequencies, while larger k values create narrower filters with more aggressive high-frequency suppression.
On the other hand, the inverse filter $1/(|H_{cont}(u, v)| + \epsilon)$ exhibits dramatically different behavior with an inverted intensity pattern. It shows minimum values at the center where the original Gaussian has maximum response, and extremely high amplification at the periphery where the Gaussian naturally attenuates. This creates a high-pass amplification effect that theoretically should restore frequencies suppressed by Gaussian blurring.

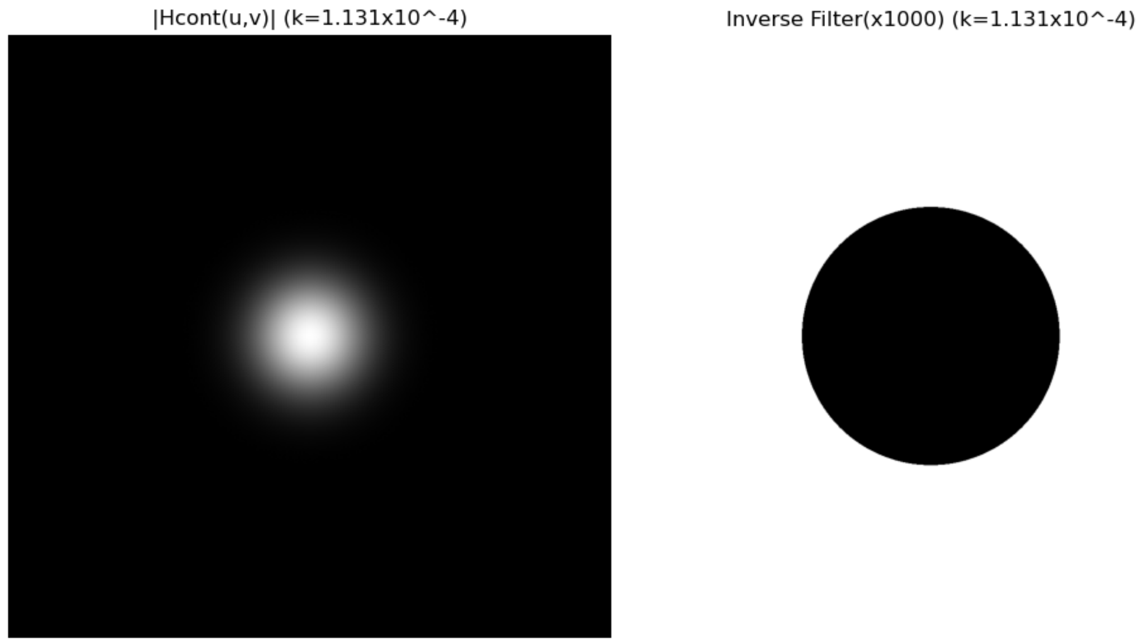

Figure 12: Plots of $|H_{cont}(u, v)|$ and $1/(|H_{cont}(u, v)| + \epsilon)$

d. **Restore the original image:** We have restored the original image from both filters. Figure 13 shows the restored images along with the original image. The mean square errors as

- Mean Squared Error (Continuous Gaussian Inverse Filter): 15.60
- Mean Squared Error (Direct DFT Inverse Filter): 21.95

The lower error for the **continuous method** is because it balances restoration quality with noise control, whereas the direct method may achieve perfect frequency inversion at the cost of catastrophic noise amplification, resulting in higher overall reconstruction error when applied to real images.

Figure 13: Image restored using direct kernel DFT and the Gaussian fit