

Monte Carlo Optimization of Non-Local Means Denoising

Robu Petru-Razvan Verzotti Matteo-Alexandru Voaides-Negustor Robert-Ionut

Monte Carlo NLM Denoising Project

Motivation and Noise Model

- Image noise is commonly modeled as additive white Gaussian noise.
- Goal: suppress noise while preserving textures and edges.
- Non-Local Means (NLM) is effective but computationally heavy.

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

Non-Local Means (NLM)

Patch-based weighted average

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

Standard similarity weight

$$w_i = \exp\left(-\frac{\|\mathbf{y} - \mathbf{x}_i\|^2}{2h_r^2}\right)$$

- Complexity: $\mathcal{O}(mnd)$ (or $\mathcal{O}(mD^2d)$ with a window).

Monte Carlo NLM (MCNLM) Sampling

- Internal denoising: sample reference patches from the noisy image.
- For each patch j , sample $I_j \sim \text{Bernoulli}(p_j)$.
- Average sampling ratio:

$$\xi = \frac{1}{n} \sum_{j=1}^n p_j$$

- Reduced complexity: $\mathcal{O}(mkd)$ for $k \ll n$ samples.

MCNLM Estimator

Unbiased estimators for numerator and denominator

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

$$Z = \frac{A}{B}$$

- A and B are unbiased; Z is biased but converges as n grows.
- Error probability decays exponentially with sampling size.

Spatial Sampling (Semi-Local NLM)

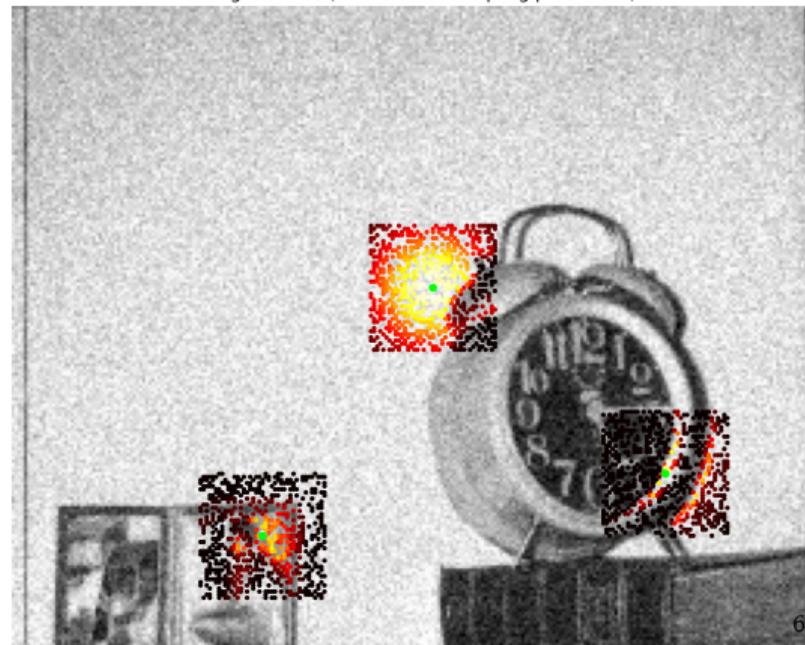
- Combine structural similarity with spatial proximity.

$$w_j = w_j^r \cdot w_j^s, \quad w_j^s = \exp\left(-\frac{(d_2^j)^2}{2h_s^2}\right) \cdot \mathbb{I}\{d_\infty^j \leq \rho\}$$

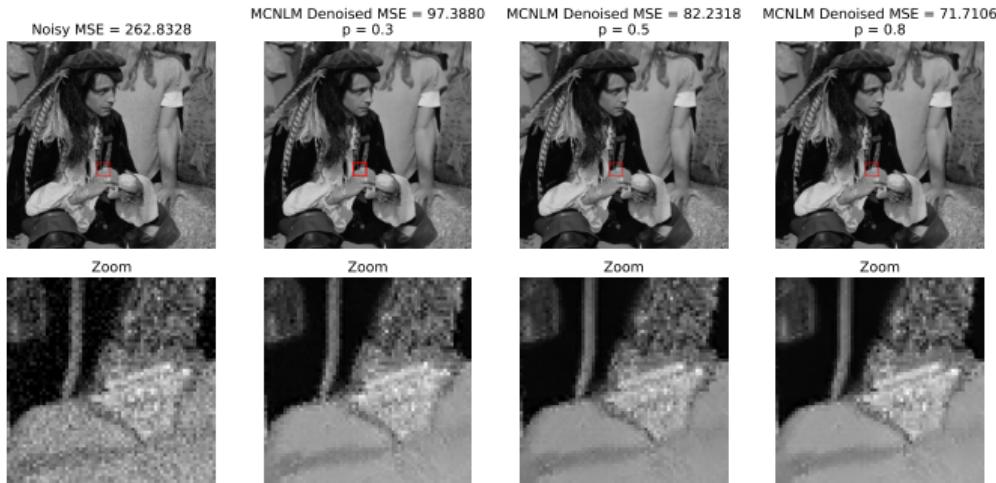
Strong matches (Monte-Carlo sampling prob = 1.0)



Strong matches (Monte-Carlo sampling prob = 0.4)



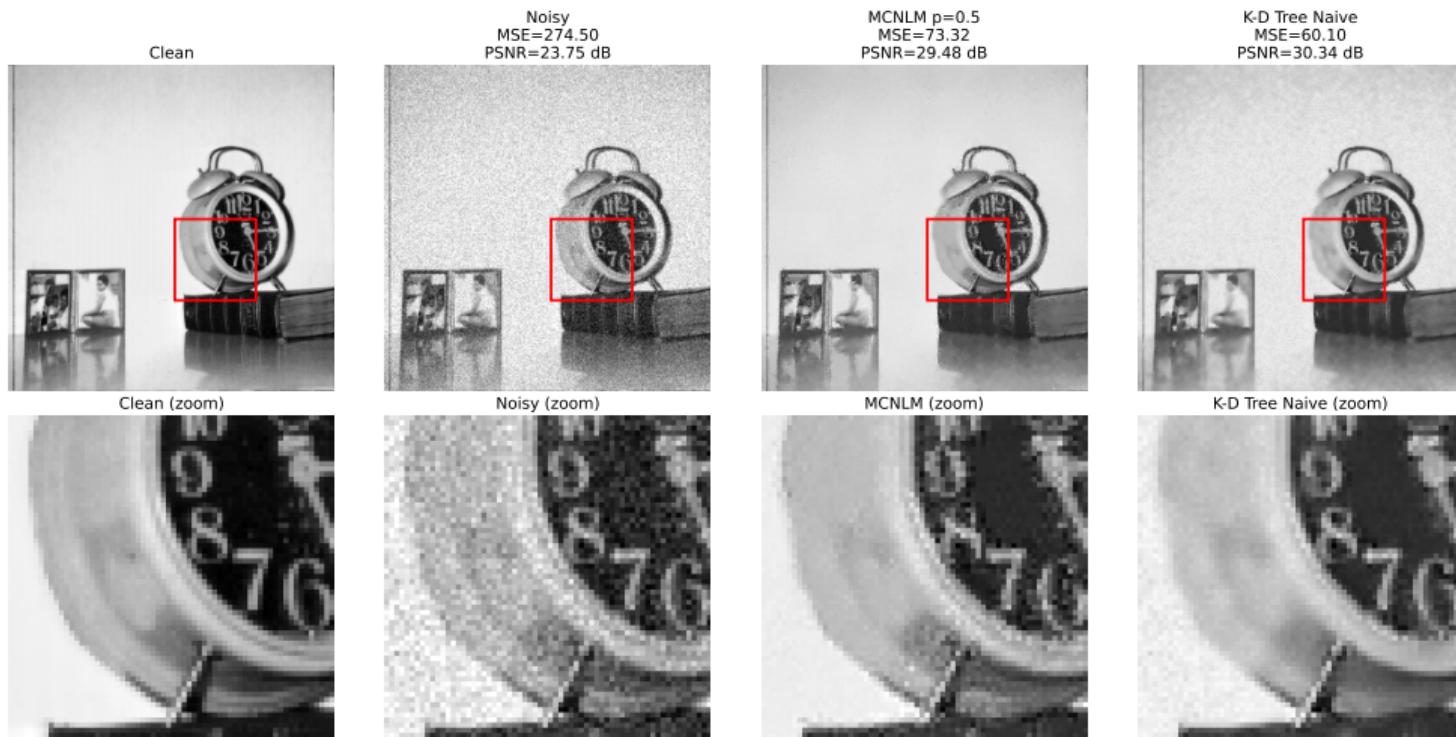
Results: Sampling Ratio ξ



- Test image: 1024×1024 with $\sigma = 17/255$.
- Uniform sampling pattern $p_j = \xi$.
- Higher ξ improves quality with diminishing returns.

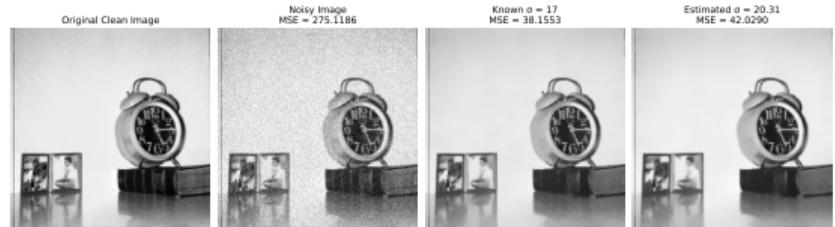
KD-Tree Accelerated NLM

- Build a KD-Tree of patches and query K nearest neighbors.
- Faster similarity search, but harder to parallelize.
- Trade-off: slightly different artifacts vs MCNLM.



Noise Estimation via FFT

- Estimate σ from high-frequency components.
- FFT → mask high frequencies → inverse FFT.
- Use $\sigma = \text{std}(\text{noise})$ for denoising.



Conclusion and Next Steps

- MCNLM reduces NLM cost while preserving image structure.
- Spatial weighting improves robustness in structured regions.
- KD-Tree search is a viable alternative with different artifacts.
- Future work: adaptive sampling, hybrid MCNLM + KD-Tree, GPU parallelism.