

DENOISING AN IMAGE BY DENOISING ITS CURVATURE IMAGE

MARCELO BERTALMIÓ AND STACEY LEVINE

Abstract. In this article we argue 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 speculate 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.

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 ,

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

We will denote the curvature image of I by $\kappa(I) = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right)$. For each pixel x , $\kappa(I)(x)$ is the value of the curvature of the level line of I passing through x . Figure 1.1(a) shows the standard lena image a and figure 1.1(b) its corresponding curvature image $\kappa(a)$; in figures 1.1(c) and (d) we see I and $\kappa(I)$, where I has been obtained by adding Gaussian noise of $\sigma = 25$ to a . Notice that it's nearly impossible 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 in figure 1.2 in order to highlight the differences, and now some structures of the grayscale images become apparent, such as the outline of her face and the textures in her hat. 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 $\kappa(I)$.

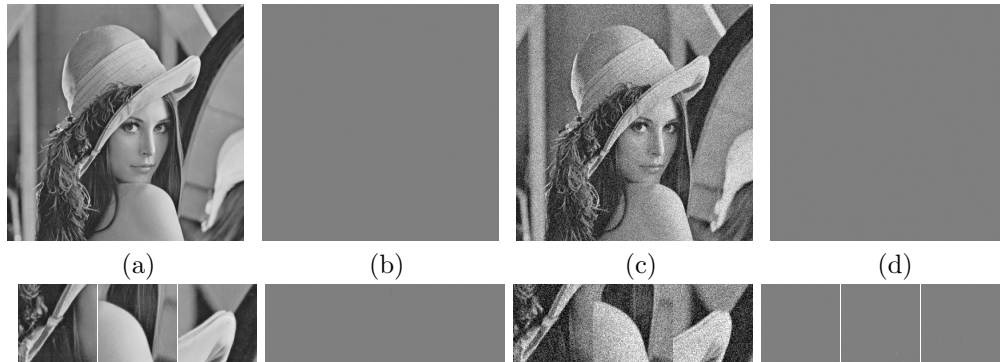


FIG. 1.1. (a), (b): image and its curvature. (c), (d): after adding noise.

This last observation is corroborated in figure 1.3 which shows, for Gaussian noise and different values of σ , 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

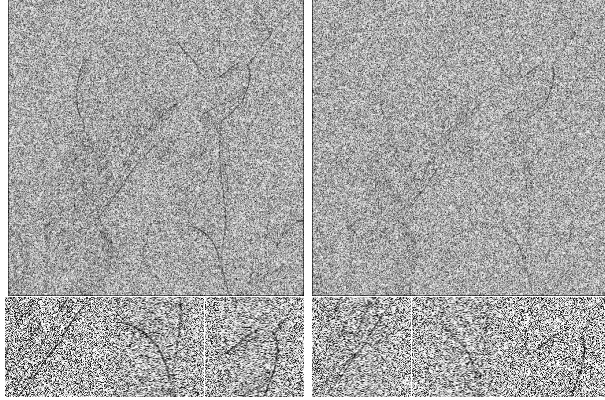


FIG. 1.2. Close-ups of the clean curvature (left) and noisy curvature (right) from figures 1.1(b) and 1.1(d) respectively, with non-linear scaling to highlight the differences.

a distribution resembling the Laplace distribution, with standard deviation smaller than σ . Consistently, in terms of Peak Signal to Noise Ratio (PSNR) the curvature image is better (higher PSNR, less noisy) than I , as is noted in the figure plots.

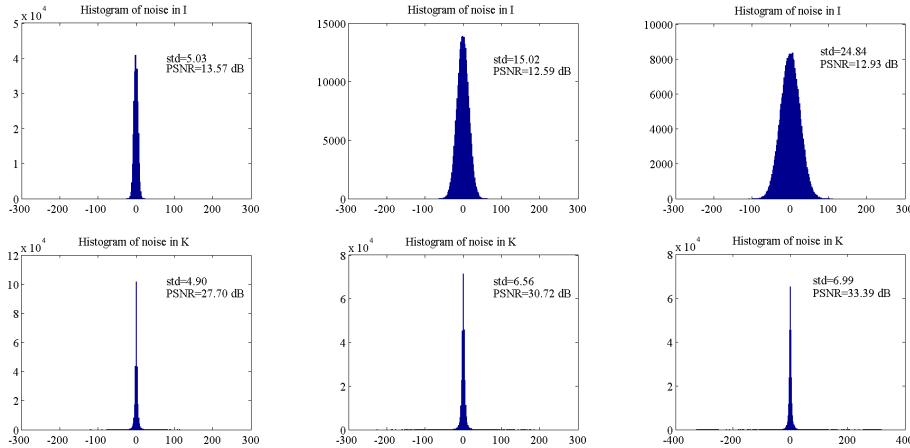


FIG. 1.3. Noise histograms for I (top) and $\kappa(I)$ (bottom). From left to right: $\sigma = 5, 15, 25$.

Another important observation is the following. All geometric information of an image is contained in its curvature, so we can fully recover the former if having the latter, up to a change in contrast. This notion was introduced as early as 1954 by Attneave [1], as Ciomaga et al. point out in a recent paper [2]. Thus, if we have the clean curvature $\kappa(a)$ and the given noisy data I (which should have the same average contrast along level lines as a), then we should be able to recover the clean image a almost perfectly. One such approach for doing this could be to solve for the steady state of

$$u_t = \kappa(u) - \kappa(a) + \lambda(I - u), \quad u(0, \cdot) = I \quad (1.2)$$

where $\lambda > 0$ is a Lagrange multiplier that depends on the noise level. As $t \rightarrow \infty$, one can expect that $u(t, \cdot)$ reaches a steady state \hat{u} (this is discussed further in section 3).

In this case, $\kappa(\hat{u})$ should be close to $\kappa(a)$ and the average value of \hat{u} (along each level line) stays close to that of a . In section 3 we discuss related models, and in section 4.6 we suggest an alternative to equation (1.2).

Figure 1.4 shows, on the left, the noisy image I , in the middle the result u , the solution of (1.2) using the stopping criteria described later in (3.4)-(3.5), and on the right the original clean image a . The images u and a look very much alike, although there are slight numerical differences among them (the Mean Squared Error, MSE, between both images is 3.7).



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

In addition to the above observations, in [3] the authors proposed a variational approach for fusing a set of exposure bracketed images (a set of images of the same scene taken in rapid succession with different exposure times) that had a related, and initially somewhat perplexing, denoising effect. The energy functional fuses the colors of a long exposure image, I_{LE} , with the details from a short exposure image, I_{SE} , while attenuating noise from the latter. The denoising effect is surprisingly similar to that produced by state-of-the-art techniques directly applied to I_{SE} , such as Non-Local Means [4]. The term in the energy functional that generates this effect is $\int \left(|\nabla u| - \nabla u \cdot \frac{\nabla I_{SE}}{|\nabla I_{SE}|} \right)$ which was initially intended to preserve the details (i.e. gradient direction) of I_{SE} . The flow of the corresponding Euler-Lagrange equation for this term, $u_t = \kappa(u) - \kappa_\epsilon(I_{SE})$, is very similar to (1.2). Here $\kappa_\epsilon(I_{SE})$ is the curvature of I_{SE} which is computed using a small positive constant ϵ to avoid division by zero, and hence we could say that it has a regularizing effect on the actual curvature $\kappa(I_{SE})$; we found that as ϵ increases the final output of the fusion process becomes less noisy, therefore $\kappa_\epsilon(I_{SE})$ appears to be playing the role of the curvature of the clean image.

Motivated by the preceding observations, we propose the following general denoising framework. Given a noisy image $I = a + n$, instead of directly denoising I with some algorithm \mathcal{F} to obtain a denoised image $I_{\mathcal{F}} = \mathcal{F}(I)$, do the following:

- Denoise the curvature image $\kappa(I)$ with method \mathcal{F} to obtain $\kappa_{\mathcal{F}} = \mathcal{F}(\kappa(I))$.
- Generate an image $\hat{I}_{\mathcal{F}}$ that satisfies the following criteria:
 1. $\kappa(\hat{I}_{\mathcal{F}}) \simeq \kappa_{\mathcal{F}}$; that is, the level lines of $\hat{I}_{\mathcal{F}}$ are well described by $\kappa_{\mathcal{F}}$.
 2. The overall contrast of $\hat{I}_{\mathcal{F}}$ matches that of the given data $I = a + n$ in the sense that the intensity of any given level line of $\hat{I}_{\mathcal{F}}$ is close to the average value of I along that contour.

The resulting image $\hat{I}_{\mathcal{F}}$ described above will be a clean version of I , and one that we claim will generally have a higher PSNR and Q-index [5] than $I_{\mathcal{F}}$. It is important to point out that what we propose here is *not* necessarily a PDE-based denoising method, but rather a general denoising framework.

This approach is closely related to the body of work inspired by Lysaker et. al. [6] in which the authors proposed a two step denoising process. In the first step, instead of smoothing a noisy image, they smooth its unit normals. They use the Rudin-Osher-Fatemi functional [7] for this smoothing process, but in a sense this could be thought of more generally as computing $\mathcal{F}(\vec{\eta}(I))$ where $\vec{\eta}(I)$ is the unit normal vector field of I and \mathcal{F} is the approach from [7]. In the second step, they compute a new (denoised) image whose unit normals match those found in the first step. This approach led to the work of Osher et.al. [8] which was motivated by using a different mechanism for denoising the unit normals. They smooth the noisy image I using the Rudin-Osher-Fatemi functional [7] and then compute the unit normals before doing the matching. In a sense, their first step was to compute $\vec{\eta}(\mathcal{F}(I))$ and then match unit normals. This led to the interesting (and now much studied) Bregman iterative approach for image processing. We discuss these works in more detail as well as their connection with our proposed approach in the subsequent sections. This also begs the question of whether we should be computing $\kappa_{\mathcal{F}}$ using $\kappa(\mathcal{F}(I))$ instead of $\mathcal{F}(\kappa(I))$. There is ample motivation for doing the former, but we found that in practice this leads to over smoothing. We discuss this further in section 4.5.

The organization of the paper is as follows. In section 2 we argue that along contours the curvature of a noisy image, $\kappa(I) = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right)$, generally has a higher PSNR than both the unit normal field of the noisy image, $\vec{\eta}(I) = \frac{\nabla I}{|\nabla I|}$, and the noisy image itself, I . Section 3 formally proposes our framework and discusses its relationship with previous work. To illustrate the broad applicability of our approach, in section 4 we provide experimental results demonstrating that the regularizer \mathcal{F} can come from vastly different schools for denoising, including variational methods as well as patch-based approaches. We also consider several different approaches for reconstructing the image from $\kappa_{\mathcal{F}}$. Our experiments corroborate the hypothesis that if one uses the same approach for denoising the curvature image to obtain an approximation $\kappa_{\mathcal{F}}$ of $\kappa(a)$ and then solves for a function whose curvature is approximated by $\kappa_{\mathcal{F}}$ and whose average value along level lines matches that of a , a better result is obtained than if the denoising algorithm was applied directly to the noisy image. In sections 5 and 6 we discuss some open questions and future work.

2. Comparing the noise power in I and in its curvature image $\kappa(I)$.

2.1. PSNR along image contours. From (1.1) and basic calculus, the curvature of I can be written

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

First we consider the situation where

$$|\nabla a| \gg |\nabla n|, \quad (2.2)$$

which is likely the case at image contours. At an edge, where (2.2) holds, we have that $\frac{|\nabla a|}{|\nabla I|} \simeq 1$ and so the first term of the right-hand side of equation (2.1) can be approximated by $\kappa(a)$, the second term $\frac{\nabla a}{|\nabla a|} \cdot \nabla \left(\frac{|\nabla a|}{|\nabla I|} \right) \simeq 0$ so it can be discarded

(except in the case where the image contour separates perfectly flat regions, a scenario we discuss in section 2.2), and finally the third term $\nabla \cdot \left(\frac{\nabla n}{|\nabla I|} \right)$ remains unchanged and is the main source of noise in the curvature image. So for now we approximate

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

and consider the difference between the curvatures of the original and observed images in (2.3) as “curvature noise”

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

In what follows, we approximate the curvature $\kappa(I)$ and unit normal field $\eta(I) = (\eta^1, \eta^2)$ of the image I using forward-backward differences, so

$$\kappa(I(x, y)) \simeq \Delta_-^x \left(\frac{\Delta_+^x I(x, y)}{|\nabla I(x, y)|} \right) + \Delta_-^y \left(\frac{\Delta_+^y I(x, y)}{|\nabla I(x, y)|} \right) \quad (2.5)$$

and

$$\vec{\eta}(I(x, y)) = (\eta^1(x, y), \eta^2(x, y)) \simeq \left(\frac{\Delta_+^x I(x, y)}{|\nabla I(x, y)|}, \frac{\Delta_+^y I(x, y)}{|\nabla I(x, y)|} \right), \quad (2.6)$$

where

$$\Delta_\pm^x I(x, y) = \pm (I(x \pm 1, y) - I(x, y)), \quad \Delta_\pm^y I(x, y) = \pm (I(x, y \pm 1) - I(x, y))$$

and where the discrete gradient is implied we use forward differences, so

$$|\nabla I(x, y)| = \sqrt{(\Delta_+^x I(x, y))^2 + (\Delta_+^y I(x, y))^2 + \epsilon^2}$$

for a small $\epsilon > 0$. In this setting, we have the following.

PROPOSITION 2.1. *At locations in the image domain where $I = a + n$ satisfies (2.2) and (2.3) (likely the case at contours of I), and where the noise standard deviation satisfies $\sigma > \frac{|\nabla I|}{10.32}$, if the curvature $\kappa(I)$ is approximated by (2.5), then*

$$PSNR(I) < PSNR(\kappa).$$

Furthermore, if the unit normal field $\eta(I) = (\eta^1, \eta^2)$ is approximated by (2.6) and $\sigma > \frac{|\nabla I|}{3.64}$, then for $i = 1, 2$ we also have

$$PSNR(I) < PSNR(\eta^i) < PSNR(\kappa).$$

Proof. First we approximate the Peak Signal to Noise Ratio (PSNR) of $\kappa(I)$. Assuming I lies in the range $[0, 255]$ and that $\kappa(I)$ is computed using directional differences as described in (2.5), we have that $|\kappa| \leq 2 + \sqrt{2}$ and therefore the amplitude of the signal $\kappa(I)$ is $4 + 2\sqrt{2}$.

To compute $Var(n_\kappa)$, first observe that

$$n_\kappa = \nabla \cdot \left(\frac{n_x}{|\nabla I|}, \frac{n_y}{|\nabla I|} \right) = \left(\frac{n_x}{|\nabla I|} \right)_x + \left(\frac{n_y}{|\nabla I|} \right)_y. \quad (2.7)$$

Using forward-backward differences as described in (2.5) we have that

$$\begin{aligned}
\left(\frac{n_x}{|\nabla I|}\right)_x &\simeq \Delta_-^x \left(\frac{\Delta_+^x(n(x, y))}{|\nabla I(x, y)|}\right) \\
&= \frac{\Delta_+^x(n(x, y))}{|\nabla I(x, y)|} - \frac{\Delta_+^x(n(x-1, y))}{|\nabla I(x-1, y)|} \\
&= \frac{\Delta_+^x(n(x, y))|\nabla I(x-1, y)| - \Delta_+^x(n(x-1, y))|\nabla I(x, y)|}{|\nabla I(x, y)||\nabla I(x-1, y)|} \\
&= \frac{\Delta_-^x(\Delta_+^x(n(x, y)))}{|\nabla I(x, y)|} - \frac{\Delta_+^x(n(x-1, y))\Delta_-^x|\nabla I(x, y)|}{|\nabla I(x, y)||\nabla I(x-1, y)|}
\end{aligned} \tag{2.8}$$

Without loss of generality, assume the edge is vertical, so $I_y \simeq 0$. If the edge discontinuity occurs between x and $x+1$, then

$$|\nabla I(x-1, y)| \simeq |\Delta_+^x I(x-1, y)| \simeq |\Delta_+^x n(x-1, y)|$$

and thus

$$\Delta_-^x |\nabla I(x, y)| = |\nabla I(x, y)| - |\nabla I(x-1, y)| \simeq |\nabla I(x, y)| - |\Delta_+^x n(x-1, y)|.$$

From the above calculations, the second term on the right hand side of (2.8) satisfies

$$\frac{\Delta_+^x(n(x-1, y))\Delta_-^x|\nabla I(x, y)|}{|\nabla I(x, y)||\nabla I(x-1, y)|} \simeq \pm \frac{|\nabla I(x, y)| - |\Delta_+^x n(x-1, y)|}{|\nabla I(x, y)|} \tag{2.9}$$

which is bounded above by 1 due to (2.2). Since an upper bound is sufficient for our argument, by (2.8) and (2.9) we can approximate

$$\left(\frac{n_x}{|\nabla I|}\right)_x \simeq \frac{\Delta_-^x(\Delta_+^x(n(x, y)))}{|\nabla I(x, y)|} + T_x, \quad \text{where } T_x \in [0, 1]. \tag{2.10}$$

Similar to (2.8),

$$\left(\frac{n_y}{|\nabla I|}\right)_y \simeq \frac{\Delta_-^y(\Delta_+^y(n(x, y)))}{|\nabla I(x, y)|} - \frac{\Delta_+^y(n(x, y-1))\Delta_-^y|\nabla I(x, y)|}{|\nabla I(x, y)||\nabla I(x, y-1)|}.$$

At a vertical edge we would expect that $|\nabla I(x, y)| \simeq |\nabla I(x, y-1)| \gg \Delta_+^y(n(x, y-1))$ and $\Delta_-^y|\nabla I(x, y)| \simeq 0$. Therefore

$$\left(\frac{n_y}{|\nabla I|}\right)_y \simeq \frac{\Delta_-^y(\Delta_+^y(n(x, y)))}{|\nabla I(x, y)|}. \tag{2.11}$$

By (2.7), (2.10), and (2.11) we have that

$$\begin{aligned}
n_\kappa &\simeq \frac{\Delta_-^x(\Delta_+^x(n(x, y)))}{|\nabla I(x, y)|} + \frac{\Delta_-^y(\Delta_+^y(n(x, y)))}{|\nabla I(x, y)|} + T_x \\
&= \frac{1}{|\nabla I|}(n(x+1, y) + n(x-1, y) + n(x, y+1) + n(x, y-1) - 4n(x, y)) + T_x.
\end{aligned} \tag{2.12}$$

Assuming $n \sim \mathcal{N}(0, \sigma^2)$, the (numerical) variance of n_κ is then

$$\begin{aligned}
&Var(n_\kappa) \\
&\simeq \frac{Var(n(x+1, y) + n(x-1, y) + n(x, y+1) + n(x, y-1)) + 16Var(n(x, y))}{|\nabla I|^2} + Var(T_x) \\
&= \frac{1}{|\nabla I|^2}(4Var(n) + 16Var(n)) + Var(T_x) = \frac{1}{|\nabla I|^2}20Var(n) + Var(T_x) = \frac{20}{|\nabla I|^2}\sigma^2 + Var(T_x).
\end{aligned}$$

Therefore, we typically have that

$$\text{Var}(n_\kappa) \simeq \frac{20}{|\nabla I|^2} \sigma^2 + \text{Var}(T_x) \quad \text{where} \quad \text{Var}(T_x) \in [0, 0.25]. \quad (2.13)$$

Now we can compute the PSNR of $\kappa(I)$, as the peak amplitude of the curvature signal is $4 + 2\sqrt{2}$ and the variance of the noise is given by (2.13), so

$$\text{PSNR}(\kappa(I)) \simeq 20 \log_{10} \left(\frac{4 + 2\sqrt{2}}{\sqrt{\frac{20\sigma^2}{|\nabla I|^2} + \text{Var}(T_x)}} \right). \quad (2.14)$$

Since $\text{Var}(T_x) \in [0, 0.25]$, at locations where $\sigma > \frac{|\nabla I|}{10.32}$ we have that

$$\text{PSNR}(\kappa(I)) \in \left(20 \log_{10} \left(\frac{|\nabla I|}{\sigma} \right), 20 \log_{10} \left(1.53 \frac{|\nabla I|}{\sigma} \right) \right] \quad (2.15)$$

If we go to the original grayscale image I and compute *locally* its PSNR, we get that the 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

$$\text{PSNR}(I) = 20 \log_{10} \left(\frac{|\nabla I|}{\sigma} \right). \quad (2.16)$$

This would be saying that, along the contours of a , the curvature image $\kappa(I)$ will be up to 3.7dB *less noisy* than the image I .

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

$$\vec{\eta} = (\eta^1, \eta^2) = \frac{\nabla I}{|\nabla I|} = \frac{\nabla a}{|\nabla I|} + \frac{\nabla n}{|\nabla I|}. \quad (2.17)$$

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

$$\text{Var}\left(\frac{n_x}{|\nabla I|}\right) \simeq \frac{1}{|\nabla I|^2} \text{Var}(n_x), \quad (2.18)$$

and, using directional differences

$$n_x(x, y) = n(x+1, y) - n(x, y), \quad (2.19)$$

so

$$\text{Var}\left(\frac{n_x}{|\nabla I|}\right) \simeq \frac{1}{|\nabla I|^2} 2 \text{Var}(n) = \frac{1}{|\nabla I|^2} 2\sigma^2. \quad (2.20)$$

Therefore, the PSNR of the first component of the normal field is

$$\text{PSNR}(\eta^1) = 20 \log_{10} \left(\frac{2}{\sqrt{2} \frac{\sigma}{|\nabla I|}} \right) = 20 \log_{10} \left(1.41 \frac{|\nabla I|}{\sigma} \right). \quad (2.21)$$

If $\sigma > \frac{|\nabla I|}{3.64}$ then

$$PSNR(\kappa(I)) \in \left(20 \log_{10} \left(\frac{1.41 |\nabla I|}{\sigma} \right), 20 \log_{10} \left(1.53 \frac{|\nabla I|}{\sigma} \right) \right] \quad (2.22)$$

From (2.16), (2.21) and (2.22) we get $PSNR(I) < PSNR(\eta^i) < PSNR(\kappa)$. \square

REMARK 2.2. *The restrictions on σ in Proposition 2.1 are fairly conservative given the experimental results that follow in section 2.3 and section 4 (e.g. see figures 2.1 and 2.2). But the overall conclusion is still the same, so we included these hypotheses for ease of argument.*

REMARK 2.3. *Note that if instead of using forward-backward 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 $4 + 2\sqrt{2}$ and hence the difference in PSNR 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. [2] for alternate ways of estimating the curvature).

The above conclusions suggest that, given any denoising method, for best results on the contours it may be better to denoise the curvature rather than directly denoise I (or the normal field).

2.2. Correction for contours separating flat regions. As we mentioned earlier, if we have an image contour that separates perfectly flat regions then the second term of the right-hand side of equation (2.1) cannot be discarded. The reason is that while $\frac{|\nabla a|}{|\nabla I|} \simeq 1$ holds on the contour, we also have $\frac{|\nabla a|}{|\nabla I|} \simeq 0$ on its sides because these regions are flat (and hence $|\nabla a| \simeq 0$). Consequently, we can no longer approximate the term $\frac{\nabla a}{|\nabla a|} \cdot \nabla \left(\frac{|\nabla a|}{|\nabla I|} \right)$ by zero, but we can bound its variance.

Consider a 100×100 square image a with value 0 for all pixels in the columns 0–49 and value 255 for all pixels in the columns 50–99. Image a then has a vertical edge that separates flat regions. Using backward differences, the term $\frac{|\nabla a|}{|\nabla I|} \simeq 0$ everywhere except at column 50, where $\frac{|\nabla a|}{|\nabla I|} \simeq 1$. Therefore, the term $\nabla \left(\frac{|\nabla a|}{|\nabla I|} \right)$ is close to $(0, 0)$ everywhere except at column 50, where it is $(1, 0)$, and column 51, where it is $(-1, 0)$ (always approximately). So the second term of the right-hand side of equation (2.1), $\frac{\nabla a}{|\nabla a|} \cdot \nabla \left(\frac{|\nabla a|}{|\nabla I|} \right)$, is close to zero everywhere except at column 50, where it is close to 1. Exactly the same result holds if the image a is flipped and takes the value 255 on the left and 0 on the right, because now $\nabla \left(\frac{|\nabla a|}{|\nabla I|} \right)$ is approximately $(-1, 0)$ at column 50 but there the normalized gradient $\frac{\nabla a}{|\nabla a|} \simeq (-1, 0)$ as well.

The conclusion is that, in practice, the second term of the right-hand side of equation (2.1) is in the range $[0, 1]$, so we may bound its variance by 0.25. This leads to a correction of equation (2.13) for this type of contour

$$Var(n_\kappa) \simeq \frac{20}{|\nabla I|^2} \sigma^2 + T'_x \quad \text{where} \quad T'_x \in [0, 0.5]. \quad (2.23)$$

2.3. PSNR along contours: numerical experiments. We have performed tests on two very simple synthetic images, one binary and the other textured, where we add noise with different σ values to them and compute the PSNR of the image, curvature and normal field along the central circumference (for the normal we average PSNR values of the vertical and horizontal components).

Figure 2.1 shows the results for the textured image, where we can see that the PSNR values are consistent with our estimates.

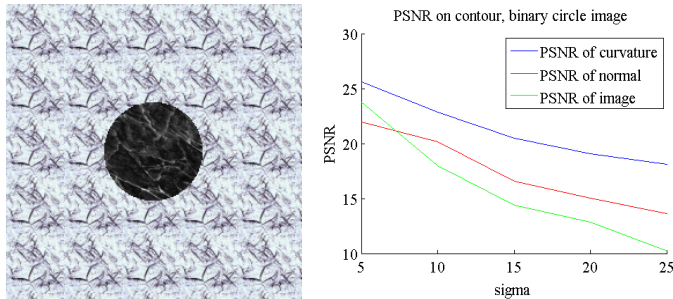


FIG. 2.1. Left: test image. Right: PSNR values of image, curvature and normal along contour.

Figure 2.2 shows the results for the binary image, which are also consistent with our estimates once we introduce the correction term of equation (2.23). As the equation predicts, for this case we see that if σ is small then the PSNR along the contours of the image may be larger than that of the curvature. Nonetheless, this does not affect the results of our denoising framework, which we will detail in section 3: with our approach we obtain denoised results with higher PSNR, computed over the whole image, even for binary images and small values of σ . In particular, for the binary circle image of fig. 2.2, for noise of standard deviation $\sigma = 5$ and for total variation (TV) based denoising with $\mathcal{F} = ROF$ [7] we obtain, with our proposed framework (i.e. by applying TV denoising to the curvature), a denoised image result with PSNR=47.85, whereas direct TV denoising on the image yields PSNR=46.77. The influence of homogeneous regions on the PSNR is discussed next.

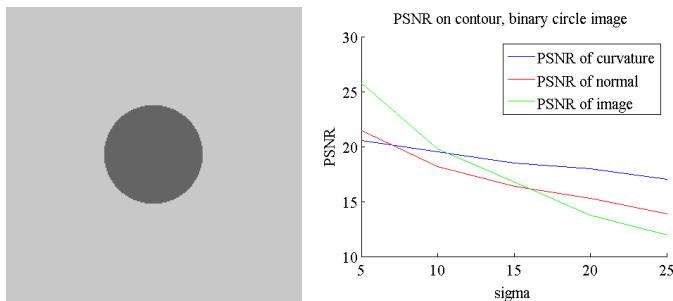


FIG. 2.2. Left: test image. Right: PSNR values of image, curvature and normal along contour.

2.4. PSNR in homogeneous regions. On homogeneous or slowly varying regions, (2.2) is no longer valid and we have instead

$$|\nabla a| \ll |\nabla n|, \quad (2.24)$$

so now

$$\kappa(I) \simeq \kappa(n) + \nabla \cdot \left(\frac{\nabla a}{|\nabla I|} \right). \quad (2.25)$$

In this case $\kappa(I)$ cannot 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 (2.24) and (2.25) we see that $\kappa(I)$ behaves like $\kappa(n)$ plus a perturbation. Since n is random noise with mean zero, so is $\kappa(n)$ and thus so is $\kappa(I)$. Therefore, any simple denoising method applied to $\kappa(I)$ will result in values of $\kappa_{\mathcal{F}}$ close to zero in homogeneous or slowly varying regions. So after running *Step 2* of the proposed approach below in Algorithm 2 (for e.g. we could use (1.2)), the reconstructed (denoised) image $\hat{I}_{\mathcal{F}}$ 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.

3. Proposed Algorithm.

3.1. The Model. The observations in the previous sections have motivated us to perform a number of experiments comparing the following two Algorithms.

Algorithm 1 Direct approach

Apply a denoising approach \mathcal{F} to directly smooth an image I , obtaining a denoised image $I_{\mathcal{F}} = \mathcal{F}(I)$.

Algorithm 2 Proposed approach

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

Step 2: Generate an image $\hat{I}_{\mathcal{F}}$ that satisfies the following criteria:

1. $\kappa(\hat{I}_{\mathcal{F}}) \simeq \kappa_{\mathcal{F}}$; that is, the level lines of $\hat{I}_{\mathcal{F}}$ are well described by $\kappa_{\mathcal{F}}$.
 2. The overall contrast of $\hat{I}_{\mathcal{F}}$ matches that of the given data $I = a + n$ in the sense that the intensity of any given level line of $\hat{I}_{\mathcal{F}}$ is close to the average value of I along that contour.
-

We have tested both variational and patch based approaches for the denoising method \mathcal{F} . So $\kappa_{\mathcal{F}}$ has been generated from fairly diverse methods in *Step 1*.

The precise method of reconstruction for *Step 2* should potentially be related to the nature of the smoothed curvature $\kappa_{\mathcal{F}}$ from *Step 1*, and thus the choice of denoising method \mathcal{F} as well as the discretization of $\kappa(I)$. For simplicity, for all of the tests in this paper we have performed *Step 2* by solving

$$u_t = \kappa(u) - \kappa_{\mathcal{F}} + 2\lambda(I - u), \quad (3.1)$$

with initial data $u(0, \cdot) = I$ or $u(0, \cdot) = I_{\mathcal{F}}$ where λ is a positive parameter depending on the noise level (and possibly depending on time). This is just one choice and in section 4.6 we discuss other alternatives. But we chose to use (3.1) as a baseline for our experiments since its behavior is well-understood. In particular, (3.1) is the flow

of the Euler-Lagrange equation associated with minimization problem

$$\begin{aligned}\hat{u} &= \arg \min_{u \in BV(\Omega) \cap L^2(\Omega)} \int (|\nabla u| + \kappa_{\mathcal{F}} u) + \lambda \int (I - u)^2 \\ &= \arg \min_{u \in BV(\Omega) \cap L^2(\Omega)} \int |\nabla u| + \lambda \int \left(\left(I - \frac{1}{2\lambda} \kappa_{\mathcal{F}} \right) - u \right)^2 =: \arg \min_{u \in BV(\Omega) \cap L^2(\Omega)} \Phi(u)\end{aligned}\quad (3.2)$$

which is the well known problem proposed by Rudin-Osher-Fatemi [7], with $I - \frac{1}{2\lambda} \kappa_{\mathcal{F}}$ used in the data fidelity term instead of the noisy data I . If $I, \kappa_{\mathcal{F}} \in L^2(\Omega)$, the above functional has a unique minimizer [9]. Furthermore, extending the definition of Φ in (3.2) to all of $L^2(\Omega)$ by setting $\Phi(u) := +\infty$ for $u \in L^2(\Omega) \setminus BV(\Omega)$, the functional is proper, convex and lower semi-continuous and thus by the theory of maximal monotone operators ([10] Theorem 3.1) there exists a unique solution $u(t, \cdot)$ in the semigroup sense to (3.1) for *a.e.* $t \in (0, \infty)$. The argument in Vese [11], Theorem 5.4 guarantees that at $t \rightarrow \infty$, $u(t, \cdot)$ converges strongly in $L^2(\Omega)$ and weakly in $L^1(\Omega)$ to the minimizer \hat{u} of (3.2), that satisfies $0 \in \partial\Phi(\hat{u})$ where $\partial\Phi(u) := \{p \in L^2 | \Phi(v) \geq \Phi(u) + \langle p, v - u \rangle \forall v \in L^2\}$ is the subdifferential of Φ at \hat{u} .

We solve (3.1) by iterating for $m = 1, 2, 3, \dots$

$$u_m = u_{m-1} + \Delta t (\kappa(u_{m-1}) - \kappa_{\mathcal{F}} + 2\lambda(I - u_{m-1})) \quad (3.3)$$

where $\kappa(u) = \kappa(u(x, y))$ is computed using the classical numerical scheme of [7], with forward-backward differences and the minmod operator to ensure stability. Our initial condition is either $u(0, \cdot) = I$ or $u(0, \cdot) = I_{\mathcal{F}}$, each leading to slightly different solutions since in practice we don't necessarily solve for the minimizer \hat{u} . Rather, we stop the iterations when the mean squared error at iteration m ,

$$MSE(m) = \frac{1}{|\Omega|} \sum_{x \in \Omega} (I(x) - u(t = m, x))^2, \quad (3.4)$$

or root mean squared error $RMSE(m) = \sqrt{MSE(m)}$ satisfies

$$MSE(m) \geq \sigma^2 \quad \text{or} \quad \epsilon(m) := |RMSE(m+1) - RMSE(m)| \leq 0.0005, \quad (3.5)$$

whichever happens first. Therefore, the solution of Algorithm 2 is $\hat{I}_{\mathcal{F}} = u(T_{\sigma}, \cdot)$ where $T_{\sigma} = \min\{t > 0 | MSE(t) \geq \sigma^2 \text{ or } \epsilon(t) \leq 0.0005\}$. The curvature $\kappa(\hat{I}_{\mathcal{F}})$ will not precisely be equal $\kappa_{\mathcal{F}}$, but at this steady state described above it will be a good approximation.

But for now we wish to emphasize that equation (3.1) is just one option to use for Algorithm 2, which we have chosen given its simplicity and its well understood behavior. In section 4.6 we discuss an alternate reconstruction equation that also yields a solution $\hat{I}_{\mathcal{F}}$ satisfying properties 1 and 2 in *Step 2* Algorithm 2 and may potentially work better. We also discuss some future work related to *Step 2* in section (5.2).

3.2. Relationship with Previous Work. Lysaker et.al. [6] proposed a two step denoising algorithm in which they first approximate a smooth normal field, $\vec{\eta}_1$, to the noisy image, I , using

$$\vec{\eta}_1 = \arg \min_{|\vec{\eta}|=1} \int |\nabla \vec{\eta}| + \lambda \int \left(\frac{\nabla I}{|\nabla I|} - \vec{\eta} \right)^2 \quad (3.6)$$

and then obtain the denoised image via the minimization problem

$$u_2 = \arg \min_{u \in BV} \int (|\nabla u| - \vec{\eta}_1 \cdot \nabla u) + \lambda \int (I - u)^2. \quad (3.7)$$

Note that (3.2) only differs from (3.7) in that (3.2) uses denoised curvature, while (3.7) uses denoised unit normals. The functionals in (3.7) and (3.2) are directly related to the one introduced in Ballester et.al. [12] 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

$$F(u) = \int |\nabla u| - \theta \cdot \nabla u \quad (3.8)$$

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 .

The authors in [13] proposed a similar algorithm to the one in [6], but first they solve for a divergence-free, noise-free approximate unit tangent field, $\vec{\xi} = (\xi^1, \xi^2)$ (a more mathematically sound minimization problem than (3.6)), use this to compute $\vec{\eta}_1 = (-\xi^2, \xi^1)$, and then solve for the clean image using (3.7). Other works have built on this model. For example, the authors in [14] suggest replacing (3.7) with a more direct feature orientation-matching functional. From our argument in section 2, the denoised curvature should be easier to obtain than the denoised unit normals (similarly, the denoised unit tangents) given it generally has a higher PSNR at the edges. We should point out that another key difference between the proposed approach and the others described here is that we are suggesting one should be able to modify *any* denoising algorithm to obtain $\kappa_{\mathcal{F}}$, not only variational approaches.

The results in [6] inspired several other works that are related to our proposed approach. One of them, the Bregman iterative algorithm of Osher et. al. [8], has made a particular impact on the field of variational based image processing. The motivation was to replace (3.6) with $\vec{\eta}_1 = \frac{\nabla u_1}{|\nabla u_1|}$ where u_1 is the denoised image obtained from minimizing the Rudin-Osher-Fatemi (ROF) functional [7], then solve (3.7). The authors observed that the same solution could be obtained by minimizing the ROF functional to obtain u_1 , computing the residual noise $v_1 = f - u_1$, and finally minimizing the ROF functional again but with data $f + v_1$. They also discovered that better results could be obtained by starting with an image of all zeros and iteratively repeating this process until the solution was within a distance of σ from the noisy image. This process can be formulated in terms of the Bregman distance [15], and a more efficient version, the linearized Bregman method [16], was proposed several years later. Its formulation and connection with the reconstruction equation (3.1) is as follows.

Given a convex functional $J(\cdot)$ defined on BV , its subdifferential is defined to be $\partial J(u) = \{p \in BV^* | J(v) \geq J(u) + \langle p, v - u \rangle \forall v \in BV\}$, and for $p \in \partial J(u)$, the Bregman distance between u and v is $D_J^p(u, v) := J(u) - J(v) - \langle p, u - v \rangle$. Then starting with $u^0 = 0$ and $p^0 = 0 \in \partial J(u^0)$, the linearized Bregman method [16]

iterates for $k = 0, 1, 2, \dots$

$$u^{k+1} = \arg \min_{u \in BV} \{D_J^{p^k}(u, u^k) + \frac{1}{2\delta} \|u - (u^k - \delta(u^k - I))\|_{L^2}^2\} \quad (3.9)$$

$$p^{k+1} = p^k - \frac{1}{\delta}(u^{k+1} - u^k) - (u^{k+1} - I). \quad (3.10)$$

Writing (3.10) as

$$\frac{1}{\delta}(u^{k+1} - u^k) = -p^{k+1} + p^k + (I - u^{k+1}), \quad (3.11)$$

and noting that if $J(u)$ is the total variation of u then $\partial J(v) = -\kappa(v)$, (3.11) can be interpreted as a discretized version of (3.1) (with $\lambda = \frac{1}{2}$ and $\Delta t = \delta$) with one seemingly small, yet critical, difference. The 'denoised' curvature $-\kappa_{\mathcal{F}} = -\mathcal{F}(\kappa(I))$ (a smoothed version of the curvature of I) in (3.1) plays the role of $p^k \in \partial J(u^k) = -\kappa(u^k)$ (the curvature of a smoothed version of I , in a sense, $\kappa(\mathcal{F}(I))$) in (3.11). We discuss the difference between using $\mathcal{F}(\kappa(I))$ and $\kappa(\mathcal{F}(I))$ in section 4.5.

In summary, the proposed approach described in Algorithm 2 is closely related to the work in [6, 8, 13, 14] but with two main differences. First, in our first step we denoise the curvature instead of the unit normals or the image itself. Second, the approaches in e.g. [6, 8, 13, 14] provide precise algorithms, while our approach is intended to be quite general. This is due to our speculation that if it is possible to modify an image denoising approach so it is applicable to curvature images, Algorithm 2 should yield better results than Algorithm 1. We demonstrate in the next section that the type of denoising approaches that can be used include (but are not necessarily limited to) variational approaches and patch-based methods.

4. Experiments. The image database used in our experiments is the set of grayscale images (range $[0, 255]$) obtained by computing the luminance channel of the images in the Kodak database [17] (at half-resolution). We tested five denoising methods: TV denoising [7], the Bregman iterative algorithm [8], orientation matching using smoothed unit tangents [14], Non-local Means [4], and Block-matching and 3D filtering (BM3D) [18]. Our experiments show that for all of these algorithms, we obtain better results by denoising the curvature image $\kappa(I)$ rather than directly denoising the image I .

To compute $\kappa(u)$ in the reconstruction equation (3.1) we have used the classical numerical scheme of [7], with forward-backward differences and the minmod operator, to ensure stability. Therefore, we also use this for the initialization of the noisy curvature $\kappa(I)$.

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

$$u_t = \nabla \left(\frac{\nabla u}{|\nabla u|} \right) + 2\lambda(t)(I - u), \quad u(0, \cdot) = I \quad (4.1)$$

where $\lambda(t)$ is estimated at each iteration, knowing the value σ of the standard deviation of the noise. The stopping criterion is based on $MSE(I, u(t))$ as described in (3.4)-(3.5), and thus $I_{ROF} = u(T_\sigma, \cdot)$ where $T_\sigma = \min\{t > 0 \mid MSE(I(x), u(t, x)) \geq \sigma^2 \text{ or } \epsilon(t) \leq 0.0005\}$.

To fit this into our framework, we do the following:

Step 1: Perform TV denoising of $\kappa(I)$

$$\kappa_t = \nabla \left(\frac{\nabla \kappa}{|\nabla \kappa|} \right), \quad \kappa(0, \cdot) = \operatorname{div} \left(\frac{\nabla I}{|\nabla I|} \right) \quad (4.2)$$

which we iterate for a fixed number of steps, obtaining κ_{ROF} . 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 $\sigma = 10, 15, 20, 25$.

Step 2: Iterate the equation

$$u_t = \kappa(u) - \kappa_{ROF} + 2\lambda(t)(I - u), \quad u(0, \cdot) = I, \quad (4.3)$$

where $\lambda(t)$ is estimated at each iteration with time step $\Delta t = 0.1$, finally obtaining $\hat{I}_{ROF} = u(T_\sigma, \cdot)$, the solution satisfying the stopping criteria described in (3.5).

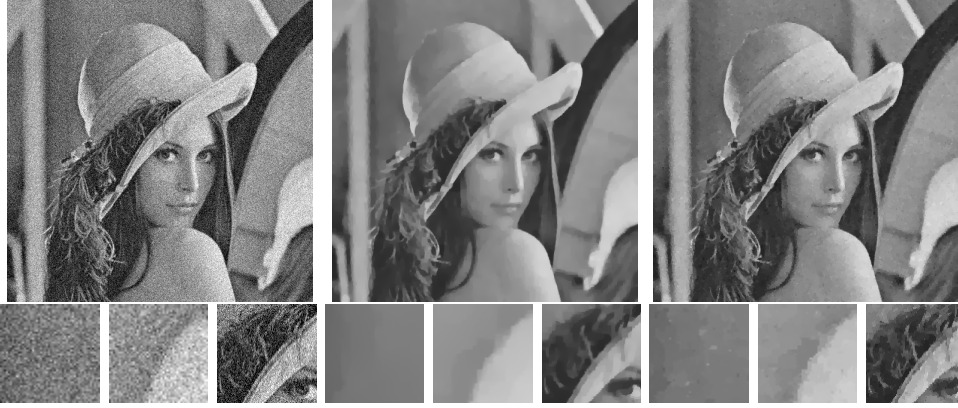


FIG. 4.1. Left: noisy image. Middle: result obtained with TV denoising of the image (PSNR=29.20 and PIQ=82.45). Right: result obtained with TV denoising of the curvature image (PSNR=29.36 and PIQ=95.64).

Figure 4.1 shows one example comparing the outputs of TV denoising of I and $\kappa(I)$ for the Lena image and noise with $\sigma = 25$. It is useful to employ for image quality assessment, apart from the PSNR, the Q-index of [5], which is reported as having higher perceptual correlation than PSNR and SNR-based metrics [19]; in our case we use the percentage increase in Q,

$$PIQ(I_{ROF}) = 100 \times \frac{Q(I_{ROF}) - Q(I)}{Q(I)} \quad \text{and} \quad PIQ(\hat{I}_{ROF}) = 100 \times \frac{Q(\hat{I}_{ROF}) - Q(I)}{Q(I)}.$$

In this image we obtain PSNR=29.36 and PIQ=95.64 for TV denoising of the curvature, while the values are PSNR=29.20 and PIQ=82.45 for TV denoising of the image.

In fig. 4.1 it's important to note that when using the proposed approach, while the PSNR and Q-index are both higher, details are better preserved (e.g. in the feathers of the hat), and edges have higher contrast than with TV (e.g. in the middle

close-up), smooth regions look worse than denoising I directly. We believe this is partly due to the fact that TV denoising is good at smoothing piecewise constant images, and $\kappa(I)$ certainly does not fall in that class as one can see in figure 1.2. We will see in subsequent sections that this is typically not an issue when using patch based approaches. Also, in figure 4.7 we demonstrate that better visual results can be obtained by an alternate reconstruction equation, which yields a much smoother reconstruction of uniform regions.

Figure 4.2 compares, on the left, the average increase in PSNR, computed over the entire Kodak database, obtained with both approaches: $\text{PSNR}(I_{ROF}) - \text{PSNR}(I)$ (in magenta), $\text{PSNR}(\hat{I}_{ROF}) - \text{PSNR}(I)$ (in blue). On the right, we plot the average percentage increase in Q-index.

Both plots in figure 4.2 show that TV denoising of the curvature allows us to obtain a denoised image \hat{I}_{ROF} 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.

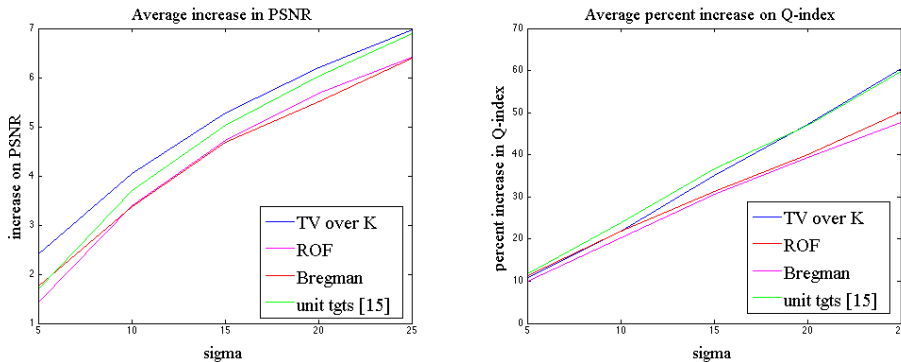


FIG. 4.2. Comparison of TV denoising of $\kappa(I)$ (blue), smoothing unit tangents [14] (green), TV denoising of I (ROF) (magenta), the Bregman iterative approach [20] (red). Left: PSNR increase for each method. Right: percentage increase on Q-index [5]. Values averaged over Kodak database (only luminance channel, images reduced to half-resolution).

4.2. Smoothing unit normals and the Bregman iterative approach.

Since the approach (4.2)-(4.3) is closely related to the approaches (3.6)-(3.7) and (3.9)-(3.10), given our discussion in section 2, a comparison with smoothing unit normals, $\mathcal{F}(\vec{\eta}(I))$, as well the Bregman iterative approach, which in a sense performs $\vec{\eta}(\mathcal{F}(I))$, is particularly relevant here. We showed in section 2 that, although better than direct denoising of I , denoising of the normalized-gradient field $\vec{\eta}(I)$ 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 4.2. This figure shows that the Bregman iterative approach fares better than ROF in terms of Q-index, although not in PSNR, and that TV denoising of $\kappa(I)$ outperforms both the Bregman iterative approach 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 iteration method of [8]; the values used for λ : 0.033, 0.013, 0.009, 0.005, 0.00425, corresponding to σ : 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 $\Delta t = 0.1$.

We also compared with one of the newer algorithms for matching unit normals, in which the unit tangents $\vec{\xi} = (\xi^1, \xi^2)$ are smoothed before matching unit normals

$\vec{\eta} = (-\xi^2, \xi^1)$ [14]. Comparisons with this approach are also included in figure 4.2. Smoothing unit tangents before matching unit normals produces results whose PSNR lie directly in between those for which I was smoothed before matching (Bregman) and our approach, which corresponds to our discussion in section 2. However the Q-measure was very similar between [14] and the proposed method, and were slightly better for the results in [14] at lower noise levels.

4.3. Non-Local Means. To illustrate a comparison with patch-based methods, we incorporated Non-Local Means denoising [4] into our general framework as follows. First we performed Non-local Means denoising on the original noisy image I using the code from [21] (with their choice of parameters) obtaining the denoised image I_{NLM} .

For our method, we have done the following:

Step 1: Apply NLM to $\kappa(I)$, but with the following two modifications.

1. Compute the weights from I instead of $\kappa(I)$ (i.e. compare image patches, not curvature patches).
2. Use $\sigma_\kappa = \sigma + 5$ as the standard deviation.

We obtain the denoised curvature κ_{NLM} .

Step 2: Starting with $u(0, \cdot) = I_{NLM}$, solve

$$u_t = \kappa(u) - \kappa_{NLM} + 2\lambda(I - u),$$

to obtain \hat{I}_{NLM} , the solution satisfying the stopping criterion described in (3.5). The values used for $\lambda : 0.2, 0.075, 0.05, 0.04, 0.03$, correspond to $\sigma : 5, 10, 15, 20, 25$ respectively.

Note that in *Step 1* we perform a weighted average of the curvature patches but compute the weights by comparing image, not curvature, patches. This is due to the nature of curvature patches, in which the curvature of a noisy but homogeneous patch takes random, large values. Therefore, comparing curvature patches directly would not be the best representation of the contours because the noise would be attributed equal importance. However, averaging curvature patches in the spirit of NLM, but with a different criterion for computing the weights, is quite effective. So the nature of the NLM algorithm in which 'similar' patches are averaged is still preserved with this adjustment.

Figure 4.3 shows one example comparing the outputs of NLM denoising of I and $\kappa(I)$. Figure 4.4 (left) compares the average increase in PSNR over the entire Kodak database of the denoised image over the original noisy image, obtained with both approaches: NLM applied to I (in magenta) and NLM applied K with different initial conditions (blue and green).

Note that if the starting condition were $u(0, \cdot) = I$ as usual then denoising the curvature performs worse, in terms of PSNR, than denoising the image. A starting condition closer to the solution, such as $u(0, \cdot) = I_{NLM}$, provides a better result and this highlights a limitation of the specific reconstruction method (3.1) chosen for *Step 2*. Although now it could be argued that what we are doing may just be TV denoising of I_{NLM} (in fact, if we over-process $\kappa(I)$ we obtain $\kappa_{NLM} \cong 0$ and in that case we would actually be applying ROF denoising to I_{NLM}). But this is not the case. If we apply ROF to I_{NLM} as explained in section 4.1 (with variable $\lambda(t)$ and the stopping criteria mentioned there), the outputs have lower PSNR and Q-index (see figure 4.4).



FIG. 4.3. *Left: noisy image. Middle: result obtained with NLM denoising of the image. Right: result obtained with NLM denoising of the curvature image.*

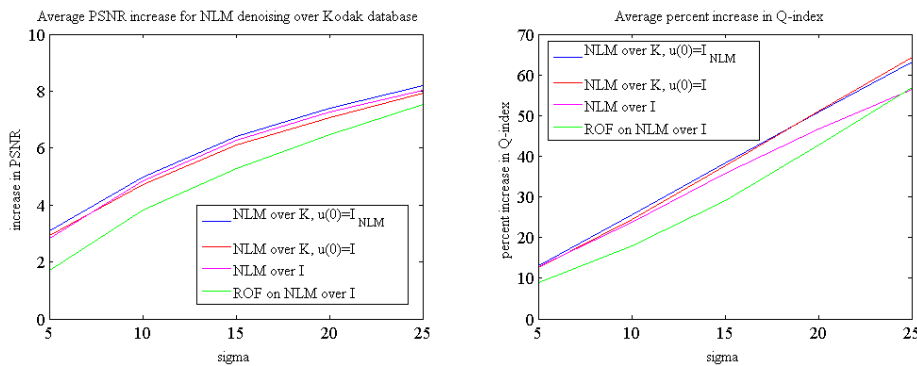


FIG. 4.4. *Comparison of NLM denoising on I and NLM denoising on $\kappa(I)$. Left: PSNR increase for each method; also pictured: PSNR increase for ROF applied to the output of NLM on I . Right: percentage increase on Q-index [5]. Values averaged over Kodak database (only luminance channel, images reduced to half-resolution).*

Figure 4.4 (right) compares the average percent increase in Q-index of the denoised image over the original noisy image, obtained with both approaches. Note that applying NLM to the curvature gives a better result, regardless of the initial condition.

Both plots in figure 4.4 show that NLM denoising of the curvature allows us to obtain a denoised image \hat{I}_{NLM} which is better in terms of PSNR and Q-index than the image I_{NLM} , obtained directly by applying NLM to the original noisy image.

4.4. BM3D. We’ve also applied our framework to the BM3D denoising algorithm [18], which is arguably the best denoising method available. As with Non-local Means, first we applied BM3D to the original noisy image I using the code from [20] (with their choice of parameters) obtaining the denoised image I_{BM3D} .

For our method, we have done the following:

Step 1: Apply BM3D to $\kappa(I)$ (actually to $\kappa(I) + 127.5$ to ensure positive values), but with these three modifications:

1. Compute the weights from I instead of $\kappa(I)$ (i.e. compare image patches, not

curvature patches; this is for the same reason as described for NLM in section 4.3).

2. Use the threshold value $\lambda_{3D} = 1.0$ (instead of the suggested value $\lambda_{3D} = 2.7$).
3. Run only the first step (basic estimate), omitting the collaborative Wiener filtering stage (this was done for simplicity).

We obtain the denoised curvature κ_{BM3D} .

Step 2: Starting with $u(0, \cdot) = I_{BM3D}$, solve

$$u_t = \kappa(u) - \kappa_{BM3D} + 2\lambda(I - u),$$

to obtain \hat{I}_{BM3D} , the solution satisfying the stopping criterion described in (3.5).

We have used two sets of values for λ , depending on the image content. For images with more texture and significant variation, $\lambda : 0.3, 0.15, 0.1, 0.07, 0.045$, corresponding to $\sigma : 5, 10, 15, 20, 25$ respectively. For images with large homogeneous regions, $\lambda : 0.2, 0.075, 0.05, 0.04, 0.03$, corresponding to $\sigma : 5, 10, 15, 20, 25$ respectively.

Figure 4.5 (left) compares the average increase in PSNR of the denoised image over the original noisy image, obtained with both approaches: BM3D applied to I (in magenta) and BM3D applied to $\kappa(I)$ (in blue). We checked two cases. In the first, we did the same experiment as in the other comparisons where we averaged over the entire database. These results are the solid lines. The PSNR was almost identical, but we did see a slight increase in Q-index using our approach. We then considered the images in the Kodak database that were more heavily textured. For both measures the increments in quality with our approach are modest, although we perform consistently better than direct BM3D denoising of I . Moreover, our very modest improvement is consistent with the bound on optimal denoising of Levin and Nadler [22] and Levin et al. [23], although Lebrun et al. [24] point out that the actual bound might be larger, because the performance bounds in [22] are computed considering a generic class of patch-based algorithms with stronger assumptions than those corresponding to BM3D.

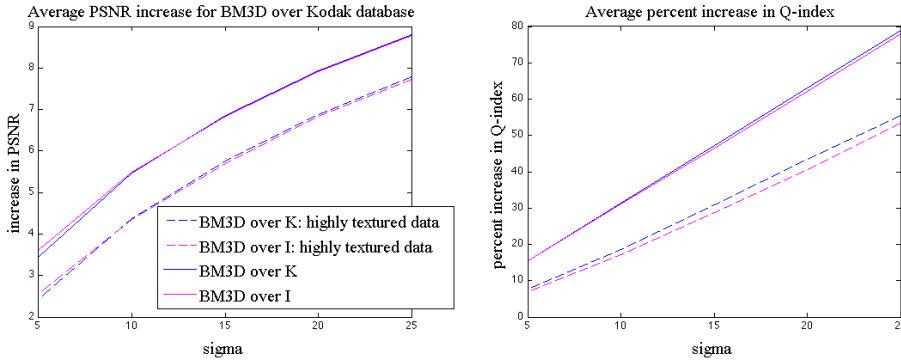


FIG. 4.5. Comparison of BM3D denoising on I and BM3D denoising on $\kappa(I)$, using both the entire database as well as focusing on just the highly textured images. Left: PSNR increase for each method. Right: percentage increase on Q-index [5]. The solid lines represent the results from averaging over the entire 24 image Kodak database. The dotted lines represent the results from averaging over images 1, 2, 5, 11-14, 18, 22 and 24 from the Kodak database. In both cases only the luminance channel was used and the images were reduced to half-resolution.

4.5. Computing $\mathcal{F}(\kappa(I))$ -vs- $\kappa(\mathcal{F}(I))$. Computing $\kappa(\mathcal{F}(I))$ in *Step 1* of the proposed approach seems more reasonable than what we suggest to do, which is to compute $\mathcal{F}(\kappa(I))$, for a number of reasons. To start, because I is noisy, one typically would want to regularize I before computing its curvature [2]. Furthermore, the denoising approaches we use here were developed for denoising image data, not for denoising curvature data. We performed the experiment of comparing the results when we use different choices for $\kappa_{\mathcal{F}}$, and some examples can be found in figure 4.6.



FIG. 4.6. Reconstructions using different choices for $\kappa_{\mathcal{F}}$ in *Step 1* of the proposed approach for the noisy image in figure 1.4. (a) $\kappa_{ROF} = \kappa(\text{ROF}(I))$, $PSNR=29.39$, $PIQ=69$. (b) $\kappa_{ROF} = \text{ROF}(\kappa(I))$ (proposed approach), $PSNR=29.41$, $PIQ=92$. (c) $\kappa_{NLM} = \kappa(\text{NLM}(I))$, $PSNR=30.32$, $PIQ=80$. (d) $\kappa_{NLM} = \text{NLM}(\kappa(I))$ (proposed approach), $PSNR=30.86$, $PIQ=103$.

All the images in figure 4.6 were generated using Algorithm 2, but whereas images (a) and (c) used $\kappa_{\mathcal{F}} = \kappa(\mathcal{F}(I))$ in *Step 1*, images (b) and (d) were generated using $\kappa_{\mathcal{F}} = \mathcal{F}(\kappa(I))$. We can see that for both denoising using the Rudin-Osher-Fatemi functional and denoising with Non-local Means, the results have higher quality when we use the proposed approach of $\kappa_{\mathcal{F}} = \mathcal{F}(\kappa(I))$. This also reflects the comparisons we found between computing $\mathcal{F}(\vec{\eta}(I))$ and $\vec{\eta}(\mathcal{F}(I))$ reported in section 4.2.

4.6. Alternate reconstruction equations. We have also tried alternate reconstruction equations and have found some improvements over (3.1). For example, given the denoised curvature $\kappa_{\mathcal{F}} = \mathcal{F}(\kappa(I))$, one could solve for

$$\hat{I}_{\mathcal{F}} = \arg \min_u \int_{\Omega} |\kappa(u) - \kappa_{\mathcal{F}}| + \frac{\lambda}{2} \int_{\Omega} (I - u)^2 \quad (4.4)$$

in *Step 2* of the proposed approach. A minimizer of (4.4) should satisfy that both $\kappa(\hat{I}_{\mathcal{F}})$ should be close to $\kappa_{\mathcal{F}}$ and the average value of $\hat{I}_{\mathcal{F}}$ (along level lines) should be close to the average value of I , and thus the average value of a . So both clean level lines and contrast should be preserved. This is related to the model for denoising an image by directly minimizing its mean curvature proposed by Zhu and Chan [25] in which the authors minimize

$$\int_{\Omega} |H(u)| + \frac{\lambda}{2} \int_{\Omega} (I - u)^2$$

where $H(u) = \text{div} \left(\frac{\nabla u}{\sqrt{\epsilon + |\nabla u|^2}} \right)$ with $\epsilon = 1$. A fast multigrid algorithm for computing the above equation was proposed in Brito and Chen [26] that works for small values of ϵ , making it close to $\kappa(u)$.

Brito and Chen modified their algorithm to solve (4.4), and using this as our reconstruction equation for the Lena image with noise $\sigma = 10$ showed a clear improvement when choosing $\mathcal{F} = ROF$ (as in section 4.1), and a smaller but still notable improvement when $\mathcal{F} = NLM$ (as in section 4.3). These results can be seen in figure 4.7. Performing extensive tests with other images and different noise levels requires a more careful study of Brito and Chen’s algorithm, with an adequate selection of parameters, and it will be the subject of further work.

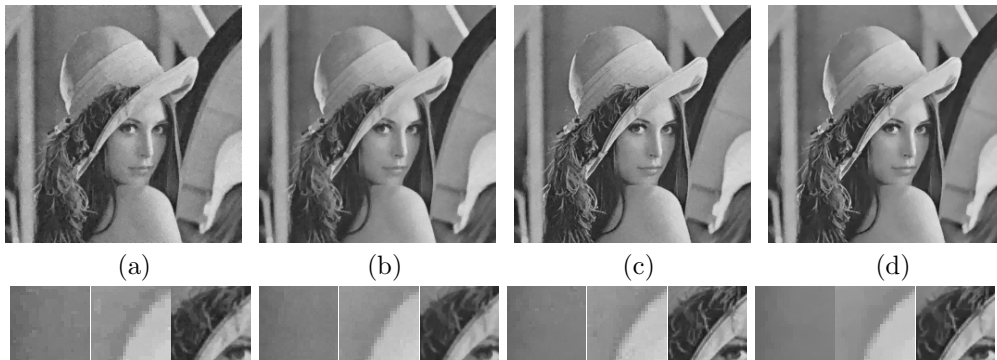


FIG. 4.7. Reconstruction of Lena image with additive noise of $\sigma = 10$, using different reconstruction equations. (a) TV denoising of curvature, Step 2 using (3.1), $PSNR=32.74$, $PIQ=28$. (b) TV denoising of curvature, Step 2 using (4.4), $PSNR=33.94$, $PIQ=35$. (c) NLM denoising of curvature, Step 2 using (3.1), $PSNR=34.20$, $PIQ=33$. (d) NLM denoising of curvature, Step 2 using (4.4), $PSNR=34.55$, $PIQ=37$.

5. Discussion.

5.1. Computing the curvature. Kovalevsky shows in [27] 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 [28] points out that the smaller the curvature, the larger the error in estimating it. Ciomaga et al. [2] 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 figure 1.4 and our other experiments show. We would like to test other computational techniques for the curvature, and their impact in the quality of the results. This is non-trivial, as the numerical approximation of $\kappa(u)$ in the reconstruction equations (3.1) and (4.4) is directly related to the stability of the algorithm.

5.2. The reconstruction equation. The equations we have tested for reconstructing an image from a clean curvature image, equations (3.1) and (4.4), show promise for this general approach, but there are still a number of questions. For instance, if one set $\lambda = 0$ in (3.1) and assumed the initial data was in $L^1(\Omega)$, the reconstruction equation is similar to one in which Andreu et al. [29] established the well-posedness and characterized the long time behavior of the solutions. This might be another approach which could ensure a more accurate and quantifiable depiction of

$\kappa_{\mathcal{F}}$ in $\kappa(u)$, although one would need to add a different mechanism to ensure contrast of the level lines is preserved.

However, even with some improvement there is the inherent challenge of studying a reconstruction equation in the continuous domain since curvature is incredibly poorly behaved in this setting. It takes on infinite values at corners and cusps of the level lines of the image data, and is highly oscillatory at noise. However, in the discrete domain, $\kappa(I)$ is bounded (e.g. using directional differences as we have done here, $\kappa(I) \in [-(2 + \sqrt{2}), 2 + \sqrt{2}]$) and quite manageable. The ability of a number of image denoising algorithms to be adapted for denoising curvature images in practice is some evidence of this. But now this also begs the question of computing the curvature more accurately than (2.5), such as using the curvature microscope work of Ciomaga et.al. [2]. The challenge here is that we do not currently know of a way of reconstructing an image from this discretized curvature information; the approach must be quite different than what we proposed if we compute the curvature along level lines, such as in [2], rather than using finite differences, as we do here. Thus there is some balance between the accuracy of the discretization of $\kappa(I)$ and the ability to easily reconstruct u from this information.

So although in practice we obtained promising results, finding a solid, mathematically sound methodology that fits into our approach would preferably require that the method \mathcal{F} of smoothing $\kappa(I)$ to obtain $\kappa_{\mathcal{F}}$ in *Step 1* should be intimately related to the method of reconstructing $\hat{I}_{\mathcal{F}}$ from $\kappa_{\mathcal{F}}$ in *Step 2*. And both depend on the discretization of $\kappa(I)$. The numerical approach proposed here is intended to illustrate the principle we derived in section 2, although we plan to explore some of these questions in future work.

5.3. Real curvature images. After we apply a given denoising method \mathcal{F} to the curvature image $\kappa(I)$ we obtain an image $\kappa_{\mathcal{F}} = \mathcal{F}(\kappa(I))$ which we call (and treat as) “denoised curvature”, i.e. as being the curvature of some given image. Indeed, if $\lambda = 0$ in (3.1) and the equation were run to convergence, the steady state solution $\hat{I}_{\mathcal{F}}$ should satisfy $\kappa_{\mathcal{F}} = \kappa(\hat{I}_{\mathcal{F}})$. However, we don’t expect this to be true when $\lambda > 0$ and the stopping criteria (3.5) is used. So in that case we cannot formally say that $\kappa_{\mathcal{F}}$ is actually a *curvature image*, or at least that it is the curvature image of $\hat{I}_{\mathcal{F}}$. This does not seem to hinder the approach from improving on denoising methods in general, but we are still exploring more precisely what effect this has on our solution. This also further begs the question from section 5.2 of whether a method other than equations (3.1) or (4.4) would yield a more optimal reconstruction. In the case that we can guarantee that $\kappa_{\mathcal{F}}$ is the curvature of some image, there may be some interesting connections with a Bregman type approach.

6. 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 speculate 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. 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. First and foremost, we would like to dedicate this paper to Vicent Caselles. We would like to thank these researchers for their very helpful com-

ments and suggestions: Stanley Osher, Guillermo Sapiro, Jesus Ildefonso Díaz, Ron Kimmel, Jean-Michel Morel and Alfred Bruckstein. We would also like to thank the following researchers for their inestimable help with our experiments: Adina Ciomaga and Wei Zhu for providing their code, Joouyoung Hahn for providing code and for generating the numerical results from [14] that were used in the comparisons in figure 4.2, and Carlos Brito and Ke Chen for adapting their algorithm and code to solve (4.4). We also thank the anonymous reviewers for their valuable comments and suggestions. We 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 European Research Council, Starting Grant ref. 306337, and by Spanish grants AACC, ref. TIN2011-15954-E, and Plan Nacional, ref. TIN2012-38112. The second author was supported in part by NSF-DMS #0915219.

REFERENCES

- [1] F. Attneave, “Some informational aspects of visual perception.” *Psychological review*, vol. 61, no. 3, p. 183, 1954.
- [2] A. Ciomaga, P. Monasse, and J.-M. Morel, “Level lines shortening yields an image curvature microscope,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 4129–4132.
- [3] M. Bertalmío and S. Levine, “A variational approach for the fusion of exposure bracketed images,” *Image Processing, IEEE Transactions on*, to appear, 2012.
- [4] A. Buades, B. Coll, and J.-M. 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.
- [5] Z. Wang and A. Bovik, “A universal image quality index,” *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, 2002.
- [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] R. Acar and C. R. Vogel, “Analysis of bounded variation penalty methods for ill-posed problems,” *Inverse Problems*, vol. 10, no. 6, pp. 1217–1229, 1994. [Online]. Available: <http://stacks.iop.org/0266-5611/10/1217>
- [10] H. Brézis, *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*. Amsterdam: North-Holland Publishing Co., 1973, north-Holland Mathematics Studies, No. 5. Notas de Matemática (50).
- [11] L. Vese, “A study in the BV space of a denoising-deblurring variational problem,” *Appl. Math. Optim.*, vol. 44, no. 2, pp. 131–161, 2001.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] L. Bregman, “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming,” *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200 – 217, 1967.
- [16] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, “Bregman iterative algorithms for l_1 -minimization with applications to compressed sensing,” *SIAM J. Imaging Sci.*, vol. 1, no. 1, pp. 143–168, 2008.
- [17] Kodak, “<http://r0k.us/graphics/kodak/>.”
- [18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-

- domain collaborative filtering,” *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [19] 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.
- [20] Marc Lebrun, “An Analysis and Implementation of the BM3D Image Denoising Method,” *Image Processing On Line*, 2012.
- [21] A. Buades, B. Coll, and J.-M. Morel, “Non-local Means Denoising,” *Image Processing On Line*, 2011.
- [22] 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.
- [23] 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.
- [24] M. Lebrun, M. Colom, A. Buades, and J. M. Morel, “Secrets of image denoising cuisine,” *Acta Numer.*, vol. 21, pp. 475–576, 2012.
- [25] W. Zhu and T. Chan, “Image denoising using mean curvature of image surface,” *SIAM J. Imaging Sci.*, vol. 5, no. 1, pp. 1–32, 2012.
- [26] C. Brito-Loeza and K. Chen, “Multigrid algorithm for high order denoising,” *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 363–389, 2010.
- [27] V. Kovalevsky, “Curvature in digital 2d images,” *IJPRAI*, vol. 15, no. 7, pp. 1183–1200, 2001.
- [28] S. Utcke, “Error-bounds on curvature estimation,” in *Scale Space Methods in Computer Vision*. Springer, 2003, pp. 657–666.
- [29] F. Andreu, V. Caselles, J. Diaz, and J. Mazón, “Some qualitative properties for the total variation flow,” *Journal of Functional Analysis*, vol. 188, no. 2, pp. 516–547, 2002.