Assignment No. 6

Problem Statement

Consider any image dataset with raw images containing noise, distortion, etc. Apply filtering techniques and implement **Inverse filter** and **Wiener filter** over the set of images in the dataset for image restoration and comment on these two filtering effects.

Aim

To study and implement image restoration techniques, specifically Inverse and Wiener filtering, and to differentiate their purpose from other image processing tasks like **image compression** and **representation**.

Objectives

1. To understand the concept of image degradation and the models that represent it.
2. To apply the **Inverse filter** to a degraded image and analyze its performance.
3. To apply the **Wiener filter** to overcome the limitations of inverse filtering.
4. To visually and conceptually compare the restoration results.
5. To distinguish the goal of image restoration from that of image compression techniques like **Huffman coding** or **JPEG**.

Expected Outcomes

1. Understand the fundamental difference between restoring an image's quality and compressing its data.
2. Gain proficiency in applying frequency-domain filters to restore images.
3. Analyze the critical weakness of the Inverse filter in the presence of noise.
4. Appreciate the superiority of the Wiener filter in practical scenarios.
5. Gain a practical understanding of how restoration fits into a larger workflow that might include **image representation** and compression.

Theory

1. Image Degradation/Restoration Model

Image restoration aims to recover an original image f(x,y) that has been degraded. The process is modeled using a degradation function h(x,y) (representing effects like blur) and additive noise n(x,y).

The degraded image g(x,y) is represented as:

$$\mathbf{g(x,y)=h(x,y)*f(x,y)+n(x,y)}$$

where '$*$' denotes convolution.

In the frequency domain, this relationship simplifies to:

$$G(u,v)=H(u,v)F(u,v)+N(u,v)$$

where G, H, F, and N are the Fourier transforms of g, h, f, and n, respectively.
It's important to distinguish image restoration from **image compression**. While restoration aims to recover the original image from degradation, compression techniques, such as lossless **Huffman coding** or lossy **JPEG**, aim to reduce the data required to store or transmit an image by exploiting redundancies. Often, an image is restored first to improve its quality before it is compressed.

2. Inverse Filtering

Inverse filtering is the most direct approach to restoration. It attempts to reverse the degradation by dividing the Fourier transform of the degraded image by the degradation function.

Formula:
The estimate of the original image, $\hat{F}(u,v)$, is found by:

$$\hat{F}(u,v)=H(u,v)G(u,v)$$

Substituting the degradation model gives:

$$\hat{F}(u,v)=H(u,v)H(u,v)F(u,v)+N(u,v)=F(u,v)+H(u,v)N(u,v)$$

Problem with Inverse Filtering: The main issue lies with the noise term $H(u,v)N(u,v)$. If the degradation function $H(u,v)$ has values close to zero for certain frequencies, this term can become very large, **amplifying the noise** to a point where it completely overwhelms the restored image. This makes the simple Inverse filter impractical for most real-world applications where noise is present.

3. Wiener Filtering

The Wiener filter is a more advanced restoration filter that addresses the noise amplification problem of the Inverse filter. It provides an optimal estimate of the original image by incorporating statistical knowledge about the noise and the original image signal, aiming to minimize the mean square error.

Formula:
The Wiener filter is defined in the frequency domain as:

$$\hat{F}(u,v)=|H(u,v)|2+Sf(u,v)Sn(u,v)|H(u,v)|2H(u,v)G(u,v)$$

Where:

- $H(u,v)$ is the degradation function.
- $|H(u,v)|2=H*(u,v)H(u,v)$, where $H*$ is the complex conjugate.
- $Sn(u,v)=|N(u,v)|2$ is the **power spectral density** of the noise.
- $Sf(u,v)=|F(u,v)|2$ is the **power spectral density** of the original image.

The term $Sf(u,v)Sn(u,v)$ is the ratio of noise power to signal power. In practice, this ratio is often unknown and is approximated by a constant, K.

How it Works: The filter attenuates frequencies with a low signal-to-noise ratio, preventing noise amplification while still performing deconvolution. Once an image is successfully restored, it can be further processed. For example, objects within the clean image can be described using **boundary descriptors** or the entire image can be efficiently stored using **Run-Length Encoding** or **Huffman coding**.

Comparison

Feature	Inverse Filter	Wiener Filter
Approach	Simple deconvolution (division in the frequency domain).	Statistical approach that minimizes mean square error.
Noise Handling	Very poor. Amplifies noise significantly where H(u,v) is small.	Excellent. Suppresses noise by considering the signal-to-noise power ratio.
Required Info	Only requires the degradation function H(u,v).	Requires H(u,v) and estimates of the power spectra of noise and image.
Complexity	Computationally simple and fast.	More complex due to the need for statistical information.
Result Quality	Often produces unusable results if any noise is present.	Produces a much better and more stable restoration.

Pseudocode

Step 1: Read the original image and create a degraded version

```
f_original ← ReadImage("input_image.jpg")
Display(f_original, "Original Image")
// Create a degradation function (e.g., motion blur)
H ← CreateBlurFilter()
// Apply blur in the frequency domain and add random noise
g_degraded ← ApplyBlur(f_original, H) + CreateNoise()
Display(g_degraded, "Degraded Image")
```

Step 2: Apply Inverse Filtering for restoration

```
G_degraded_fft ← FFT(g_degraded)
```

F_restored_inv_fft ← G_degraded_fft / H
f_restored_inv ← IFFT(F_restored_inv_fft)
Display(f_restored_inv, "Inverse Filter Restoration")

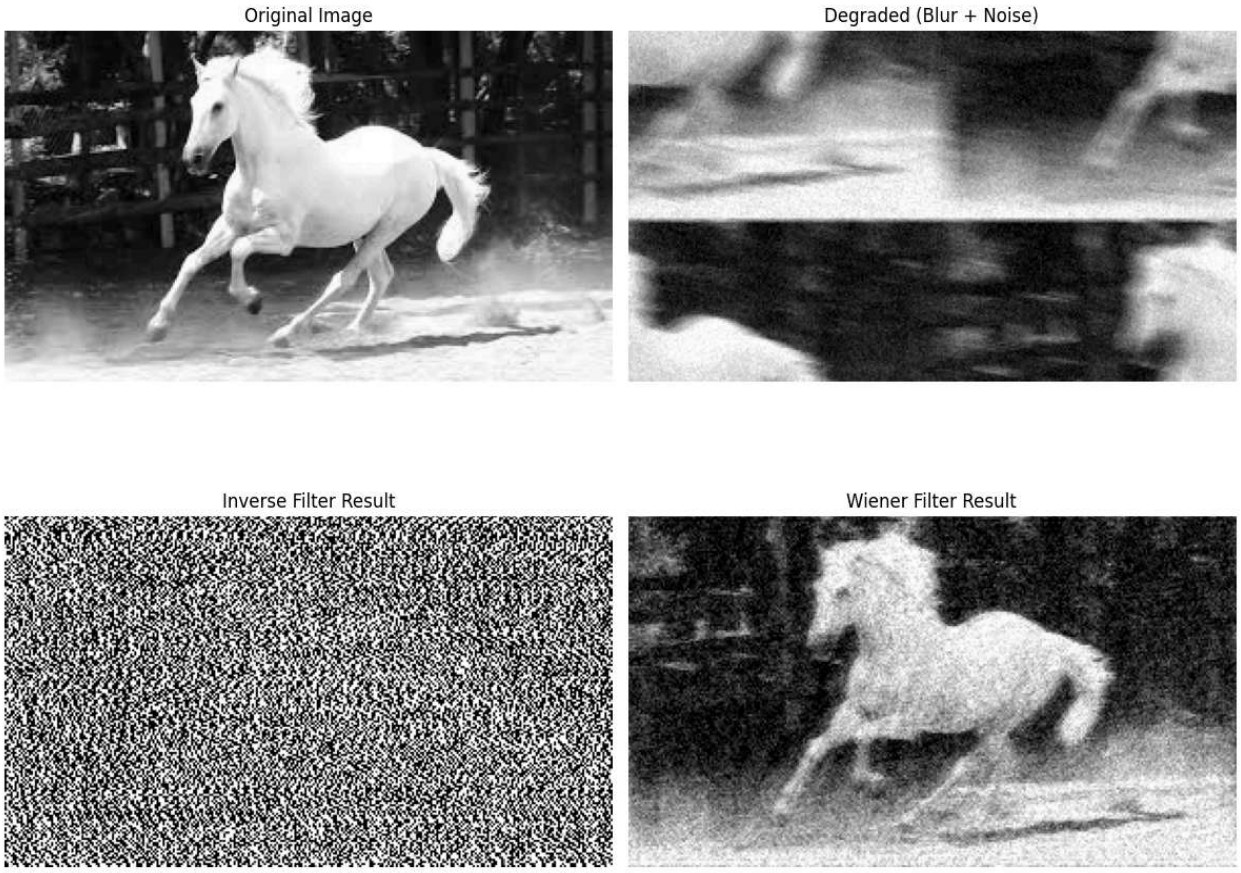
Step 3: Apply Wiener Filtering for restoration
// Estimate noise-to-signal power ratio (K)
K ← estimate_nsr(g_degraded)
// Wiener filter formula
wiener_factor ← (|H|^2) / (|H|^2 + K)
F_restored_wie_fft ← wiener_factor * (G_degraded_fft / H)
f_restored_wie ← IFFT(F_restored_wie_fft)
Display(f_restored_wie, "Wiener Filter Restoration")

Step 4: Compare results
Print("Inverse filter fails due to noise amplification.")
Print("Wiener filter provides a superior, balanced restoration.")

Results: Add the Output Images in Following Format.

Upload your ZIP file containing images...

archive (3).zip(application/x-zip-compressed) - 3837948 bytes, last modified: 10/6/2025 - 100% done
Saving archive (3).zip to archive (3) (1).zip
Extracted images to folder: dataset
Found 1221 images.



Conclusion

This experiment confirmed the superiority of the Wiener filter over the Inverse filter for restoring noisy images. While the Inverse filter amplified noise to unusable levels, the Wiener filter provided a balanced and effective restoration. This process is a critical pre-processing step in any complete image processing pipeline. A successfully restored image provides a reliable foundation for subsequent tasks, whether it's analyzing shapes using **region descriptors** or preparing the image for efficient storage and transmission using compression standards like **JPEG** or lossless methods like **Huffman coding**.