Image denoising using diffusion on curvelet-scaled Gabor filter responses

Jim Bremer [*] Yoel Shkolnisky [†] Arthur Szlam [‡]

October 29, 2007

1 Introduction

In [5], a general framework for adaptive function regularization was introduced, and this framework was demonstrated in several applications, including image denoising. The basic idea of the method applied to image denoising is to choose a set of features and consider the pixels of the image as lying in feature space. We then try to use the heat equation on the points in feature space to smooth the image. In this note, we would like to give a more in depth account of image denoising in this framework using curvelet scaled Gabor filter responses as features.

1.1 Weights from images

To build the image-dependent weights we first associate a feature vector to each location x in the image I by convolving I with a filter bank. Here we will be mostly concerned with the Gabor filters described in 1.2. If $g = (g_1, \cdots, g_d)$ are our filters, $Q = [0,1] \times [0,1]$, and $I : Q \mapsto \mathbb{R}$ is our (noisy) image, map Q into $\mathbb{R}^d$ by

$$\begin{aligned} \mathcal{F}_g(I) : Q &\to \mathbb{R}^d \\ x &\mapsto (I * g_1(x), \cdots, I * g_d(x)) \end{aligned} \tag{1}$$

Once we have features $\mathcal{F}(\mathcal{I})$, let

$$\rho(x,y) = ||\mathcal{F}_g(I)(x) - \mathcal{F}_g(I)(y)||\,,$$

where $||\cdot||$ is the Euclidean norm in $\mathbb{R}^d$. Now we pick a (usually exponentially decreasing) function h and variance parameter ϵ, and define

$$W(x,y) = h\left(\frac{\rho(x,y)^2}{\epsilon}\right). \tag{2}$$

[*]Department of Mathematics, U.C. Davis
[†]Department of Mathematics, Yale University,
[‡]Department of Mathematics, U.C.L.A.

A common choice is $h(a) = \exp(-a)$. The idea is that we expect that very close data points (with respect to ρ) will be similar, but do not want to assume that far away data points are necessarily different. The exponential weight h in (2) gives a large preference to very close points.

It is computationally prohibitive to find $W(x, y)$ for all pairs of pixels. In addition, unless there are patterns repeated in many locations in the image, far away pixels are unlikely to be useful in determining the denoising. So we modify ρ by choosing sets $S = S(x) \subset Q$ so that

$$\rho(x, y) = \begin{cases} \rho_g(x, y) & \text{if } y \in S(x); \\ \infty & \text{otherwise.} \end{cases} \quad . \tag{3}$$

A simple and effective choice for $S(x)$ is a square of fixed side length centered at x. Even within the coarse search box S there may be many pixels that are not too similar to x. To further decrease the computational complexity and to insure that x only communicates with pixels very similar to it, we fix a small number k and set $\rho(x, y) = \infty$ for any y which is not one of the k nearest ρ neighbors of x. Finally, we follow [4] and modify ρ so that the distance between x and its four nearest spatial neighbors is not set to ∞ regardless of whether they are among the k nearest points to x in feature space; this leads to a dramatic reduction in artifacts. With h as above, building W with the modified ρ results in a matrix with at most $k + 4$ entries per row.

1.2 Construction of curvelet-scaled Gabor filters

To build the filters we partition the frequency plane into radial bands, and then the radial bands into polar rectangular tiles, doubling the number of tiles every two bands, as in [2]. In each frequency tile, we place a Gaussian bump with mean in the centroid of the tile, and variances scaled as the ratio of the radial length of the tile to the angular width of the tile. To make real filters, we also place a bump in the tile reflected through the origin.

More precisely: the radial variable is split into $N_B + 1$ bands

$$\begin{aligned} B_0 &= \frac{1}{2^{(N_B+1)}}[0, 2] \\ B_j &= \frac{1}{2^{(N_B+1)}}[2^j, 2^{j+1}] \quad \text{for } j = 2, \ldots, N_B \end{aligned}$$

giving $N_B + 1$ annuli. For each annulus, $[-\pi, \pi]$ is split into $N_\theta(B_i)$ pieces, where $N_\theta(B_0) = 1$, $N_\theta(B_1) = 4$, and then $N_\theta(B_1)$ is doubled every two bands after $N_\theta(B_1)$ giving the sequence $1, 4, 4, 8, 8, \cdots$.

An elliptical Gaussian is placed in each polar rectangle. To do this, first we define center points:

$$\begin{aligned} C_\theta &= \frac{\theta_1 + \theta_2}{2} \\ C_\rho &= \frac{\rho_1 + \rho_2}{2}. \end{aligned}$$

Pick a number δ and choose variances such that the value of the Gaussian centered at (C_θ, C_ρ) is δ at the points $\left(\frac{\rho_1+\rho_2}{2}, \theta_2\right)$ and $\left(\rho_2, \frac{\theta_1+\theta_2}{2}\right)$. In our experiments below, we take $N_B = 6$ and $\delta = .3$.

1.3 Evolving the heat equation in feature space

We interpret the weight $W(i,j)$ as a measure of similarity between the pixels i and j. A natural averaging filter acting on functions on Q can be defined by normalization of the weight matrix as follows: let

$$D(x) = \sum_{y \in V} W(x,y)^1 ,$$

and let the filter be

$$K(x,y) = D^{-1}(x) W(x,y) , \tag{4}$$

so that $\sum_{y \in V} K(x,y) = 1$. This filter acts on a function f on Q via

$$Kf(x) = \sum_{z \in Q} K(x,z) f(z) .$$

and hence it is a local averaging operation, with locality measured by the similarities W. One can also think of the matrix $K = D^{-1}W$ as a diffusion or random walk on Q which is run for one step, by multiplying from the other side. This filter can be iterated several times by considering the power K^n; from the point of view of the diffusion process, this corresponds to taking n steps of the random walk, whose transition probabilities are the transpose of K. We can think of applying the powers of K as running the heat equation on Q embedded in the feature coordinates. This heat equation is nonlinear because of the use of the image in the definition of K, but is linear in the sense that we do not update K after applying it. Thus, as $t \mapsto \infty$, $K^n f$ tends to a constant.

We can balance smoothing by K with fidelity to the original noisy function by setting

$$f_{n+1} = (Kf_n + \beta f)/(1+\beta) \tag{5}$$

where $\beta > 0$ is a parameter to be chosen, and large β corresponds to less smoothing and more fidelity to the noisy image. This is a standard technique in PDE based image processing, see [3] and references therein. If we consider iteration of K as evolving a heat equation, the fidelity term sets the noisy function as a heat source, with strength determined by β. Note that even though when we smooth in this way, the steady state is no longer the constant function, we still do not usually wish to smooth to equilibrium.

2 Experiments

We now show the results of some denoising experiments. We build the Gabor filters and K as above, setting $h(a) = -a$. We choose the parameters ϵ and n by

[1] Note that $D(x) = 0$ if and only if x is not connected to any other vertex, in which case we trivially define $D^{-1}(x) = 0$, or simply remove x from the graph.

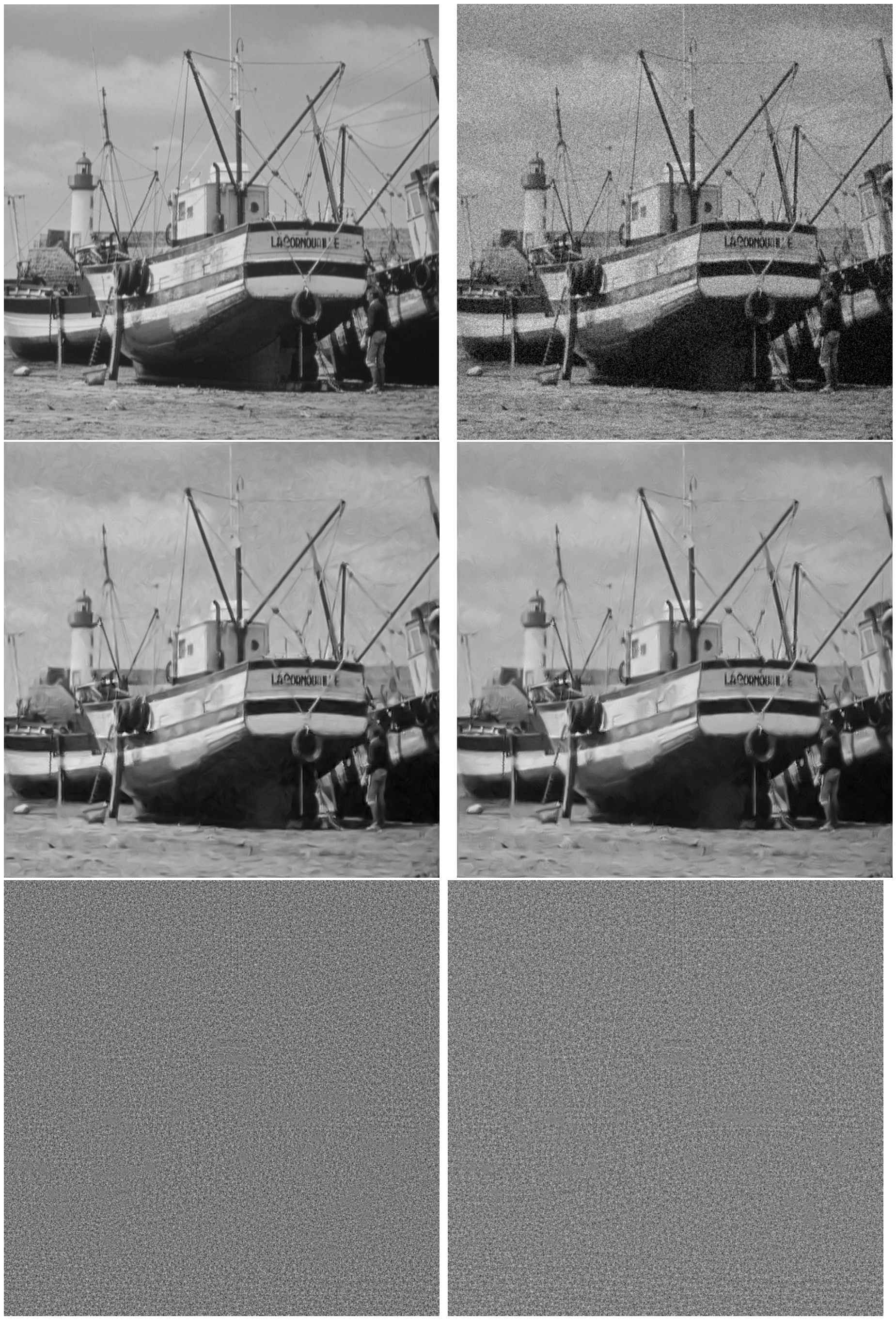

Figure 1: Top left: clean boat image. Top right: noisy boat with $\sigma = 20$. Middle right:denoising using diffusion in NL-means type patch embedding, SNR=16.75. Middle left: denoising using diffusion on curvelet-Gabor features averaged with the NL-means type denoising, SNR=17.13. Bottom right: residual from the NL-means type denoising. Bottom left is the residual from the denoising using diffusion on curvelet-Gabor features averaged with the NL-means type denoising.

Figure 2: Top left: clean Lena image. Top right: noisy Lena with $\sigma = 20$. Middle right:denoising using diffusion in NL-means type patch embedding, SNR=17.65. Middle left: denoising using diffusion on curvelet-Gabor features averaged with the NL-means type denoising, SNR=17.93. Bottom right: residual from the NL-means type denoising. Bottom left is the residual from the denoising using diffusion on curvelet-Gabor features averaged with the NL-means type denoising.

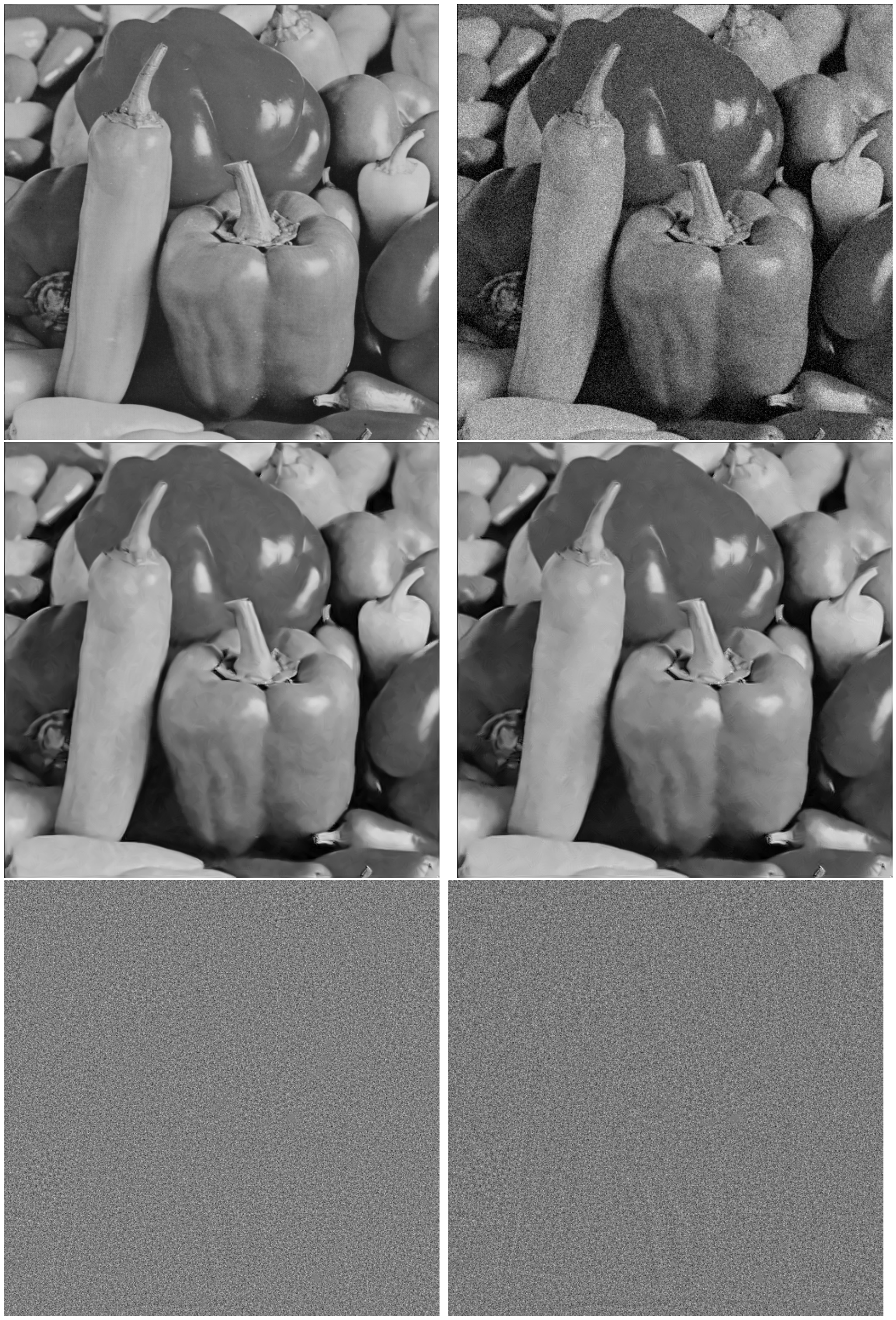

Figure 3: Top left: clean peppers image. Top right: noisy peppers with $\sigma = 20$. Middle right:denoising using diffusion in NL-means type patch embedding, SNR=17.88. Middle left: denoising using diffusion on curvelet-Gabor features averaged with the NL-means type denoising, SNR=18.28. Bottom right: residual from the NL-means type denoising. Bottom left is the residual from the denoising using diffusion on curvelet-Gabor features averaged with the NL-means type denoising.

hand to maximize recovered SNR. As a baseline in our experiments, we will use a 7×7 NL-means type denoising. This NL-means denoising is obtained exactly as the Gabor-filter denoising, except with a different choice of filters. e.g. for 3×3 patches, use the 9 filters

$$f_{1,0} = \begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \quad , \quad f_{1,2} = \begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \quad \dots \quad f_{3,3} = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix} \quad ; \tag{6}$$

for 7×7 patches, we get 49 filters. Once we have the filter responses from the image (which are just the 7×7 patches surrounding a given pixel), we proceed as above with the construction of ρ_P using the filters $f_{i,j}$, and K_P using ρ_P, again choosing all parameters to maximize recovered SNR. We note that in terms of SNR, and in our subjective assesment, in terms of visual quality, this form of the NL-means algorithm (using a coarse search with a small fixed number of neighbors for each pixel chosen from inside the coarse search, multiple iterations with a relatively small ϵ in equation (2), and forcing the four nearest spatial neighbors of a pixel to be neighbors in the weights) outperforms [1] and gives results equivalent to [4].

It has been reported that experimentally [4] that choosing the parameter β to be 0 is optimal for the NL means type denoising; our experience corrobrates this. In terms of SNR, the Gabor feature denoising tends to perform better with a non-zero fidelity, but visual artifacts are more pronounced. For simplicity of comparison for between both sets of weights, we choose β in equation (5) to be 0.

The images cleaned by diffusion on the Gabor filters and by the patch filters have essentially equivalent SNR's. But more important than a simple ranking is the fact that the two choices of filters distill different and complementary information. For example, the Gabor filters diffusion does a much better job preserving the thin, faint cables in the boats image, and the soft edged pole in the Lena image. On the other hand, the patch filters smooth content free portions of the images much more completely, and are superior for preserving repeating textures, such as the rapidly alternating light and dark stripes in the bottom left of Lena's hat. This suggests taking the additive mean of the denoisings obtained by the two methods; and in fact, doing so results in a significant and consistent increase in recovered SNR as well as in subjective image quality over either of the methods alone.

Figures 1, 2, and 3 display examples of denoising with a diffusion on the curvelet-Gabor features averaged with an NL-means denoising, and the NL-means denoising for comparison. On the top left of each figure we have the clean image. On the top right is the noisy image f_0 with $\sigma = 20$. On the middle right of the each figure is a denoising using diffusion in NL-means type patch embedding, and then on the middle left is the denoising using diffusion on curvelet-Gabor features averaged with the NL-means denoising. On the bottom right of the each figure is the residual from the NL-means type patch embedding, and on the bottom left is the residual from the denoising using diffusion on curvelet-Gabor features averaged with the NL-means denoising.

References

[1] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 4(2):490–530 (electronic), 2005.

[2] E. J. Candes and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise smooth singularities. *Comm. Pure Appl. Math.*, 2004.

[3] T. F. Chan and J. Shen. *Image processing and analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005. Variational, PDE, wavelet, and stochastic methods.

[4] Guy Gilboa and Stanley Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling Simulation*, 6(2):595–630, 2007.

[5] A. Szlam, Mauro Maggioni, and Ronald Coifman. Regularization on graphs with function-adapted diffusion processes. *submitted*, 2006.