CONDY: ULTRA-FAST HIGH PERFORMANCE RESTORATION USING MULTI-FRAME L2-RELAXED-L0 SPARSITY AND CONSTRAINED DYNAMIC HEURISTICS

J. Portilla∗, E. Gil-Rodrigo
Instituto de Optica
Consejo Superior de Investigaciones Científicas
Madrid, Spain

D. Miraut, R. Suarez-Mesa
E.T.S. Ingenieria Informatica
Universidad Rey Juan Carlos
Madrid, Spain

ABSTRACT
Recently we proposed an efficient technique based on analysis-based sparsity in tight frames to restore images affected by linear blur and additive white Gaussian noise. Such technique performed a joint optimization of a cost function in the image and coefficient space by alternating their corresponding marginal minimizations, until convergence. Here we propose, first, a more standard Bayesian (MAP) re-formulation of the model/method, yielding the same cost function. Second, we propose a heuristical twist in the method, consisting of running only a few iterations (too few to reach convergence), with a dynamic shrinkage plan obtained by maximizing a performance measure in a set of deblurring tests. Third, we propose a multi-frame prior scheme allowing for different thresholds in different frames. Compared to the original method, we achieve a very important speed-up, while keeping high robustness and state-of-the-art performance.

Index Terms— Ultra-fast deblurring, sparsity, alternate marginal optimization, dynamic hard thresholding, L2-relaxed L0 pseudo norm.

1. INTRODUCTION
An extended trend in the image processing field during the last years is to address an every time larger set of ill-posed inverse problems within a common conceptual and computational framework. In this context, a Bayesian or regularization-based approach is typically based on using: (1) an appropriate image transformation, in terms of enhancing the saliency of its peculiar features (sparse edges, lines, etc.); (2) a prior (or regularization term) for the features in that representation; and (3) an observation model of how the degradation has affected the ideal image. These three components are used to build a cost function expressing a measure of the “goodness” of a solution to the inverse problem in terms both of how typical it is and how well it fits the observation. What sets a new trend is that (1) the used representations are, typically, multi-scale and multi-orientation, of the kind of ”x-let” redundant frames, and they are very efficiently implementable; (2) the prior models typically express a characterization of the image sparsity in the chosen representation, and they can be extremely simple (e.g., ℓ1 norm, ℓ0 pseudo-norm, etc.); (3) the fitness of the solution to the prior and observation can be imposed iteratively, through very simple operations, for a wide variety of degradation cases, until convergence to a (local, in general) optimum of the cost function. Based on a wide variety of conceptual approaches and mathematical tools, a number of techniques have been proposed during the last years which fit, functionally, this scheme (see, e.g. [1, 13, 2]).

On the other hand, many authors (e.g. [14, 13, 1, 15]) have used dynamic shrinkage operations through the sparse optimization loop. This can be interpreted, from a global optimization point of view, as a means of avoiding non-favourable local optima by progressively increasing the cost function’s sharpness (see, e.g., [3]). Typical dynamic shrinkage results, for the case of simple priors like ℓ1 or ℓ0 pseudo-norms, in a monotonically decreasing thresholding, as the iterations progress. Some of the cited authors have chosen, because of its simplicity and good behavior, an exponentially decay. Here we truncate its tail, not letting it to converge.

2. USING A ℓ2-RELAXED ℓ0 PSEUDO-NORM TO CHARACTERIZE IMAGE SPARSITY
To use the ℓ0 pseudo-norm (i.e., counting the non-zero coefficients) to directly characterize real-world sparse signals conveys some problems, because: (1) real images are not strictly sparse in any given dictionary; (2) ℓ0 pseudo-norm is not only non-convex, but also discontinuous. In previous models (see, e.g. [2]) we used this pseudo-norm as a tool to describe a hidden, auxiliary strictly sparse vector, in such a way that discontinuity of the pseudo-norm was not an issue. However, including a strictly sparse hidden random vector in the model makes it less standard and more difficult to interpret. Here we propose a more standard MAP formulation, giving raise to the

∗JP, EGR and RSM have been funded by grants CICYT TEC2009-13696 and CSIC PIE201050E021. DM has been funded by grant CICYT TIN2010-21289-C02-01. All grants from the Spanish Education and Science&Innovation Ministrys.
By using Eq. (2) and Eq. (3) we can write:

$$p(x) \propto \exp \left( -\frac{1}{\alpha} \|\Phi^* x\|_{(0:T)} \right),$$  \hspace{1cm} (1)

where $x \in \mathbb{R}^N$, $\Phi^* x \in \mathbb{R}^M$, with $M \geq N$, and $\|v\|_{(0:T)}$ is an $\ell_2$-relaxed form of $\ell_0$ with parameter $T$, defined by:

$$\|v\|_{(0:T)} = \min_{a} \|v\|_2 + \frac{1}{T^2} \|v - a\|_2^2$$  \hspace{1cm} (2)

and

$$\|v\|_{(0:T)} = \|\Theta_h(v, T)\|_0 + \frac{1}{T^2} \|v - \Theta_h(v, T)\|_2^2,$$

where $\Theta_h(v, \theta)$ denotes the vector resulting from setting to zero the $v$ components smaller than $\theta$ in magnitude $^1$. Thus, it is the $\ell_0$ pseudo-norm of a strictly sparse approximation $a$ to $v$, plus its (scaled by $1/T^2$) squared Euclidean error. The chosen $a$ is the one minimizing that sum. It may be computed as a sum of saturated-to-one parabola centered at zero, fulfilling $\lim_{T \to 0} \|v\|_{(0:T)} = \|v\|_0$.

In [4] we used a combined Parseval frame (PF) by building a vector with the coefficients of two scaled PFs. This produced a significant performance gain. Nevertheless, in terms of designing a prior, forcing to use the same $T$ for every sub-frame is too rigid. A more flexible version of Eq. (1) is:

$$p(x) \propto \exp \left( -\sum_{j=1}^{J} \frac{1}{\alpha_j} \|\Phi_j^* x\|_{(0:T_j)} \right),$$  \hspace{1cm} (3)

which allows to use a different thresholding for each frame $^2$.

3. MAXIMUM-A-PSEUDO-ESTIMATION

By using Eq. (2) and Eq. (3) we can write:

$$\hat{x}(y) = \arg\max_x p(x|y)$$  \hspace{1cm} (4)

$$= \arg\min_x -\log p(y|x) - \log p(x)$$

$$= \arg\min_x -\log p(y|x) +$$

$$\sum_{j=1}^{J} \frac{1}{\alpha_j} \min_{a_j} \left( \|a_j\|_0 + \frac{1}{T^2} \|\Phi_j^* x - a_j\|_2^2 \right),$$  \hspace{1cm} (5)

which, for $y = Hx + w$ representing a linear blur plus additive Gaussian white noise, becomes:

$$\hat{x}(y) = \arg\min_x \mu \|Hx - y\|_2^2 +$$

$$+ \sum_{j=1}^{J} \frac{1}{\alpha_j} \min_{a_j} \left( \|a_j\|_0 + \lambda_j \|\Phi_j^* x - a_j\|_2^2 \right),$$  \hspace{1cm} (6)

with $\mu = 1/(2\sigma_0^2)$ and $\lambda_j = 1/T_j^2$. We also impose

$$\lambda_j = \alpha_j/(2\sigma_j^2), \hspace{1cm} j = 1..J,$$  \hspace{1cm} (7)

with $\sigma_j^2$ a positive constant ("variance") controlling the amount of $\ell_2$ relaxation of our modified $\ell_0$ pseudo-norm. Under the Eq. (7) constraint, Eq. (6), is an extension to the multiple frame case of the problem formulation in [4]. Besides the multi-frame extension, now we can write the new model as a MAP estimation with the prior of Eq. (3). Analogously as in [4], Eq. (6) can be solved very efficiently by alternate marginal optimizing $\{a_j\}$ (hard thresholding) and $x$ (linear transform), yielding:

**Step 0:** $x^{(0)} = y$

**Step 1:** $b^{(n)}_j = \Phi_j^* x^{(n)}$

**Step 2:** $a^{(n)}_j = \Theta_h(b^{(n)}_j, T_j)$

**Step 3:** $z^{(n)} = 2\sigma_j^2 \sum_{j=1}^{J} \lambda_j \Phi_j a^{(n)}_j = \frac{1}{J} \sum_{j=1}^{J} \Phi_j a^{(n)}_j$

**Step 4:** $x^{(n+1)} = \left( \sum_{j=1}^{J} \lambda_j \Phi_j^* \Phi_j + \mu H^* H \right)^{-1}$

$$\times \left( \sum_{j=1}^{J} \lambda_j \Phi_j a^{(n)}_j + \mu H^* y \right)$$

$$= \left( \frac{1}{J} \sum_{j=1}^{J} \Phi_j^* \Phi_j + \nu H^* H \right)^{-1} (z^{(n)} + \nu H^* y)$$

*back to Step 1,*

with $\nu = \sigma_j^2/(J\sigma_0^2)$, and where we have applied Eq. (7) to obtain the last equalities in Step 3 and 4. If $H$ implements a 2D convolution, Step 4 can be easily done in the Fourier domain. If, besides, $\Phi_j^*$'s are Parseval frames ($\Phi_j^* \Phi_j = I$), then Step 4 further simplifies to:

$$X(u, v) = \frac{Z(u, v) + \nu H^*(u, v) Y(u, v)}{1 + \nu |H(u, v)|^2},$$

with $Z(u, v) = F \{ z^{(n)} \}$ and $x^{(n+1)} = F^{-1} \{ X(u, v) \}$.

4. CONSTRAINED DYNAMIC THRESHOLDING

Although an effective cost function minimization requires the convergence of the dynamic parameters as well as the solution, here we explore the empirical question, of obvious practical interest, of how far we can go reducing the number of

$^1$We use here the term "magnitude" (instead of "absolute value") so we can apply it to complex transforms as well.

$^2$In the single frame model one can obtain different effective thresholds using a single $T$, just by weighting differently the sub-frames within. However, this weighting will affect (negatively, in general) the performance.
iterations without disrupting the good properties (robustness, high performance) of the method - no matter we may get far from convergence conditions. In this work we have explored the effect of fixing the number of iterations, \(N_i\), and using a truncated exponential decay for \(\sigma^2_i\), as the only independent dynamic parameter in the loop:

\[
\sigma^2_i(n) = \sigma^2_{r0} \beta^n, \quad n = 1 \ldots N_i
\]

with \(0 < \beta \leq 1\). We first fix the proportions \(k_{i,j} = T_i/T_j\) among the different frames, e.g., based on their relative mean square value, or based on that some are complex and others are real (as we have done here) etc. Then, noting that \(T_j(n) = \sqrt{2/\alpha_j} \sigma_j(n)\), we obtain that \(\alpha_i = k_{i,j} \alpha_j\). Thus, once we have set \(\sigma^2_j(n)\), there is only another free parameter left, that we name \(\alpha\) (we may set \(\alpha = \alpha_0\), \(\alpha\) being the index denoting an arbitrary reference frame). Summarizing, our parameters to be trained for the model are \(\sigma^2_j(0)\), \(\beta\), and \(\alpha\). From them and the set of constants \(\{k_{j,o}\}\), for \(j = 1 \ldots J\), we can obtain all the \(\alpha_j\)'s. From them and the \(\sigma^2_j(n)\), we get the dynamic thresholds \(T_j(n)\). Note that \(\nu(n) = \sigma^2_j(n)/(J \sigma^2_w)\) changes also through iterations.

5. IMPLEMENTATION AND TESTS

5.1. Image Representation

In this work we have used two frames: \(\Phi_1\), a translation invariant Haar pyramid (TIHP, quasi-Parseval), as described in [5], and \(\Phi_2\), the Dual-Tree Complex Wavelet Transform [16], which is a PF, both with 3 scales. Unlike in [4], coefficients of the D-T CWT are now processed as \([16]\), which is a PF, both with 3 scales. Unlike in [4], co-

\(\Phi\)

variant Haar pyramid (TIHP, quasi-Parseval), as described empirically set 

\[k_4\] 

numbers, instead of 

\[5.1. Image Representation\]


\[N\]

high performance) of the method - no matter we may get far iterations without disrupting the good properties (robustness, high performance) of the method - no matter we may get far from convergence conditions. In this work we have explored the effect of fixing the number of iterations, \(N_i\), and using a truncated exponential decay for \(\sigma^2_i\), as the only independent dynamic parameter in the loop:

\[
\sigma^2_i(n) = \sigma^2_{r0} \beta^n, \quad n = 1 \ldots N_i
\]

with \(0 < \beta \leq 1\). We first fix the proportions \(k_{i,j} = T_i/T_j\) among the different frames, e.g., based on their relative mean square value, or based on that some are complex and others are real (as we have done here) etc. Then, noting that \(T_j(n) = \sqrt{2/\alpha_j} \sigma_j(n)\), we obtain that \(\alpha_i = k_{i,j} \alpha_j\). Thus, once we have set \(\sigma^2_j(n)\), there is only another free parameter left, that we name \(\alpha\) (we may set \(\alpha = \alpha_0\), \(\alpha\) being the index denoting an arbitrary reference frame). Summarizing, our parameters to be trained for the model are \(\sigma^2_j(0)\), \(\beta\), and \(\alpha\). From them and the set of constants \(\{k_{j,o}\}\), for \(j = 1 \ldots J\), we can obtain all the \(\alpha_j\)'s. From them and the \(\sigma^2_j(n)\), we get the dynamic thresholds \(T_j(n)\). Note that \(\nu(n) = \sigma^2_j(n)/(J \sigma^2_w)\) changes also through iterations.

5.2. Experiments

We show here the results of applying the resulting dynami-

\[cally constrained optimized method, for iterations \(N_i\) taking the values 5, 10 and 20. For each considered number of iter-

\[ations, \(N_i\), we obtained the triplet \((\sigma^2_{r0}(N_i), \beta(N_i), \alpha(N_i))\), shown in Table 1 by optimizing a performance criterion jointly over a set of restoration tests (we used the ten standard experiments reported in [4]). As performance measurement we used the average increment in signal-to-noise ratio (ISNR), in dBs. MFS denotes "multi-frame static", which corresponds to the method used here, but with static thresholding. CD(5), CD(10), and CD(20) corresponds to the described method (ConDy, from CONstrained DYnamic), using 5, 10 and 20 iterations. We can see that in most cases our method provides the best ISNR results, only surpassed in two of the seven experiments by [7].

Table 3 shows average performance for a subset of 5 ex-

\[periments using a single frame (a redundant version of the Haar wavelet, TIHP in our case), in ISNR, number of it-

\[erations and time, in seconds. We compare our method to recent sparsity-based methods using also a redundant Haar frame (data taken from table in [12]). The averaged results correspond to degradation tests (PSF1 and PSF3 both with \(\sigma^2_w = 2\) and \(\sigma^2_w = 8\), PSF2 with \(\sigma^2_w = 0.56^2\)). We note a substantial improvement of our method, in the three compared performance criteria, with respect to the others. In terms of number of iterations, the improvement is very important: in a factor between 3 and more than 40. With respect to our previous method based on full convergence using static parameters, the achieved speed-up factor is \(\sim 10\). Note also that in our method the number of iterations are fixed a priori, and, thus, they are predictable independently from the image and experiment. Furthermore, as the iterations of our method are extremely simple, we have also reduced typical times per iteration, which are now in the range of milliseconds. We have obtained times per iteration around 0.013 s (256² pixels) and 0.050 s (512²) for the single frame implementation (TIHP) and around 0.036 s (256²) and 0.115 s (512²) for the dual frame (TIHP+DTCWT) implementation in our combined Matlab-C++ implementation in an Intel Xeon 2-quad 2.26 GHz machine. This translates into more than 100 speed-up factor w.r.t. to our closest competitors. Note that here we only used traditional computation means (we did not use any graphical cards, FPGA, etc.)

6. RESULTS AND DISCUSSION

Table 2 shows the performance of our double frame imple-

\[mentation for Cameraman (256² pixels), for different number of iterations, and comparison with what to the best of our knowledge are the state-of-the-art methods in terms of Increment of Signal-to-Noise ratio (ISNR), in dBs. MFS denotes "multi-frame static", which corresponds to the method used here, but with static thresholding. CD(5), CD(10), and CD(20) corresponds to the described method (ConDy, from CONstrained DYnamic), using 5, 10 and 20 iterations. We can see that in most cases our method provides the best ISNR results, only surpassed in two of the seven experiments by [7].

Table 3 shows average performance for a subset of 5 ex-

\[periments using a single frame (a redundant version of the Haar wavelet, TIHP in our case), in ISNR, number of it-

\[erations and time, in seconds. We compare our method to recent sparsity-based methods using also a redundant Haar frame (data taken from table in [12]). The averaged results correspond to degradation tests (PSF1 and PSF3 both with \(\sigma^2_w = 2\) and \(\sigma^2_w = 8\), PSF2 with \(\sigma^2_w = 0.56^2\)). We note a substantial improvement of our method, in the three compared performance criteria, with respect to the others. In terms of number of iterations, the improvement is very important: in a factor between 3 and more than 40. With respect to our previous method based on full convergence using static parameters, the achieved speed-up factor is \(\sim 10\). Note also that in our method the number of iterations are fixed a priori, and, thus, they are predictable independently from the image and experiment. Furthermore, as the iterations of our method are extremely simple, we have also reduced typical times per iteration, which are now in the range of milliseconds. We have obtained times per iteration around 0.013 s (256² pixels) and 0.050 s (512²) for the single frame implementation (TIHP) and around 0.036 s (256²) and 0.115 s (512²) for the dual frame (TIHP+DTCWT) implementation in our combined Matlab-C++ implementation in an Intel Xeon 2-quad 2.26 GHz machine. This translates into more than 100 speed-up factor w.r.t. to our closest competitors. Note that here we only used traditional computation means (we did not use any graphical cards, FPGA, etc.)

7. REFERENCES


[2] Implemented code for reproducing the results reported here available to the public in Portilla’s web page, after publication of this work.
<table>
<thead>
<tr>
<th>$N_i$</th>
<th>$\alpha$</th>
<th>$\sigma_w$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4.87</td>
<td>43.80</td>
<td>0.41</td>
</tr>
<tr>
<td>10</td>
<td>6.48</td>
<td>39.16</td>
<td>0.65</td>
</tr>
<tr>
<td>20</td>
<td>7.64</td>
<td>32.06</td>
<td>0.83</td>
</tr>
<tr>
<td>50*</td>
<td>9.60</td>
<td>10.21</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. Optimized parameters, for some number of iterations ($N_i$). *Static optimization (MFS in Table 2).

<table>
<thead>
<tr>
<th>Blur</th>
<th>$\sigma_w$</th>
<th>PSF1</th>
<th>PSF2</th>
<th>PSF3</th>
<th>PSF4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_w$</td>
<td>2</td>
<td>8</td>
<td>0.56*</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>[6]</td>
<td>7.46</td>
<td>5.24</td>
<td>8.16</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[7]</td>
<td>8.29</td>
<td>6.34</td>
<td>8.58</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[8]</td>
<td>7.58</td>
<td>5.70</td>
<td>8.23</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[5]</td>
<td>7.45</td>
<td>5.55</td>
<td>7.33</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[4]</td>
<td>7.70</td>
<td>5.55</td>
<td>9.10</td>
<td>4.81</td>
<td>3.75</td>
</tr>
<tr>
<td>MFS</td>
<td>7.86</td>
<td>5.64</td>
<td>9.19</td>
<td>5.02</td>
<td>3.96</td>
</tr>
<tr>
<td>CD(5)</td>
<td>7.89</td>
<td>5.60</td>
<td>8.75</td>
<td>5.09</td>
<td>3.95</td>
</tr>
<tr>
<td>CD(10)</td>
<td>7.92</td>
<td>5.83</td>
<td>9.22</td>
<td>5.22</td>
<td>4.09</td>
</tr>
<tr>
<td>CD(20)</td>
<td>7.74</td>
<td>5.84</td>
<td>9.34</td>
<td>5.06</td>
<td>4.01</td>
</tr>
</tbody>
</table>

Table 2. ISNR, (in dB) of our double frame implementation for Cameraman, compared to state-of-the-art methods.

<table>
<thead>
<tr>
<th>Blurring</th>
<th>PSF1</th>
<th>PSF2</th>
<th>PSF3</th>
<th>PSF4</th>
</tr>
</thead>
<tbody>
<tr>
<td>FISTA [9]</td>
<td>4.66</td>
<td>180</td>
<td>108.992</td>
<td></td>
</tr>
<tr>
<td>TwIST2 [10]</td>
<td>5.16</td>
<td>62</td>
<td>28.512</td>
<td></td>
</tr>
<tr>
<td>SALSA [12]</td>
<td>5.36</td>
<td>16</td>
<td>8.164</td>
<td></td>
</tr>
<tr>
<td>ConDy(5)</td>
<td>5.98</td>
<td>5</td>
<td>0.067</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Average performance using a redundant Haar frame.

Fig. 1. From left to right, top to bottom: Original; blurred (3rd. experiment); Result from [7]; CD(20) (9.34 dB).