Statistical image processing final project:

An Iterative Linear Expansion of Thresholds for l_1 based Image Restoration

Hanjie Pan & Thierry Blu
Presented by Oren Solomon

August 7, 2016

1 Introduction and main contributions

The paper *An Iterative Linear Expansion of Thresholds for l_1 based Image Restoration* [1] presents a novel framework for image restoration under sparsity assumptions. Specifically, the authors deal with the problem of image deblurring, but similar techniques were also applied to image denoising and can also be applied in the more general context of sparse signal recovery.

The core idea of the paper can be understood intuitively by considering the linear signal acquisition model $\mathbf{y} = \mathbf{H}\mathbf{x}$ with $\mathbf{H} \in \mathbb{R}^{M \times N},\ N > M$ and the common sparse reconstruction paradigm

$$\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2 + \lambda\rho(\mathbf{x}),$$

with $\lambda \geq 0$ and a function $\rho(\cdot)$ representing a (sparsity promoting) regularizer for $\mathbf{x}$. In many cases, the most suited regularizer, or our prior assumption on $\mathbf{x}$, is not known a-priori. Consequently, we assume some prior knowledge on $\mathbf{x}$ and recover $\hat{\mathbf{x}}$ which suits this prior, hopefully close enough to the real $\mathbf{x}$. The authors of the paper suggest a new paradigm for the image restoration process which models it as a linear combination of several elementary operations. Each such operation can be considered as corresponding to a different prior assumption on $\mathbf{x}$, and the main goal of the algorithm is to determine the optimal coefficients of the linear combination. The elementary functions are termed *thresholds*, hence their linear combination is termed *Linear Expansion of Thresholds* (LET). The dimension of the optimization problem is then reduced, since the set of LET operations is usually very small compared to the original problem dimensions.

Main algorithmic benefits:

- Optimization between different priors - the algorithm determines which priors are best suited for the restored image, combining them optimally.
- Fast processing time due to optimization on a dimensionally reduced problem - this benefit is somewhat problematic, as will be explained later on.

2 Background and problem formulation

Consider an image $\mathbf{x}$ degraded by some linear operator $\mathbf{H} \in \mathbb{R}^{M \times N}$, such that the measurement is $\mathbf{y} = \mathbf{H}\mathbf{x}$. $\mathbf{H}$ can correspond to a convolution matrix with some blurring kernel, missing pixels, etc.

The authors assume that $\mathbf{x}$ is sparsely represented in a wavelet base $\mathbf{W} \in \mathbb{R}^{N \times D}$ such that $\mathbf{x} = \mathbf{W}\mathbf{c}$ with some coefficients $\mathbf{c}$. This assumption is not restrictive and other representation bases can be assumed.

A common optimization problem formulation for recovering $\mathbf{x}$ is given by

$$\min_{\mathbf{c}} J(\mathbf{c}) = \min_{\mathbf{c}} J_0(\mathbf{c}) + \lambda ||\mathbf{c}||_1 \tag{1}$$

with $J_0(\mathbf{c}) = ||\mathbf{y} - \mathbf{H}\mathbf{W}\mathbf{c}||_2^2$.

2.1 Iterative reweighted least squares approach

Problem (1) can be solved using many convex optimization solvers, such as *(fast) iterative shrinkage/soft-thresholding* (ISTA/FISTA), TwIST, NESTA etc. Another approach relies on the general framework of *majorization minimization* algorithms. In *iterative reweighted least squares* (IRLS), the core idea is to majorize the non-differentiable $||\cdot||_1$ term with a majorizing function. Specifically

$$||\mathbf{c}||_1 \leq \frac{1}{2}\mathbf{c}^T\mathbf{D}\mathbf{c}, \ \forall \mathbf{c} \in \mathbb{R}^D,$$

with diagonal $\mathbf{D}$ whose ii-th element is $D_{ii} = \frac{1}{|c_i|}$. The latter quadratic term (assuming $\mathbf{D}$ is fixed) is indeed differentiable, hence we can reformulate the minimizer of (1) as

$$\mathbf{c}^{(n+1)} = \operatorname{argmin}_{\mathbf{c}} ||\mathbf{y} - \mathbf{H}\mathbf{W}\mathbf{c}||_2^2 + \frac{\lambda}{2}\mathbf{c}^T\mathbf{D}^{(n)}\mathbf{c}, \tag{2}$$

where n is the iteration number. Each iteration of (2) admits a closed form solution

$$\mathbf{c}^{(n+1)} = \left((\mathbf{H}\mathbf{W})^T(\mathbf{H}\mathbf{W}) + \frac{\lambda}{2}\mathbf{D}^{(n)}\right)^{-1}(\mathbf{H}\mathbf{W})^T\mathbf{y},$$

where the diagonal of $\mathbf{D}^{(n)}$ is updated according to the previous iteration $\mathbf{c}^{(n)}$. The size of $\mathbf{c}$ can be very large, limiting the use of the IRLS technique, due to

- Inversion of a matrix of high dimensions.
- Storage of a large dimension matrix in memory (though in some cases efficient implementation is possible without the need to store the matrix in memory).

2.2 Linear expansion of thresholds

The key idea of LET is to represent $\mathbf{c}$ as a linear combination of "elementary basis vectors" and then optimize their weights. We denote the result of such thresholding process of the measurement vector $\mathbf{y}$ as $\mathbf{F}_k(\mathbf{y}) \in \mathbb{R}^D$, such that

$$\mathbf{c} = \sum_{k=1}^{K} a_k \mathbf{F}_k(\mathbf{y}) = \mathbf{F}\mathbf{a}. \tag{3}$$

More precisely, each vector $\mathbf{F}_k(\mathbf{y})$ is the result of some thresholding operation on the measurements. Each operation can be thought of as corresponding to some prior knowledge on $\mathbf{c}$. Again, our main goal is to determine the optimal coefficients vector $\mathbf{a}$ which in turn reconstructs $\mathbf{c}$ and $\mathbf{x}$. To do so, we plug (3) into (1),

$$\mathbf{a}^{(n+1)} = \operatorname{argmin}_{\mathbf{a}} ||\mathbf{y} - \mathbf{H}\mathbf{W}\mathbf{F}\mathbf{a}||_2^2 + \frac{\lambda}{2}(\mathbf{F}\mathbf{a})^T\mathbf{D}^{(n)}\mathbf{F}\mathbf{a}$$

with $D_{ii}^{(n)} = \frac{1}{[|\mathbf{F}\mathbf{a}^{(n)}|]_i}$. Denoting

$$\mathbf{M}^{(n)} = (\mathbf{HWF})^T(\mathbf{HWF}) + \tfrac{\lambda}{2}\mathbf{F}^T\mathbf{D}^{(n)}\mathbf{F},$$
$$\mathbf{b} = (\mathbf{HWF})^T\mathbf{y},$$

we get an explicit expression for the optimal coefficients vector at each iteration,

$$\mathbf{a}^{(n+1)} = \left(\mathbf{M}^{(n)}\right)^{-1}\mathbf{b}.$$

Typically, K is a small number, such that the latter matrix inversion is easily performed in a computer.

2.3 Iterative LET (iLET)

To refine the algorithm, we can extend the paradigm of (3) such that the LET basis is constructed not only from the measurements $\mathbf{y}$, but also from previous iterations, such as $\mathbf{c}^{(n)}, \mathbf{c}^{(n-1)}$, etc. In such way, the LET scheme becomes an *iterative* scheme,

$$a_k^{opt} = \arg\min_{a_k \in \mathbb{R}} J\left(\sum_{k=1}^{K} a_k \mathbf{F}_k(\mathbf{y}, \mathbf{c}^{(n)})\right).$$

Figure 1 introduces the general iLET scheme. At each iteration the LET vectors are constructed from the measurements and previous iterations. The optimal weights $\mathbf{a}^{(n)}$ are then found and the new wavelet coefficients $\mathbf{c}^{(n+1)}$ are constructed and the process repeats itself.

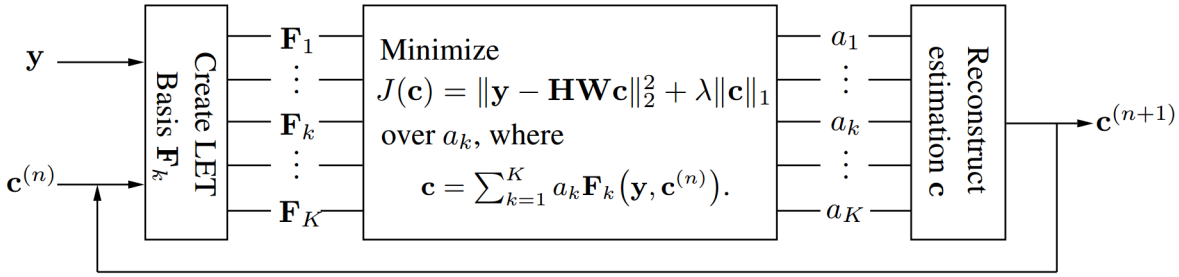


Figure 1: The iLET process, taken from [1].

2.4 Choice of LET basis elements

There are many choices of possible LET/iLET basis operations. Immediate choices are previous iterations such as $\mathbf{c}^{(n)}, \mathbf{c}^{(n-1)}$, etc. Secondly, we can look at *Tikhonov* regularizers as LET bases, i.e.

$$\mathbf{F}_k(\mathbf{y}) = \mathbf{W}^T(\mathbf{H}^T\mathbf{H} + \mu_k\mathbf{I})^{-1}\mathbf{H}^T\mathbf{y},$$

with identity matrix $\mathbf{I} \in \mathbb{R}^{N\times N}$ and some regularization parameter μ_k. Tikhonov regularization is often encountered in ill-conditioned matrix inversion problems and in some cases where $\mathbf{c}$ is assumed to be smooth in some sense. By using several basis vectors with different values of μ_k (which are chosen very crudely), the algorithm chooses the optimal combination of the Tikhonov regularizers.

Another possible type of iLET vectors are based on a "generalized gradient" of the non-differentiable function J and are defined as

$$\bar{\nabla}_\tau J(\mathbf{c}) = \frac{2}{\tau}\left(\mathbf{c} - \mathcal{T}_{\lambda\tau/2}(\mathbf{c} - \frac{\tau}{2}\nabla J_0(\mathbf{c}))\right), \tag{4}$$

with the *soft-thresholding operator*

$$\mathcal{T}_\alpha(\mathbf{x}) = \max\{|\mathbf{x}| - \alpha, 0\} \cdot \text{sign}(\mathbf{x}),\ \alpha \geq 0. \tag{5}$$

As a last example, it is also possible to look at Hessian-based pre-conditioned generalized gradients. The use of second order information can, in some cases, speed the rate of convergence,

$$\bar{\nabla}_\tau J_P(\mathbf{c}) = (\mathbf{W}^T\mathbf{H}^T\mathbf{H}\mathbf{W} + \mu\mathbf{I})^{-1}\bar{\nabla}_\tau J(\mathbf{c}),$$

with identity matrix $\mathbf{I}$ and some regularization μ. Again, it is possible to use several such pre-conditioned gradients with different values of μ and let the algorithm find the optimal combination.

2.5 Examples

Based on the available code for the paper, two examples for image deblurring are presented in figures 2 and 3. In all of the examples, the left image is the ground truth, the middle image is the blurred image (a 9×9 pixels constant square blurring kernel and SNR of 40dB) and the right image is the deblurred image. In both cases the PSNR, defined as

$$\text{PSNR}_{[dB]} = 10\log_{10}\left(\frac{\max(I_0)^2}{\text{MSE}(I_1 - I_0)}\right),$$

with I_1 the blurred image, I_0 the clean image and MSE the mean-squared-error function, is better by several dB than the blurred images. When considering the "evolution" of the iLET coefficients over

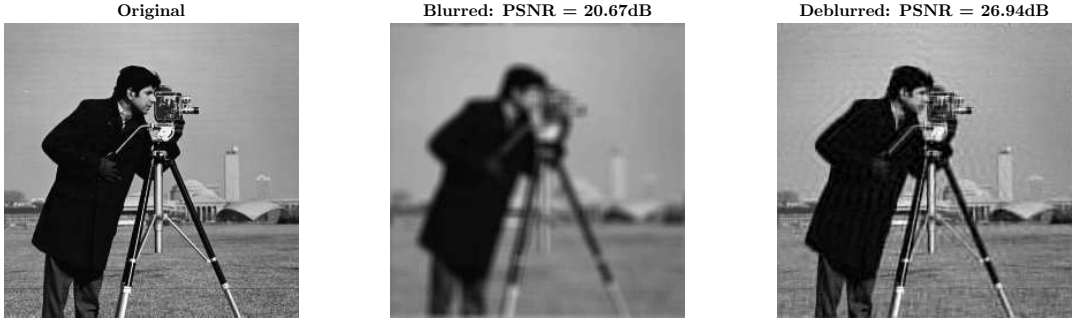


Figure 2: Deblurring the cameraman image using iLET.

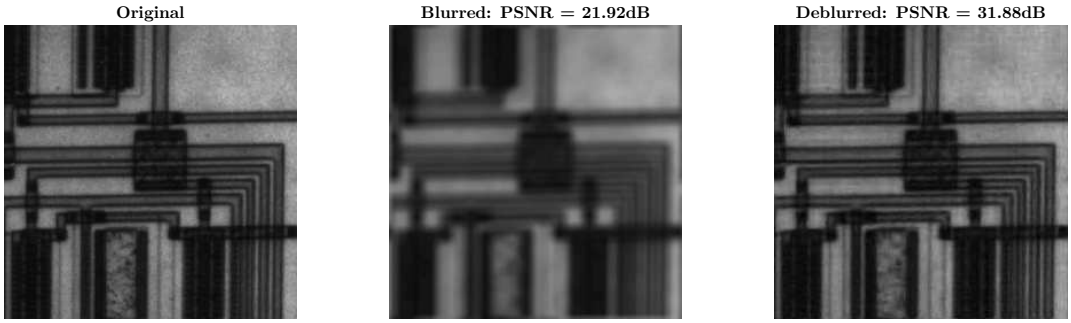
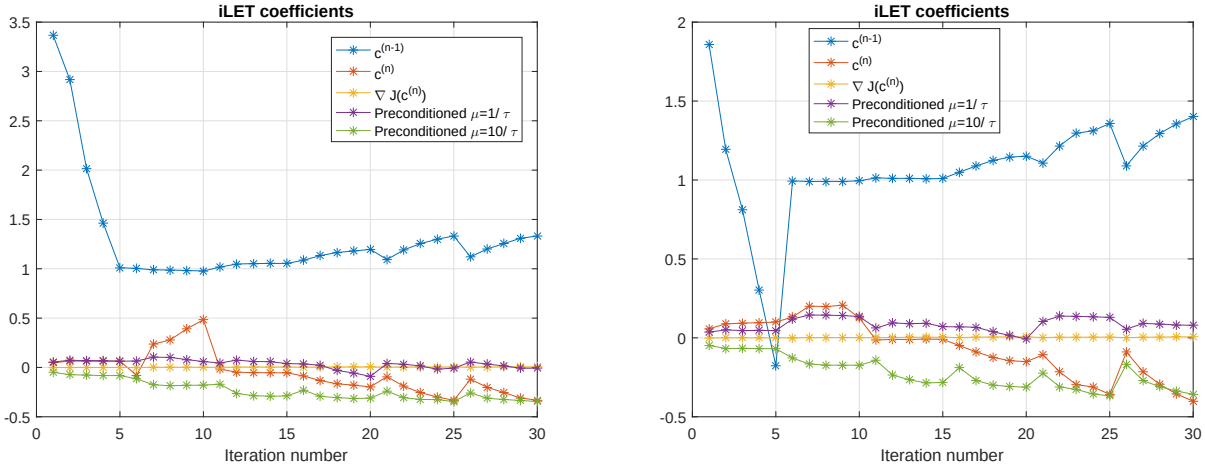


Figure 3: Deblurring the circuit image using iLET.

the iterations, as shown in figures 4a and 4b for the cameraman and circuit images respectively, it is clear that some coefficients are more dominant at some iterations, while others are more dominant at other iterations. This observation is the basis for the proposed improvements in section 3. The chosen iLET base vectors for the above examples were



(a) Cameraman coefficients

(b) Circuit coefficients

Figure 4: Evolution over time of the iLET coefficients.

- $\mathbf{F}_1 = \mathbf{c}^{(n-1)}$,
- $\mathbf{F}_2 = \mathbf{c}^{(n)}$,
- $\mathbf{F}_3 = \bar{\nabla} J(\mathbf{c}^{(n)})$,
- $\mathbf{F}_4 = (\mathbf{W}^T\mathbf{H}^T\mathbf{H}\mathbf{W} + \mu_1 I)^{-1}\bar{\nabla} J(\mathbf{c}^{(n)})$,
- $\mathbf{F}_5 = (\mathbf{W}^T\mathbf{H}^T\mathbf{H}\mathbf{W} + \mu_2 I)^{-1}\bar{\nabla} J(\mathbf{c}^{(n)})$,

with $\mu_1 = 1/\tau$, $\mu_2 = 10/\tau$ and τ a gradient step-size.

2.6 A possibly misleading claim

As mentioned in section 1, one of the main claims of the paper is fast processing time since the optimization process is performed over a low dimensional problem, such that "heavy" operations as matrix inversion are simple and fast. But, consider for example the case of $K = 2$, that is, only two iLET vectors are used and the particular choice of

$$\begin{aligned}\mathbf{F}_1(\mathbf{y}, \mathbf{c}^{(n)}) &= \mathbf{c}^{(n)},\\ \mathbf{F}_2(\mathbf{y}, \mathbf{c}^{(n)}) &= \tfrac{2}{\tau}\left(\mathbf{c} - \mathcal{T}_{\lambda\tau/2}(\mathbf{c} - \tfrac{\tau}{2}\nabla J_0(\mathbf{c}))\right),\end{aligned}$$

with $\mathbf{a} = [1, -\tau/2]^T$. In this case we get that

$$\mathbf{c}^{(n+1)} = \mathbf{F}_1(\mathbf{y}, \mathbf{c}^{(n)}) - \frac{\tau}{2}\mathbf{F}_2(\mathbf{y}, \mathbf{c}^{(n)}) = \mathcal{T}_{\frac{\lambda\tau}{2}}(\mathbf{c}^{(n)} - \frac{\tau}{2}\nabla J_0(\mathbf{c}^{(n)})).$$

This is the general step of the *iterative shrinkage/soft-thresholding algorithm* (ISTA). In that sense, ISTA can be thought of as a spacial case of iLET. In the general case (but still $K = 2$), iLET has an additional computational step which involves finding the optimal $\mathbf{a}$, but "ISTA type" calculations are still necessary, that is in terms of computational complexity iLET is more extensive. The authors claim that indeed iLET is more computationally extensive than ISTA for example, but when numerical simulations were performed against FISTA (and other solvers), iLET usually performed much faster in terms of execution speed. Since iLET requires additional computations over ISTA/FISTA, I believe that the reported improvement in execution time can arise from two possibilities:

- A more efficient implementation of the iLET code in MATLAB.
- iLET uses second order information ("preconditioned gradient"), such that the minimal cost function value is attained after much fewer iterations than a first order method such as FISTA. Thus, the smaller number of iterations required by iLET leads to an overall faster execution time, implying that the comparison between the algorithms is not "fair".

3 Personal contribution, extensions and simulations

In this section, I propose an extension of the iLET algorithm. I propose to form an iLET based scheme for TV regularization sparse recovery problems.

3.1 Analysis based TV reconstruction

The observation that the different coefficients $\mathbf{a}$ change over time, that is, as the iteration number increases some values of $\mathbf{a}$ increase while others decrease can imply that different priors are best suited for the image at different time points (iterations). Recall problem (2)

$$\min_{\mathbf{c}} ||\mathbf{y} - \mathbf{HWc}||_2^2 + \lambda||\mathbf{c}||_1,$$

which is commonly known as the *synthesis problem*. Another possible formulation, known as the *analysis problem* is

$$\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{Hx}||_2^2 + \lambda||\mathbf{W}^*\mathbf{x}||_1, \tag{6}$$

with $\mathbf{W}^*$ being the adjoint of $\mathbf{W}$. In general, the solutions to both problems is different, but in certain cases, e.g. $\mathbf{W}$ is a unitary transformation, the solutions coincide. One of the most popular analysis type regularizers is the well known *total-variation* (TV) [2, 3] regularizer, which admits piece-wise constant solutions. I decided to build on the freely available code of the paper and add another iLET vector, which performs as a "generalized gradient" step for a TV based minimization to see if an additional piece-wise constant prior can improve reconstruction results.

Problem (1) can be viewed as a minimization of a decomposition model

$$\min_{\mathbf{x}\geq\mathbf{0}} \lambda g(\mathbf{x}) + f(\mathbf{x}),$$

where f is a smooth, convex function (such as $J_0(\mathbf{x})$) with a Lipschitz continuous gradient and a possibly non-smooth but proper, closed and convex function g (such as $||\mathbf{x}||_1$). Solving (1) iteratively involves finding *Moreau's proximal* (prox) mapping [4] of αg for some $\alpha \geq 0$, defined as

$$\text{prox}_{\alpha g}(\mathbf{x}) = \arg\min_{\mathbf{u}\in\mathbb{R}^n} \left\{ \alpha g(\mathbf{u}) + \frac{1}{2}||\mathbf{u} - \mathbf{x}||_2^2 \right\}. \tag{7}$$

For $g(X) = ||\mathbf{x}||_1$, $\text{prox}_{\alpha g}(\mathbf{x})$ is given by the *soft-thresholding* operator (5). this is in fact the **closed form solution** for the l_1 minimization step of ISTA/FISTA. The generalized gradient step defined in (4) is based on the latter prox mapping. In general, for analysis type problems i.e. minimization of $||\mathbf{W}^*\mathbf{x}||_1$, there is no closed form solution for the prox operator, so usually it needs to be handled in a different way. In [5], the authors derived an iterative solution to the proximal mapping of the TV norm of an image TV($\mathbf{x}$), for both the isotropic and anisotropic cases with box constraints. Additional look on the definition of the prox mapping (7) shows that this is actually a *denoising* problem, so by running a denoising procedure in each iteration of the TV FISTA minimization algorithm [5] we recover the TV restored object. Instead of performing a gradient step along the direction of the gradient of J_0

and then performing soft-thresholding we perform a gradient step in the direction of the gradient of J_0 and then perform TV denoising on it.

Following this rationale, I decided to add the additional TV based iLET vector by using the available code from [5] (Algorithm GP in [5]) for TV denoising and test it against several examples. Figure 5 shows the reconstruction of an artificial test image without the additional TV based iLET coefficient, while figure 6 shows the reconstruction of that same image (same blurring kernel as before) with this additional vector. A mild improvement in the PSNR value (0.72 dB) and the reconstructed image itself is evident. It should be noted that the TV generalized gradient step was performed on $\mathbf{x}$ and not on $\mathbf{c}$. SNR was 40dB in both cases.

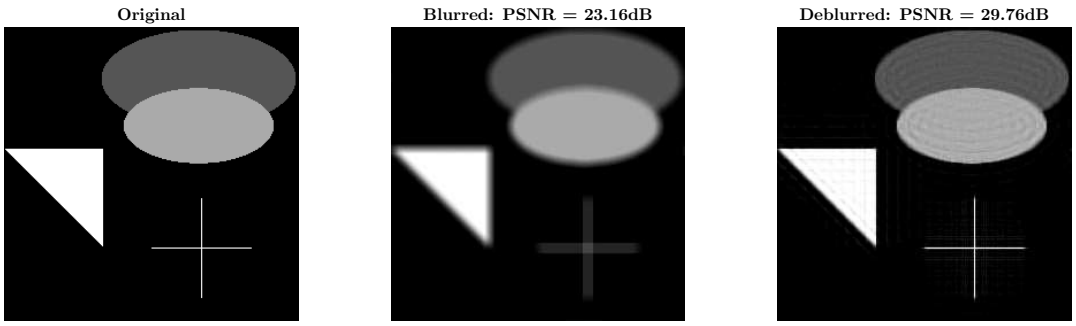


Figure 5: Artificial test: iLET without TV.

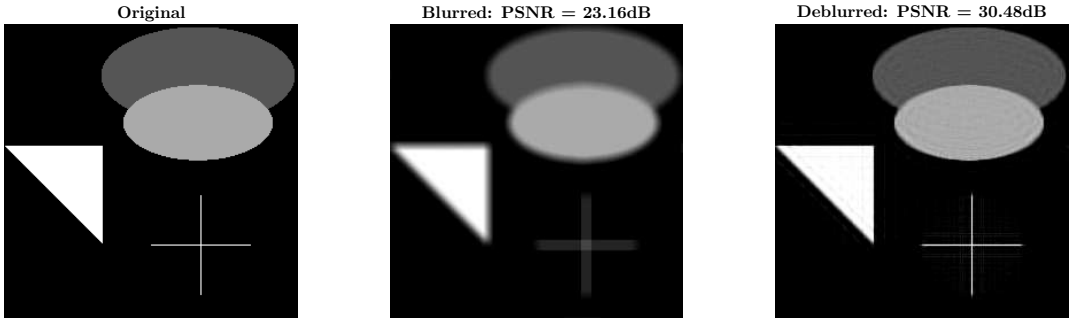


Figure 6: Artificial test: iLET with TV.

The evolution of the iLET coefficients is presented in figure 7. It is evident that the TV weight is small, corresponding to the small improvement as shown in figure 6. I run the algorithm on additional (natural) images, but reconstruction improvements were very mild at best. I believe this is due to several reasons as discussed during the presentation,

- A prior of TV is less appropriate for the images I chose (compared with priors of sparsity in the wavelet domain or smoothness).
- All of the iLET vectors suggested in [1] resulted from the synthesis formulation (2), and correspondingly the optimization procedure of $\mathbf{a}$ arises from that same formulation. Consequently, analysis based iLET vectors will possibly yield better results when used with a proper analysis formulation.

Encouraged by the previous results, I decided to concentrate on an iLET formulation for the case

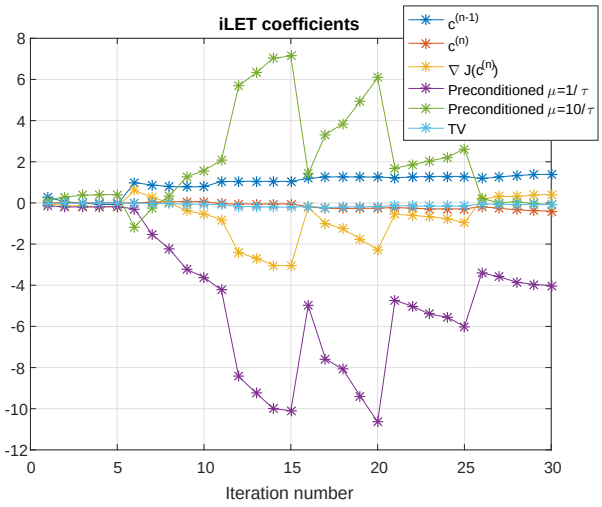


Figure 7: Artificial test: iLET with TV.

of TV minimization. Consider the TV minimization problem,

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{y}-\mathbf{H}\mathbf{x}||_2^2 + \lambda \mathrm{TV}(\mathbf{x}), \tag{8}$$

with isotropic TV term

$$\mathrm{TV}(\mathbf{x}) = \sum_i \sqrt{(\Delta_i^h \mathbf{x})^2 + (\Delta_i^v \mathbf{x})^2},$$

such that $\Delta_i^h = \mathbf{x}_i - \mathbf{x}_{r(i)}$ and $\Delta_i^v = \mathbf{x}_i - \mathbf{x}_{b(i)}$ represent horizontal and vertical first-order derivatives with cyclic boundary conditions, respectively ($r(i)$ represents the pixel on the right of pixel i while $b(i)$ represents the pixel below pixel i) [6].

According to [6], a majorizing function for $\mathrm{TV}(\mathbf{x})$ is the following function (with some constant terms, omitted here),

$$\mathrm{TV}(\mathbf{x}) \leq \mathbf{x}^T \mathbf{D}^T \mathbf{W}^{(n)} \mathbf{D}\mathbf{x},$$

were $\mathbf{D} = [(\mathbf{D}^h)^T, (\mathbf{D}^v)^T]^T$ and $\mathbf{D}^h, \mathbf{D}^v$ represent the horizontal and vertical first-order differentiation matrices (cyclic boundary conditions). Also define a diagonal matrix $\mathbf{\Lambda}^{(n)}$ whose ii-th element is

$$[\mathbf{\Lambda}^{(n)}]_{ii} = \frac{1}{\sum_i \sqrt{(\Delta_i^h \mathbf{x}^{(n)})^2 + (\Delta_i^v \mathbf{x}^{(n)})^2}},$$

with (n) being an iteration number, such that

$$\mathbf{W}^{(n)} = \begin{bmatrix} \mathbf{\Lambda}^{(n)} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}^{(n)} \end{bmatrix}.$$

Thus, (8) is replaced by

$$\min_{\mathbf{x}} \mathbf{x}^T (0.5\mathbf{H}^T\mathbf{H} + \lambda \mathbf{D}^T \mathbf{W}^{(n)} \mathbf{D})\mathbf{x} - \mathbf{x}^T \mathbf{H}^T \mathbf{y}. \tag{9}$$

Assuming that $\mathbf{x}$ is a linear combination of an iLET basis, $\mathbf{x} = \mathbf{F}\mathbf{a}$ and since (9) is differentiable, we can formulate the IRLS update rule,

$$\mathbf{a}^{(n+1)} = \left(\mathbf{F}^T (0.5\mathbf{H}^T\mathbf{H} + \lambda \mathbf{D}^T \mathbf{W}^{(n)} \mathbf{D})\mathbf{F}\right)^{-1} \mathbf{F}^T \mathbf{H}^T \mathbf{y}.$$

I decided to use the following iLET base vectors,

- $\mathbf{F}_1 = \mathbf{x}^{(n-1)}$,
- $\mathbf{F}_2 = \mathbf{x}^{(n)}$,
- $\mathbf{F}_3 = \bar{\nabla} J(\mathbf{x}^{(n)})_{\mathrm{TV}}$,
- $\mathbf{F}_4 = (\mathbf{H}^T\mathbf{H} + \mu_1 I)^{-1}\bar{\nabla} J(\mathbf{x}^{(n)})_{\mathrm{TV}}$,
- $\mathbf{F}_5 = (\mathbf{H}^T\mathbf{H} + \mu_2 I)^{-1}\bar{\nabla} J(\mathbf{x}^{(n)})_{\mathrm{TV}}$,

with $\mu_1 = 1/\tau$, $\mu_2 = 10/\tau$, τ a gradient step-size and $\bar{\nabla} J(\mathbf{x}^{(n)})_{\mathrm{TV}}$ is the same gradient step as before, only the prox-mapping of the TV norm is calculated iteratively using algorithm GP in [5] (I chose 20 iterations).

I implemented the TV based iLET algorithm and run it on some several examples (images of size 128×128 pixels) with the same blurring kernel as before. Figures 8 (MATLAB code *TV_iLET_demo.m*) and 9 (MATLAB code *iLET_demo1.m* of the paper's authors) show the TV analysis and synthesis iLET reconstructions, respectively.

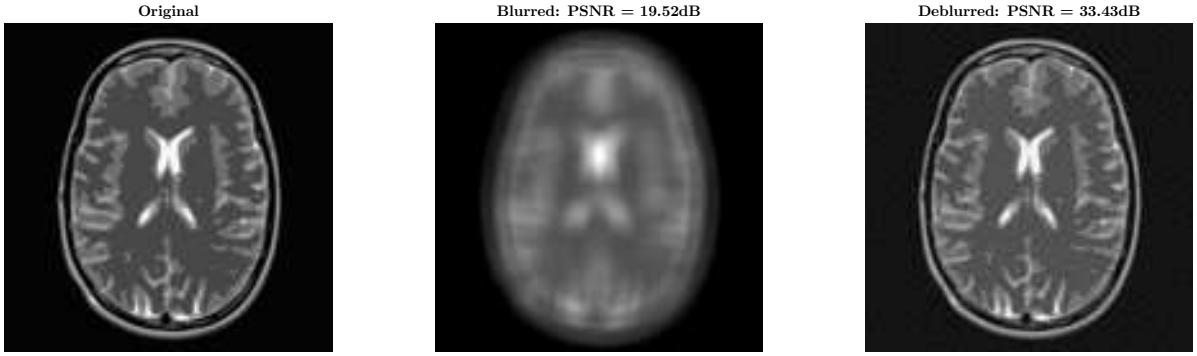


Figure 8: TV iLET reconstruction of an MRI brain phantom.

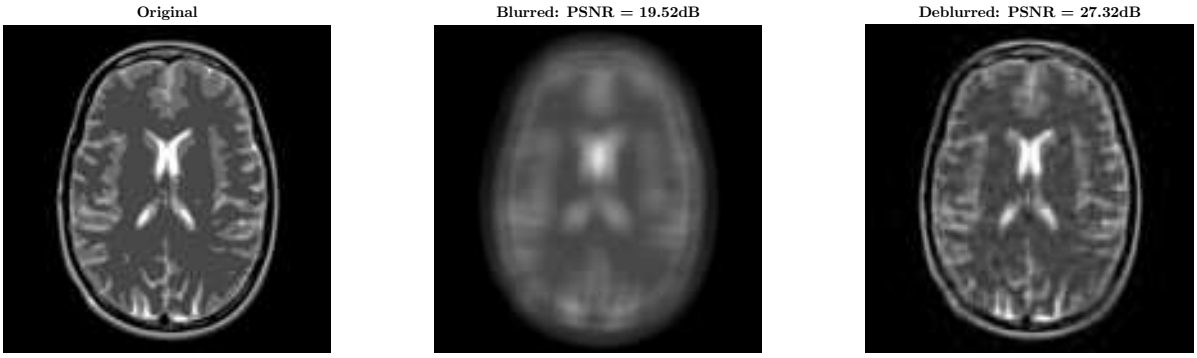
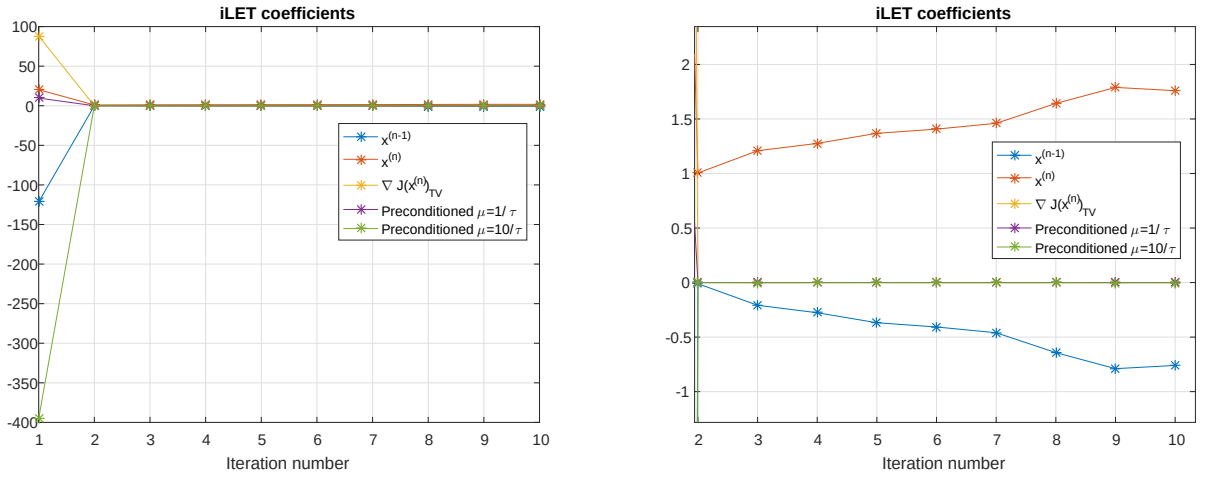


Figure 9: Synthesis iLET reconstruction of an MRI brain phantom.

Figure 10a shows the iLET coefficients of the reconstruction in Figure 8. It seems that the TV gradient and preconditioned gradient steps have a small weight.I decided to test the reconstruction when using only vectors $\mathbf{F}_1$ and $\mathbf{F}_2$ and again when using the vectors $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ only. the PSNR values for each reconstruction are summarized in Table 1. It seems that incorporating the TV gradient and pre-conditioned gradient steps has a strong influence on the reconstruction performance, even though their weight is small. Incorporation of first order and then second order information does improve reconstruction quality.

iLET vectors	PSNR [dB]
$\mathbf{F}_1$-$\mathbf{F}_2$	23.47
$\mathbf{F}_1$-$\mathbf{F}_3$	30.95
$\mathbf{F}_1$-$\mathbf{F}_5$	33.43

Table 1: PSNR values for reconstruction of an MRI brain phantom with different iLET vectors.



(a) TV iLET coefficients.

(b) TV iLET coefficients - zoom.

Figure 10: Evolution over time of the iLET coefficients for the TV analysis reconstruction of Figure 8.

Comparing the reconstruction performance of the analysis (Figure 8) and synthesis (Figure 9) reconstructions, it seems that the TV iLET reconstruction performs better for this type of example. To be fair, the algorithms I run are a bit different. Though both algorithms run for the same number of iterations (10) and with the same SNR value of 40dB, the step size used is a bit different. I used a step size of $\tau = 2.7988$ and $\lambda = 0.0001$ in the TV iLET algorithm, while the authors of the paper used an adaptive scheme to determine the step size and λ. I didn't see the need to use an adaptive step size scheme since I got preferable results and I think this example shows the main essence of my formulation (which can be further improved).

Another reconstruction example is given in Figures 11 and 12 of the circuit image of size 128×128 pixels (same blurring kernel as before), showing preferable reconstruction results for the TV based formulation.

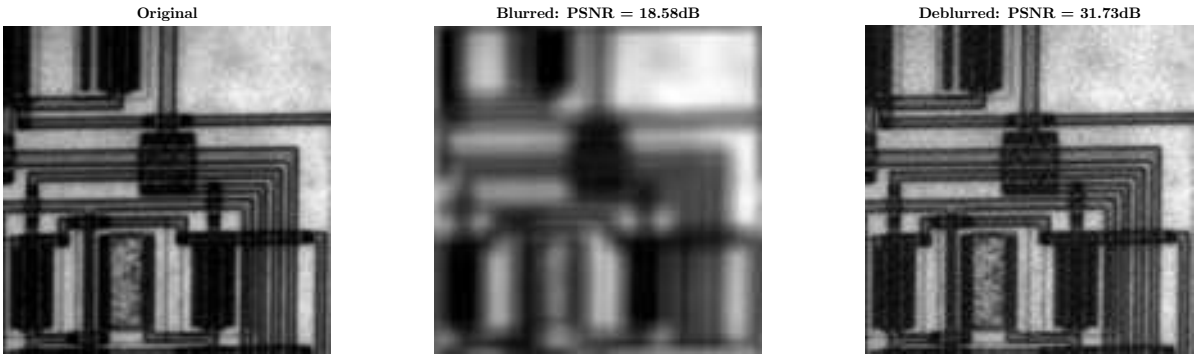


Figure 11: TV iLET reconstruction of the circuit image.

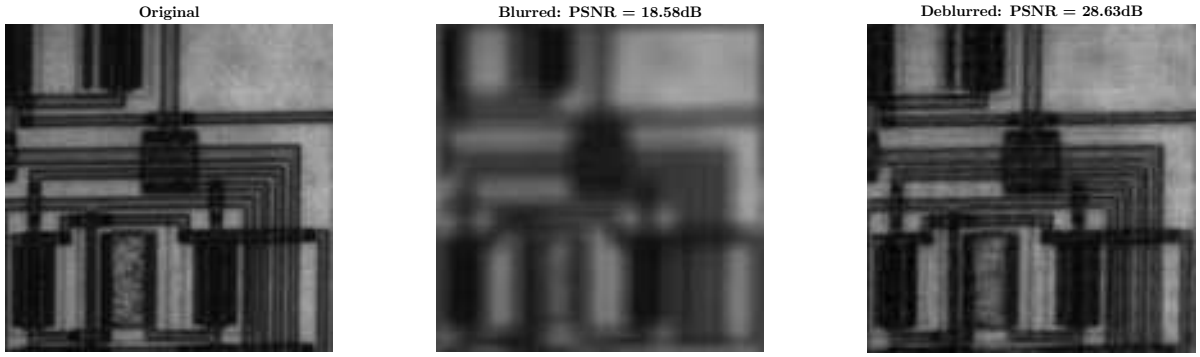


Figure 12: Synthesis iLET reconstruction of the circuit image.

When I tried to extend these results to a patch-based analysis, I got poor reconstruction results. Due to the latter results and since I ran out of pages, I decided to forgo this direction and concentrate on the TV iLET formulation instead.

4 Conclusions

In this work I've surveyed the paper in Ref. [1]. Its authors propose a novel, general framework for sparse signal recovery (applied to image deblurring as an example, but suited for general sparse signals). The main iLET idea is the combination of several "simple" steps in an optimal manner. Intuitively, each step corresponds to a different prior assumption on the recovered signal. The algorithm then combines the steps in an optimal manner. Since the iLET algorithm corresponds to a synthesis type of problem, I decided to extend its core idea to include also analysis type priors, specifically a total-variation prior and demonstrated improved reconstruction over several examples.

References

[1] Hanjie Pan and Thierry Blu. An iterative linear expansion of thresholds for-based image restoration. *IEEE Transactions on Image Processing*, 22(9):3715–3728, 2013.

[2] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[3] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[4] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

[5] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009.

[6] João P Oliveira, José M Bioucas-Dias, and Mário AT Figueiredo. Adaptive total variation image deblurring: a majorization–minimization approach. *Signal Processing*, 89(9):1683–1693, 2009.