Optimality Bounds for Denoising Image Curvature

Brady Sheehan
Department of Mathematics and Computer Science
Duquesne University, Pittsburgh, PA, USA
June 27, 2016

1 Introduction

How well can we denoise an image? Levin and Nadler from the Weizmann Institute of Science were able to determine bounds for denoising a natural image with respect to minimum mean squared error. Their approach used a nonparametric model and therefore assumed no prior distribution of the images in determining these bounds. The results of their work showed that, when considering small patch sizes, state of the art denoising algorithms are nearing optimality when working with only the natural image. Figure 1 shows the results of a state-of-the-art patch based denoising algorithm, BM3D [7], run on a noisy image. Figure 2 depicts a close-up comparison of one region from the original image and the result after running the BM3D denoising algorithm. The comparison in Figure 2 shows a situation where, despite BM3D's effectiveness, the denoised result lacks detail and sharpness in edges. Motivation for this work stems from how close patch based denoising algorithms are to optimality and the observation that patch based methods still offer room for improvement. Could we find different bounds for image denoising if we considered denoising other features of an image?

In this work we are interested in determining if there are optimality bounds for the performance of denoising algorithms that consider denoising geometric features, specifically curvature, of an image. One of the key points this work explores is if denoising an image by denoising the curvature of its level lines could produce better results than denoising the image itself. This result is important because finding bounds on curvature denoising could be impactful to the larger image processing community. If the theoretical bounds demonstrate that it is possible to recover more information from a noisy curvature image than a noisy natural image, then it could offer researchers ideas for developing better denoising algorithms. On the other hand, if the bounds on curvature denoising demonstrate that it is not possible to recover as much information from a noisy curvature image then researchers know to look for other frameworks for denoising images.

To address this question we will first discuss previous work for denoising images by denoising the curvature image [5,6] and outline the framework used by Levin and Nadler in [2] for bounding patch based denoising algorithms. Then we will introduce a model for bounding the amount of information that can be recovered with respect to minimum mean squared error when denoising the curvature image with patch based methods. Finally we will discuss preliminary results attained by this model for small sample sizes and where this work is headed next.

Figure 1: Left: Original image. Middle: Original image with Gaussian noise added ($\sigma^2 = 15$). Right: BM3D result from denoising Middle.

Figure 2: Comparison of a region from the original image in Figure 1 (Left) to the BM3D result in Figure 1 (Right).

2 Background

2.1 Modeling the Noisy Image

A noisy image y can be modeled by considering

$$y = x + n \tag{1}$$

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.

2.2 Measuring Image Quality

One metric for determining the quality of a denoising algorithm is find the mean squared error, or MSE, of the estimated clean image. The MSE is defined as

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

The MSE measures the average of the squares of the errors between the clean image and the reconstructed image. In general, a lower MSE will indicate better quality in the reconstructed image. For this discussion, x_j will be the jth pixel of the clean image, $\hat{y}_j$ will be jth pixel of the estimated clean image, y_j will be the jth pixel of the noisy image, and n will be the number of pixels in the image. One way to view the MSE is instead to consider the peak signal-to-noise ratio, or $PSNR$ values. The formula for computing $PSNR$ is given in (3). $PSNR$ is measured in terms of the logarithmic decibel scale.

$$PSNR = 10 \log_{10} \left(\frac{1}{MSE} \right) \tag{3}$$

2.3 Image Curvature

It was proposed by Bertalmío and Levine in [6] that an image could be denoised by denoising the curvature of its level lines and then reconstructing the image from the denoised curvature. The curvature of the level lines, $\kappa(y)$, of an image y can be defined as

$$\kappa(y) = \nabla \cdot \left(\frac{\nabla y}{|\nabla y|}\right). \tag{4}$$

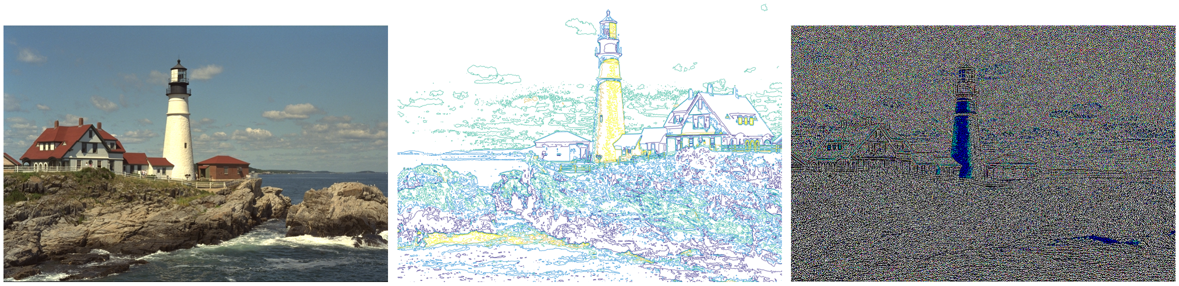

Figure 3: Left: Original image. Middle: Level lines of the original image. Right: Curvature of the level lines of the original image.

In (4), ∇y is the gradient of y and in actual computation will be approximated with finite forward and backward differences. The motivation for denoising the curvature of the level lines of an image is derived from a few important properties of curvature and of the curvature image. When thinking of an image as a surface, we can consider the level lines of that image (See the level set method). Recall that a surface can be perfectly reconstructed from its level lines. Therefore, given the curvature of the level lines of an image, the original image can be reconstructed from the curvature of its level lines. Additionally, it was shown in [6] that when an image is degraded by noise the curvature image is less degraded by it. This gives evidence to conjecture that there is possibility for reconstructing more information from a noisy curvature image than from a noisy natural image.

Another important property of curvature is that it is not a linear operator, that is $\kappa(y) \neq \kappa(x) + \kappa(n)$. It was also shown by Matuk, Levine, and Bertalmío in [5] that there is no statistically significant difference in the CDFs of $\kappa_n = \kappa(y) - \kappa(x)$ and the Laplace distribution. All of these properties aid us in constructing a model for bounding the performance of curvature based image denoising. To construct such a model it is important to have a brief understanding of the curvature denoising framework.

2.4 Curvature Denoising Framework

The process of denoising an image by denoising its curvature image is broken into three steps. The first step is to compute the curvature of the level lines of an image. Then, by applying some of the properties previously described, the noisy curvature image can be denoised by using adaptations of existing denoising algorithms to assume Laplace noise. One denoising method that works well for curvature images is applying a median filter to the image. A median filter can be applied to local windows across the image and replaces each pixel with the median of its neighbors. Once we have the denoised curvature image, the original image can be reconstructed using a reconstruction equation such as

$$\hat{y} = argmin_y \int_\Omega |\kappa(y) - \kappa_{den}| + \frac{\lambda}{2} \int_\Omega (y - x)^2, \tag{5}$$

where κ_{den} is the denoised curvature image and Ω is the domain of the image. The first term in (5) ensures that the curvature of the image's level lines, $\kappa(y)$, will approach that of the denoised curvature κ_{den}. The second term in (5) ensures that the average intensity values of y are close to those of x. Figure 4 shows images that are generated during the curvature denoising process.

Figure 4: Left: Original image. Middle Left: Curvature of original image. Middle Right: Denoised curvature of original image. Right: Reconstructed image from denoised curvature.

2.5 Bounding Natural Image Denoising Performance

Recall that the minimum mean squared error, $MMSE$, is a lower bound on the MSE [1]. To find a bound on how much information can be recovered using patch based denoising algorithms, one can estimate the $MMSE$ of an image patch y. To do this, assume we are given $p(x)$ as the density of assumed clean $k \times k$ patches and $p(y)$ as the density of noisy $k \times k$ patches. We can then define the lowest possible denoising error by any denoising algorithm with the expected value of the conditional variance given as

$$MMSE = \mathbb{E}[\nu(y)] = \int p(y)\, \nu(y)\, dy. \tag{6}$$

It is explained in [2] that we can quantify the inherent ambiguity of the denoising problem and the statistics of natural images by considering the $MMSE$. This is true because any number of natural images that have a noise level within that of y could have generated y. To compute $MMSE$ value, we would need to know the true density $p(x)$. Since this is unknown, we will bound this theoretical error above and below. The conditional variance $\nu(y)$ and conditional mean $\mu(y)$ are given in (7) and (8).

$$\nu(y) = \mathbb{E}[(x_c - \mu(y))^2 | y] = \int p(y|x)(x_c - \mu(y))^2\, dx \tag{7}$$

$$\mu(y) = \mathbb{E}[x_c | y] = \int \frac{p(y|x)}{p(y)}\, p(x) x_c\, dx \tag{8}$$

Consider a set of M clean and noisy pairs of patches $\{(\tilde{x}_j, y_j)\}_{j=1}^{M}$ and another independent set of N clean patches $\{x_i\}_{i=1}^{N}$ that are both independently randomly sampled from natural images. An important idea in this framework is that the distribution $p(x)$ is unknown and classifies this framework as nonparametric. However, we can use bootstrapping to sample a large subset of image patches from the unknown population achieving an accurate representative of $p(x)$. The MMSE can be bounded above and below, as in [1], by

$$MMSE^U = \frac{1}{M} \sum_j (\hat{\mu}(y_j) - \tilde{x}_{j,c})^2 \quad \text{and} \quad MMSE^L = \frac{1}{M} \sum_j \hat{\nu}(y_j). \tag{9}$$

These bounds are defined when $\hat{\nu}(y_j)$ is the approximate variance and $\hat{\mu}(y_j)$ is the approximate mean,

$$\hat{\nu}(y_j) = \frac{\frac{1}{N} \sum_i p(y_j|x_i)(\hat{\mu}(y_j) - x_{i,c})^2}{\frac{1}{N} \sum_i p(y_j|x_i)} \quad \text{and} \quad \hat{\mu}(y_j) = \frac{\frac{1}{N} \sum_i p(y|x_i) x_{i,c}}{\frac{1}{N} \sum_i p(y|x_i)}, \tag{10}$$

with Gaussian noise,

$$p(y|x) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{||x-y||^2}{2\sigma^2}}, \tag{11}$$

where $d = k^2$ and σ^2 is the variance.

Computationally, the $MMSE^U$ and $MMSE^L$ will be similar for large values of N and will give an accurate estimate for the true optimal $MMSE$. Since the $MMSE^U$ measures the error of the estimator $\hat{\mu}(y_j)$, this quantity provides an upper bound on the true $MMSE$. It is also shown in [1] that, for a finite set of samples, the $MMSE^L$ provides a lower bound on the optimal $MMSE$ by showing that $\hat{\nu}(y_j)$ provides a lower bound on the true variance $\nu(y_j)$.

3 Model Description

Rather than considering the $MMSE$ in terms of natural image patches, we will want to consider curvature of image patches found with (4). We will now assume that κ_n is distributed as Laplacian instead of Gaussian. Given M clean and noisy pairs of patches, $\{(\tilde{x}_j, y_j)\}_{j=1}^{M}$, and N assumed clean patches, $\{x_i\}_{i=1}^{N}$, we can bound curvature denoising similarly to [1]. We will define the $MMSE_\kappa^U$ and $MMSE_\kappa^l$ as

$$MMSE_\kappa^U = \frac{1}{M}\sum_j (\hat{\mu}_\kappa(\kappa(y_j)) - \kappa(\tilde{x}_j)_c)^2 \quad \text{and} \quad MMSE_\kappa^L = \frac{1}{M}\sum_j \hat{\nu}_\kappa(\kappa(y_j)), \tag{12}$$

where $\kappa(x)_c$ is the center pixel of the curvature of image patch x. The approximate mean, $\hat{\mu}_\kappa$, and variance, $\hat{\nu}_\kappa$, will then be given as

$$\hat{\mu}_\kappa(y_j) = \frac{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))\kappa(x_i)_c}{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))} \quad \text{and} \quad \hat{\nu}_\kappa(y_j) = \frac{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))(\hat{\mu}(y_j) - \kappa(x_i)_c)^2}{\frac{1}{N}\sum_i p(\kappa(y_j)|\kappa(x_i))}. \tag{13}$$

The $p(\kappa(y)|\kappa(x))$ will be given by the Laplace distribution,

$$p(\kappa(y)|\kappa(x)) = \frac{1}{2\hat{b}} \exp\left(-\frac{|\kappa(y) - \kappa(x) - \hat{\mu}|}{\hat{b}}\right), \tag{14}$$

where the estimators $\hat{\mu}$ and $\hat{b}$ are defined as

$$\hat{\mu} = median(\kappa(y) - \kappa(x)) \quad \text{and} \quad \hat{b} = \frac{1}{n}\sum_{i=1}^{n} |\kappa(y) - \kappa(x) - \hat{\mu}|. \tag{15}$$

Ultimately, these bounds should give an estimate of how well we can expect to recover curvature information with respect to mean squared error from a noisy image using patch based denoising algorithms. Not only do we need to consider a large number of image patches, but we also need to compute the $MMSE_\kappa^U$ and $MMSE_\kappa^L$ for multiple $k \times k$ patch sizes and multiple added noise distributions. Computing these values allows us to make valid claims for optimality of denoising algorithms that consider the same parameters.

4 Model Implementation

The bounds on the $MMSE$ will be meaningful only as the number of image patches approaches infinity. Therefore, we need to consider a large number of images and sample from this set a large number of image patches for the experiment to be statistically significant. Color images will be selected from the LabelMe dataset [3]. This dataset is diverse and has a MATLAB compatible tool for working with the dataset. We will select from this dataset about $20,000$ images. The images used for this experiment are the same images used in [1] for bounding natural image denoising with patch based methods and all images will be converted to grayscale before processing.

We will consider the set of $20,000$ natural images to be an unbiased representative of noise free natural image statistics. We will use bootstrapping to simulate one population of curvature patches, $\{x_i\}_{i=1}^{N}$, and another set of clean and noisy pairs of curvature patches, $\{(\tilde{x}_j, y_j)\}_{j=1}^{M}$. To generate these sets, we will first downsample all the images by a factor of two and low-pass filter them with a Gaussian filter. Next we will compute an approximation of the curvature defined in (4) with forward-backward differences of each image in this set. Then the set of N images is generated by randomly selecting a $k \times k$ patch from the set of images and then randomly selecting a patch from within that image, $N = 10^{10}$ times. The set of M image patch pairs is generated the same way for $M = 2,000$. However, this set of M image patches is sampled and then the duplicated half has random Gaussian noise with mean zero and σ^2 variance added to it. The set of N images is assumed to be noise free because the amount of noise added to half the patches in M is large in comparison.

The $MMSE_\kappa^U$ and $MMSE_\kappa^L$ were programmed in MATLAB. Implementing this experiment posed multiple challenges. One problem was in computing $\hat{\mu}$. The curvature image is scaled in the range $[-4, 4]$ instead of the usual range $[0, 255]$. As a result, early experiments produced means that were negative and resulted in loss of information. To fix this we needed to rescale the curvature image in the range $[0, 255]$. Another challenge this project faced resulted from the time complexity of the MATLAB implementation. Early versions of the MATLAB program took about one week on a single computer with approximately $M = 1,000$ and $N = 10,000$. These runs only considered $k^2 = 25$ and $\sigma^2 = 18$. As a result, the code was vectorized to improve performance. Currently, the model can be run with $M = 500$ and $N = 50,000$ in about ten minutes. However, future implementations will still need to be parallelized to handle a computation on the same scale as [2] because one computer runs out of memory when considering $N > 100,000$.

5 Preliminary Results

Results found with small samples emphasize the need for a larger sample size. In the experiment performed in [1] for natural image patches, they claim that the bounds in (9) will only be similar for large sample sizes. We believe the same should hold true for the bounds defined in (12). The bounds we found with samples sizes of approximately $M = 1,000$ and $N = 10,000$ were not tight but reasonable. These bounds were reasonable because the PSNR values were between 0dB and 90dB.

After vectorizing the code for improved efficiency, the code ran fast enough to be executed on a larger sample size. With $M = 500$ and $N = 50,000$, we were able to calculate another set of upper and lower bounds. With this sample size, the bounds found were between a lower bound of 13dB and an upper bound of 40dB. These bounds do differ from the bounds found with the incremental code, however they are getting much tighter and approaching the results cited in [1]. Evidence to suggest that this project is nearing an appropriate sample size is supported by the histogram of means show in Figure 5. This histogram shows the distribution of mean values for each of the M image patches and is approximately normal. As the sample size grows, this histogram will appear to be even more normal. Both of these observations support that the model is working and merely needs to be considered with larger sample sizes to make similar claims to [1].

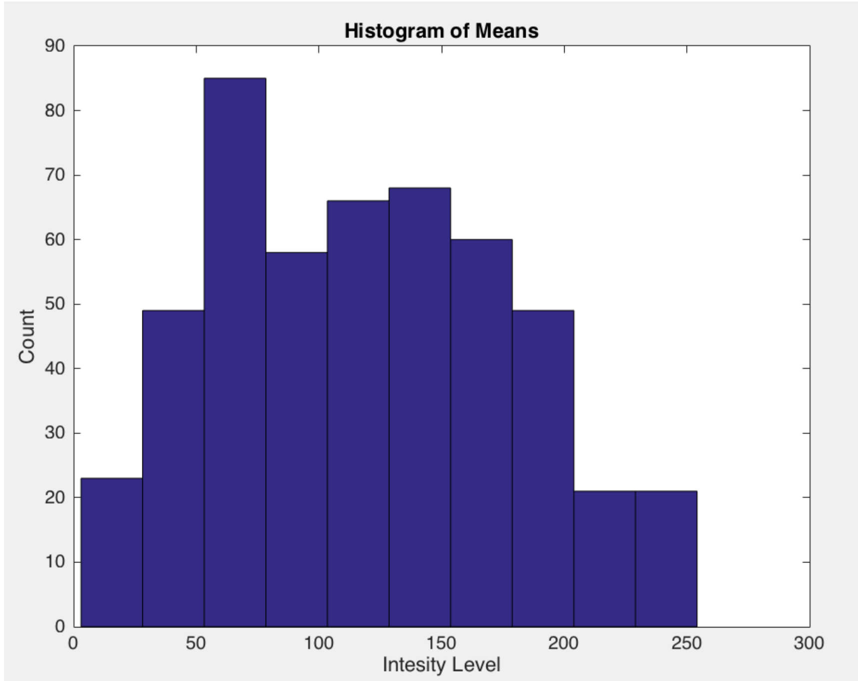

Figure 5: Histogram of the means computed with equation (10) for $M = 500$ and $N = 50,000$.

6 Conclusion and Future Work

The preliminary results discussed show how this framework has promise for bounding the performance of patch based curvature denoising algorithms with respect to minimum mean squared error. Since the distribution of means appears approximately normal, I believe that this project will be able to make strong claims in the near future. However, until bounds are computed for a larger set of samples, limitations on this work in terms of patch size and level of noise are unknown.

In the future, we would like to use a parallel computing cluster to compute the bounds on the minimum mean squared error for curvature denoising for multiple patch sizes and multiple added σ^2 noise distributions. Such a cluster would help distribute the processing of a large sample, such as $M = 2000$ and $N = 10^{10}$. Once bounds are determined with this model, we would like to apply an adaption of the framework used in [2] to curvature denoising. After doing this and demonstrating that this model is capable of bounding the recovery of geometric information, we would like to adapt this framework for bounding other denoising algorithms. One interesting framework to consider bounding the performance of would be denoising an image by denoising its components in a moving frame [4]. Additionally, it would be interesting to determine bounds with respect to other error metrics such as the structural similarity (SSIM) index.

7 References

[1] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds", in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference, pp. 2833–2840.

[2] A. Levin, B. Nadler, F. Durand, and W. T. Freeman. "Patch complexity, finite pixel correlations and optimal denoising", In ECCV, 2012.

[3] B. Russell, A. Torralba, K. Murphy, and 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.

[4] G. Ghimpeteanu, T. Batard, M. Bertalmío, and S. Levine, "Denoising an image by denoising its components in a moving frame", in Image and Signal Processing (Lecture Notes in Computer Science), vol. 8509. Berlin, Germany: Springer-Verlag, 2014, pp. 375–383.

[5] J. Matuk, S. Levine, and M. Bertalmío, "The Curvature Noise Distribution and Applications", in Midstates Conference for Undergraduate Research in Computer Science and Mathematics. http://www.cs.bgsu.edu/MCURCSM/proceedings/A-3.pdf, 2015, pp. M1–M4.

[6] M. Bertalmío and S. Levine, "Denoising an Image by Denoising Its Curvature Image", SIAM J. Imaging Sciences, Vol.7, pp. 187-211 (2014).

[7] Marc Lebrun, An Analysis and Implementation of the BM3D Image Denoising Method, Image Processing On Line, 2 (2012), pp. 175–213. http://dx.doi.org/ .5201/ipol.2012.l-bm3d.