Image Denoising with Shrinkage and Redundant Representations

Michael Elad
Department of Computer Science
The Technion - Israel Institute of Technology
Haifa 32000 Israel
elad@cs.technion.ac.il

Boaz Matalon
Department of Electrical Engineering
mboaz17@techunix.technion.ac.il

Michael Zibulevsky
Department of Electrical Engineering
mzib@ee.technion.ac.il

Abstract

Shrinkage is a well known and appealing denoising technique. The use of shrinkage is known to be optimal for Gaussian white noise, provided that the sparsity on the signal's representation is enforced using a unitary transform. Still, shrinkage is also practiced successfully with non-unitary, and even redundant representations. In this paper we shed some light on this behavior. We show that simple shrinkage could be interpreted as the first iteration of an algorithm that solves the basis pursuit denoising (BPDN) problem. Thus, this work leads to a novel iterative shrinkage algorithm that can be considered as an effective pursuit method. We demonstrate this algorithm, both on synthetic data, and for the image denoising problem, where we learn the image prior parameters directly from the given image. The results in both cases are superior to several popular alternatives.

1 Introduction

One way to pose the maximum a-posteriori probability (MAP) estimator for the denoising problem is the minimization of the function

$$f(\mathbf{x}) = \frac{1}{2} \cdot \|\mathbf{x}-\mathbf{y}\|_2^2 + \lambda \cdot \mathbf{1}^T \cdot \rho\{\mathbf{T}\mathbf{x}\} \ . \qquad (1)$$

The first term is known as the log-likelihood, describing the relation between the desired (clean) signal, $\mathbf{x} \in \mathbb{R}^N$, and a noisy version of it, $\mathbf{y} \in \mathbb{R}^N$. We assume the model $\mathbf{y} = \mathbf{x} + \mathbf{v}$, with $\mathbf{v} \in \mathbb{R}^N$ a Gaussian zero mean white noise. The term $\mathbf{1}^T \cdot \rho\{\mathbf{T}\mathbf{x}\}$ stands for the prior posed on the unknown signal $\mathbf{x}$, based on sparsity of the unknown signal with respect to its transformed ($\mathbf{T}$) representation. The function ρ is a scalar robust measure (e.g., $\rho(z) = |z|$), and when operating on a vector, it does so entry-wise. The multiplication by $\mathbf{1}^T$ sums those robust measures.

Donoho and Johnstone pioneered a wavelet based signal denoising algorithm in line with the above structure. They advocated the use of sparsity of the wavelet coefficients $\mathbf{W}\mathbf{x}$ (i.e., here $\mathbf{T}$ is the unitary matrix $\mathbf{W}$) as a driving force in recovering the desired signal [1, 2]. Later work in [3, 4, 5] simplified these ideas and related them to the MAP formulation as presented above. Interestingly, using such a prior in Equation (1) leads to a *simple closed-form solution, known as shrinkage*. This solution amounts to a wavelet transform on the noisy signal, a look-up-table (LUT) function on the coefficients (that depends on the function ρ), $\mathcal{S}\{\mathbf{W}\mathbf{y}\}$, and an inverse wavelet transform to produce the outcome $\hat{\mathbf{x}}$. The LUT operation promotes sparsity by nulling small coefficients to zero, which explains the name shrinkage. This optimality depends strongly on the ℓ^2-norm used in evaluating the distance $\mathbf{x}-\mathbf{y}$, and this has direct roots in the white Gaussianity assumptions on the noise. Also, crucial to the optimality of this method is the orthogonality of $\mathbf{W}$.

A new trend of recent years is the use of overcomplete transforms, replacing the traditional unitary ones – see [6, 7, 8, 9, 11, 12] for representative works. This trend was partly motivated by the growing realization that orthogonal wavelets are weak in describing the singularities found in images. Another driving force in the introduction of redundant representations is the sparsity they can provide, which many applications find desirable [22]. Finally, we should mention the desire to obtain shift-invariant transforms, again calling for redundancy in the representation. In these methods the transform is defined via a non-square full rank matrix $\mathbf{T} \in \mathbb{R}^{L \times N}$, with $L > N$. Such redundant methods, like the un-decimated wavelet transform, curvelet, contourlet, and steerable-wavelet, were shown to be more

effective in representing images, and other signal types.

Given a noisy signal $\mathbf{y}$, one can still follow the shrinkage procedure, by computing the forward transform $\mathbf{Ty}$, putting the coefficients through a shrinkage LUT operation $\mathcal{S}\{\mathbf{Ty}\}$, and finally applying the inverse transform to obtain the denoised outcome, $\mathbf{T}^{+}\mathcal{S}\{\mathbf{Ty}\}$. Will this be the solution of (1)? The answer is no! As we have said before, the orthogonality of the transform plays a crucial role in the construction of the shrinkage as an optimal procedure. Still, shrinkage is practiced quite often with non-unitary, and even redundant representations, typically leading to satisfactory results – see [6, 7, 8] for representative examples. Naturally, we should wonder why this is so.

In this paper we shed some light on this behavior. Our main argument is that such a shrinkage could be interpreted as the first iteration of a converging algorithm that solves the basis pursuit denoising (BPDN) problem [22]. The BPDN forms a similar problem to the one posed in (1), replacing the analysis prior with a generative one. While the desired solution of BPDN is hard to obtain in general, a simple iterative procedure that amounts to step-wise shrinkage can be employed with quite successful performance. Thus, beyond showing that shrinkage has justified roots in solid denoising methodology, we also show how shrinkage can be iterated in a simple form, to further strengthen the denoising effect. As a byproduct, we get an effective pursuit algorithm that minimizes the BPDN functional via simple steps.

In the next section we bridge between an analysis based objective function and a synthesis one, leading to the BPDN. Section 3 then develops the iterated shrinkage algorithm that minimizes it. In Section 4 we present few simulations to illustrate the algorithm proposed on both synthetic and image data.

2 From Analysis to Synthesis-Based Prior

Starting with the penalty function posed in (1), we define $\mathbf{x}_T = \mathbf{Tx}$. Multiplying both sides by $\mathbf{T}^T$, and using the fact that $\mathbf{T}$ is full-rank, we get[1] $\mathbf{x} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{x}_T = \mathbf{T}^{+}\mathbf{x}_T$. Using these relations to rearrange Equation (1), we obtain a new function of the representation vector $\mathbf{x}_T$,

$$\tilde{f}(\mathbf{x}_T) = \frac{1}{2}\cdot\|\mathbf{D}\mathbf{x}_T - \mathbf{y}\|_2^2 + \lambda\cdot\mathbf{1}^T\cdot\rho\{\mathbf{x}_T\}, \qquad (2)$$

where we have defined $\mathbf{D} = \mathbf{T}^{+}$.

Denoising can be done by minimizing f and obtaining a solution $\hat{\mathbf{x}}_1$. Alternatively, we can minimize $\tilde{f}$ with respect to $\mathbf{x}_T$ and deduce the denoised outcome by $\hat{\mathbf{x}}_2 = \mathbf{D}\hat{\mathbf{x}}_T$. Interestingly, *these two results are not expected to be the same in the general case*, since in the conversion from f to $\tilde{f}$ we have expanded the set of feasible solutions by allowing $\mathbf{x}_T$ to be an arbitrary vector in $\mathbb{R}^L$, whereas the original definition $\mathbf{x}_T = \mathbf{Tx}$ implies that it must be confined to the column space of $\mathbf{T}$. Notice that this difference between the two formulations disappears when $\mathbf{T}$ is full rank square matrix, which explains why this dichotomy of methods does not bother us for the regular unitary wavelet transform.

Still, the formulation posed in (2) is a feasible alternative Bayesian method that uses a generative prior. Indeed, for the choice $\rho\{z\} = |z|$, this formulation is known as the basis pursuit denoising (BPDN) [22]. Referring to $\mathbf{D}$ as a dictionary of signal prototypes (atoms) being its columns, we assume that the desired signal $\mathbf{x}$ is a linear construction of these atoms, with coefficients drawn independently from a probability density function proportional to $\exp\{-\text{Const}\cdot\rho\{x_T(j)\}\}$. In the case of $\rho(z) = |z|$ this is the Laplace distribution, and we effectively promote sparsity in the representation.

3 Proposed Algorithm

3.1 A sequential approach

We desire the minimization of (2). Assume that in an iterative process used to solve the above problem, we hold the k-th solution $\hat{\mathbf{z}}_k$. We are interested in updating its j-th entry, $z(j)$, assuming all the others as fixed. Thus, we obtain a one-dimensional optimization problem of the form

$$\min_w \frac{1}{2}\cdot\|[\mathbf{D}\mathbf{z}_k - \mathbf{d}_j z_k(j)] + \mathbf{d}_j w - \mathbf{y}\|_2^2 + \lambda\cdot\rho\{w\}. \quad (3)$$

In the above expression, $\mathbf{d}_j$ is the j-th column in $\mathbf{D}$. The term $\mathbf{D}\mathbf{z}_k - \mathbf{d}_j z_k(j)$ uses the current solution for all the coefficients, but discards of the j-th one, assumed to be replaced with a new value, w.

Since this is a 1D optimization task, it is relatively easy to solve. If $\rho(w) = |w|$, the derivative is

$$0 = \mathbf{d}_j^T(\mathbf{D}\mathbf{z}_k - \mathbf{d}_j z_k(j) + \mathbf{d}_j w - \mathbf{y}) + \lambda\cdot\text{sign}\{w\}, \quad (4)$$

leading to

$$\begin{aligned} w &= z_k(j) + \frac{\mathbf{d}_j^T(\mathbf{y} - \mathbf{D}\mathbf{z}_k)}{\|\mathbf{d}_j\|_2^2} - \frac{\lambda\cdot\text{sign}\{w\}}{\|\mathbf{d}_j\|_2^2} \quad (5)\\ &= v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j) - \hat{\lambda}(j)\cdot\text{sign}\{w\}. \end{aligned}$$

Here we have defined

$$\begin{aligned} v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j) &= \frac{\mathbf{d}_j^T(\mathbf{y} - \mathbf{D}\mathbf{z}_k)}{\|\mathbf{d}_j\|_2^2} + z_k(j) \text{ and } \quad (6)\\ \hat{\lambda}(j) &= \frac{\lambda}{\|\mathbf{d}_j\|_2^2}. \end{aligned}$$

Both $v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j)$ and $\hat{\lambda}(j)$ are computable using the known ingredients. Similarly to [2] for example, this leads

[1] If $\mathbf{T}$ is a tight frame ($\alpha\mathbf{T}^T\mathbf{T} = \mathbf{I}$), then $\mathbf{x} = \alpha\mathbf{T}^T\mathbf{x}_T$.

to a closed form formula for the optimal solution for w, being a shrinkage operation on $v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j)$,

$$w_{opt} = \mathcal{S}\left\{v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j)\right\} \quad (7)$$
$$= \begin{cases} v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j) - \hat{\lambda}(j) & \text{for } v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j) > \hat{\lambda}(j) \\ 0 & \text{for } |v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j)| \leq \hat{\lambda}(j) \\ v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j) + \hat{\lambda}(j) & \text{for } v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j) < -\hat{\lambda}(j) \end{cases}.$$

A similar LUT result can be developed for any many other choices of the function $\rho(\cdot)$.

It is tempting to suggest an algorithm that applies the above procedure for $j = 1, 2, \ldots, , L$, updating one coefficient at a time in a sequential coordinate descent algorithm, and cycle such process several times. While such algorithm necessarily converges, and could be effective in minimizing the objective function using scalar shrinkage operations only, it is impractical in most cases. The reason is the necessity to draw one column at a time from $\mathbf{D}$ to perform this computation. Consider, for example, the curvelet dictionary. While the transform and its inverse can be interpreted as multiplications by the dictionary and its transpose (because it is a tight frame), this matrix is never explicitly constructed, and an attempt to draw basis functions from it or store them could be devastating. Thus we take a different route.

3.2 A parallel approach

Given the current solution $\mathbf{z}_k$, let us assume that we use the above update formulation to update *all the coefficients* in parallel, rather than doing this sequentially. Obviously, this process must be slower in minimizing the objective function, but with this slowness comes a blessed simplicity that will be evident shortly.

First, let us convert the terms $v(\mathbf{D},\mathbf{y},\mathbf{z}_k,j)$ in Equation (6) to a vector form that accounts for all the updates at once. Gathering these terms for all $j \in [1, L]$, this reads

$$\mathbf{v}(\mathbf{D},\mathbf{y},\mathbf{z}_k) = \text{diag}^{-1}\left\{\mathbf{D}^T\mathbf{D}\right\}\mathbf{D}^T(\mathbf{y} - \mathbf{D}\mathbf{z}_k) + \mathbf{z}_k. \quad (8)$$

If the transform we use is such that multiplication by $\mathbf{D}$ and its adjoint are fast, then computing the above term is easy and efficient. Notice that here we do not need to extract some columns from the dictionary, and need not use these matrices explicitly in any other way. The normalization by the norms of the columns is simple to obtain and can be kept as fixed parameters of the transform, computed once offline. In the case of tight frames, applying multiplications by $\mathbf{D}^T$ and $\mathbf{D}$ are the forward and the inverse transforms, up to a constant. For a non-tight frame, the above formula says that we need to be able to apply the adjoint *and not the pseudo-inverse* of $\mathbf{D}$.

There is also a natural weakness to the above strategy. One cannot take a shrinkage of the above vector with respect to the threshold vector $\lambda \cdot \text{diag}^{-1}\left\{\mathbf{D}^T\mathbf{D}\right\} \cdot \mathbf{1}$, and expect the objective function to be minimized well. While updating every scalar entry w_j using the above shrinkage formula is necessarily decreasing the function's value, applying all those at once is likely to diverge, and cause an ascent in the objective. Thus, instead of applying a complete shrinkage as Equation (7) suggests, we consider a relaxed step of the form

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \mu\left[\mathcal{S}\left\{\mathbf{v}(\mathbf{D},\mathbf{y},\mathbf{z}_k)\right\} - \mathbf{z}_k\right] = \mathbf{z}_k + \mu\mathbf{h}_k. \quad (9)$$

This way, we compute the shrinkage vector as the formula suggests, and use it to define a descent direction. The solution is starting from the current solution $\mathbf{z}_k$ and updates it by "walking" towards the shrinkage result. For a sufficiently small $\mu > 0$, this step *must* lead to a feasible descent in the penalty function, because this direction is a non-negative combination of L descent directions.

We can apply a line search to find the proper choice for the value of μ. In general, a line search seeks the best step-size as a 1D optimization procedure that solves

$$\min_{\mu} \frac{1}{2} \cdot \|\mathbf{D}\left[\mathbf{z}_k + \mu\mathbf{h}_k\right] - \mathbf{y}\|_2^2 + \lambda \cdot \mathbf{1}^T \cdot \rho\left\{\mathbf{z}_k + \mu\mathbf{h}_k\right\}, \quad (10)$$

where $\mathbf{h}_k$ is a computable vector. Following the previous reasoning, the solution in this case is also given as a shrinkage-like procedure.

As a final step, we consider the first iteration, and assume that the algorithm is initialized with $\mathbf{z}_0 = \mathbf{0}$. Thus, the term in Equation (8) becomes

$$\mathbf{v}(\mathbf{D},\mathbf{y},\mathbf{0}) = \text{diag}^{-1}\left\{\mathbf{D}^T\mathbf{D}\right\}\mathbf{D}^T\mathbf{y}. \quad (11)$$

The solution $\mathbf{z}_1$ is obtained by first applying shrinkage to the above vector, using $\lambda\text{diag}^{-1}\left\{\mathbf{D}^T\mathbf{D}\right\}\mathbf{1}$ as the threshold vector, and then relaxing it, as in Equation (9). The denoised outcome is thus

$$\mathbf{D}\mathbf{z}_1 = \mu\mathbf{D}\mathcal{S}\left\{\text{diag}^{-1}\left\{\mathbf{D}^T\mathbf{D}\right\}\mathbf{D}^T\mathbf{y}\right\}, \quad (12)$$

and the resemblance to the heuristic shrinkage is evident. In fact, for tight frames with normalized columns the above becomes exactly equal to the heuristic shrinkage.

4 Experimental Results

4.1 Synthetic data

We start with simple synthetic experiments that correspond to the case of a tight frame with normalized columns. We build $\mathbf{D}$ as a union of 10 random unitary matrices of size 100×100 each. We synthesize a sparse representation $\mathbf{z}_0$ with 15 non-zeros in random locations and Gaussian i.i.d.

entries, so as to match the sparsity prior we use. The clean signal is defined as $\mathbf{x}_0 = \mathbf{D}\mathbf{z}_0$, and it is contaminated by a Gaussian i.i.d noise $\sigma = 0.3$ (parallels an SNR of ≈ 1.3dB).

We consider several denoising algorithms: (A) a heuristic shrinkage as described in the introduction; (B) the IRLS algorithm, known to be computationally heavy, but for low dimensions it is an effective algorithm, and thus good as a reference [13]; (C) the sequential shrinkage algorithm developed above; and (D) the parallel counterpart. We assume $\rho(z) = |z|$, and the results are reported in Figures 1- 3.

First, we show how effective are these algorithms (B-D) in minimizing the objective in Equation (2). Figure 1 presents the value of the objective as a function of the iteration number. Here we have implemented the parallel shrinkage algorithm both with a fixed $\mu = 1/\alpha$ [2] and with a line-search. As expected, the IRLS performs the best in terms of convergence speed. The sequential and the parallel (with line-search) coordinate descent are comparable to each other, being somewhat inferior to the IRLS.

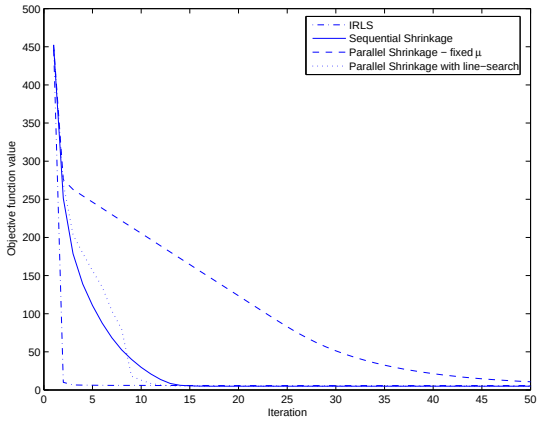

Figure 1. The objective as a function of the iteration – algorithms B-D.

When implementing these algorithms for denoising, we sweep through the possible values of λ to find the best choice. In assessing the denoising effect, we use the measure $r(\hat{\mathbf{x}}, \mathbf{x}_0, \mathbf{y}) = \|\hat{\mathbf{x}} - \mathbf{x}_0\|_2^2 / \|\mathbf{y} - \mathbf{x}_0\|_2^2$, which gives the ratio between the final reconstruction error and the noise power. Thus, a value smaller than 1 implies a decay in the noise, and the closer it is to zero the better the result.

We compare the IRLS results (after the 1-st and the 5-th iterations) to the simple shrinkage algorithm. The simple shrinkage in this case uses a threshold being $\lambda/\alpha = 10\lambda$, so as to match to the objective function that uses λ in its formulation. Figure 2 presents this comparison, showing the noise decay factor versus λ. Interestingly, it appears than the simple shrinkage manages to utilize most of the denoising potential, and 5 iterations of the IRLS give only slightly better results.

[2] See its definition at the beginning of Section 2.

Figure 3 presents a similar comparison of the simple shrinkage with the parallel shrinkage with a fixed μ chosen as $\mu = 1/\alpha$. We see that the first iteration of the parallel shrinkage aligns perfectly with the simple shrinkage when $\mu = 1/\alpha$, as predicted, and having 5 iterations gives a slight improvement. Other experiments with the other algorithms were done with similar results, and are omitted here due to space constraints.

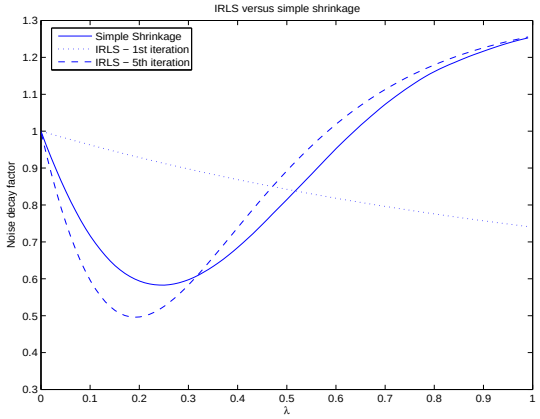

Figure 2. The denoising effect of the IRLS versus simple shrinkage.

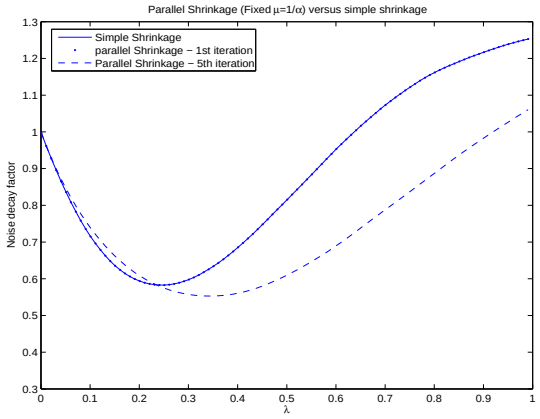

Figure 3. The denoising effect of the parallel coordinate descent algorithm versus simple shrinkage.

4.2 Image denoising

The BPDN formulation in Equation (2) can be used for removing noise from images. We use the Contourlet Transform [8] (CT), which is one of several transforms developed in recent years, aimed at improving the representation sparsity of images over the Wavelet Transform (WT). The main feature of these transforms is the potential to efficiently handle 2-D singularities, i.e. edges, unlike wavelets which can

deal with point singularities exclusively. A newer version of the CT, allowing better performance, was recently developed in [10], and was thus employed throughout our simulations (where the inverse transform operator is used as the dictionary $\mathbf{D}$).

Experiments made on natural images show that the Contourlet coefficients at different scales and directions have different average variance. Hence the variance σ_i^2 of each coefficient should depend on the scale and direction, and perhaps on the spatial position as well. This observation justifies a modification in the BPDN formulation, such that each coefficient $z(i)$ is normalized by σ_i, yielding

$$\tilde{f}(\mathbf{z}) = \frac{1}{2} \cdot \|\mathbf{D}\mathbf{z} - \mathbf{y}\|_2^2 + \lambda \cdot \sum_i |z(i)/\sigma_i|, \qquad (13)$$

after replacing $\mathbf{x}_T$ by $\mathbf{z}$ and assuming $\rho(z) = |z|$.

The implementation of this algorithm requires learning of the image prior parameters directly from the given image, i.e. estimation of the variances $\{\sigma_i^2\}$. We employ here a method introduced by Chang *et al.* [14] for the WT, though it remains valid for any multiscale transform like the CT. To explain the idea behind this method, first take a look at Figure 4, showing the distribution of a coefficient conditioned on its neighbor's value. According to this figure, which resembles a bow-tie shape, a coefficients' standard deviation scales roughly linearly with its neighbor's absolute value. Consequently, all of the coefficients whose neighbors have roughly the same absolute value, may be attributed with the same variance, which is then estimated.

In detail, consider a subband with M coefficients, and denote $\bar{\mathbf{z}}_i$ as a $p \times 1$ vector containing the *absolute values* of p neighbors of $z(i)$, e.g. its eight nearest spatial neighbors and one parent. The *context* of $z(i)$ is defined as a weighted average of its neighbors' absolute values, $c(i) = \mathbf{w}^t \bar{\mathbf{z}}_i$.The weights vector $\mathbf{w}$ is calculated by the least squares (LS) estimate over the whole subband, i.e.

$$\mathbf{w}_{LS} = (Z^t Z)^{-1} Z^t |\mathbf{z}_{sb}|, \qquad (14)$$

where Z is a $M \times p$ matrix with rows $\{\bar{\mathbf{z}}_i\}$, and $\mathbf{z}_{sb}$ is a $M \times 1$ vector of the subband's coefficients.

Following the context calculation, a coefficients' variance is estimated based on all of the coefficients in the subband with similar context, just as we have explained earlier. More precisely, the contexts $\{c(j)\}$ in each subband are sorted in an increasing order, and the coefficients $\{z(j)\}$ whose context are at most L values away from $c(i)$ are chosen (i.e. $2L+1$ coefficients). The variance estimate of $z(i)$ is then given by

$$\hat{\sigma}_i^2 = \max\Big\{\frac{1}{2L+1}\sum_{j(i)} z^2(j) - \sigma_{n,i}^2 \quad , 0\Big\}, \qquad (15)$$

where $\sigma_{n,i}^2$ is the noise variance at the i-th coefficient (it is in fact constant over a subband). This is a simple heuristic variance estimator, compensating for the additive Gaussian noise and imposing nonnegativity. In practice, we take some small $\varepsilon > 0$ instead of 0 in the above formula, to prevent zero-division in the objective function (13). Similarly to [14], we choose $L = \max\{100, 0.02M\}$ to guarantee a reliable estimation, along with adaptivity to varying characteristics, as well as $p = 9$ (eight spatial neighbors and one parent in the immediate coarser scale).

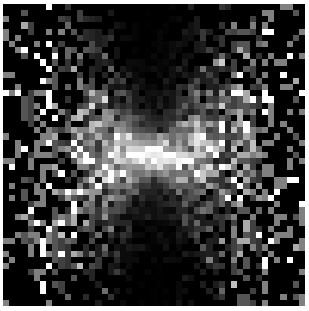

Figure 4. Distribution of a coefficient (vertical) conditioned on its neighbor's value (horizontal), estimated from one CT subband of *Peppers* (each column has been separately rescaled to fit the display range).

To choose the parameter λ, recall that Equation (13) corresponds to the MAP estimator, assuming an independent Laplacian prior model, i.e. $p(z(i)) \propto \exp(-\frac{\sqrt{2}}{\sigma_i}|z(i)|)$. This yields the objective function

$$\tilde{f}(\mathbf{z}) = \frac{1}{2\sigma_n^2} \cdot \|\mathbf{D}\mathbf{z} - \mathbf{y}\|_2^2 + \sqrt{2} \cdot \sum_i |z(i)/\sigma_i|, \qquad (16)$$

where σ_n^2 is the noise variance at the image domain. By comparing (13) and (16), the a-priori value of λ should be $\lambda_0 = \sqrt{2}\sigma_n^2$. This value turned out to be indeed the optimal one performance-wise. Notice that our iterative-shrinkage algorithm should be modified slightly to account for the inclusion of $\{\lambda_j\}$. This is done by simply replacing λ in Equation (4) with λ_j, and continuing accordingly.

To evaluate the performance of our algorithm in minimizing the function in (13), we have compared several algorithms (see [15]): (i) Steepest Descent; (ii) Conjugate Gradients; (iii) Truncated Newton; and (iv) our parallel iterative-shrinkage algorithm. All of the algorithms were employed with exact line-search, for comparability. We did not simulate the IRLS algorithm, because of its impractical complexity for large images. Similarly, the sequential shrinkage algorithm was not experimented, since obtaining the columns of the CT dictionary is computationally expensive.

Although the hereafter results were confirmed for many images and noise levels, because of space considerations we present only representative results. Figure 5 shows the objective function minimization progress of the tested algorithms for the image *Peppers*, contaminated by a white Gaussian noise of $\sigma_n = 20$. Normally, the more complex the calculation of the descent direction, the faster the minimization (in terms of the iteration's number). Apart from the Truncated Newton algorithm, all algorithms involve only a synthesis and an analysis operation per iteration (in addition to some insignificant calculations), thus being numerically equivalent. In contrast, Truncated Newton requires some inner Conjugate Gradients iterations (the number of which typically increases with the signal size). Nevertheless, we let the inner optimization to converge (requiring at least 20 iterations), to eliminate any ambiguities. The figure clearly shows that our iterative shrinkage algorithm minimizes the objective function the fastest, while being as cheap as any other method.

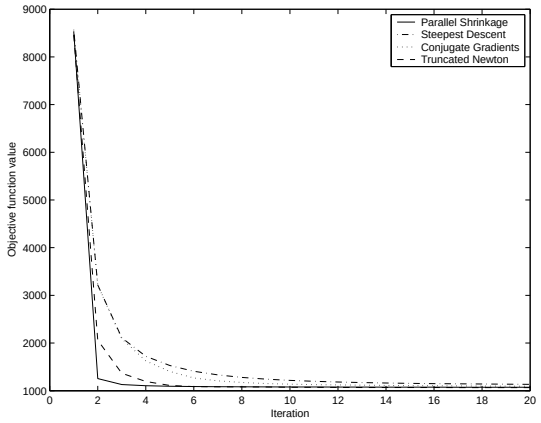

Figure 5. Image denoising: the objective vs. the iteration number (starting from 1).

Our new algorithm excels not only in terms of minimization rate, but also in terms of PSNR. As Figure 6 demonstrates, the PSNR rises faster using our algorithm than using any other minimization method. In fact, only the much more computationally expensive Truncated Newton algorithm comes close to it. The denoising results of our algorithm are presented in Figures 7 and 8. These results show that excellent denoising can be obtained with as little as two successive shrinkage operations. It should be noted that the CT is not well-suited for textured images like *Barbara*, and a different dictionary (or a mixture two) may further improve the results.

Although this paper does not claim that BPDN is the best denoising tool (in terms of PSNR), we nonetheless present here a brief comparison with two other denoising methods. One simple method is hard-thresholding (HT), namely zero-forcing $z(i)$ if it is smaller than a threshold $K\sigma_{n,i}$. As in

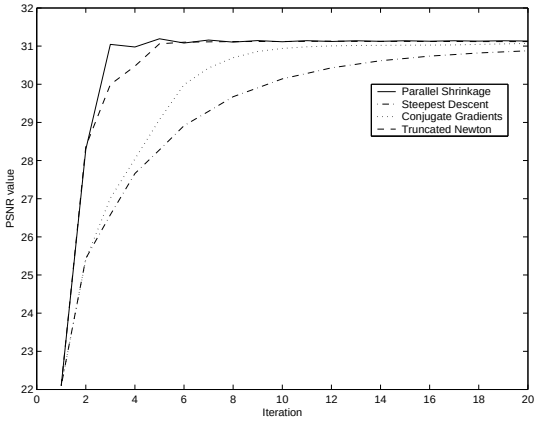

Figure 6. Image denoising: the PSNR vs. the iteration number (starting from 1).

[8], we set $K = 4$ for the finest scale, and $K = 3$ otherwise. Another method is the state-of-the-art BLS-GSM (see [11]), adapted to the CT (see [21] for details). The results, summarized in Table 1, show that BPDN is a reasonable tool for image denoising, especially considering its fast convergence with our iterative shrinkage method.

Table 1. PSNR values comparison ($\sigma_n = 20$)

	Peppers256	*Peppers*	*Lena*	*Barbara*
HT	28.21	30.87	31.46	28.36
BPDN	29.21	31.14	31.49	29.61
BLS-GSM	29.27	31.69	32.06	30.19

5 Related Work

Interestingly, a sequence of recent contributions proposed a similar iterative shrinkage algorithm. First, the work reported in [16, 17] uses such an algorithm for finding the sparsest representation over redundant dictionaries (such as the curvelet, or combination of dictionaries). These papers motivated such algorithm heuristically, relying on the resemblance to the unitary case, on one hand, and on the block-coordinate-relaxation method, on the other [18].

The work by Steidl et. al. [24] studied the connections between Total-Variation (TV) techniques and wavelet-shrinkage. They showed that shift-invariant wavelet shrinkage is equivalent to a single step diffusion filtering or regularization of the Laplacian pyramid of the signal. Moreover, iterating the wavelet shrinkage may improve the denoising performance, a concept lying at the heart of our method as well. However, their work is currently relevant to the 1D Haar wavelet case, and is yet to be fully generalized, although a first step at this direction was made in [25].

Two recent works have proposed similar algorithms to ours, albeit with an entirely different motivation. Figueiredo

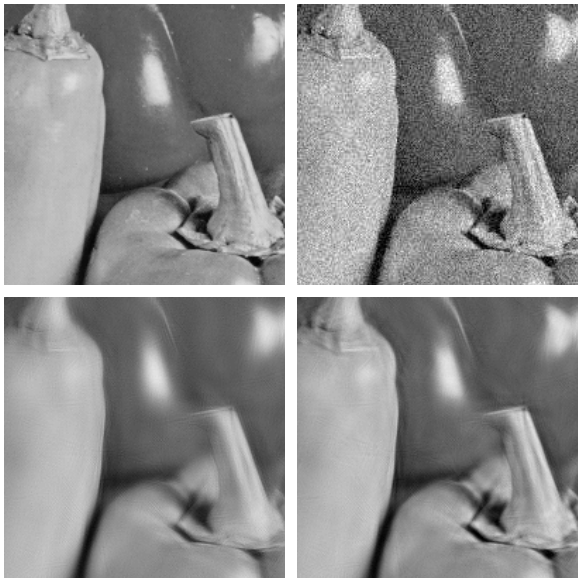

Figure 7. Denoising results (using iterative-shrinkage) of a 200×200 slice of *Peppers* (for $\sigma_n = 20$). From left to right and top to bottom: Original; Noisy ($PSNR = 22.10dB$); Iteration no. 1 ($PSNR = 28.30dB$); Iteration no. 2 ($PSNR = 31.05dB$).

Figure 8. Denoising results (using iterative-shrinkage) of a 256×256 slice of *Barbara* (for $\sigma_n = 20$). From left to right and top to bottom: Original; Noisy ($PSNR = 22.10dB$); Iteration no. 1 ($PSNR = 27.03dB$); Iteration no. 2 ($PSNR = 29.46dB$).

and Nowak suggested a constructive method for image deblurring, based on iterated shrinkage [19]. Their algorithm aims at minimizing the penalty function

$$f_B(\mathbf{x}) = \frac{1}{2} \cdot \|\mathbf{K}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \cdot \mathbf{1}^T \cdot \rho\{\mathbf{W}\mathbf{x}\}, \qquad (17)$$

where $\mathbf{K}$ represents a blur operator, and $\mathbf{W}$ is a unitary wavelet transform. Their sequential shrinkage method is derived via expectation-maximization (EM), and its corresponding structure is very similar to our method. Another work, by Daubechies, Defrise, and De-Mol [20], addresses the same objective function as in (17). However, their way of developing the algorithm is entirely different, leaning on the definition of a sequence of surrogate functions that are minimized via shrinkage.

Both algorithms can be generalized to handle the minimization of the objective posed in (2). By defining $\mathbf{x}_W = \mathbf{W}\mathbf{x}$, the above penalty function becomes

$$\tilde{f}_B(\mathbf{x}_W) = \frac{1}{2} \cdot \|\mathbf{K}\mathbf{W}^T\mathbf{x}_W - \mathbf{y}\|_2^2 + \lambda \cdot \mathbf{1}^T \cdot \rho\{\mathbf{x}_W\}. \qquad (18)$$

Defining $\mathbf{D} = \mathbf{K}\mathbf{W}^T$, both methods can cope with the very same problem we have discussed here. The subsequent algorithms are disparate from ours only in the role the norms of the atoms play, the thresholds chosen in the shrinkage, and the necessity of a line-search. The implications of these differences will be thoroughly discussed in a future work.

6 Conclusion

We have shown that the heuristic shrinkage has origins in Bayesian denoising, being the first iteration of a sequential shrinkage denoising algorithm. This leads to several consequences: (i) we are able to extend the heuristic shrinkage and get better denoising; (ii) we obtain alternative shrinkage algorithms that use the transform and its adjoint, rather than its pseudo-inverse; (iii) the new interpretation may help in addressing the question of choosing the threshold in shrinkage, and how to adapt it between scales; (iv) the obtained algorithm can be used as an effective pursuit for the BPDN for other applications; and (v) due to the close relation to [20], the proposed algorithm can handle general inverse problems of the form (here $\mathbf{K}\mathbf{D}$ is the effective dictionary):

$$\tilde{f}(\mathbf{x}_T) = \frac{1}{2} \cdot \|\mathbf{K}\mathbf{D}\mathbf{x}_T - \mathbf{y}\|_2^2 + \lambda \cdot \mathbf{1}^T \cdot \rho\{\mathbf{x}_T\}. \qquad (19)$$

7 Acknowledgements

The authors thank Mr. Yue Lu for supplying his new Contourlet toolbox implementation.

References

[1] Donoho, D.L and Johnstone, I.M. (1994) Ideal spatial adaptation by wavelet shrinkage, *Biometrika* Vol. 81 No.

3, pp. 425–455, September.

[2] Donoho, D.L. (1995) De-noising by soft thresholding, *IEEE Transactions on Information Theory*, Vol. 41, No. 3, pp. 613–627, May.

[3] Simoncelli, E.P. and Adelson, E.H. (1996) Noise removal via Bayesian wavelet coring, Proceedings of the *International Conference on Image Processing*, Laussanne, Switzerland. September.

[4] Chambolle, A., DeVore, R.A., Lee, N.-Y., and Lucier, B.J. (1998) Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage, *IEEE Trans. Image Process.*, Vol. 7, No. 3, 319–335.

[5] Moulin, P. and Liu, J. (1999) Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors, *IEEE Transactions on Information Theory*, Vol. 45, No. 3, pp. 909-919, April.

[6] Starck, J.-L, Candes, E.J, and Donoho, D.L. (2002) The curvelet transform for image denoising, *IEEE Transactions On Image Processing*, Vol. 11, No. 6, pp. 670–684, June.

[7] Starck, J.-L., Elad, M., and Donoho D.L. (2004) Redundant multiscale transforms and their application for morphological component separation, *Advances in Imaging And Electron Physics*, Vol. 132, pp. 287–348.

[8] Do, M.N. and Vetterli, M. (2004) The Contourlet Transform: An Efficient Directional Multiresolution Image Representation, *IEEE Transaction on Image Processing*, to appear.

[9] Do, M.N. and Vetterli, M. (2003) Framing pyramids, em IEEE Transactions On Signal Processing, Vol. 51, No. 9, pp. 2329–2342, September.

[10] Lu, Y. and Do, M. N. (2005) Constructing Contourlets with Spatial/Frequency Localization, to appear.

[11] Portilla, J., Strela, V., Wainwright, M.J, and Simoncelli, E.P. (2003) Image denoising using scale mixtures of gaussians in the wavelet domain *IEEE Transactions On Image Processing*, Vol. 12, No. 11, pp. 1338–1351, November.

[12] Guleryuz, O.G. (2003) Weighted overcomplete denoising, Proceedings of the *Asilomar Conference on Signals and Systems*, Pacific Grove, CA, November.

[13] Gorodnitsky, I.F. and Rao, B.D. (1997) Sparse signal reconstruction from limited data using focuss: A reweighted norm minimization algorithm. IEEE Trans. On Signal Processing, Vol. 45, No. 3, pp. 600–616.

[14] Chang, S. G., Yu B., and Vetterli, M. (2000) Spatially Adaptive Wavelet Thresholding with Context Modeling for Image Denoising, *IEEE Transactions on Image Processing*, Vol. 9, No. 9, pp. 1522–1531, September.

[15] Gill, P. E., Murray, M., and Wright, M. H. (1981) Practical Optimization, *Academic Press*, New York.

[16] Starck, J.-L., Candes, E., and Donoho, D.L. (2003) Astronomical image representation by the curvelet transform, *Astronomy and Astrophysics*, Vol. 398, pp. 785–800.

[17] Starck, J.-L., Elad, M., and Donoho, D.L. (2004) Redundant multiscale transforms and their application for morphological component analysis, *Journal of Advances in Imaging and Electron Physics*, Vol. 132, pp. 287–348.

[18] Sardy, S., Bruce, A., and Tseng, P. (2000) Block coordinate relaxation methods for nonparametric wavelet denoising, *J. Comput. Graph. Stat.*, Vol. 9, pp. 361–379.

[19] Figueiredo, M.A. and Nowak, R.D. (2003) An EM algorithm for wavelet-based image restoration, *IEEE Trans. Image Process.* Vol. 12, No. 8, pp. 906–916, August.

[20] Daubechies, I., Defrise, M., and De-Mol, C. (2004) An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Communications on Pure and Applied Mathematics*, Vol. LVII, pp. 1413–1457.

[21] Matalon, B., Elad, M., and Zibulevsky, M. (2005) Image Denoising with the Contourlet Transform, *Proceedings of SPARSE'05*, Rennes, France.

[22] Chen, S.S., Donoho, D.L. and Saunders, M.A. (1998) Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 33–61.

[23] Shapiro, J. (1992) Embedded image coding using zerotrees of wavelet coefficients, *IEEE Transactions on Signal Processing*, Vol. 41, pp. 3445-3462, December.

[24] Steidl, G., Weickert, J., Brox, T., Mrzek, P., and Welk, M. (2004) On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs, *SIAM Journal on Numerical Analysis*, Vol. 42, No. 2, pp. 686–713.

[25] Mrazek, P., and Weickert, J. (2003) Rotationally invariant wavelet shrinkage, *Pattern Recognition, Lecture Notes in Computer Science*, Springer, Berlin, pp. 156-163.