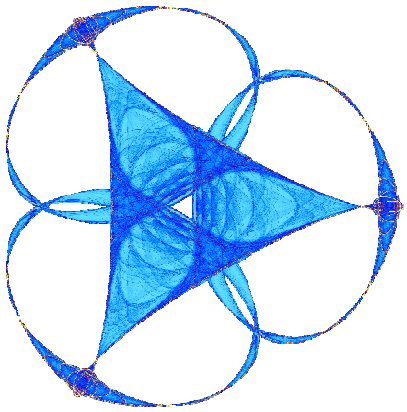
DENOISING AN IMAGE BY DENOISING ITS CURVATURE IMAGE

By

Marcelo Bertalmío and Stacey Levine

IMA Preprint Series #2411

(September 2012)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS
UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455-0436
Phone: 612-624-6066 Fax: 612-626-7370
URL: http://www.ima.umn.edu

DENOISING AN IMAGE BY DENOISING ITS CURVATURE IMAGE

MARCELO BERTALMÍO AND STACEY LEVINE

Abstract. In this article we show that when an image is corrupted by additive noise, its curvature image is less affected by it, i.e. the PSNR of the curvature image is larger. We conjecture that, given a denoising method, we may obtain better results by applying it to the curvature image and then reconstructing from it a clean image, rather than denoising the original image directly. Numerical experiments confirm this for several PDE-based and patch-based denoising algorithms. The improvements in the quality of the results bring us closer to the optimal bounds recently derived by Levin et al. [1, 2].

1. Introduction. We start this work trying to answer the following question: when we add noise of standard deviation σ to an image, what happens to its curvature image? Is it altered in the same way?

Let's consider a grayscale image I, the result of corrupting an image a with additive noise n of zero mean and standard deviation σ:

$$I = a + n. \tag{1.1}$$

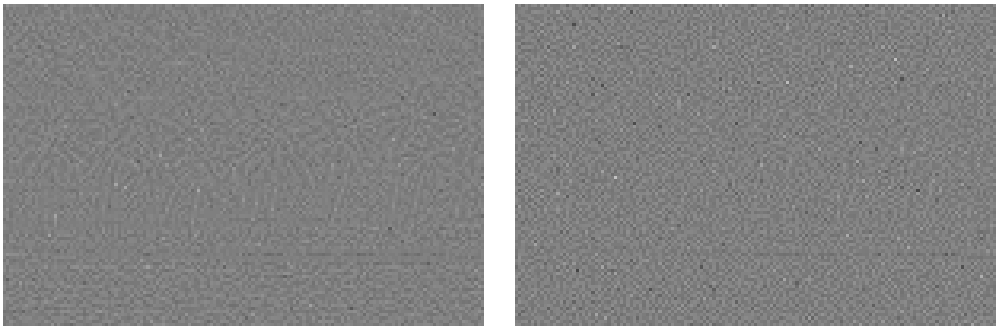
Figure 1.1 shows on the left an image a from the Kodak database [3] and its corresponding curvature image $\kappa(a)$; on the right we see I and $\kappa(I)$, where I has been obtained adding Gaussian noise of $\sigma = 25$ to a. Notice that is difficult to tell the curvature images apart because they look mostly gray, which shows that their values lie mainly close to zero (which corresponds to the middle-gray value in these pictures). We have performed a non-linear scaling[1] in fig. 1.2 in order to highlight the differences, and now some structures of the grayscale images become apparent, like the boundary between sky and water, or the contours of the palm trees. However, when treating the curvature images as images in the usual way, they appear less noisy than the images that originated them; that is, the difference in noise between a and I is much more striking than that between $\kappa(a)$ and $\kappa(I)$.

(a) (b) (c) (d)

FIG. 1.1. *(a), (b): image and its curvature. (c), (d): after adding noise. For visualization purposes we have linearly scaled the curvature images so that the mid-gray value represents zero curvature, black is minimum curvature and white is maximum curvature.*

This last observation is corroborated in figure 1.3 which shows, for Gaussian noise and different values of σ, and for the same image in fig. 1.1, the *noise* histograms of I and $\kappa(I)$, i.e. the histograms of $I - a$ and of $\kappa(I) - \kappa(a)$. We can see that, while the noise in I is $N(0, \sigma^2)$ as expected, the curvature image is corrupted by noise that, if we model as additive, has a distribution resembling the Laplace distribution, with standard deviation smaller than σ. Consistently, in terms of Signal to Noise Ratio

[1] We have used the formula $y = 127.5 * sign(x) * \sqrt{|x|} + 127.5$, where x is the curvature linearly scaled into the range $[-1, 1]$.



(a) (b)

FIG. 1.2. *Close ups of the clean curvature (left) and noisy curvature (right) with non-linear scaling to highlight the differences.*

(SNR) the curvature image is better (higher SNR, less noisy) than I, as is noted in the figure plots.

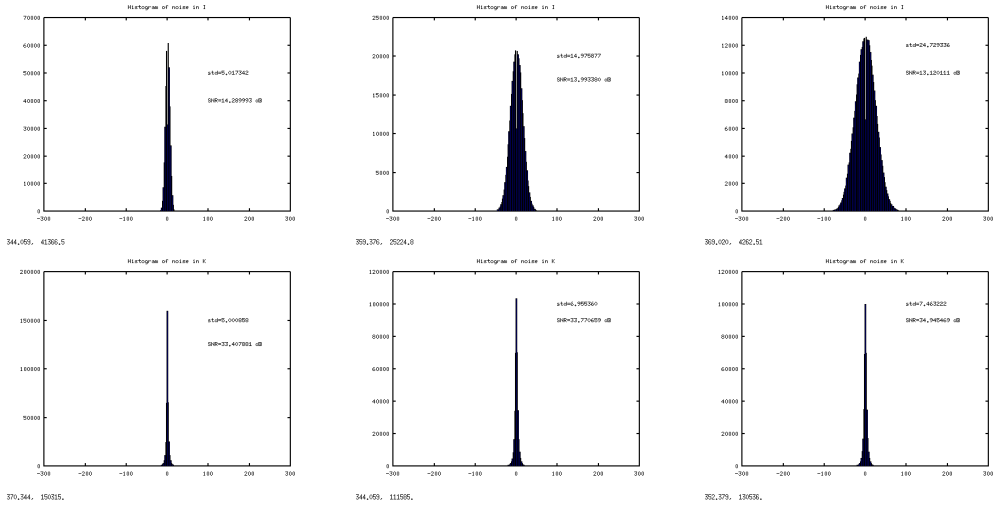


FIG. 1.3. *Noise histograms for I (top) and $\kappa(I)$ (bottom). From left to right: $\sigma = 5, 15, 25$.*

Another important observation is the following. If we have the noisy image I and the clean curvature $\kappa(a)$, then we can recover the clean image a almost perfectly by looking for the steady state of this PDE:

$$u_t = \kappa(u) - \kappa(a), \tag{1.2}$$

with initial condition $u(t = 0) = I$. Notice that as u evolves, $\kappa(u)$ gets progressively closer to $\kappa(a)$ until they coincide when $u_t = 0$ and the steady state $\hat{u}$ is reached. This notion that all the relevant information on an image is contained in its curvature (so we can fully recover the former if having the latter) was introduced by Attneave [4], as Ciomaga et al. point out in a recent paper [5].

Figure 1.4 shows, on the left, the noisy image I, in the middle the result $\hat{u}$ obtained as the steady state of (1.2), and on the right the original clean image a. The images

$\hat{u}$ and a look very much alike, although there are slight numerical differences among them (the MSE between both images is 11.5).

FIG. 1.4. *Left: noisy image I. Middle: the result $\hat{u}$ obtained with* (1.2)*. Right: original clean image a.*

In this work we argue that the curvature of a noisy image, $\kappa(I) = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right)$ has a higher signal-to-noise ratio than both the unit normals of the noisy image, $\overrightarrow{n} = \frac{\nabla I}{|\nabla I|}$, and the noisy image itself, I. In fact, for a number of denoising algorithms, we demonstrate that if one uses the same algorithm for denoising the curvature image to obtain an approximation of $\kappa(a)$ and then solves (4.1), a better result is obtained than if the denoising algorithm was applied directly to the noisy image.

It is important to point out that what we propose here is *not* a PDE-based denoising method, but rather a general denoising framework. Specifically, instead of directly denoising I with some algorithm $\mathcal{F}$ to obtain a clean approximation $u = \mathcal{F}(I)$, one can do the following:

Step 1: Given a noisy image, I, denoise $\kappa(I)$ to obtain $\kappa_d = \mathcal{F}(\kappa(I))$.

Step 2: Construct a new image $\hat{u}$ whose curvature matches that of κ_d.

The resulting image $\hat{u}$ from *Step 2* will be a clean version of I, and one that we claim will generally have a higher SNR than u. To illustrate the broad applicability of this approach, in Section 3 we demonstrate that the regularizer $\mathcal{F}$ can come from vastly different schools for denoising, including variational methods as well as patch-based approaches.

2. Comparing the noise power in I and in its curvature image $\kappa(I)$. From (1.1) and basic calculus we get this formula for the curvature of I, $\kappa(I)$:

$$\kappa(I) = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right) = \kappa(a) \frac{|\nabla a|}{|\nabla I|} + \frac{\nabla a}{|\nabla a|} \cdot \nabla \left(\frac{|\nabla a|}{|\nabla I|} \right) + \nabla \cdot \left(\frac{\nabla n}{|\nabla I|} \right). \tag{2.1}$$

If

$$|\nabla a| \gg |\nabla n|, \tag{2.2}$$

then

$$\kappa(I) \simeq \kappa(a) + \nabla \cdot \left(\frac{\nabla n}{|\nabla I|} \right), \tag{2.3}$$

and we can consider the difference between the curvatures of the original and observed images in (2.3) as "curvature noise", n_κ:

$$n_\kappa = \nabla \cdot \left(\frac{\nabla n}{|\nabla I|} \right). \tag{2.4}$$

In this set-up, we now compute the Signal to Noise Ratio (SNR) of $\kappa(I)$. If we use directional differences to compute the curvature κ on images which are in the range $[0, 255]$, then $|\kappa| \leq 4$ and therefore the amplitude of the signal $\kappa(I)$ is 8.

Now we need to compute the variance of n_κ. Let vector $\vec{v}$ be the gradient of the noise:

$$\vec{v} = (v_1, v_2) = \nabla n, \tag{2.5}$$

so we can re-write n_κ as:

$$n_\kappa = \nabla \cdot (\frac{v_1}{|\nabla I|}, \frac{v_2}{|\nabla I|}) = (\frac{v_1}{|\nabla I|})_x + (\frac{v_2}{|\nabla I|})_y. \tag{2.6}$$

Expanding:

$$(\frac{v_1}{|\nabla I|})_x = \frac{(v_1)_x}{|\nabla I|} - v_1 \frac{|\nabla I|_x}{|\nabla I|^2}, \tag{2.7}$$

and:

$$(\frac{v_2}{|\nabla I|})_y = \frac{(v_2)_y}{|\nabla I|} - v_2 \frac{|\nabla I|_y}{|\nabla I|^2}. \tag{2.8}$$

Recall that:

$$|\nabla I|_x = \frac{I_x I_{xx} + I_y I_{yx}}{|\nabla I|}, \tag{2.9}$$

and

$$|\nabla I|_y = \frac{I_x I_{xy} + I_y I_{yy}}{|\nabla I|}. \tag{2.10}$$

Observe that (2.2) tells us that we are presently at a contour of a. Let us now assume, without loss of generality, that this contour is oriented in the direction of $\vec{y}$, i.e. it is vertical and therefore $I_y = 0$ and $|\nabla I| = |I_x|$. It also tells us that, for both v_1 and v_2:

$$|v_i| < |\nabla n| \ll |\nabla a| < |\nabla I|, i = 1, 2. \tag{2.11}$$

Taking all this into account and considering (2.7) and (2.8), now we have:

$$v_1 \frac{|\nabla I|_x}{|\nabla I|^2} = \frac{v_1}{|\nabla I|} \frac{I_{xx}}{I_x}, \tag{2.12}$$

and

$$v_2 \frac{|\nabla I|_y}{|\nabla I|^2} = \frac{v_2}{|\nabla I|} \frac{I_{xy}}{I_x}. \tag{2.13}$$

From (2.11), $\frac{v_i}{|\nabla I|} \simeq 0, i = 1, 2$ and therefore we should be able to disregard the second terms of the right-hand side of (2.7) and (2.8), *as long as* $\frac{I_{xx}}{I_x}$ *and* $\frac{I_{xy}}{I_x}$ *are not very large*. In practice, we can always ensure this.

To see this, we introduce the notation: $I(x-1, y+1) = A, I(x+1, y+1) = B, I(x-1, y-1) = C, I(x+1, y-1) = D, \alpha = B - A, \delta = D - C$. Then using centered differences, $I_{xy} = \frac{1}{4}(\alpha - \delta)$ and $I_x = \frac{1}{4}(\alpha + \delta)$.

Therefore:

$$\left|\frac{I_{xy}}{I_x}\right| = \left|\frac{\alpha-\delta}{\alpha+\delta}\right| = \left|1-\frac{2\delta}{\alpha+\delta}\right|, \tag{2.14}$$

which can only be large if $\alpha+\delta \simeq 0$. But this would imply that $I_x \simeq 0$, contradicting our hypothesis of (2.2). As a result:

$$\left(\frac{v_2}{|\nabla I|}\right)_y \simeq \frac{(v_2)_y}{|\nabla I|}. \tag{2.15}$$

The same reasoning can be applied to the horizontal direction. Now we use this notation: $I(x,y) = A, I(x+1,y) = B, I(x-1,y) = C$. Using centered differences, $I_{xx} = B + C - 2A$ and $I_x = \frac{B-C}{2}$.

Therefore:

$$\left|\frac{I_{xx}}{I_x}\right| = 2\left|1+\frac{2(C-A)}{B-C}\right|, \tag{2.16}$$

and the only way this can have a very large value is if $B - C \simeq 0$, but this would imply that $I_x \simeq 0$, which contradicts our hypothesis of (2.2). As a result:

$$\left(\frac{v_1}{|\nabla I|}\right)_x \simeq \frac{(v_1)_x}{|\nabla I|}. \tag{2.17}$$

From (2.17) and (2.15) we get:

$$n_\kappa \simeq \frac{1}{|\nabla I|}((v_1)_x + (v_2)_y) = \frac{1}{|\nabla I|}\nabla\cdot(v_1,v_2) = \frac{1}{|\nabla I|}\nabla\cdot\nabla n = \frac{1}{|\nabla I|}\Delta n. \tag{2.18}$$

Using directional differences:

$$n_\kappa \simeq \frac{1}{|\nabla I|}(n(x+1,y)+n(x-1,y)+n(x,y+1)+n(x,y-1)-4n(x,y)), \tag{2.19}$$

so the (numerical) variance of the curvature noise is:

$$Var(n_\kappa) \simeq \frac{1}{|\nabla I|^2}20Var(n) = \frac{20}{|\nabla I|^2}\sigma^2. \tag{2.20}$$

Now we can compute the SNR of $\kappa(I)$:

$$SNR(\kappa(I)) = 20log_{10}\left(\frac{8}{\sqrt{20}\frac{\sigma}{|\nabla I|}}\right) = 20log_{10}\left(1.79\frac{|\nabla I|}{\sigma}\right). \tag{2.21}$$

If we go to the original grayscale image I and compute *locally* its SNR, we get that its amplitude is approximately $|\nabla I|$ (because the local amplitude is the magnitude of the jump at the boundary, and using directional differences $|\nabla I|$ is the value of this jump) and the standard deviation of the noise is just σ, therefore:

$$SNR(I) = 20log_{10}\left(\frac{|\nabla I|}{\sigma}\right). \tag{2.22}$$

This would be saying that, along the contours of a, the curvature image $\kappa(I)$ is some 5dB *less noisy* than the image I. Note that if instead of using backward-forward differences to compute κ we had used central differences and the formula:

$$\kappa = \frac{I_x^2 I_{yy} + I_y^2 I_{xx} - 2 I_x I_y I_{xy}}{(I_x^2 + I_y^2)^{\frac{3}{2}}},$$

then the amplitude of κ would be much larger than 8 and hence the difference in SNR with respect to I would also be much larger. But we have preferred to consider the case of directional differences, because in practice the curvature is usually computed this way, for numerical stability reasons (see Ciomaga et al. [5] for alternate ways of estimating the curvature).

What happens if we want to denoise the normals, as Lysaker et al. [6] do? Let $\vec{w}$ be the normal vector:

$$\vec{w} = (w_1, w_2) = \frac{\nabla I}{|\nabla I|} = \frac{\nabla a}{|\nabla I|} + \frac{\nabla n}{|\nabla I|}. \tag{2.23}$$

Let's compute the SNR for any of the components of $\vec{w}$, say w_1. Its amplitude is 2, since $w_1 \in [-1, 1]$. Using similar arguments as before, we can approximate the variance of the "noise" in w_1 as:

$$Var(\frac{n_x}{|\nabla I|}) \simeq \frac{1}{|\nabla I|^2} Var(n_x), \tag{2.24}$$

and, using directional differences:

$$n_x(x, y) = n(x + 1, y) - n(x - 1, y), \tag{2.25}$$

so

$$Var\left(\frac{n_x}{|\nabla I|}\right) \simeq \frac{1}{|\nabla I|^2} 2Var(n) = \frac{1}{|\nabla I|^2} 2\sigma^2. \tag{2.26}$$

Therefore, the SNR of the first component of the normal field is:

$$SNR(w_1) = 20log_{10}\left(\frac{2}{\sqrt{2}\frac{\sigma}{|\nabla I|}}\right) = 20log_{10}\left(1.41\frac{|\nabla I|}{\sigma}\right). \tag{2.27}$$

From equations 2.21,2.22 and 2.27 we get $SNR(I) < SNR(w_i) < SNR(\kappa)$ and therefore we can conjecture that, given any denoising method, for best results *on the contours* it's better to denoise the curvature rather than directly denoise I (or the normal field).

As mentioned in section 1, from the *denoised* curvature κ_d we can obtain the final denoised image I_d by looking for the steady state of this PDE:

$$u_t = \kappa(u) - \kappa_d, \tag{2.28}$$

with initial condition $u(t = 0) = I$. Notice that as u evolves, $\kappa(u)$ gets progressively closer to κ_d (which is constant in time) until they coincide when $u_t = 0$ and the steady state is reached.

On homogeneous or slowly varying regions, though, (2.2) is no longer valid and we have instead:

$$|\nabla a| \ll |\nabla n|, \tag{2.29}$$

so now

$$\kappa(I) \simeq \kappa(n) + \nabla \cdot (\frac{\nabla a}{|\nabla I|}). \tag{2.30}$$

In this case then $\kappa(I)$ can't be expressed as the original curvature $\kappa(a)$ plus some *curvature noise*, unlike in (2.3). So in homogeneous regions $\kappa(I)$ is a poor estimation of $\kappa(a)$, but we can argue that this is not a crucial issue, with the following reasoning.

From eqs. 2.29 and 2.30 we see that $\kappa(I)$ behaves as $\kappa(n)$ plus a perturbation; since n is random noise, $\kappa(I)$ will also be random noise and its mean will be zero. Therefore, any simple denosing method applied to $\kappa(I)$ will result in curvature values of κ_d close to zero, so after running (2.28) the reconstructed (denoised) image I_d will have, in these homogeneous regions, curvature close to zero, which means that these regions will be approximated by planes (not necessarily horizontal). This is not a bad approximation given that these regions are, precisely, homogeneous or slowly varying. Another observation is that the plots in fig. 1.3 show that the SNR of $\kappa(I)$ is higher than the SNR of I, despite the fact that the contour pixels represent a small percentage of the image area. This means that the poor estimation of the curvature in the homogeneous regions (i.e. at the non-contour pixels) is still largely compensated by the high SNR at the contours.

3. Experiments. The observations in the previous sections have led us to perform the following experiments:

-run a denoising method on an image I obtaining a denoised image $\hat{I_d}$;

-run the same method on the curvature image $\kappa(I)$ obtaining a denoised curvature image κ_d, then iterate the equation:

$$u_t = \kappa(u) - \kappa_d + 2\lambda(I - u), \tag{3.1}$$

where λ is a positive parameter and I is the original noisy image, finally obtaining $I_d = \lim_{t\to\infty} u$.

We have used as image database the grayscale images (range $[0, 255]$) obtained by computing the luminance channel of the images in the Kodak [3] database (at half-size), and tested three denoising methods: TV denoising [7], Bregman iterations [8] and Non-local Means [9]. Our experiments show that in all three cases we obtain better results by denoising the curvature image $\kappa(I)$ rather than directly denoising the image I. Furthermore, the improvement in terms of PSNR is in the range predicted by Levin et al. [1, 2], which suggests that with our technique we could be approaching the optimality bounds estimated in those works.

3.1. TV denoising with ROF. We have compared with the Rudin-Osher-Fatemi (ROF) TV denoising method [7]:

$$u_t = \nabla(\frac{\nabla u}{|\nabla u|}) + 2\lambda(t)(I - u), \tag{3.2}$$

where $\lambda(t)$ is estimated at each iteration, knowing the value σ of the standard deviation of the noise. Initialization: $u(t = 0) = I$. The stopping criterion is: $MSE(I, u) \geq \sigma^2$. The obtained result is $I_{ROF} = \lim_{t\to\infty} u$.

We perform TV denoising of $\kappa(I)$:

$$\kappa_t = \nabla(\frac{\nabla \kappa}{|\nabla \kappa|}), \tag{3.3}$$

which we iterate for a fixed number of steps, obtaining κ_d. The parameter values are: time step $\Delta t = 0.025$, number of steps $T = 25$ for noise value $\sigma = 5$, $T = 15$ for noise values $\sigma = 10, 15, 20, 25$.

Then we iterate the equation:

$$u_t = \kappa(u) - \kappa_d + 2\lambda(t)(I - u), \tag{3.4}$$

where $\lambda(t)$ is estimated at each iteration, finally obtaining I_d as the steady state. Initialization: $u(t = 0) = I$. Time step: $\Delta t = 0.1$.

Defining $\epsilon^2(m)$ as the MSE difference with respect to the original noisy image at iteration step m,

$$\epsilon^2(m) = \frac{1}{|\Omega|} \sum_{x \in \Omega} (I(x) - u(x)(t = m))^2, \tag{3.5}$$

we can express the stopping criterions for (3.4) as follows: stop when

$$\epsilon^2(m) \geq \sigma^2 \quad \text{or} \quad |\epsilon(m + 1) - \epsilon(m)| \leq 0.0005, \tag{3.6}$$

whichever happens first.

Fig. 3.1 shows one example comparing the outputs of TV denoising of I and $\kappa(I)$.

FIG. 3.1. *Left: noisy image. Middle: result obtained with TV denoising of the image. Right: result obtained with TV denoising of the curvature image.*

Figure 3.2 compares, on the left, the average increase in PSNR (over the original noisy image) obtained with both approaches: PSNR(I_{ROF})-PSNR(I) (in magenta), PSNR(I_d)-PSNR(I) (in blue). As with all the plots in this article, the values have been averaged over all the images in the database. On the right we use another image quality measure, the Q-index of [10], which is reported as having higher perceptual correlation than PSNR and SNR-based metrics [11]; in this case we plot the average percentage increase in Q:

$$100 \times \frac{Q(I_{ROF}) - Q(I)}{Q(I)} \quad \text{and} \quad 100 \times \frac{Q(I_d) - Q(I)}{Q(I)}.$$

Both plots in fig. 3.2 show that TV denoising of the curvature allows us to obtain a denoised image I_d which is better in terms of PSNR and Q-index than I_{ROF}, the image obtained by directly applying TV denoising to the original noisy image.

3.2. Bregman iterations. Lysaker et.al. [6] proposed a two step denoising algorithm in which they first approximate a smooth normal field, $\overrightarrow{n_1}$, to the noisy image, I, using

$$\overrightarrow{n_1} = \arg\min_{|\overrightarrow{n}|=1} \int |\nabla \overrightarrow{n}| + \lambda \int \left(\frac{\nabla I}{|\nabla I|} - \overrightarrow{n} \right)^2 \tag{3.7}$$

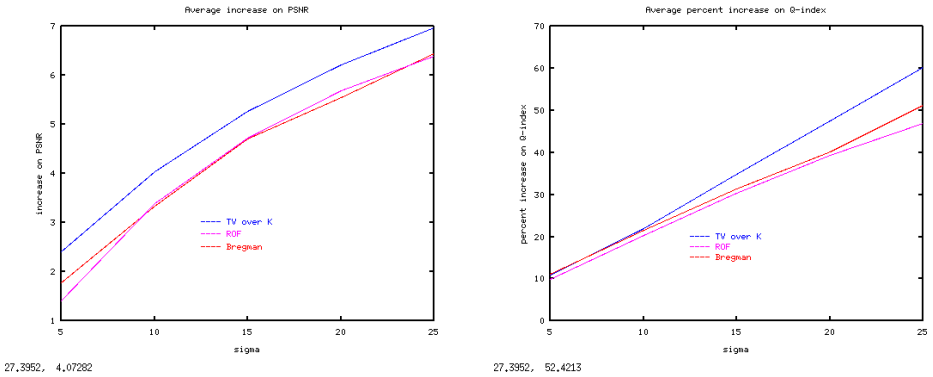


FIG. 3.2. *Comparison of TV denoising on I, TV denoising on $\kappa(I)$ and Bregman iterations. Left: PSNR increase for each method. Right: percentage increase on Q-index [10]. Values averaged over Kodak database (only luminance channel, images reduced to half-size).*

and then obtain the denoised image via the minimization problem

$$u_2 = \underset{u \in BV(\Omega)}{\arg\min} \int (|\nabla u| - \overrightarrow{n_1} \cdot \nabla u) + \lambda \int (I - u)^2. \tag{3.8}$$

This idea motived several other related works. The authors in [12] proposed a similar algorithm, but in step 1 they solve for a divergence-free, noise-free approximate unit tangent field, $\overrightarrow{t} = (u, v)$ (a more mathematically sound minimization problem than (3.7)), use this to compute $\overrightarrow{n}_1 = (-v, u)$, and then solve for the clean image using (3.8). Other works have built on this model. For example, the authors in [13] suggest replacing (3.8) with a more direct feature orientation-matching functional.

The Bregman iterations of Osher et.al. [8] was also motivated by [6], and has made a particular impact on the field of variational based image processing. The idea is to replace (3.7) with a step that first generates a smooth approximation of the image, u_1, then compute an approximate unit normal, $\overrightarrow{n}_1 = \frac{\nabla u_1}{|\nabla u_1|}$, and finally solve (3.8). The authors discovered this is equivalent to solving the Rudin-Osher-Fatemi (ROF) functional [7], adding the residual noise back to the original noisy image, and then solving ROF again. They also discovered that even better results could be obtained by starting with an image of all zeros and iteratively repeating this process until the solution is within a distance of σ from the noisy image.

We wanted to compare also with this method, because of its relationship with the denoising of normalized-gradient fields. We showed in section 2 that, although better than direct denoising of I, denoising of normalized-gradient fields would not perform as well as the denoising of $\kappa(I)$, at least on the image contours. Comparisons in term of PSNR and Q-index can be seen in figure 3.2. This figure shows that Bregman iterations fare better than ROF in terms of Q-index, although not in PSNR, and that TV denoising of κ outperforms both Bregman iterations and ROF, as predicted, and it does so both in terms of PSNR and Q-index.

The implementation details are as follows. We have compared with the original Bregman iterations method of [8]; the values used for $\lambda : 0.033, 0.013, 0.009, 0.005, 0.00425$, corresponding to $\sigma : 5, 10, 15, 20, 25$ respectively, have been chosen following the suggestions given in [8] in order to obtain optimum results. The time step is $\Delta t = 0.1$.

3.3. Non-Local Means. To illustrate a comparison with patch-based methods, we incorporated Non-Local Means denoising [9] into our general framework as follows. First we performed Non-local Means denoising on the original noisy image I_n using the code from [14] (with their choice of parameters) obtaining the denoised image I_{IPOL}.

For our method, we have done the following:

- Apply NLM to $\kappa(I)$, but with these two modifications:
  1. Compute the weights from I instead of κ (i.e. compare image patches, not curvature patches).
  2. Use $\sigma_\kappa = \sigma + 5$ as the standard deviation.

  We obtain the denoised curvature κ_d.
- Starting with $u(t=0) = I_{IPOL}$, solve for the steady state of:

$$u_t = \kappa(u) - \kappa_d + 2\lambda(I - u),$$

  with the stopping criterion described in (3.6).

The values used for $\lambda : 0.2, 0.075, 0.05, 0.04, 0.03$, correspond to $\sigma : 5, 10, 15, 20, 25$ respectively.

Fig. 3.3 shows one example comparing the outputs of NLM denoising of I and $\kappa(I)$.

FIG. 3.3. *Left: noisy image. Middle: result obtained with NLM denoising of the image. Right: result obtained with NLM denoising of the curvature image.*

Figure 3.4 (left) compares the average increase in PSNR of the denoised image over the original noisy image, obtained with both approaches: NLM on I (in magenta), NLM on K (in blue).

If the starting condition were $u(t=0) = I$ as usual then denoising the curvature performs worse, in terms of PSNR, than denoising the image, see fig. 3.5. This is why we are starting with $u(t=0) = I_{IPOL}$, but now one could argue that what we are doing is basically TV denoising on the output of NLM; in fact, if we over-process κ we obtain $\kappa_d \cong 0$ and in that case we would actually be doing ROF denoising on I_{IPOL}. But that is not the case: if we apply ROF on I_{IPOL} as explained in section 3.1 (with variable $\lambda(t)$ and the stopping criteria mentioned there), the outputs have lower PSNR, see fig. 3.4 (left). Figure 3.4 (right) compares the average percent increase in Q-index of the denoised image ogver the original noisy image, obtained with both approaches.

Both plots in fig. 3.4 show that NLM denoising of the curvature allows us to obtain a denoised image I_d which is better in terms of PSNR and Q-index than the image I_{IPOL}, obtained directly by applying NLM tos the original noisy image.

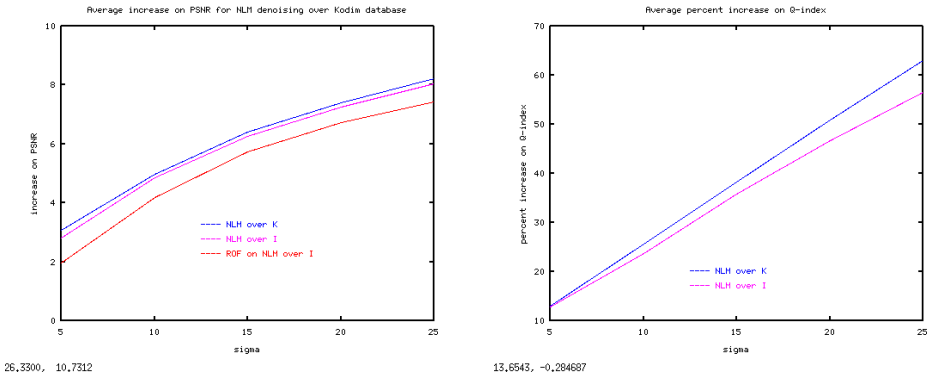


FIG. 3.4. *Comparison of NLM denoising on I and NLM denoising on $\kappa(I)$. Left: PSNR increase for each method; also pictured: PSNR increase for ROF on the output of NLM on I. Right: percentage increase on Q-index [10]. Values averaged over Kodak database (only luminance channel, images reduced to half-size).*

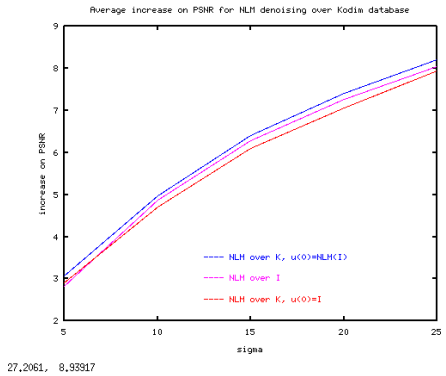


FIG. 3.5. *PSNR comparison of NLM denoising on I and NLM denoising on $\kappa(I)$ with different starting conditions.*

4. Discussion.

4.1. Computing the curvature. Kovalevsky shows in [15] that it is difficult to compute the curvature with errors smaller than 40% without subpixel accuracy and numerical optimization, even in high resolution images. The reason is that small errors require very long curves. Utcke [16] points out that the smaller the curvature, the larger the error in estimating it. Ciomaga et al. [5] propose a method to increase the accuracy in estimating a curvature image by decomposing the image in its level lines and computing the curvature at each of these curves with subpixel accuracy.

All the tests in this article have been performed using very simple numerical schemes for the computation of the curvature hence the error must be very significant, but this does not seem to affect the final result dramatically as fig. 1.4 shows. We would like to test other computational techniques for the curvature, and their impact in the quality of the results.

4.2. The reconstruction equation. The equation that we use to reconstruct a clean image from a denoised curvature image, eq. (1.2), is the flow associated with

the minimization problem

$$\min_{u \in BV(\Omega)} \int |\nabla u| + \kappa(a) u. \tag{4.1}$$

The functional in (4.1) was used in Ballester et.al. [17] for the purpose of image inpainting, and in particular, for propagating the level lines of the known parts of an image into the inpainting region. The functional they use is

$$F(u) = \int |\nabla u| - \theta \cdot \nabla u \tag{4.2}$$

where θ is a gradient field that determines the direction of the level lines. Intuitively, when considering the denoising problem, if one starts with a noisy image for which the noise has mean zero and propagates the level lines of the clean image (ideally using $\theta = \frac{\nabla a}{|\nabla a|}$) while smoothing with a total variation based regularizer, one would expect a relatively accurate reconstruction of the original clean image, a. Existence and uniqueness results, as well as fast convergence, were formally proven in Andreu et al. [18] for a very similar equation, one in which the term κ_d is constant or time-varying. If it is also spatially varying, as it is our case, then convergence is achieved as time goes to infinity. In practice we have found that convergence with eq. (3.1) takes three times as many iterations as convergence with eq. (3.2).

We are not aware of any other method to recover an image from its curvature image, and it would be interesting to study whether the accuracy of the reconstruction could be improved by using another technique.

4.3. Real curvature images. After we apply a given denoising method $\mathcal{F}$ to the curvature image $\kappa(I)$ we obtain an image $\kappa_d = \mathcal{F}((\kappa(I))$ which we call (and treat as) "denoised curvature". But we can't formally say that κ_d is actually a *curvature image*, i.e. there may not exist an image I_d such that $\kappa_d = \nabla \cdot \left(\frac{\nabla I_d}{|\nabla I_d|}\right)$. Although this does not seem to hinder the approach from improving on denoising methods in general, we are currently exploring more precisely what effect this has on our solution.

5. Conclusions and future work. In this article we have shown that when an image is corrupted by additive noise, its curvature image is less affected. This has led us to conjecture that, given a denoising method, we may obtain better results applying it to the curvature image and then reconstructing a clean image from it, rather than denoising the original image directly. Numerical experiments confirm this for several PDE-based and patch-based denoising algorithms. The improvements in the quality of the results bring us closer to the optimal bounds recently derived by Levin et al. [1, 2]. Many open questions remain, concerning the accuracy in the computation of the curvature, the reconstruction method used and the nature of the denoised curvature image, which will be the subject of further work.

Acknowledgments. We would like to thank the following researchers for their very helpful comments and suggestions: Vicent Caselles, Stanley Osher, Guillermo Sapiro, Jesús Ildefonso Díaz, Ron Kimmel, Jean-Michel Morel and Alfred Bruckstein. We also want to acknowledge the Institute for Mathematics and its Applications (IMA) at Minneapolis, where the authors were visitors in 2011. The first author acknowledges partial support by AACC project, reference TIN2011-15954-E. The second author was supported in part by NSF-DMS #0915219.

REFERENCES

[1] A. 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.

[2] A. Levin, B. Nadler, F. Durand, and W. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," MIT - Computer Science and Artificial Intelligence Laboratory, Tech. Rep., 2012.

[3] Kodak, "http://r0k.us/graphics/kodak/."

[4] F. Attneave, "Some informational aspects of visual perception." *Psychological review*, vol. 61, no. 3, p. 183, 1954.

[5] A. Ciomaga, P. Monasse, and J. Morel, "Level lines shortening yields an image curvature microscope," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 4129–4132.

[6] M. Lysaker, S. Osher, and X. Tai, "Noise removal using smoothed normals and surface fitting," *Image Processing, IEEE Transactions on*, vol. 13, no. 10, pp. 1345–1357, 2004.

[7] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[8] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling and Simulation*, vol. 4, no. 2, p. 460, 2005.

[9] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. Ieee, 2005, pp. 60–65.

[10] Z. Wang and A. Bovik, "A universal image quality index," *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, 2002.

[11] M. Pedersen and J. Hardeberg, "Full-reference image quality metrics: Classification and evaluation," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 1, pp. 1–80, 2011.

[12] T. Rahman, X.-C. Tai, and S. Osher, "A tv-stokes denoising algorithm," in *SSVM*, ser. Lecture Notes in Computer Science, F. Sgallari, A. Murli, and N. Paragios, Eds., vol. 4485. Springer, 2007, pp. 473–483.

[13] J. Hahn, X.-C. Tai, S. Borok, and A. M. Bruckstein, "Orientation-matching minimization for image denoising and inpainting," *Int. J. Comput. Vis.*, vol. 92, no. 3, pp. 308–324, 2011. [Online]. Available: http://dx.doi.org/10.1007/s11263-010-0371-5

[14] J.-M. M. Antoni Buades, Bartomeu Coll, "Non-local Means Denoising," *Image Processing On Line*, 2011.

[15] V. Kovalevsky, "Curvature in digital 2d images," *IJPRAI*, vol. 15, no. 7, pp. 1183–1200, 2001.

[16] S. Utcke, "Error-bounds on curvature estimation," in *Scale Space Methods in Computer Vision*. Springer, 2003, pp. 657–666.

[17] C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, 2001.

[18] F. Andreu, V. Caselles, J. Dıaz, and J. Mazón, "Some qualitative properties for the total variation flow," *Journal of Functional Analysis*, vol. 188, no. 2, pp. 516–547, 2002.