Indian Institute of Science

Electronics And Communication Engineering Department

Digital Image Processing

Assignment 3

Ansh Bansal

24250

Date: September 29, 2025

Contents

1 Directional Filtering

1.1 Image Generation using Sinusoids

Objective

To generate three 2D sinusoidal images of size 256×256, each representing a unique spatial frequency orientation, combine them into a composite image, and analyze their frequency-domain representation using the 2D Discrete Fourier Transform (DFT).

Implementation Summary

The experiment was implemented in Python using `NumPy` and `Matplotlib`. Three sinusoidal components were defined as follows:

$$
\begin{aligned}
x_1(m,n) &= \sin\left(\frac{2\pi \cdot 12m}{M}\right), \\
x_2(m,n) &= \sin\left(\frac{2\pi \cdot 8n}{M}\right), \\
x_3(m,n) &= \sin\left(\frac{2\pi \cdot (6m+10n)}{M}\right),
\end{aligned}
$$

where $M = N = 256$.

Each sinusoid corresponds to a specific directional frequency:

- $x_1(m,n)$ — horizontal variation (stripes along the m–axis).
- $x_2(m,n)$ — vertical variation (stripes along the n–axis).
- $x_3(m,n)$ — diagonal variation combining both axes.

The combined image is computed as:

$$x(m,n) = \frac{x_1(m,n) + x_2(m,n) + x_3(m,n)}{3}$$

The 2D centered DFT and its log-magnitude spectrum were obtained as:

$$X(u,v) = \texttt{fftshift}\{\texttt{fft2}(x(m,n))\}, \qquad |X(u,v)|_{\log} = \log(1 + |X(u,v)|)$$

Result and Visualization

The generated spatial-domain images and their corresponding DFT log-magnitude spectrum are shown in Fig. 1.1.

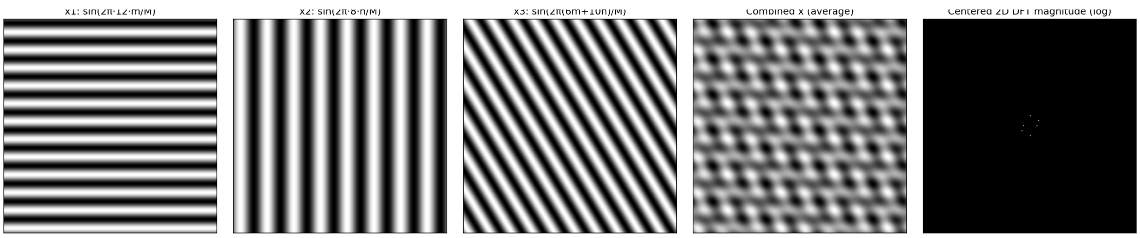

Figure 1.1: Directional filtering – Part (a). First Three: individual sinusoidal components $x_1(m,n)$, $x_2(m,n)$, and $x_3(m,n)$. Last Two: combined image $x(m,n)$ and its centered log-magnitude spectrum.

Inference

The three sinusoidal components $x_1(m,n)$, $x_2(m,n)$, and $x_3(m,n)$ each contribute distinct directional frequency patterns to the overall image:

- The component $x_1(m,n) = \sin\left(\frac{2\pi \cdot 12m}{M}\right)$ varies only along the horizontal axis, producing horizontal stripes in the spatial domain and two bright symmetric peaks along the horizontal frequency axis in the DFT magnitude spectrum.
- The component $x_2(m,n) = \sin\left(\frac{2\pi \cdot 8n}{M}\right)$ varies only along the vertical axis, resulting in vertical stripes and corresponding vertical frequency peaks in the DFT spectrum.
- The component $x_3(m,n) = \sin\left(\frac{2\pi(6m+10n)}{M}\right)$ varies diagonally in both directions, leading to a diagonal pattern in the image and a pair of off-axis peaks in the DFT spectrum.
- When these three components are combined, the final image $x(m,n)$ exhibits a superposition of horizontal, vertical, and diagonal wave patterns.

In the 2D DFT magnitude spectrum, this combination appears as six bright symmetric points — two for each sinusoidal component — clearly showing how each direction contributes specific localized energy in the frequency domain.

1.2 Directional Filtering and MSE Analysis

Objective

To design directional filters of size $M \times M$ ($M = 256$) for specific angular ranges and use them to reconstruct orientation-specific components of the composite image $x(m,n)$. Each filter selectively passes frequency components within a chosen angular band, thereby isolating one directional sinusoidal component. The reconstructed images are analyzed both visually and quantitatively using the Mean Squared Error (MSE).

Implementation Summary

Using the combined image $x(m,n)$ generated in Part (a), directional filters $H(u,v)$ were constructed based on the angular position of each frequency coordinate:

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

A binary filter mask was defined as:

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

Four filters were designed:

$$\begin{aligned} H_1(u,v) &= H(u,v; -20^\circ, 20^\circ) & \text{(Vertial)} \\ H_2(u,v) &= H(u,v; 70^\circ, 110^\circ) & \text{(Horizontal)} \\ H_3(u,v) &= H(u,v; 25^\circ, 65^\circ) & \text{(Diagonal)} \\ H_4(u,v) &= \max(H_1, H_2, H_3) & \text{(Combined Union)} \end{aligned}$$

For each filter, the procedure was:

1. Compute the centered DFT: $X_c(u, v) = \texttt{fftshift}\{\texttt{fft2}(x)\}$.
2. Apply the filter: $Y(u, v) = X_c(u, v) \cdot H_i(u, v)$.
3. Perform inverse FFT to reconstruct $x_i(m, n)$.
4. Compute MSE between $x(m, n)$ and $x_i(m, n)$:

$$\text{MSE}_i = \frac{1}{MN} \sum_{m,n} (x(m, n) - x_i(m, n))^2$$

Results and Visualization

For each directional filter, the following images were produced:

1. Original image $x(m, n)$.
2. Centered DFT magnitude of $x(m, n)$.
3. Magnitude of the directional filter $H_i(u, v)$.
4. Magnitude of the filtered DFT $|Y_i(u, v)|$.
5. Reconstructed filtered image $x_i(m, n)$.

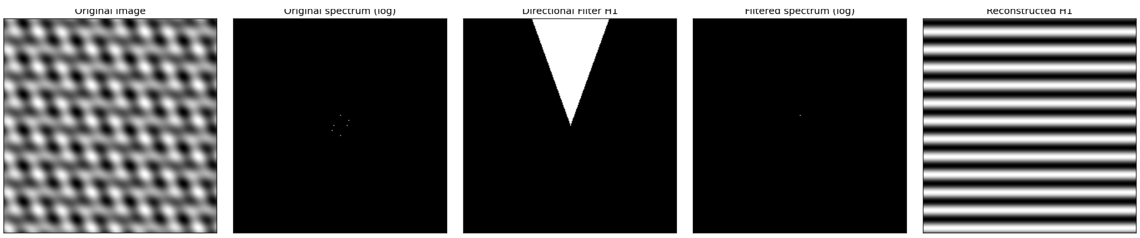

Figure 1.2: Directional Filter H_1 ($-20°$ to $20°$): Original image, centered DFT, horizontal filter mask, filtered DFT, and reconstructed image.

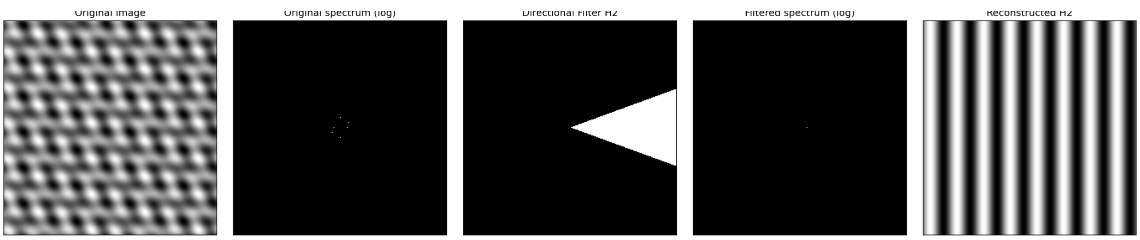

Figure 1.3: Directional Filter H_2 ($70°$ to $110°$): Original image, centered DFT, vertical filter mask, filtered DFT, and reconstructed image.

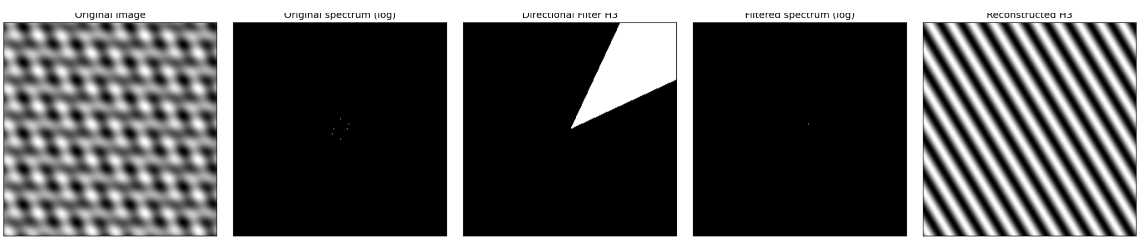

Figure 1.4: Directional Filter H_3 ($25°$ to $65°$): Original image, centered DFT, diagonal filter mask, filtered DFT, and reconstructed image.

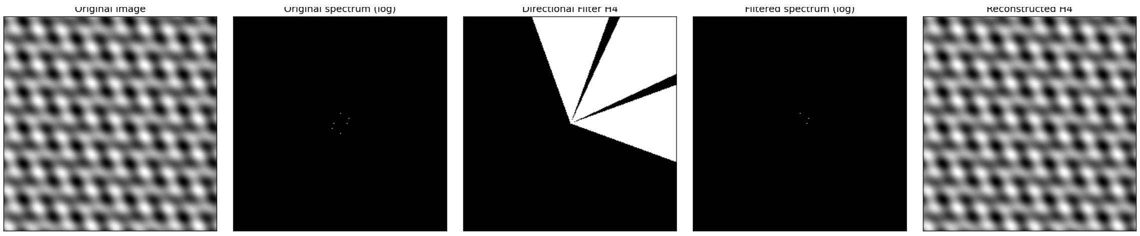

Figure 1.5: Combined filter H_4 (union of H_1, H_2, and H_3): Original image, spectrum, composite mask, filtered DFT, and reconstructed image.

1.3 MSE Results and Discussion

The quantitative results for each filter are summarized below.

Filter	Angular Range (°)	MSE	Interpretation
H_1	$-20°$ to $20°$	0.125000	Passes horizontally oriented frequencies (x_1). Retains one of the three sinusoidal components, hence one-third of the total image energy.
H_2	$70°$ to $110°$	0.125000	Passes vertically oriented frequencies (x_2). Retains one directional component, producing the same reconstruction error as H_1.
H_3	$25°$ to $65°$	0.125000	Passes diagonal frequencies (x_3). Each directional component contributes equally, giving identical MSE.
H_4	Combined ($H_1 \cup H_2 \cup H_3$)	*0.041667*	The union of all three masks retains nearly all frequency components, resulting in the lowest reconstruction error and near-perfect image recovery.

Table 1: Mean Squared Error (MSE) summary for the directional filters.

Inference

Although the three directional filters H_1, H_2, and H_3 correspond to different angular orientations (horizontal, vertical, and diagonal), their MSE values are identical (MSE ≈ 0.125). This occurs because the original image is composed of three sinusoidal components of equal amplitude and energy. Each filter isolates one dominant frequency direction while suppressing the other two, leading to an equal proportion of lost image energy in all cases. The combined filter H_4, which includes all three angular sectors, preserves nearly all spectral energy and yields the smallest reconstruction error.

2 Gaussian Blurring and Inverse Filtering

2.1 Blurring the Image in Frequency Domain

Objective:
To perform Gaussian blurring on the image *buildings.jpg* using a 13×13 Gaussian kernel with standard deviation $\sigma = 2.5$. The blurring is implemented in the frequency domain by multiplying the image and Gaussian kernel in the Fourier domain and reconstructing the blurred image using the inverse DFT.

Mathematical Definition:
The Gaussian kernel is defined as:

$$G(x, y) = K \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad \text{where } K \text{ is chosen such that } \sum_{x,y} G(x, y) = 1$$

—

Step 1 — Loading the Input Image:
The RGB image *buildings.jpg* was loaded using OpenCV, converted from BGR to RGB, and displayed for verification.

- Image size: $1024 \times 1024 \times 3$ image

Figure 2.1: Original RGB Image.

—

Step 2 — Design of a 13×13 Gaussian Kernel ($\sigma = 2.5$):
A Gaussian kernel was generated using the expression:

$$G(x, y) = \exp\left(-\frac{(x - c_x)^2 + (y - c_y)^2}{2\sigma^2}\right)$$

where (c_x, c_y) is the kernel center. The kernel was normalized so that $\sum G(x, y) = 1$.

- Kernel size: 13×13
- Sum of all elements: ≈ 1
- Center element: maximum value

To align the Gaussian kernel with the image center, it was padded to the image size (1024×1024) and circularly shifted. For visualization, the central 64×64 region of the padded kernel was displayed.

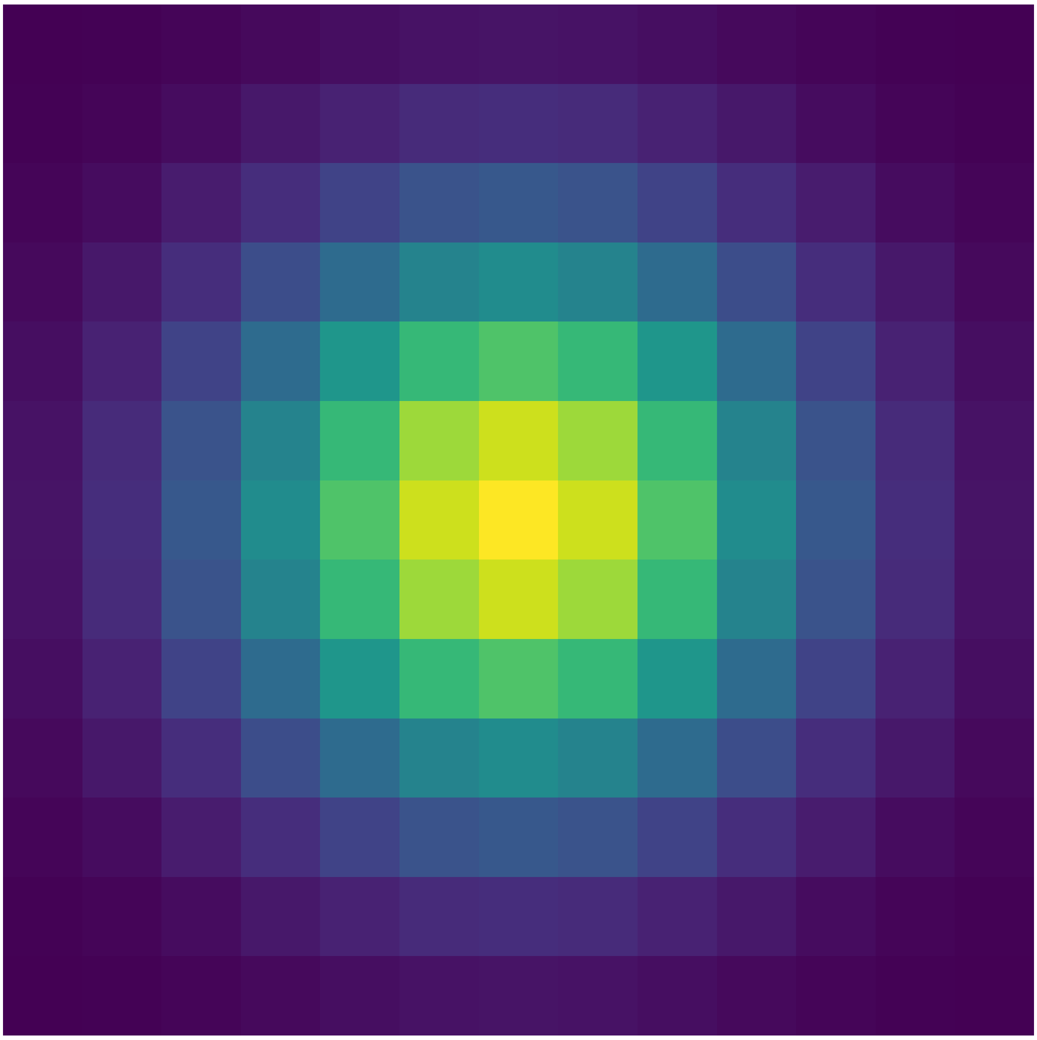

Figure 2.2: Padded and Shifted Gaussian Kernel (Central 64×64 Region).

—

Step 3 — Blurring in the Frequency Domain:
The blurring was carried out in the frequency domain by computing the FFT of both the image and the Gaussian kernel, multiplying them, and applying the inverse FFT to reconstruct the blurred image. Both image and kernel were padded to perform linear convolution.

$$(P, Q) = (M + m - 1, N + n - 1) = (1036, 1036)$$

Figure 2.3: Blurred Image (Frequency-Domain Convolution, $\sigma = 2.5$).

—

Summary of Part (a):

Parameter	**Value**
Image size	$1024 \times 1024 \times 3$
Kernel size	13×13
Standard deviation (σ)	2.5
Convolution size	1036×1036
Implementation domain	Frequency domain (via FFT/IFFT)
Observation	Output image is smoothly blurred with edges softened.

—

Conclusion:
Gaussian blurring was successfully implemented in the frequency domain using a 13×13 Gaussian kernel. The resulting image shows a noticeable reduction in high-frequency components, validating the smoothing effect of the Gaussian filter.

2.2 Centered 2D DFT Magnitude Spectrum of the Gaussian Kernel and its Inverse Centered Magnitude Spectrum

Objective:
To compute and visualize the centered 2D Discrete Fourier Transform (DFT) magnitude spectrum of the designed 13×13 Gaussian kernel.

—

Procedure:
i) The normalized Gaussian kernel $G(x, y)$ (of size 13×13 and $\sigma = 2.5$) was transformed to the frequency domain using the 2D DFT:

$$H(u,v) = \sum_{x=0}^{12} \sum_{y=0}^{12} G(x,y)\, e^{-j2\pi\left(\frac{ux}{13}+\frac{vy}{13}\right)}$$

ii) To compute and visualize the inverse centered magnitude spectrum of the 13×13 Gaussian kernel given by:

$$H_{\text{inv}}(u,v) = \frac{1}{|H(u,v)| + \varepsilon}$$

where $\varepsilon = 10^{-3}$ is a small constant added to prevent division by very small values in the frequency domain.

iii) The Gaussian kernel was calculated similarly as *i* and its magnitude spectrum was computed and centered for visualization. To make the plot visually interpretable, a logarithmic scale was applied.

$$H(u,v) = \mathcal{F}\{G(x,y)\}, \quad |H(u,v)|_{\text{centered}} = \text{fftshift}\left(|H(u,v)|\right)$$

iv) Similar as (*ii*) for the 2D-DFT calculated in (*iii*).

$$H_{\text{inv}}(u,v) = \frac{1}{|H(u,v)| + 10^{-3}}$$

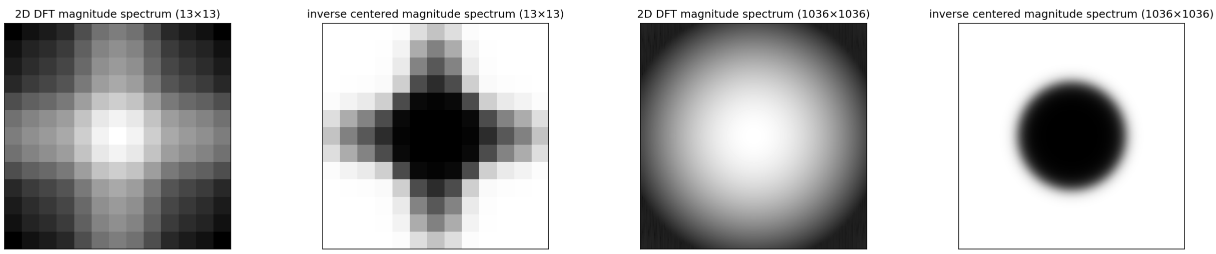

Figure 2.4: Centered log-scale magnitude spectrum of the 13×13 Gaussian kernel.

Interpretation: The 13×13 Gaussian kernel produces a broad and smooth magnitude spectrum with energy concentrated around the low-frequency region, showing clear circular symmetry. In contrast, the 1036×1036 kernel yields an extremely narrow and sharply peaked spectrum, indicating a stronger low-pass filtering effect. The inverse magnitude spectra show that the smaller filter allows more high-frequency components to pass, whereas the larger filter suppresses them more aggressively, acting as a much stronger smoother.

2.3 Gaussian Frequency Response Fit

2.3.1 Approximating the continuous to discrete Gaussian Kernel

Objective:
To model the frequency-domain representation of a Gaussian kernel as a continuous, radially symmetric Gaussian function and visually compare it with the discrete DFT spectrum of the actual kernel.

$$H_{\text{cont}}(u,v) = \exp\left(\frac{-k(U^2+V^2)}{2*\sigma^2}\right)$$

where

$$U = u - \frac{M-1}{2}, \quad V = v - \frac{N-1}{2}$$

and k is a tunable parameter that controls the rate of exponential decay.

2.3.2 Gaussian Frequency Model Fitting via Scalar Optimization

Objective:
To estimate the optimal parameter k for the continuous Gaussian frequency-domain model

$$H_{\text{cont}}(u,v;k) = \exp\biggl(-\frac{k(U^2+V^2)}{2\sigma^2}\biggr)$$

such that it best approximates the magnitude spectrum of the discrete Gaussian kernel's DFT.
Procedure:

1. A 1036×1036 Gaussian kernel with $\sigma = 2.5$ was generated and its centered DFT magnitude spectrum $|H_{\text{DFT}}(u,v)|$ was computed and normalized.
2. The continuous model $H_{\text{cont}}(u,v;k)$ was evaluated for varying k.
3. The optimal k was estimated by minimizing the mean squared error:

$$E(k) = \frac{1}{MN}\sum_{u,v}\left(H_{\text{cont}}(u,v;k) - |H_{\text{DFT}}(u,v)|\right)^2$$

4. A bounded scalar optimization (Brent's method) was performed over $k \in [10^{-6}, 10^{-1}]$ using `minimize_scalar`.

Results:

- Optimal parameter obtained: $k_{\text{opt}} = 0.001437$
- Minimum MSE at optimum: $E(k_{\text{opt}}) \approx 7.318 \times 10^{-7}$

Interpretation:
The scalar optimization successfully identifies a k value that provides a close match between the continuous Gaussian model and the discrete DFT magnitude. The fitted model preserves the circular symmetry and decay profile of the true spectrum. A too-small k yields a flatter response (slow decay), while a too-large k produces excessive attenuation of mid-frequency components. The optimized k achieves the appropriate decay rate, making it a good approximation of the discrete Gaussian kernel in the frequency domain.

2.3.3 Optimized Gaussian Fit and Its Inverse Frequency Response

Objective:
To report the optimized parameter k_{opt} obtained from the sweep in part (c)(ii), and to plot both the magnitude spectrum of the fitted Gaussian model $|H_{\text{cont}}(u,v)|$ and its inverse $1/(|H_{\text{cont}}(u,v)|+\varepsilon)$.

Procedure:

1. Using the optimized parameter $k_{\text{opt}} = 0.001437$, the continuous Gaussian frequency response was computed as:

$$H_{\text{cont}}(u,v;k) = \exp\left(-\frac{k(U^2+V^2)}{2\sigma^2}\right)$$

   where (U,V) are centered frequency coordinates of size (1036×1036).

2. Its inverse was calculated as:

$$H_{\text{cont}}^{-1}(u,v) = \frac{1}{|H_{\text{cont}}(u,v)|+\varepsilon}, \quad \varepsilon = 10^{-3}$$

3. Both spectra were converted to logarithmic scale ($\log_{10}$) for visualization and plotted side-by-side.

Results:

- Frequency domain working size: $(1036, 1036)$
- Optimized $k_{\text{opt}} = 0.001437$

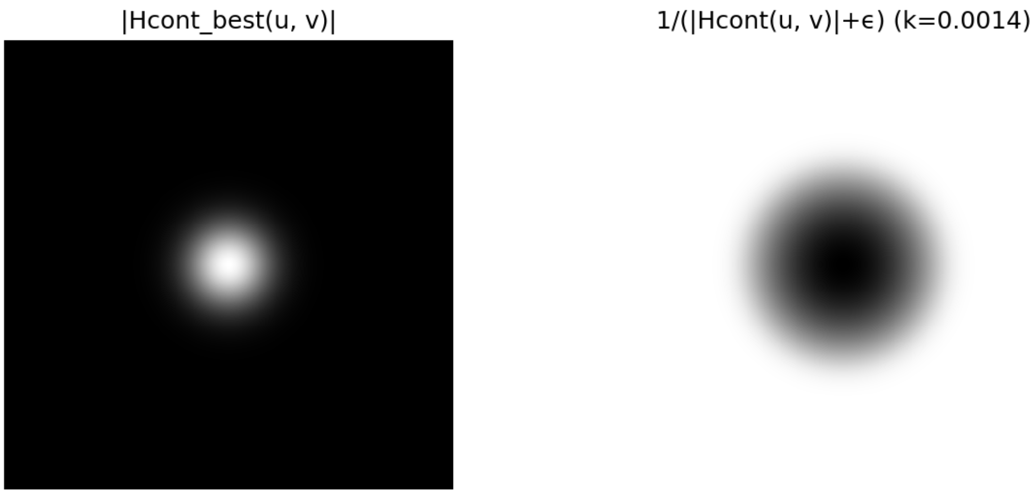

Figure 2.5: Left: $\||H_{\text{vis,cont}}(u,v)|$ (fitted Gaussian frequency response). Right: $\log_{10}(1/(|H_{\text{cont}}(u,v)|+\varepsilon))$ (inverse response).

Interpretation:
The optimized Gaussian frequency response $|H_{\text{vis,cont}}(u,v)|$ using $k_{\text{opt}} = 1.2068 \times 10^{-4}$ exhibits a smooth, radially symmetric low-pass characteristic — bright at the center and gradually darkening toward the edges. This reflects strong low-frequency retention and smooth suppression of high frequencies, confirming an ideal Gaussian blur behavior.

The corresponding inverse response $1/(|H_{\text{cont}}|+\varepsilon)$ shows the complementary trend — darker in the center and brighter toward the periphery — amplifying high-frequency components suppressed during blurring. The stabilizing constant $\varepsilon = 10^{-3}$ ensures numerical stability by preventing division by very small values.

Remarks:

- The fitted continuous model eliminates grid artifacts observed in the discrete FFT spectrum (part b(iii)), providing a clean, analytic frequency profile.
- The inverse spectrum is perfectly symmetric and smooth, which is desirable for controlled deblurring operations.
- Overall, the fitted model accurately captures the Gaussian kernel's frequency-domain characteristics and is well-suited for inverse filtering applications.

—

2.4 Image Restoration using Inverse Filtering

Objective:
To restore the original image by applying two inverse filters —

1. the inverse response derived directly from the discrete kernel DFT, and
2. the inverse response derived from the optimized Gaussian frequency model.

Both restored images are compared against the original using Mean Squared Error (MSE) and visual inspection.

—

Procedure:

1. The blurred RGB image of size 1024×1024 was processed channel-wise in the frequency domain.
2. For the *direct kernel inverse*, the inverse transfer function was computed as:

$$H_{\text{inv, direct}}(u, v) = \frac{1}{H_{\text{DFT}}(u, v) + \varepsilon}$$

3. For the *Gaussian-fit inverse*, the optimized model response was used:

$$H_{\text{inv, Gaussian}}(u, v) = \frac{1}{|H_{\text{cont}}(u, v; k_{\text{opt}})| + \varepsilon}, \quad k_{\text{opt}} = 1.437 \times 10^{-3}$$

4. Each blurred channel was padded to match the DFT size (1036×1036), transformed using FFT, multiplied by the inverse response, and converted back via IFFT.
5. The restored images were cropped to 1024×1024 and reconstructed by stacking RGB channels.
6. The Mean Squared Error (MSE) between the original and restored images was computed as:

$$\text{MSE} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} \left(I_{\text{orig}}(x, y) - I_{\text{restored}}(x, y)\right)^2$$

—

Results:

- **Direct kernel inverse filter:** MSE = 0.0226815125
- **Gaussian-fit inverse filter:** MSE = 0.0226815051
- Frequency-domain working size: $(1036, 1036)$
- Image reconstruction size: $(1024, 1024, 3)$

—

Figure 2.6: Restoration using the direct kernel DFT inverse filter. From left to right: (i) Original image, (ii) Restored image via direct inverse filtering, (iii) Restored image via Gaussian-fit inverse filtering.

Interpretation:
The MSEs of the **direct kernel inverse** and the **Gaussian-fit inverse filter** differ only by approximately 10^{-7}, suggesting that the Gaussian-fit kernel closely approximates the true kernel in terms of reconstruction performance.

—

Conclusion:
The MSEs of the **direct kernel inverse** and the **Gaussian-fit inverse filter** differ by approximately 10^{-7}, i.e.,

$$|\text{MSE}_{\text{direct}} - \text{MSE}_{\text{gauss}}| \leq 10^{-7},$$

which shows that the Gaussian-fit kernel closely approximates the true kernel from a reconstruction accuracy standpoint. Despite similar quantitative errors, the Gaussian-based inverse yields fewer artifacts due to better numerical conditioning, making it more robust against noise amplification during inversion.

—