Assignment 1

Signal Processing in Practice

Tejash More
ID: 27077

January 19, 2026

1 Introduction

This assignment covers the fundamental operations of convolution for 1D and 2D. Implementation of convolution from scratch is done to understand boundary conditions, then apply linear and time-invariant techniques to restore signals degraded by reverberation (1D) and optical blur (2D). This project aims to:

1. Implement discrete linear convolution from scratch, handling various boundary conditions, such as full, valid, and same.
2. Establish the linear algebraic framework of convolution via Toeplitz and Circulant matrices.
3. Perform deconvolution to restore audio and images using Inverse and Wiener filters.
4. Analyze the sensitivity of restoration quality to the regularization parameter K.

2 Methodology and Implementation

2.1 Part 1: Discrete Convolution Implementation

The discrete convolution of a signal $x[n]$ of length N and a filter $h[n]$ of length M is defined as:

$$y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{1}$$

2.1.1 Algorithm Implementation

A custom function `manual_convolution_loops` was implemented to bypass optimized libraries and demonstrate the underlying mechanics. The algorithm employs a double-loop structure:

- **Outer Loop:** Iterates over the output index n from 0 to $N + M - 2$.
- **Inner Loop:** Iterates over the filter index k.

- **Condition:** A check `if n-k < N and n-k >= 0` ensures indices remain within valid signal bounds.

Three boundary modes were implemented:

1. **Full:** Returns the complete output of size $N + M - 1$.
2. **Same:** Returns the central section of size N, centering the alignment.
3. **Valid:** Returns only the segment where the filter and signal fully overlap, size $N - M + 1$.

2.2 Part 2: Matrix Formulation (Toeplitz & Circulant)

Convolution is a linear operation and can be expressed as a matrix-vector product $y = Hx$.

2.2.1 Linear Convolution as Toeplitz Matrix

For linear convolution, the matrix H takes a Toeplitz structure where each row is a shifted copy of the previous row. For inputs x (length N) and h (length M), the matrix H has dimensions $(N + M - 1) \times N$.

$$H = \begin{bmatrix} h[0] & 0 & \dots & 0 \\ h[1] & h[0] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & h[M-1] \end{bmatrix} \tag{2}$$

2.2.2 Circular Convolution as Circulant Matrix

Circular convolution assumes periodic extension of the signals. The matrix C is Circulant, where each row is a cyclic shift of the filter h. This is fundamental to FFT-based convolution, as a Circulant matrix is diagonalized by the Discrete Fourier Transform (DFT) matrix.

For a filter h of length N (assuming padding to match signal length), the Circulant matrix C ($N \times N$) is defined as:

$$C = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \dots & h[1] \\ h[1] & h[0] & h[N-1] & \dots & h[2] \\ h[2] & h[1] & h[0] & \dots & h[3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \dots & h[0] \end{bmatrix} \tag{3}$$

2.3 Part 3: Deconvolution and Restoration

2.3.1 Inverse Filtering

In the frequency domain, convolution becomes multiplication: $Y(\omega) = X(\omega)H(\omega)$. The naive estimate is:

$$\hat{X}(\omega) = \frac{Y(\omega)}{H(\omega)} = X(\omega) + \frac{N(\omega)}{H(\omega)} \tag{4}$$

Theoretical Limit: If $H(\omega) \approx 0$ at any frequency (common in low-pass filters like blurs), the term $\frac{N(\omega)}{H(\omega)}$ amplifies noise exponentially, destroying the signal.

2.3.2 Wiener Filtering

The Wiener filter minimizes the Mean Square Error (MSE) between the estimated and true signal. It introduces a regularization term K (related to the Signal-to-Noise Ratio):

$$\hat{X}(\omega) = \left[\frac{H^*(\omega)}{|H(\omega)|^2 + K}\right] Y(\omega) \tag{5}$$

- When $|H(\omega)|^2 \gg K$ (strong signal), it behaves like the Inverse Filter.
- When $|H(\omega)|^2 \ll K$ (noise/zeros), the gain is attenuated, suppressing noise.

3 Analysis and Observations

3.1 Validation of Manual Implementation

The manual convolution implementation was rigorously tested against `numpy.convolve`.

Mode	**Input Sizes**	**Output Size**	**Verification**
Full	$N = 5, M = 4$	8	MATCH
Same	$N = 5, M = 4$	5	MATCH
Valid	$N = 5, M = 4$	2	MATCH

Table 1: Validation of manual loops against built-in library.

The derived output vector for the test case $x = [1, 2, 3, 4, 5]$ and $h = [1, -1, 2, 3]$ was:

$$y_{full} = [1, 1, 3, 8, 13, 12, 22, 15]$$

This confirms the correctness of the index logic used in the implementation.

3.2 Audio De-reverberation Analysis

Observations:

1. **Degradation:** The convolution with RIR created a distinct "echo" effect, reducing clarity. The addition of Gaussian noise ($\sigma = 0.02$) further masked high-frequency details.

2. **Naive Recovery ($K = 0$):** The output was dominated by high-amplitude static. This confirms the theoretical prediction that inverting small values in the RIR frequency response amplifies the noise floor.

3. **Wiener Recovery ($K = 0.5$):** The restoration was significantly cleaner. While some reverberation artifacts remained (due to the limitation of linear filtering on non-stationary noise), the speech was understandable, and the background noise was suppressed.

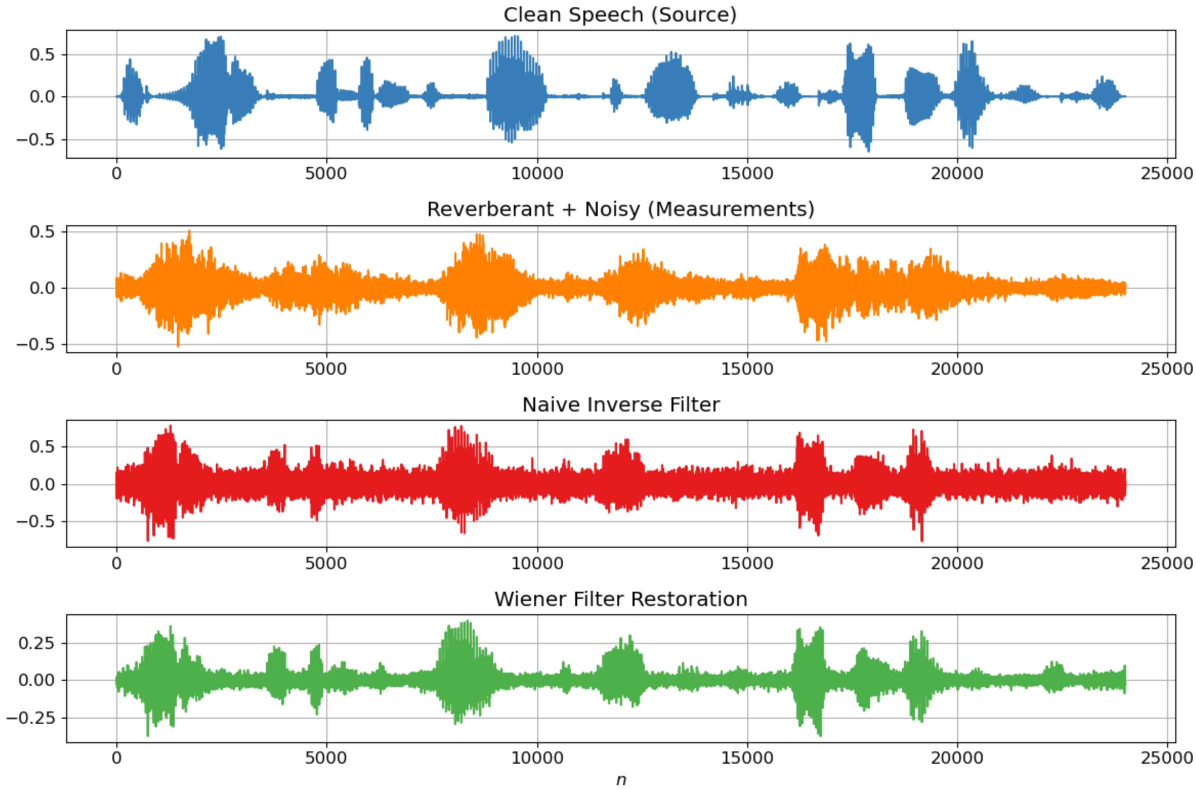

Figure 1: Time-domain visualization of audio restoration

4 Image Restoration and Parameter Estimation

This section explores the restoration of 2D images degraded by a Gaussian Point Spread Function (PSF) and additive white Gaussian noise. The restoration quality is evaluated qualitatively (visual inspection) and quantitatively (PSNR).

4.1 Degradation Model and Inverse Filtering

The degradation was modeled as a convolution with a Gaussian kernel (h) followed by additive noise (η):

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y) \tag{6}$$

As seen in 2, Weiner filter with K = 0.09 is able to recover most of the image as compared to the inverse filter (K = 0). The inverse filter amplifies noise, leading to the wavy pattern in the image and rendering it unrecognizable, whereas the Wiener filter preserves structural details.

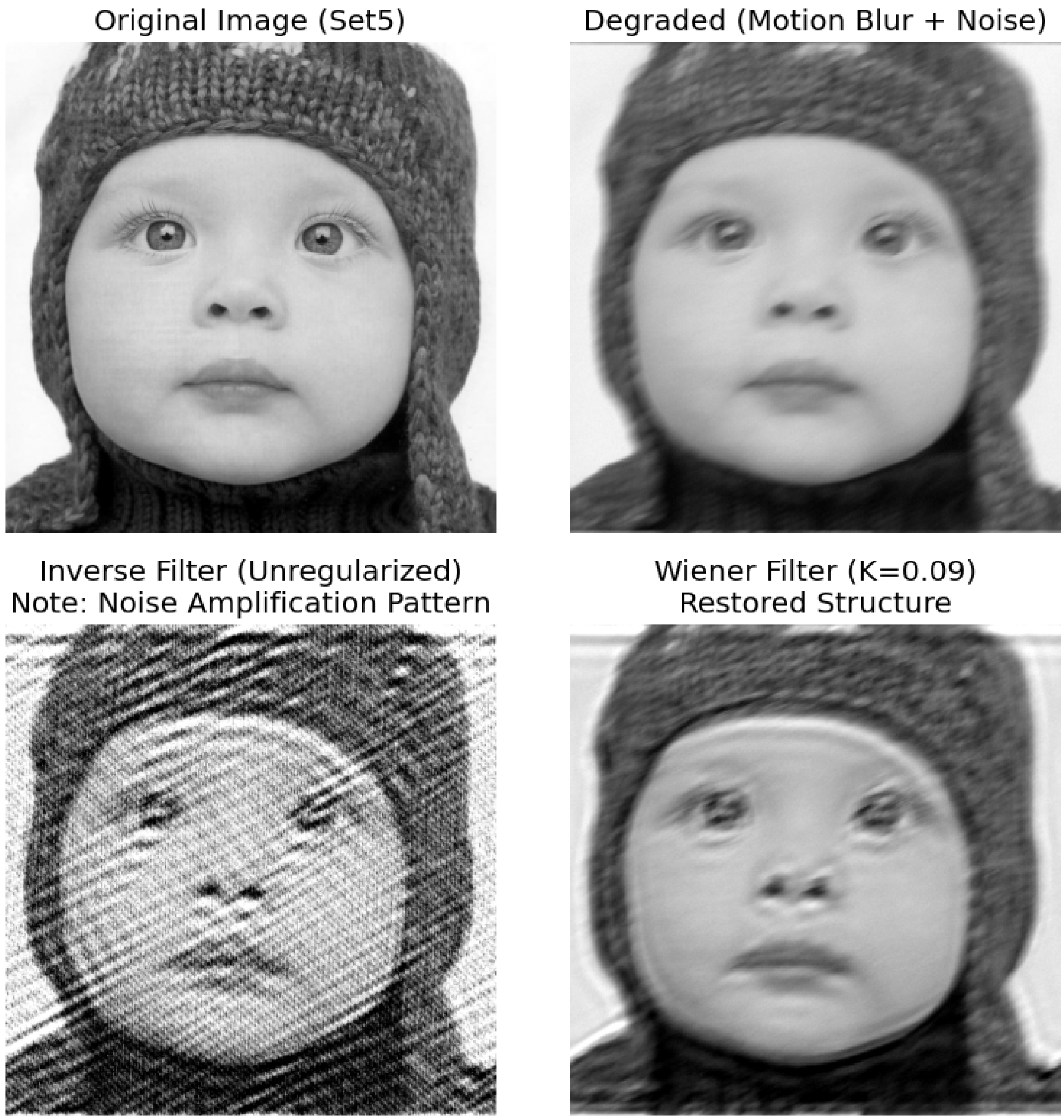

Figure 2: Visual comparison of degradation and restoration

4.2 Blind Deconvolution: Hyperparameter Grid Search

As shown in 3, We performed a grid search over two parameters:

1. **Blur Width (σ):** Varying the estimated sigma size against the true blur.
2. **Regularization (K):** Varying the noise-to-signal ratio estimate.
3. **Window Size:** Varying the window size for the Gaussian kernel.

Visual Analysis of the Grid:

- **Small K (Under-regularized):** The result resembles the Inverse Filter. Noise is prevalent.
- **Large K (Over-regularized):** The filter suppresses high frequencies too aggressively. The image appears blurry, effectively failing to deconvolve the original blur.

- **Optimal K:** A balance is struck where edges are sharpened without amplifying the background noise.
- **Optimal Window Size:** Windows size of 9x9 was found to be optimal.

Figure 3: Qualitative effects of parameter tuning. Moving left-to-right (increasing K) transitions the image from noisy to blurry for gaussian window size of 9x9

4.3 Quantitative Analysis (PSNR Curves)

To rigorously determine the optimal parameters, we plotted the Peak Signal-to-Noise Ratio (PSNR) as a function of K for various σ values.

Observations from the PSNR Plot:

- The curves exhibit a distinct **convex shape** (or concave down), confirming the existence of a global maximum for restoration quality.
- **Sensitivity to σ:** If the assumed σ deviates from the true blur width, the maximum achievable PSNR drops significantly. This highlights that accurate estimation of the sigma is as critical as the regularization term.
- For the optimal σ, the peak PSNR is achieved at approximately $K \approx 10^{-2}$ to 10^{-3}, which aligns with the visual "sweet spot" observed in the grid search.

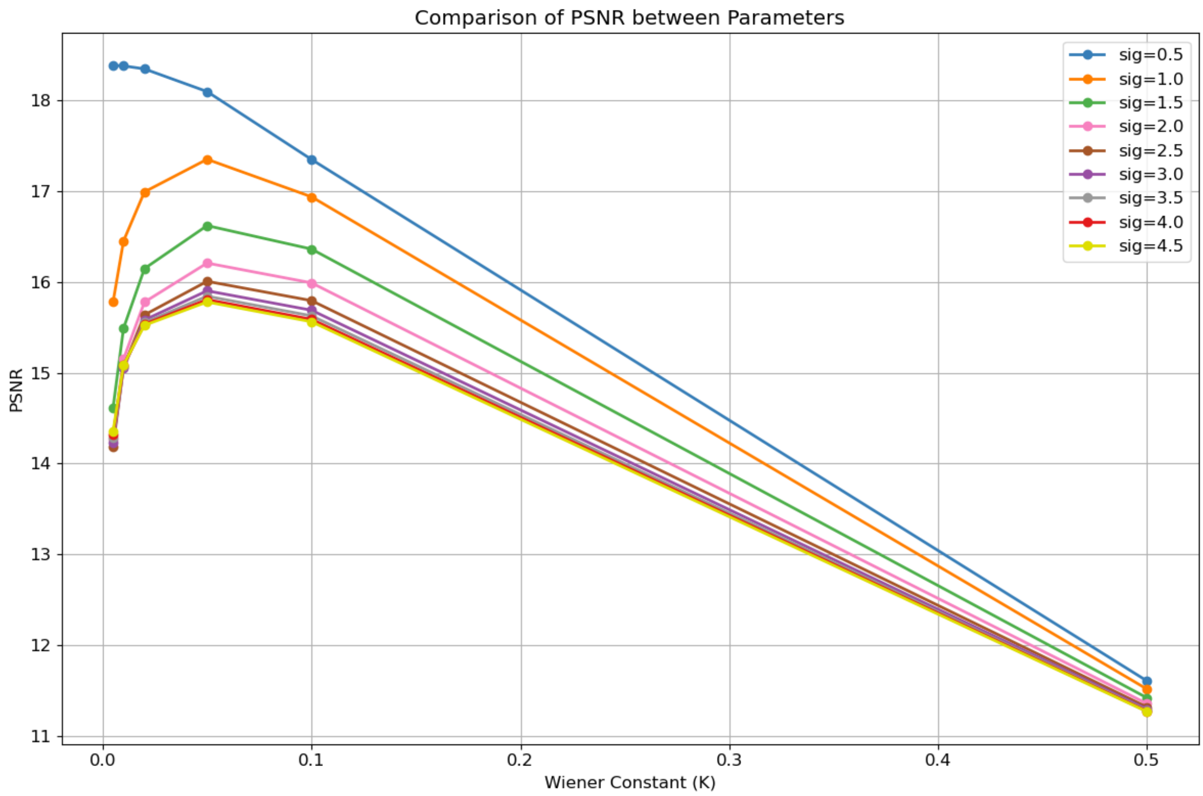

Figure 4: PSNR vs K for the optimal window size of 9x9

4.4 Optimal Restoration Result

Based on the maxima of the PSNR curves, the optimal parameters were selected to produce the final restored image with

$$\boxed{\text{Window Size} = 9,\ \sigma = 0.5\ \&\ K = 0.05 \text{ with PSNR} = 18.3836}$$

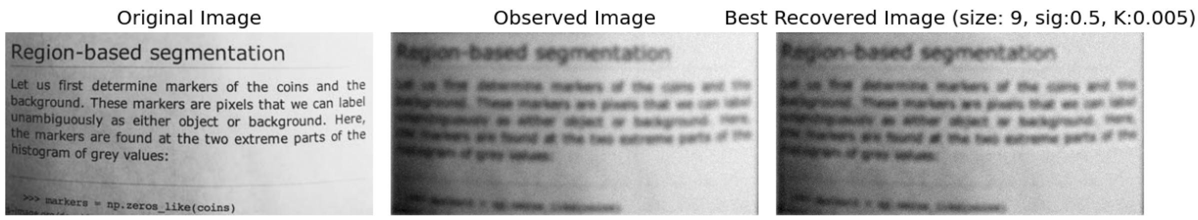

Figure 5: Final restoration result using the parameters $WindowSize = 9, \sigma = 1$ & $K =$ $= 0.05$

5 Conclusion

This project successfully demonstrated the implementation of convolution from first principles and validated its linear algebraic properties via Toeplitz matrices. The transition from 1D to 2D applications highlighted the universality of these operations.