Optimality Bounds for Recovering Geometric Information in Images

Brady Sheehan, Advised by Stacey Levine*

I. Motivation

"Is Denoising Dead?"[1]

Techniques for denoising natural images have seen incredible advances over the past several decades. Recent developments in denoising techniques have not been improving image quality much beyond the current state of the art. This realization led to a question: how well can we denoise a natural image? Work toward answering this question has been performed by multiple researchers. Recent statistical analyses have shown that these algorithms are approaching optimality with respect to minimum mean squared error when denoising natural images directly. Figure 1 shows the results of a state of the art denoising algorithm.

One notable approach to answering this question was found by Anat Levin and Boaz Nadler who found optimality bounds for denoising natural image patches without using any assumptions about the distribution of natural images. In this work we are studying optimality bounds for denoising geometric features of an image with the hope of discovering if there is more potential for improvement when working within this framework. More specifically, we want to consider denoising image curvature and then reconstructing the original image. Finding optimal denoising bounds for this new framework could be a breakthrough that would allow for the development of algorithms that far surpass the state of the art when denoising natural images directly.

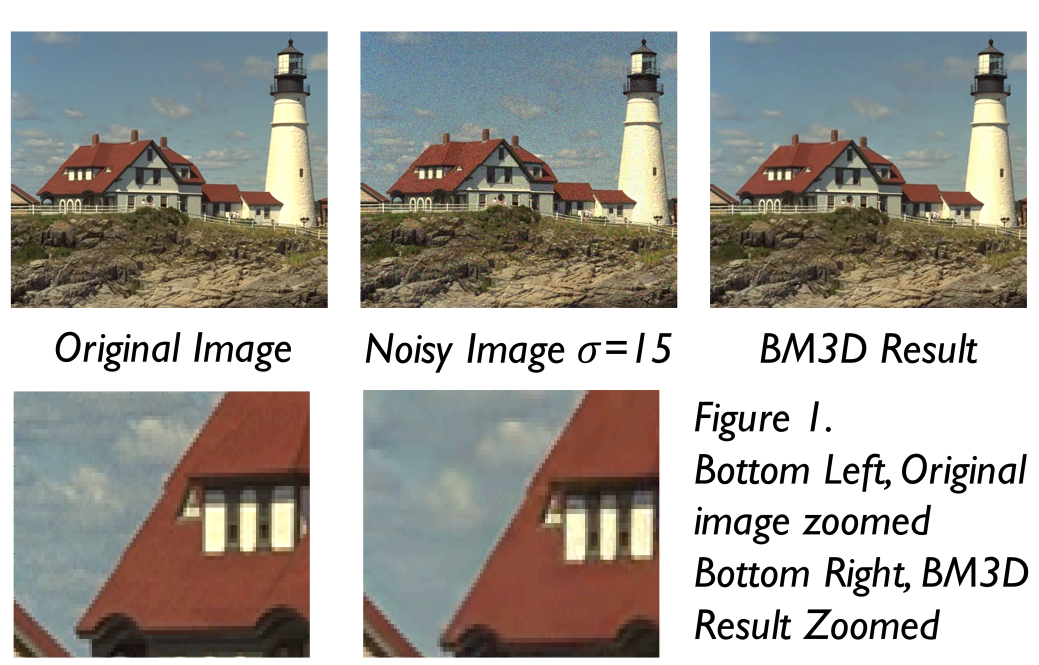


Figure 1.
Bottom Left, Original image zoomed
Bottom Right, BM3D Result Zoomed

II. Background

A noisy image y can be modeled by considering

$$y = x + n$$

where x is the clean image and n is a random noise vector with entries that are independent and identically distributed according to a Gaussian with zero mean and variance σ^2. The goal of image denoising is to estimate a clean image $\hat{y}$ from y.

[1] *"Is Denoising Dead?", Chatterjee and Millanfar.*

One measure of quality for an image denoising algorithm is the mean squared error, MSE, defined as

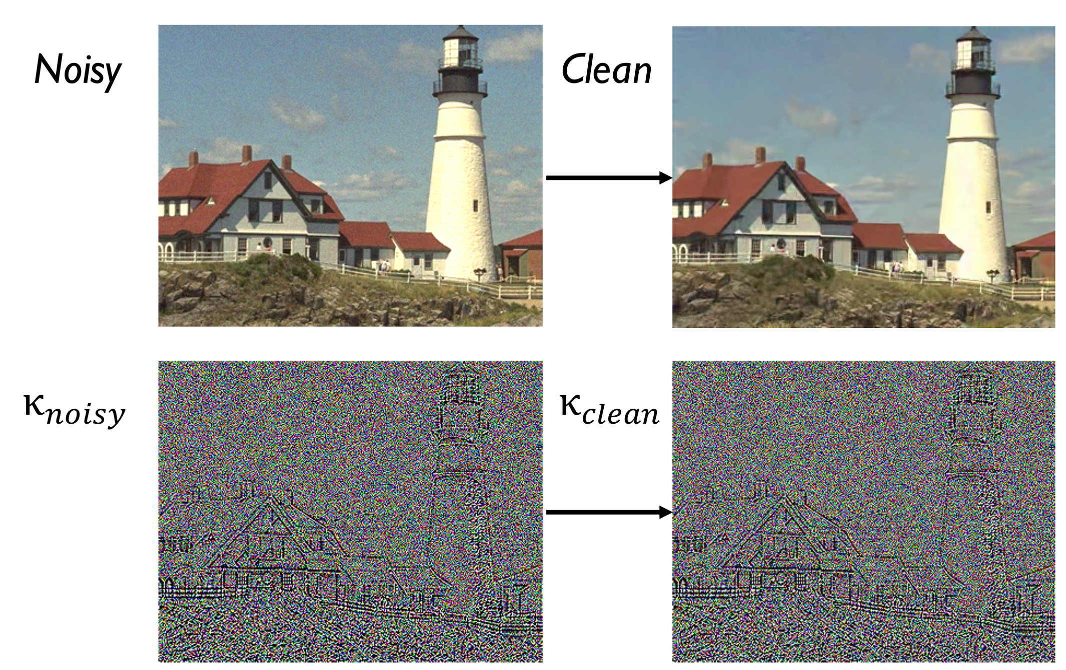
$$MSE = \frac{1}{n}\sum_{j=1}^{n}(x_j - \hat{y}_j)^2,$$

where n is the number of pixels in the images.

It was proposed by Bertalmío and Levine that an image could be denoised by denoising the curvature of its level lines and then reconstructing the original image from the denoised curvature, Figure 3. The curvature of the level lines of an image $y, \kappa(y)$, can be defined as

$$\kappa(y) = \nabla \cdot \left(\frac{\nabla y}{|\nabla y|}\right).$$

Figure 2. How Good Can we Denoise Data?



III. Model

The minimum mean squared error, MMSE, is a lower bound on the MSE. The MMSE can be defined as

$$MMSE = \mathbb{E}[v(y)] = \int p(y)v(y)dy$$

where $v(y)$ is the conditional variance. This MMSE is the lowest achievable denoising error by any denoising algorithm. The goal is to estimate the MMSE of an image patch y with $p(x)$ as the density of assumed clean patches and $p(y)$ the density of noisy patches. This estimation can be adapted for curvature images by replacing y with $\kappa(y)$. To estimate the MMSE, we can approximate it and bound it above and below. Given M clean and noisy pairs of patches $\{(\tilde{x}_j, y_j)\}_{j=1}^{M}$ and N assumed clean patches $\{x_i\}_{i=1}^{N}$, both independently randomly sampled from natural images, we can define the upper and lower bounds as

$$MMSE_\kappa^U = \frac{1}{M}\sum_j \left(\left(\hat{\mu}(\kappa(y_j)) - \kappa(\tilde{x}_{j,c}\right)\right)^2$$

and

$$MMSE_\kappa^L = \frac{1}{M}\sum_j \hat{v}(\kappa(y_j)).$$

These are defined when

$$\hat{\mu}_\kappa(y_j) = \frac{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))\kappa(x_{i,c})}{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))}$$

and

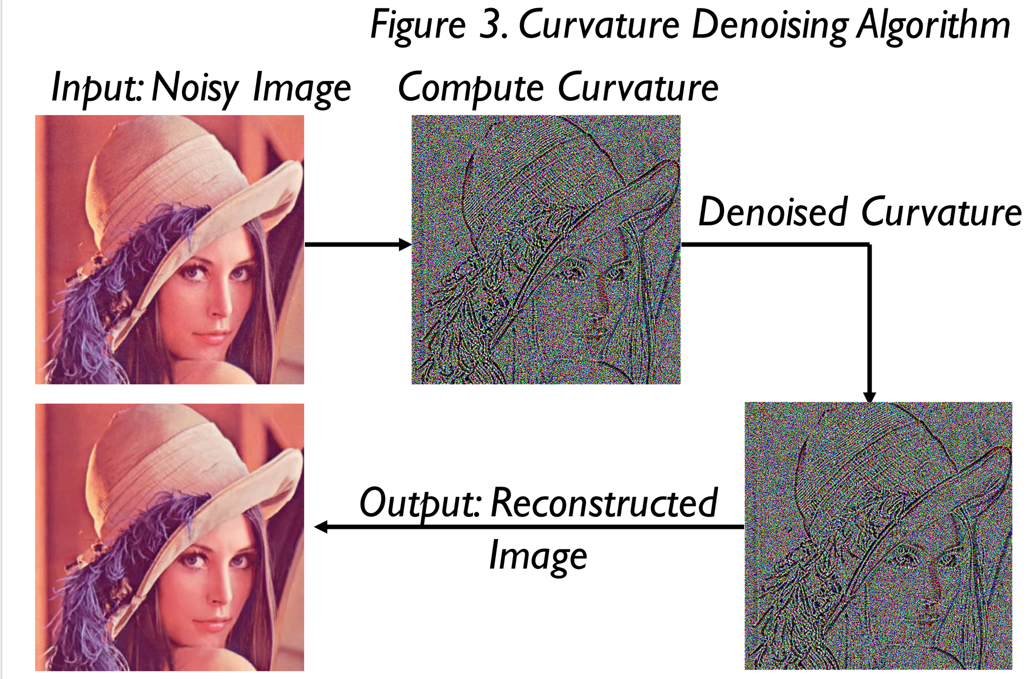
$$\hat{v}_\kappa(y_j) = \frac{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))\left(\hat{\mu}(\kappa(y_j)) - \kappa(x_{i,c})\right)^2}{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))}$$

where $p(\kappa(y_j)|\kappa(x_i))$ is the conditional Laplace distribution,

$$p(\kappa(y_j)|\kappa(x_i)) = \frac{1}{2\hat{b}}e^{-\frac{|\kappa(x)-\kappa(y)-\hat{\phi}|}{\hat{b}}},$$

with estimators $\hat{\phi} = median(\kappa(y) - \kappa(x))$ and $\hat{b} = \frac{1}{n}\sum_{i=1}^{n}|\kappa(x) - \kappa(y) - \hat{\phi}|$. Note that $d = k^2$ is the patch size. Computationally, the $MMSE_\kappa^U$ and $MMSE_\kappa^L$ will be similar for large sample size and will give an accurate estimate for the true optimal $MMSE$.

Figure 3. Curvature Denoising Algorithm

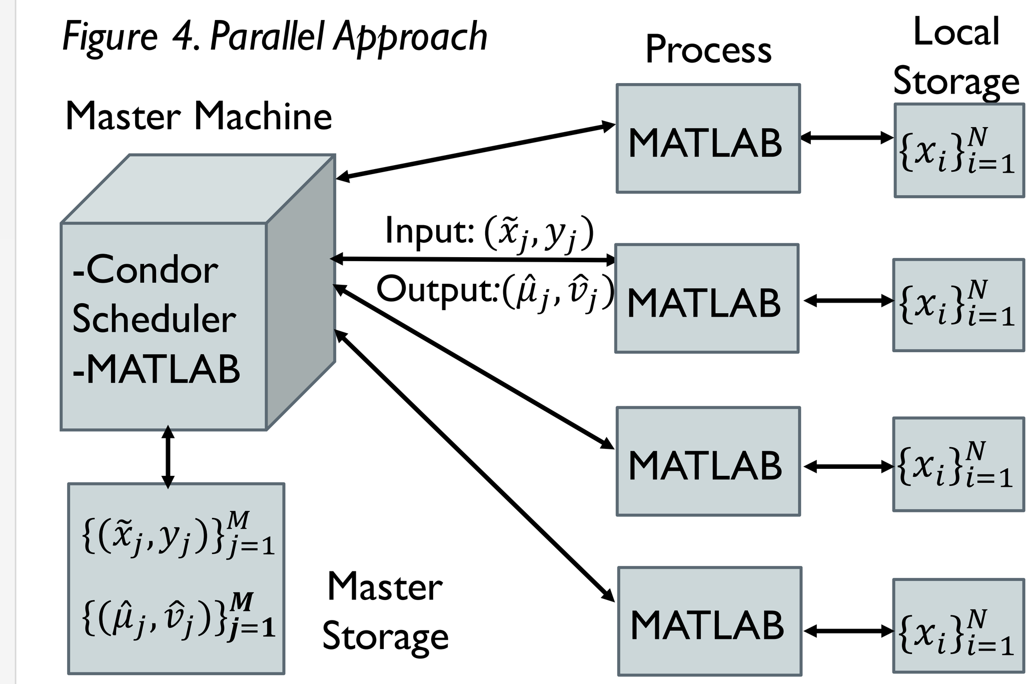


IV. Current Work

We are currently trying to compute these bounds for large sample sizes. When Levin and Nadler performed this experiment with natural image patches, they independently and randomly sampled $M = 2{,}000$ and $N = 10^{10}$ image patches from a set of about $20{,}000$ natural images. Here we need to consider similar sized samples for the bounds to be statistically significant. The experiment as described is computationally intensive. Small scale experiments using MATLAB ran in about 80 hours with $M \approx 100$ and $N \approx 1{,}000$. Therefore, parallelizing the code is a necessity. We started with small sample sizes as a proof of concept and are now working towards parallelizing the computation across a cluster with multiple nodes to make it tractable.

To parallelize this computation, we will use the Condor job scheduler running on one machine inside a parallel cluster, see Figure 4. After computing the conditional mean and variance for each patch, we can easily compute the $MMSE_\kappa^U$ and $MMSE_\kappa^L$.

Figure 4. Parallel Approach



V. Preliminary Results

Setting up a large parallel cluster with Condor and MATLAB has proven difficult. Until this process is finalized, we have run trials on small sample sizes on a single machine. One trial was run with $M = 1{,}000$ and $N = 10{,}000$. The results from this trial gave us peak signal-to-noise bounds for the $MMSE_\kappa^U$ and $MMSE_\kappa^L$ that were not very tight. This evidence emphasizes the need for much larger sample sizes.

VI. Future Work

The next step is to finish processing the $N = 10^{10}$ image patches and attempt to run the experiment on the entire data set. Details from Levin and Nadler's paper suggest that this computation will take about one week with a 100 node cluster. If this holds true for the curvature experiment, the code will need to be further optimized and we will need to add many more nodes to the existing cluster. After running the experiment, we would like to attempt to apply methods used in "Patch complexity, finite pixel correlations and optimal denoising" by Levin and Nadler to bounding curvature denoising.

VII. References

1. B. Russell, A. Torralba, K. Murphy, W. T. Freeman. LabelMe: A Database and Web-based Tool for Image Annotation. *International Journal of Computer Vision*, pages 157-173, Volume 77, Numbers 1-3, May, 2008.
2. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *Computer Vision and Pattern Recognition* (CVPR), 2011 *IEEE Conference* on. IEEE, 2011, pp. 2833–2840.
3. J. Matuk, S. Levine, and M. Bertalmío, "The Curvature Noise Distribution and Applications," in *Midstates Conference for Undergraduate Re- search in Computer Science and Mathematics*. http://www.cs.bgsu.edu/MCURCSM/proceedings/A- 3.pdf, 2015, pp. M1–M4.
4. M. Bertalmío and S. Levine, Denoising an Image by Denoising Its Curvature Image, *SIAM J. Imaging Sciences*, Vol.7, pp. 187-211 (2014).

***This work was partially funded by NSF-DMS #1320829.**