Monte Carlo Optimization of Non-Local Means Denoising Algorithm

Petru-Răzvan Robu, Matteo-Alexandru Verzotti,
Robert-Ionuț Voaideș-Negustor

January 13, 2026

Introduction: The Noise Problem

- **Image Noise:** Modeled as additive white Gaussian noise.

$$y = x + \eta, \quad \eta \sim \mathcal{N}(0, \sigma)$$

- **Traditional Techniques:**
  - Gaussian smoothing, Median filtering.
  - *Issue:* Relies on locality, causing blurred edges and lost textures.
- **Non-Local Means (NLM):**
  - Preserves texture by using similar patches from *anywhere* in the image.
  - *Issue:* High computational complexity.

Non-Local Means (NLM) Formulation

The denoised value $z(p)$ is a weighted average of pixels $y(q)$:

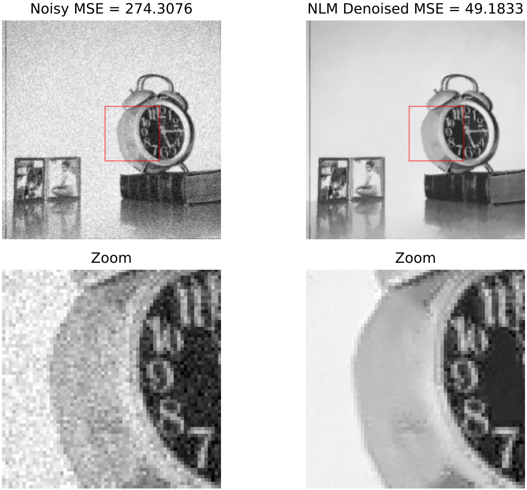
$$z(p) = \frac{1}{C(p)} \sum_{q \in \Omega} w(p,q) y(q)$$

- **Weights (w):** Measure similarity between patches centered at p and q.

$$w_i = e^{-||y - x_i||^2/(2h_r^2)}$$

- **Complexity:** $\mathcal{O}(mnd)$
  - m: Pixels to denoise
  - n: Number of patches (search space)
  - d: Patch dimensions

NLM Denoising Results



Monte Carlo Non-Local Means (MCNLM)

The Goal

Reduce complexity from $\mathcal{O}(mnd)$ to $\mathcal{O}(mkd)$ where $k \ll n$.

Core Strategy:

- Instead of computing weights for *all* n patches, select a random subset k.
- Use Bernoulli sampling to generate a reference set.
- **Sampling Pattern p:** Probability vector determining if a patch is selected.

Mathematical Approximation

We approximate the NLM fraction using random variables A (numerator) and B (denominator):

$$A = \frac{1}{n}\sum_{j=1}^{n} w_j x_j \frac{I_j}{p_j}, \quad B = \frac{1}{n}\sum_{j=1}^{n} w_j \frac{I_j}{p_j}$$

- $I_j \sim \text{Bernoulli}(p_j)$: Indicator variable for sampling.
- $Z = A/B$ is the estimator for the true pixel value z.
- **Note:** Z is a *biased* estimator, but the error drops exponentially as sample size increases.

Improving Spatial Locality

Pure NLM ignores spatial distance. We reintroduce locality for better results.

Combined Weight Function:

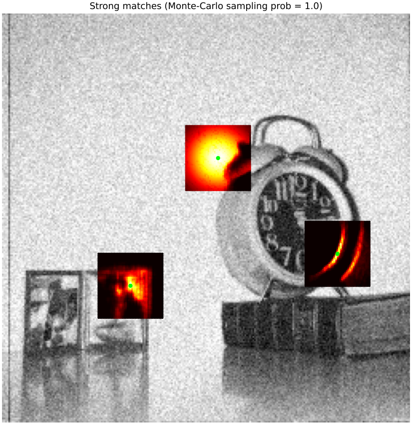
$$w_j = w_j^r \cdot w_j^s$$

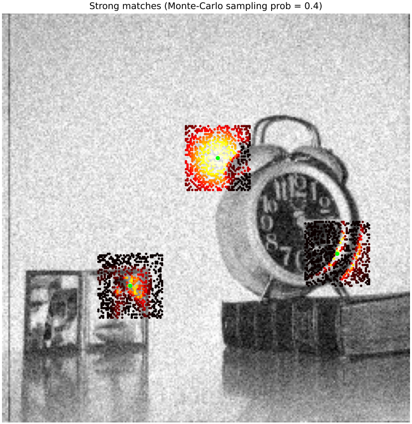
- w_j^r: Structural similarity (standard NLM).
- w_j^s: Spatial proximity.

$$w_j^s = e^{-(d_2^j)^2/(2h_k^2)} \cdot \mathbb{I}\{d_\infty^j \leq \rho\}$$

- Helps distinguish features (e.g., a bright star vs. noise) based on surroundings.

Non-local matches

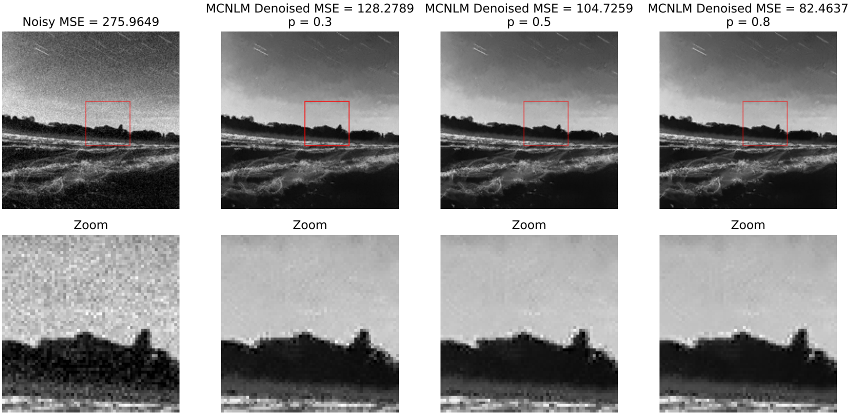


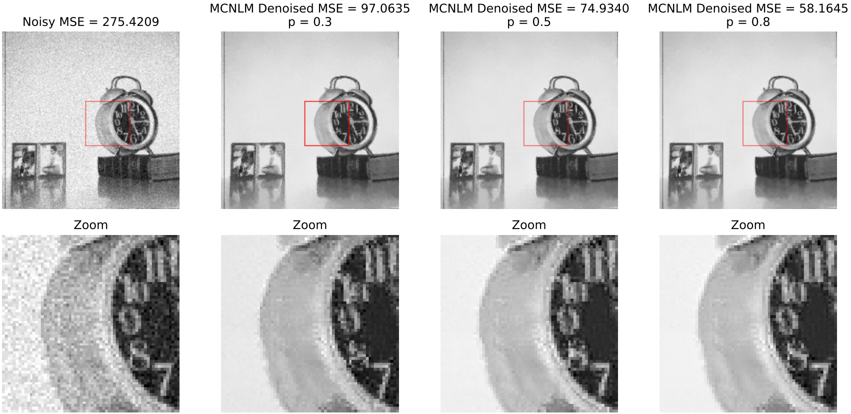


Algorithm: Monte Carlo NLM

1. **Input:** Noisy patch y, reference set X.
2. **Loop** through references $j = 1 \ldots n$:
   - Generate random variable $I_j \sim \text{Bernoulli}(p_j)$.
   - **If** $I_j = 1$ (Sampled):
   - Compute weight w_j.
   - Update sums for Numerator (A) and Denominator (B).
3. **Output:** $Z = A/B$.

MCNLM Results




Noisy MSE = 275.4209
MCNLM Denoised MSE = 97.0635
p = 0.3
MCNLM Denoised MSE = 74.9340
p = 0.5
MCNLM Denoised MSE = 58.1645
p = 0.8
Zoom
Zoom
Zoom
Zoom

Why it Works: Error Bounds

Does the approximation converge?

Exponential Error Decay

For sample size n and error tolerance ϵ:

$$\mathbb{P}(|Z - z| > \epsilon) \leq \text{Exponential Decay Terms}$$

- **Empirical Proof:** Even with only **5%** of samples ($p = 0.05$):

$$\mathbb{P}(\text{Error} > 0.01) \leq 6.5 \times 10^{-6}$$

- High reliability with fraction of the cost.

Experimental Results

MSE Convergence:

- MSE drops rapidly as sampling ratio increases.
- Diminishing returns after $p = 0.3$.

Visual Quality:

- Preserves edges better than Gaussian.
- Comparable visual quality to full NLM.

(Placeholder for Figure 4/6: MSE Comparison Charts)

Comparison Results

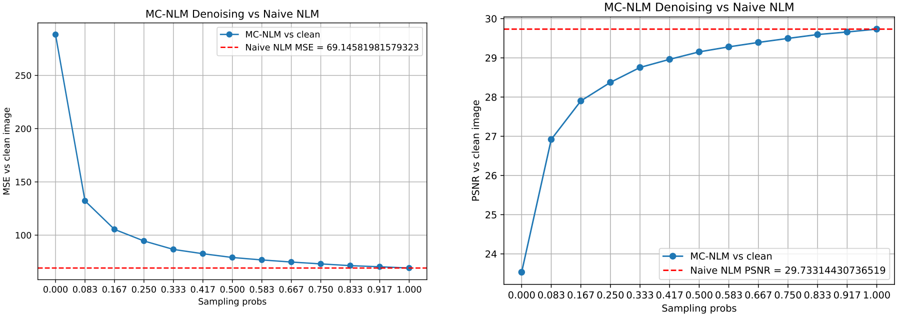


Figure: Visual comparison showing MSE and PSNR convergence.

Increasing window size

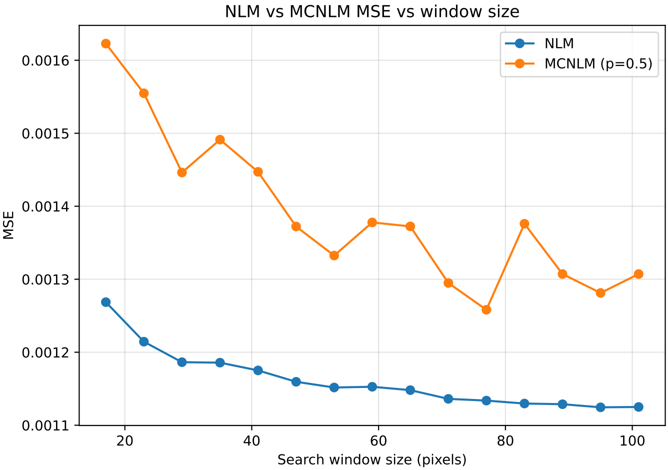


Figure: Plotting MSE as window size is increased

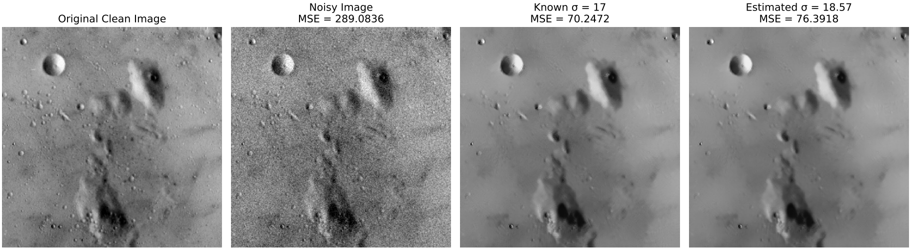
Noise Estimation (FFT)

In real applications, noise deviation (σ) is unknown.
Solution: Fast Fourier Transform (FFT)

1. Convert image to Frequency Domain.
2. High-frequency components $\approx$ Noise.
3. Mask low frequencies and compute standard deviation of the remainder.

- **Result:** Estimated $\sigma \approx 20.33$ vs True $\sigma = 17$.
- Sufficient for effective denoising.

Noise Comparison



Conclusion

- **Efficiency:** MCNLM significantly reduces computational cost ($\mathcal{O}(mkd)$).
- **Quality:** Maintains high Peak Signal-to-Noise Ratio (PSNR).
- **Robustness:** Theoretical bounds prove reliability even with low sampling rates (e.g., 5-10%).
- **Practicality:** Can be combined with spatial weights and FFT noise estimation for real-world usage.

Thank You!