

ON TOTAL VARIATION DENOISING: A NEW MAJORIZATION-MINIMIZATION ALGORITHM AND AN EXPERIMENTAL COMPARISON WITH WAVELET DENOISING

M. A. T. Figueiredo, J. B. Dias, J. P. Oliveira

R. D. Nowak

Instituto de Telecomunicações
Instituto Superior Técnico
1049-001 Lisboa, Portugal

Electrical and Computer Engineering Dept.
University of Wisconsin,
Madison, WI 53706, U.S.A.

ABSTRACT

Image denoising is one of the classical problems in image processing, which has been addressed using a variety of conceptual frameworks and computational tools. Most approaches use some form of penalty/prior as a regularizer, expressing a preference for images with some form of (generalized) “smoothness”. Total variation (TV) and wavelet-based methods have received a great deal of attention in the last decade and are among the state of the art in this problem. However, as far as we know, no experimental studies have been carried out, comparing the relative performance of the two classes of methods. In this paper, we present the results of such a comparison. Prior to that, we introduce a new majorization-minimization algorithm to implement the TV denoising criterion. We conclude that TV is outperformed by recent state of the art wavelet-based denoising methods, but performs competitively with older wavelet-based methods.

1. INTRODUCTION: WAVELET-BASED AND TOTAL-VARIATION DENOISING

In image denoising problems, the goal is to estimate an original image \mathbf{x} from an observed noisy version \mathbf{y} , modelled as

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \quad (1)$$

where \mathbf{n} is a sample of a zero-mean white Gaussian field of variance σ^2 . In this paper, we overload the notation \mathbf{x} and \mathbf{y} with different meanings: **(i)** a 2D (say, $M \times N$) array of image pixel values, **(ii)** the same values lexicographically stacked into a vector (say MN -dimensional), **(iii)** real-valued functions, $\mathbf{x} : \Omega \rightarrow \mathbb{R}$, $\mathbf{y} : \Omega \rightarrow \mathbb{R}$, defined on some domain $\Omega \subset \mathbb{R}^2$ (say $\Omega = [0, 1] \times [0, 1]$). Which notation is being used at each point will be clear from the context.

Good denoising performance can only be achieved if some form of regularization (prior knowledge, from a Bayesian viewpoint) is used to penalize “undesirable” solutions. Accordingly, typical criteria have the form

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \{ \|\mathbf{y} - \mathbf{x}\|^2 + \lambda P(\mathbf{x}) \}, \quad (2)$$

where $\|\mathbf{y} - \mathbf{x}\|^2$ has an obvious meaning for vectors or 2D arrays (sum of all the squared element-wise differences) and stands for

$$\|\mathbf{y} - \mathbf{x}\|^2 = \int_{\Omega} (\mathbf{y}(\mathbf{t}) - \mathbf{x}(\mathbf{t}))^2 dt,$$

when \mathbf{x} and \mathbf{y} are defined on the continuous domain Ω , while $P(\mathbf{x})$ denotes a penalty function(al) which is designed to have small values for “desirable” estimates.

In wavelet-based (WB) formulations, (1) becomes

$$\mathbf{y} = \mathbf{W}\boldsymbol{\theta} + \mathbf{n}, \quad (3)$$

as a result of writing $\mathbf{x} = \mathbf{W}\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the vector of representation coefficients and the set of columns of \mathbf{W} is a wavelet basis or a redundant dictionary. The penalized estimate of $\boldsymbol{\theta}$ is thus

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \{ \|\mathbf{y} - \mathbf{W}\boldsymbol{\theta}\|^2 + \lambda P(\boldsymbol{\theta}) \}, \quad (4)$$

where $P(\boldsymbol{\theta})$ is a penalty function expressing certain properties of the wavelet coefficients of the “desirable” images. In Bayesian approaches, $P(\boldsymbol{\theta})$ is minus de logarithm of a prior, usually expressing the sparseness of the wavelet coefficients of natural images [5, 8]. Choosing $P(\boldsymbol{\theta})$ and solving (4) have been the focus of a vast literature; let us simply mention the seminal reference [4] and some more recent publications where comprehensive literature reviews can be found: [5, 8, 10]. Wavelet based methods hold the current state of the art in image denoising [10]. A key feature of wavelet-based methods is their ability to remove noise while keeping important image detail, such as edges.

In *total variation* (TV) denoising (introduced in [11], see also [3] for a review of recent advances and literature), a continuous domain formulation is adopted and the estimation criterion is a variational problem,

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \left\{ \int_{\Omega} (\mathbf{y}(\mathbf{t}) - \mathbf{x}(\mathbf{t}))^2 dt + \lambda \text{TV}(\mathbf{x}) \right\}, \quad (5)$$

where $\text{TV}(\mathbf{x})$ is the integral of the norm of the gradient of \mathbf{x} , called the *total variation* of function \mathbf{x} ,

$$\text{TV}(\mathbf{x}) \equiv \int_{\Omega} |\nabla \mathbf{x}(\mathbf{t})| dt, \quad (6)$$

and where the minimization is over all square integrable functions defined on Ω . It has been shown that TV denoising is able to remove noise, but does so while preserving image edges, which are perceptually important. Solving (5) is usually considerably more difficult than solving (4) and has been the focus of a considerable amount of work over the last decade. Most of the approaches adopted to deal with (5) fall into one of three classes [3]: **(i)** solving the associated Euler-Lagrange equation, which is a nonlinear *partial differential equation* (PDE); **(ii)** using methods based on duality, still formulated in the continuous domain, which avoid some

This work was partially supported by *Fundação para a Ciência e Tecnologia* (FCT), Portuguese Ministry of Science and Higher Education, under project POSC/EEA-CPS/61271/2004.

of the difficulties of (5) at the cost of replacing it by a constrained variational problem; **(iii)** direct optimization methods applied to a discrete version of (5). Almost all the literature on TV denoising follows the first two approaches; however, in practice, since computer implementations can only handle images on discrete lattices, the solution methods derived on the continuous domain have to be replaced by discrete approximations. Consequently, all classes of approaches have to undergo some discretization. The choice to be made is between: **(a)** deriving a solution method on the continuous domain and then discretizing it; **(b)** discretizing the problem and then using a finite-dimensional optimization algorithm. In this paper, we propose a method of type **(b)**, which belongs to the class of *majorization-minimization* (MM) algorithms [6]. Recently, we have proposed a similar MM algorithm for image deconvolution, a related (but different) problem [1]. Another finite-dimensional algorithm for TV denoising has been recently proposed in [2].

The remaining sections of the paper are organized as follows. Section 2 introduces the discrete version of the TV denoising criterion and the proposed MM algorithm. Experimental results are presented in Section 3, and Section 4 concludes the paper.

2. AN MM ALGORITHM FOR TV DENOISING

2.1. Discretization of the TV Denoising Criterion

A discrete version of (5)-(6) can be obtained by considering that the functions \mathbf{x} and \mathbf{y} have been uniformly sampled. At this point, we switch the notation and start assuming that \mathbf{x} and \mathbf{y} denote vectors containing all these samples arranged in (say) column lexicographic ordering.

Using local differences to approximate the two orthogonal components of the gradient, we can write a discrete version of the TV penalty (denoted as TV_d),

$$\text{TV}_d(\mathbf{x}) = \sum_i \sqrt{(\Delta_i^h \mathbf{x})^2 + (\Delta_i^v \mathbf{x})^2}, \quad (7)$$

where Δ_i^h and Δ_i^v are operators corresponding to, respectively, horizontal and vertical first order differences, at pixel i ; that is, $\Delta_i^h \mathbf{x} \equiv x_i - x_{l(i)}$ and $\Delta_i^v \mathbf{x} \equiv x_i - x_{a(i)}$, where $l(i)$ and $a(i)$ denote the nearest neighbors of i , to the left and above, respectively.

The resulting denoising criterion is thus a finite dimensional optimization problem:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \{ \|\mathbf{x} - \mathbf{y}\|^2 + \lambda \text{TV}_d(\mathbf{x}) \}. \quad (8)$$

It is important to observe that although $\text{TV}_d(\mathbf{x})$ is a convex function of \mathbf{x} , it is not everywhere differentiable, which makes solving (8) a difficult task.

2.2. MM Algorithms

Let $L(\mathbf{x})$ be some function to be minimized (e.g., the objective function in (8)). Consider the class of iterative algorithms that produce a sequence $\{\hat{\mathbf{x}}^{(t)}, t = 1, 2, \dots\}$ according to

$$\hat{\mathbf{x}}^{(t+1)} = \arg \min_{\mathbf{x}} Q(\mathbf{x}, \hat{\mathbf{x}}^{(t)}). \quad (9)$$

Consider that function $Q(\cdot, \cdot)$ satisfies the following properties: $Q(\mathbf{x}, \mathbf{x}') \geq L(\mathbf{x})$, for any \mathbf{x}, \mathbf{x}' , and $Q(\mathbf{x}', \mathbf{x}') = L(\mathbf{x}')$. Then,

$$L(\hat{\mathbf{x}}^{(t+1)}) \leq Q(\hat{\mathbf{x}}^{(t+1)}, \hat{\mathbf{x}}^{(t)}) \leq Q(\hat{\mathbf{x}}^{(t)}, \hat{\mathbf{x}}^{(t)}) = L(\hat{\mathbf{x}}^{(t)}) \quad (10)$$

where the first inequality results from $Q(\mathbf{x}, \mathbf{x}') \geq L(\mathbf{x})$, for any \mathbf{x}, \mathbf{x}' , and the second one from the fact that, by definition (see (9)), $Q(\mathbf{x}, \hat{\mathbf{x}}^{(t)})$ attains its minimum at $\mathbf{x} = \hat{\mathbf{x}}^{(t+1)}$. This shows that the algorithm defined by (9) produces a sequence of monotonically descendent values of the objective function. The origin of the term *majorization-minimization* (MM) also becomes clear: MM algorithms work by considering, at each iteration, a *majorizer* (or upper bound) of the objective function which is *minimized* to obtain the next estimate [6].

Under mild conditions, namely that $Q(\mathbf{x}, \mathbf{x}')$ is continuous in $(\mathbf{x}, \mathbf{x}')$, all limit points of the MM sequence $\mathbf{x}^{(t)}$ are stationary points of L , and $L(\mathbf{x}^{(t)})$ converges monotonically to $L(\mathbf{x}^*)$, for some stationary point \mathbf{x}^* . If, additionally, L is strictly convex, $\mathbf{x}^{(t)}$ converges to the global minimum of L . Proofs of these properties are similar to those of similar properties of the EM algorithm [12].

MM algorithms have three (trivially shown) properties, of which we will make use below:

Property 1: Any function $Q_a(\mathbf{x}, \mathbf{x}')$ differing from $Q(\mathbf{x}, \mathbf{x}')$ by an additive term and/or a multiplicative factor (both independent of \mathbf{x}) defines the same MM algorithm.

Property 2: Let $L(\mathbf{x}) = L_1(\mathbf{x}) + L_2(\mathbf{x})$; consider two majorizers, $Q_1(\mathbf{x}, \mathbf{x}') \geq L_1(\mathbf{x})$ and $Q_2(\mathbf{x}, \mathbf{x}') \geq L_2(\mathbf{x})$, both with equality if $\mathbf{x} = \mathbf{x}'$. Then, all the following functions majorize $L(\mathbf{x})$ (with equality for $\mathbf{x} = \mathbf{x}'$) and can thus be used in an MM algorithm: $Q_1(\mathbf{x}, \mathbf{x}') + Q_2(\mathbf{x}, \mathbf{x}')$, $L_1(\mathbf{x}) + Q_2(\mathbf{x}, \mathbf{x}')$, and $Q_1(\mathbf{x}, \mathbf{x}') + L_2(\mathbf{x})$.

Property 3: The monotonicity property of MM algorithms is kept if, instead of exactly minimizing $Q(\mathbf{x}, \hat{\mathbf{x}}^{(t)})$ at each iteration (as in (9)), the following weaker condition is satisfied:

$$\hat{\mathbf{x}}^{(t+1)} \text{ is such that } Q(\hat{\mathbf{x}}^{(t+1)}, \hat{\mathbf{x}}^{(t)}) \leq Q(\hat{\mathbf{x}}^{(t)}, \hat{\mathbf{x}}^{(t)}).$$

Notice that this is the only property of $\hat{\mathbf{x}}^{(t+1)}$ that was invoked in (10) to show monotonicity. A similar reasoning underlies generalized EM (GEM) algorithms [12]. We thus adopt the term *generalized MM* (GMM) algorithms.

2.3. Majorizing the TV Penalty

We now derive a quadratic majorizer for λTV_d . The choice of a quadratic upper bound is motivated by the fact that if that bound is added to the data term $\|\mathbf{x} - \mathbf{y}\|^2$, the resulting function will be an upper bound for the complete objective function (see Property 2, in Subsection 2.2). Moreover, this upper bound will be quadratic, its minimization leading to a linear system of equations.

The square root function, for non-negative arguments, is strictly concave, thus upper-bounded by any of its tangents, i.e., for any $a \geq 0$ and $a' \geq 0$,

$$\sqrt{a} \leq \sqrt{a'} + \frac{a - a'}{2\sqrt{a'}}, \quad (11)$$

with equality if and only if $a = a'$. It follows that the function $Q_{\text{TV}}(\cdot, \cdot)$ defined as

$$\begin{aligned} Q_{\text{TV}}(\mathbf{x}, \mathbf{x}') &= \text{TV}_d(\mathbf{x}') + \frac{\lambda}{2} \sum_i \frac{[(\Delta_i^h \mathbf{x})^2 - (\Delta_i^h \mathbf{x}')^2]}{\sqrt{(\Delta_i^h \mathbf{x}')^2 + (\Delta_i^v \mathbf{x}')^2}} \\ &\quad + \frac{\lambda}{2} \sum_i \frac{[(\Delta_i^v \mathbf{x})^2 - (\Delta_i^v \mathbf{x}')^2]}{\sqrt{(\Delta_i^h \mathbf{x}')^2 + (\Delta_i^v \mathbf{x}')^2}} \end{aligned} \quad (12)$$

satisfies $Q_{\text{TV}}(\mathbf{x}, \mathbf{x}') \geq \text{TV}_d(\mathbf{x})$ for any \mathbf{x}, \mathbf{x}' , with equality for $\mathbf{x} = \mathbf{x}'$. Function $Q_{\text{TV}}(\mathbf{x}, \mathbf{x}^{(t)})$ is thus a quadratic upper bound function for $\text{TV}_d(\mathbf{x})$. Finally, notice that the terms $(\Delta_i^h \mathbf{x}')^2$ and $(\Delta_i^v \mathbf{x}')^2$ in the numerators are simply additive constants which can be ignored as they do not affect the resulting MM algorithm (see Property 1, in Subsection 2.2).

Let \mathbf{D}^h and \mathbf{D}^v denote matrices such that $\mathbf{D}^h \mathbf{x}$ and $\mathbf{D}^v \mathbf{x}$ are the vectors of all horizontal and vertical (respectively) first order differences. Define also the vector $\mathbf{w}^{(t)}$ whose i -th element is

$$w_i^{(t)} = \lambda \left(2 \sqrt{(\Delta_i^h \mathbf{x}^{(t)})^2 + (\Delta_i^v \mathbf{x}^{(t)})^2} \right)^{-1}, \quad (13)$$

the diagonal matrix

$$\mathbf{\Lambda}^{(t)} = \text{diag}(\mathbf{w}^{(t)}, \mathbf{w}^{(t)}), \quad (14)$$

and the matrix $\mathbf{D} = [(\mathbf{D}^h)^T (\mathbf{D}^v)^T]^T$. With these definitions in place, $Q_{\text{TV}}(\mathbf{x}, \mathbf{x}^{(t)})$ can be written as a quadratic form

$$Q_{\text{TV}}(\mathbf{x}, \mathbf{x}^{(t)}) = \mathbf{x}^T \mathbf{D}^T \mathbf{\Lambda}^{(t)} \mathbf{D} \mathbf{x} + K(\mathbf{x}^{(t)}), \quad (15)$$

where $K(\mathbf{x}^{(t)})$ is a constant independent of \mathbf{x} , thus irrelevant for the MM algorithm. Finally, adding $Q_{\text{TV}}(\mathbf{x}, \mathbf{x}^{(t)})$ to the data term $\|\mathbf{x} - \mathbf{y}\|^2$, we obtain the complete quadratic upper bound (after dropping additive constants),

$$Q(\mathbf{x}, \mathbf{x}^{(t)}) = \mathbf{x}^T (\mathbf{D}^T \mathbf{\Lambda}^{(t)} \mathbf{D} + \mathbf{I}) \mathbf{x} - 2 \mathbf{x}^T \mathbf{y}. \quad (16)$$

Since this is a quadratic function, minimization w.r.t. \mathbf{x} leads to

$$\hat{\mathbf{x}}^{(t+1)} = \underset{\mathbf{x}}{\text{solution}} \left\{ (\mathbf{D}^T \mathbf{\Lambda}^{(t)} \mathbf{D} + \mathbf{I}) \mathbf{x} = \mathbf{y} \right\}. \quad (17)$$

Of course, this system can not be solved analytically, due to its huge dimension: $MN \times MN$, for $M \times N$ images.

2.4. Solving the Update Equation

In addition to its size, (17) has an additional difficulty: when one (or more) term(s) $(\Delta_i^h \mathbf{x}^{(t)})^2 + (\Delta_i^v \mathbf{x}^{(t)})^2$ goes to zero (which is very likely to happen under the TV penalty), the corresponding element(s) of vector $\mathbf{w}^{(t)}$ goes to infinity and so do some elements of the system matrix $(\mathbf{D}^T \mathbf{\Lambda}^{(t)} \mathbf{D} + \mathbf{I})$. In other words, there is no numerically stable way to handle this matrix when some elements of $\mathbf{\Lambda}^{(t)}$ go to infinity. We sidestep this difficulty by invoking the well known matrix inversion lemma,

$$\mathbf{D}^T \mathbf{\Lambda}^{(t)} \mathbf{D} + \mathbf{I} = \left[\mathbf{I} - \mathbf{D}^T (\mathbf{D} \mathbf{D}^T + (\mathbf{\Lambda}^{(t)})^{-1})^{-1} \mathbf{D} \right]^{-1}$$

which leads to

$$\hat{\mathbf{x}}^{(t+1)} = \mathbf{y} - \mathbf{D}^T \mathbf{z}^{(t)} \quad (18)$$

where

$$\mathbf{z}^{(t)} = \underset{\mathbf{z}}{\text{solution}} \left\{ [\mathbf{D} \mathbf{D}^T + (\mathbf{\Lambda}^{(t)})^{-1}] \mathbf{z} = \mathbf{D} \mathbf{y} \right\}. \quad (19)$$

Notice that now, the matrix of the system to be solved involves $(\mathbf{\Lambda}^{(t)})^{-1}$ instead of $\mathbf{\Lambda}^{(t)}$; the elements of $(\mathbf{\Lambda}^{(t)})^{-1}$ go to zero, instead of infinity, when pairs of local differences go to zero. We thus have a matrix which can be handled in a numerically stable way. Notice also that we never have to explicitly invert any matrix, simply solve the corresponding system of equations. To deal with (19), we adopt the standard *conjugate gradient* (CG) algorithm

[9]. Invoking Property 3 of MM algorithms (see Subsection 2.2), we obtain a GMM algorithm by running just a few CG iterations.

In summary, the proposed algorithm for TV denoising is composed of two alternating steps. Starting with some initial estimate $\hat{\mathbf{x}}^{(1)}$, the algorithm proceeds by cyclically repeating the following steps until some stopping criterion is met:

Step 1: compute matrix $(\mathbf{\Lambda}^{(t)})^{-1}$ using (13)-(14);

Step 2: update the estimate by applying (18)-(19).

Let us make some comments about implementation of this algorithm. Computing $(\mathbf{\Lambda}^{(t)})^{-1}$ is simpler than computing $\mathbf{\Lambda}^{(t)}$ itself, as is obvious from its definition in (13)-(14). The vector $\mathbf{D} \mathbf{y}$ in (19) can be pre-computed, stored, and used during the algorithm. The only operation involving $[\mathbf{D} \mathbf{D}^T + (\mathbf{\Lambda}^{(t)})^{-1}]$ required by the CG algorithm are matrix-vector products; these products can be very efficiently computed (for an arbitrary \mathbf{v}) as

$$[\mathbf{D} \mathbf{D}^T + (\mathbf{\Lambda}^{(t)})^{-1}] \mathbf{v} = (\mathbf{D} \mathbf{D}^T) \mathbf{v} + (\mathbf{\Lambda}^{(t)})^{-1} \mathbf{v},$$

where $(\mathbf{\Lambda}^{(t)})^{-1} \mathbf{v}$ is just the product of a diagonal matrix by a vector and matrix $(\mathbf{D} \mathbf{D}^T)$ can be pre-computed. Finally, computation of $(\mathbf{D} \mathbf{D}^T) \mathbf{v}$, when \mathbf{v} is some image, corresponds to the convolution of \mathbf{v} (in its 2D form) by a small local kernel, which can be carried out much more efficiently (with linear cost) than the explicit matrix-vector multiplication (which has quadratic cost). The same thing is true for the product by \mathbf{D}^T in (18).

The results presented in the next section were obtained with a MATLAB implementation running on a 3GHz Pentium PC. The CG algorithm for (19) is stopped when the relative decrease of the objective function falls below 10^{-5} ; this always happened (in all the cases considered) after no more than about 15 CG iterations, sometimes much less than that; each CG iteration takes about 0.5 seconds for 512×512 pixels images.

3. EXPERIMENTS

3.1. Illustrative Examples

In this subsection, we show an example with the single purpose of illustrating the proposed algorithm. Figure 1 shows a noisy Lena image (with $\sigma = 15$) and the TV denoised image produced by the proposed algorithm; still in Figure 1, the evolution of the objective function and the PSNR are plotted, showing that the MM algorithm practically converges after 6 iterations.

3.2. Comparison with Wavelet-Based Methods

Comparing TV with wavelet denoising is delicate, as there are free parameters of the algorithms which can affect their performance. Namely, the choice of λ is critical; we have verified experimentally that $\lambda = \sqrt{3}\sigma$ is a good general purpose choice for a large range of values of σ (see [5] for an empirical-Bayes justification of this threshold value in wavelet-based denoising context).

Table 1 shows the results of the four methods: TV denoising, classical wavelet-based soft thresholding (SF), the methods proposed in [5], [7] (see those papers for details). For SF, the threshold value was obtained in a clairvoyant way (using the original image to obtain the best PSNR); although this gives an unfair advantage over the other methods, the result of SF are always the worst. It is remarkable that TV, despite the simplicity of the underlying criterion, can perform competitively with these wavelet-based methods

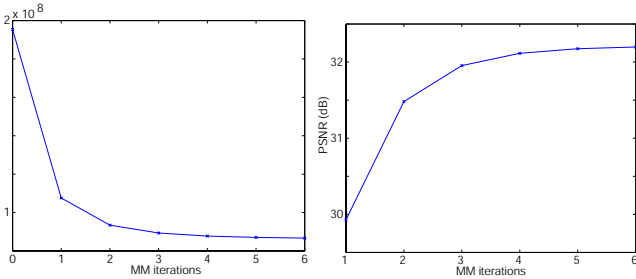


Fig. 1. Upper row: noisy ($\sigma = 15$) and TV restored images. Bottom row: evolution of the TV objective function (left) and the PSNR (right).

for the Lena image. For the Barbara image, the presence of strong texture makes the TV criterion less competitive.

Table 2 shows the results of TV and the method from [10]; in these results, TV is outperformed, at all noise levels and for all images, by this state of the art method, which uses explicit models of inter-coefficient dependencies and sophisticated wavelet representations (the steerable pyramid). Given the difference in complexity of the TV penalty and the model used in [10], this is not surprising.

Table 1. PSNR (dB) results for the methods mentioned in the text; [7]-(a) and [7]-(b) refer to two different methods proposed in [7].

Lena				
	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$
TV	33.99	32.20	30.93	29.93
soft rule	32.00	29.57	28.04	26.66
[5]	32.74	30.48	28.74	27.38
[7] (a)	33.72	31.37	29.63	28.22
[7] (b)	34.25	32.33	31.00	29.96
Barbara				
	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$	$\sigma = 25$
TV	30.56	28.25	26.79	25.73
soft rule	30.37	27.67	25.91	24.68
[5]	31.28	28.79	27.03	25.71
[7] (a)	32.32	29.72	27.93	26.53
[7] (b)	32.46	30.03	28.39	27.21

4. CONCLUSIONS

We have introduced a new MM algorithm to implement a discrete version of the total variation denoising criterion. We have also carried out a detailed experimental comparison of TV versus wavelet

Table 2. PSNR (dB) results for the wavelet-based method from [10] and for TV denoising.

σ	Lena		Barbara		Boats	
	[10]	TV	[10]	TV	[10]	TV
10	35.61	33.99	34.03	30.56	33.58	31.94
15	33.90	32.20	31.86	28.25	31.70	30.17
20	32.66	30.93	30.32	26.79	30.38	28.97
25	31.69	29.93	29.13	25.73	29.37	28.07
50	28.61	27.17	25.48	23.28	26.38	25.42
75	26.84	25.41	23.65	22.29	24.79	23.90
100	25.64	24.08	22.61	21.52	23.75	22.78

denoising. The main conclusions were that TV is outperformed by recent state of the art wavelet denoising methods, but performs competitively with older wavelet methods.

5. REFERENCES

- [1] J. Bioucas-Dias, M. Figueiredo, J. Oliveira, “Total variation image deconvolution: A majorization-minimization approach”, *IEEE ICASSP*, Toulouse, France, 2006 (to appear).
- [2] A. Chambolle, “An algorithm for total variation minimization and applications”, *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89-97, 2004.
- [3] T. Chan, S. Esedoglu, F. Park, M. Yip. “Recent developments in total variation image restoration.” In *Mathematical Models of Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Editors, Springer Verlag, 2005.
- [4] D. Donoho, I. Johnstone. “Ideal adaptation via wavelet shrinkage.” *Biometrika*, vol. 81, pp. 425–455, 1994.
- [5] M. Figueiredo, R. Nowak. “Wavelet-based image estimation: an empirical Bayes approach using Jeffreys’ noninformative prior,” *IEEE Tr. Image Proc.*, vol. 10, pp. 1322–1331, 2001.
- [6] D. Hunter, K. Lange. “A Tutorial on MM algorithms.” *The American Statistician*, vol. 58, pp. 30–37, 2004.
- [7] M. Miħak, I. Kozintsev, K. Ramchandran, P. Moulin. “Low-complexity image denoising based on statistical modeling of wavelet coefficients,” *IEEE Signal Proc. Letters*, vol. 6, pp. 300-303, 1999.
- [8] P. Moulin, J. Liu. “Analysis of multiresolution image denoising schemes using generalized-Gaussian and complexity priors,” *IEEE Tr. Inform. Theory*, vol. 45, pp. 909–919, 1999.
- [9] J. Nocedal, S. Wright. *Numerical Optimization*, Springer Verlag, 1999.
- [10] J. Portilla, V. Strela, M. Wainwright, E. Simoncelli. “Image denoising using scale mixtures of Gaussians in the wavelet domain.” *IEEE Tr. Image Proc.*, vol. 12, pp. 1338-1351, 2003.
- [11] L. Rudin, S. Osher, E. Fatemi. “Nonlinear total variation based noise removal algorithms.” *Physica D*, vol. 60, pp. 259-268, 1992.
- [12] C. Wu, “On the convergence properties of the EM algorithm,” *The Annals of Statistics*, vol. 11, pp. 95-103, 1983.