This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2019.2914773, IEEE Transactions on Computational Imaging

Image Restoration and Reconstruction using Targeted Plug-and-Play Priors

Afonso M. Teodoro, *Student Member, IEEE,* José M. Bioucas-Dias, *Fellow, IEEE,* and Mário A. T. Figueiredo, *Fellow, IEEE*

***Abstract*—Leveraging current state-of-the-art denoisers to tackle other inverse problems in imaging is a challenging task, which has recently been the topic of significant research effort. In this paper, we present several contributions to this research front, based on two fundamental building blocks: (i) the recently-proposed *plug-and-play* framework, which allows combining iterative algorithms for imaging inverse problems with state-of-the-art image denoisers, used in black-box fashion; (ii) patch-based denoisers, using Gaussian mixture models (GMM). We exploit the adaptability of GMM to learn class-adapted denoisers, which opens the door to embedding a patch classification step in the algorithmic loop, yielding simultaneous restoration and semantic segmentation. We apply the proposed approach to several standard imaging inverse problems (deblurring, compressive sensing reconstruction, and super-resolution), obtaining results that are competitive with the state of the art.**

***Index Terms*—Image restoration, image reconstruction, Gaussian mixtures, ADMM, plug-and-play, class-adapted priors.**

I. Introduction

OVER the last decade, many methods have been proposed for image denoising. The ones that perform best, although very distinct on their approach, provide results that are very close to one another. This corroborates the hypothesis that these methods are not far from the theoretically best possible performance [1]. Whereas the vast majority of state-of-the-art denoisers are patch-based [2]–[6], that is, instead of dealing with the whole image, they divide it into small blocks and treat each of them independently, in more involved inverse imaging problems it is not clear how to generalize such approaches. This is, nonetheless, an important and relevant research direction, where [7]–[10] are but a few examples.

Recently, Venkatakrishnan *et al* [11] proposed a framework that leverages the advances in image denoising to solve other inverse problems, such as image deblurring, compressive imaging, super-resolution, among others. In their so-called *plug-and-play* (PnP) approach, the authors propose to *plug* a general-purpose denoiser into the iterations an *alternating direction method of multipliers* (ADMM) algorithm [12]. In a nutshell, the denoiser is used as a *black-box* and we need only to adjust how to deal with the observation operator, which depends on the particular problem at hand. Other algorithms, such as primal-dual, *iterative shrinkage/thresholding algorithm* (ISTA), the fast versions of ISTA (FISTA [13] and TwIST [14]), *approximate message passing* (AMP), or *regularization by denoising* (RED), can also naturally leverage off-the-shelf denoisers; for more details about these methods, we refer the reader to [15]–[20].

In this paper, we build upon the original PnP framework, which uses ADMM, and combine it with learned denoisers based on *Gaussian mixture models* (GMM), to address several image restoration/reconstruction problems. Furthermore, instead of learning a generic denoiser to be used in any setting, we train models that are targeted to specific classes of images, namely: text, faces, fingerprints, medical images. The rationale is that those denoisers should be able to capture well the characteristics of each class, and hence perform better than general-purpose denoisers [21], [22]. In parallel to this work, several authors have also proposed improving denoising/restoration results by narrowing down the set of exemplars/database to particular classes of images [23]–[26].

We begin by considering the scenario where the image class is assumed to be known *a priori*, and thus we may use a GMM-based denoiser targeted to that class directly; this is the case, for example, if it is known that the image being restored is of a face or a fingerprint. We then extend the approach to the case where the image being restored is from an unknown class and/or may contain regions from different classes (e.g., text and faces, or text and natural images). The proposed method may be seen as performing simultaneous segmentation and restoration, thus exploiting synergies between these two tasks [27]. Notice that our main goal is not to obtain a good or meaningful segmentation, but to use the segmentation to allow class-specific models to be exploited at each location of the image, thus our focus is on the restoration performance of the method. Nevertheless, we believe that this type of approach may contribute to bridging the gap between low-level (restoration) and mid-level (segmentation) image processing/analysis. Other approaches to joint restoration and segmentation include the work in [28], which considers only two-region segmentation, the work in [29], which is based on a Nash-equilibrium principle, or the more recent approach in [30], using deep learning.

Combining ADMM with a GMM-based denoiser (or any other black-box denoiser, for that matter), under the PnP framework, begs an obvious question: does the resulting algorithm keep the convergence guarantees of standard ADMM? In this paper, we rely on the findings presented in [31], [32] to show how the proposed method can be modified to ensure a positive answer to this question. However, as this is not the

Afonso M. Teodoro, José M. Bioucas-Dias, and Mário A. T. Figueiredo are with Instituto de Telecomunicações and Instituto Superior Técnico, Universidade de Lisboa, Portugal. E-mail: {afonso.teodoro, mario.figueiredo}@tecnico.ulisboa.pt, and bioucas@lx.it.pt.

This work was partially supported by the *Fundação para a Ciência e Tecnologia* (FCT), grants UID/EEA/50008/2013, ERANETMED/0001/2014 and BD/102715/2014.

focus of the paper, we purposely disregard the modifications, and *plug* the GMM-denoiser as presented in [5]. In practice, we still always observe convergence to a stationary point.

The paper is organized as follows. Section II describes the ADMM algorithm and how it can be applied to tackle several inverse imaging problems. Section III explains the intuition behind the PnP framework, as well as our choice of denoiser. Section III also builds upon the concept of targeted priors, by proposing two distinct ways of leveraging models that are trained for specific classes of images. Section III-D briefly describes how to ensure the proposed algorithm converges, based on the guarantees presented in [31], [32], while Section IV reports several experiments, exhibiting very competitive results. Finally, Section V concludes the paper and discusses directions for future research.

II. Tools and Problem Formulation

A. ADMM

We begin by providing a very brief review of the ADMM algorithm. Dating back to the 1970's [33], the last decade has shown that ADMM is a flexible and efficient tool, currently widely used in inverse imaging problems, as well as in machine learning, and several other areas [12], [34]. Strong convergence properties and modular structure are some of the properties that make ADMM such a convenient tool.

In its canonical formulation [12], ADMM is used to address problems of the form

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n,\,\mathbf{v}\in\mathbb{R}^m} \quad & f(\mathbf{x})+g(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{v}=\mathbf{b}, \end{aligned} \tag{1}$$

where functions $f:\mathbb{R}^n\to\bar{\mathbb{R}}=\mathbb{R}\cup\{+\infty\}$ and $g:\mathbb{R}^m\to\bar{\mathbb{R}}$ are closed, proper, and convex, and matrices $\mathbf{A}$ and $\mathbf{B}$, and vector $\mathbf{b}$ are of appropriate dimensions. The ADMM tackles (1) iteratively, by alternating minimizations with respect to $\mathbf{x}$ and $\mathbf{v}$, while keeping the other fixed, and updating of the so-called *Lagrange multipliers*, $\mathbf{u}$, that is

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + \frac{\rho}{2}\left\|\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{v}^{(k)}-\mathbf{b}+\mathbf{u}^{(k)}\right\|_2^2, \tag{2}$$

$$\mathbf{v}^{(k+1)} = \arg\min_{\mathbf{v}} g(\mathbf{v}) + \frac{\rho}{2}\left\|\mathbf{A}\mathbf{x}^{(k+1)}+\mathbf{B}\mathbf{v}-\mathbf{b}+\mathbf{u}^{(k)}\right\|_2^2, \tag{3}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)}+\mathbf{A}\mathbf{x}^{(k+1)}+\mathbf{B}\mathbf{v}^{(k+1)}-\mathbf{b}. \tag{4}$$

An issue of practical relevance is the choice of parameter ρ, which may strongly affect the practical performance of the algorithm and which is an active research topic [35].

B. Solving Imaging Inverse Problems via ADMM

In this paper, we will consider three different inverse problems, namely: cyclic deblurring, (block) compressive imaging, and super-resolution. We start by providing a brief description of each of these problems, as well how to tackle them using ADMM. The classical formulation of these problems is

$$\mathbf{y}=\mathbf{H}\mathbf{x}+\mathbf{n}, \tag{5}$$

where $\mathbf{y}\in\mathbb{R}^m$ is the observed data, $\mathbf{x}\in\mathbb{R}^n$ is the (vectorized) underlying image to be inferred, $\mathbf{H}$ is the observation matrix, and $\mathbf{n}$ is assumed to be Gaussian zero-mean with known variance σ^2. Matrix $\mathbf{H}$ depends on the type of problem: in image deblurring, $\mathbf{H}$ represents a convolution operator; in compressive imaging, $\mathbf{H}$ is a random measurements matrix; in super-resolution, $\mathbf{H}$ represents a blurring operator followed by sub-sampling. In general, $\mathbf{H}$ is either non-invertible or severely ill-conditioned; consequently, inferring $\mathbf{x}$ requires the help of a prior/regulariser that promotes particular characteristics on the image estimate. One of the standard choices is to look for the *maximum a posteriori* (MAP) estimate (under prior $p(\mathbf{x})$)

$$\hat{\mathbf{x}} = \operatorname*{argmax}_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p(\mathbf{x}), \tag{6}$$

$$= \operatorname*{argmin}_{\mathbf{x}} -\log p(\mathbf{y}|\mathbf{x}) - \log p(\mathbf{x}), \tag{7}$$

$$= \operatorname*{argmin}_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \phi(\mathbf{x}), \tag{8}$$

where (8) follows from the hypothesis that the noise is Gaussian with zero mean and variance σ^2, and $\phi(\mathbf{x})=-\log p(\mathbf{x})$ (up to an additive constant) is also known as the regulariser.

Comparing (8) with (1), it is straightforward to identify $f(\mathbf{x})=\frac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2$, and $g(\mathbf{v})=\sigma^2\phi(\mathbf{v})$, under the constraint $\mathbf{x}=\mathbf{v}$, thus $\mathbf{A}=\mathbf{I}$, $\mathbf{B}=-\mathbf{I}$, and $\mathbf{b}=\mathbf{0}$. Injecting these equalities in (2) and (3), we conclude that the latter corresponds to the *proximity operator* [36] of $(\sigma^2/\rho)\,\phi$, computed at $\mathbf{x}^{(k+1)}+\mathbf{u}^{(k)}$:

$$\mathbf{v}^{(k+1)} = \operatorname{prox}_{(\sigma^2/\rho)\phi}\left(\mathbf{x}^{(k+1)}+\mathbf{u}^{(k)}\right). \tag{9}$$

Recall that the *proximity operator* of some convex function $\psi:\mathbb{R}^n\to\bar{\mathbb{R}}$ is defined as

$$\operatorname{prox}_{\psi}(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \psi(\mathbf{x}), \tag{10}$$

and can be seen as the MAP solution of a denoising problem, where the argument of $\operatorname{prox}_{\psi}$ is the noisy data, the noise is Gaussian i.i.d. with unit variance, and the prior is $p(\mathbf{x})\propto\exp(-\psi(\mathbf{x}))$. This interpretation is at the root of the so-called PnP framework [11], as discussed in the next section.

With $f(\mathbf{x})=\frac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2$, problem (2) becomes quadratic, with solution

$$\mathbf{x}^{(k+1)} = \left(\mathbf{H}^T\mathbf{H}+\rho\,\mathbf{I}\right)^{-1}\left(\mathbf{H}^T\mathbf{y}+\rho\left(\mathbf{v}^{(k)}-\mathbf{u}^{(k)}\right)\right), \tag{11}$$

where the computational bottleneck is the matrix inversion. Although it may seem at first that this inversion should be avoided, if it can be computed efficiently, then it is in fact an advantage, since the resulting algorithm turns out to be very fast in practice [37], [38]. Furthermore, in the problems considered in this paper, these inversions can in fact be computed efficiently, as shown in the following paragraphs.

1) Cyclic Deblurring: in the case of cyclic/periodic deblurring, $\mathbf{H}\in\mathbb{R}^{n\times n}$ is a block-circulant matrix with circulant blocks, thus the inversion can be done in the 2D discrete Fourier domain. In this case, $\mathbf{H}$ can be factored as $\mathbf{H}=\mathbf{U}^T\mathbf{D}\mathbf{U}$, where $\mathbf{U}$ represents the 2D discrete Fourier transform matrix, $\mathbf{U}^T=\mathbf{U}^{-1}$ is its inverse (with $(\cdot)^T$ denoting conjugate transpose), and $\mathbf{D}$ is a diagonal matrix. Consequently,

$$(\mathbf{H}^T\mathbf{H}+\rho\,\mathbf{I})^{-1} = (\mathbf{U}^T\mathbf{D}\mathbf{U}\mathbf{U}^T\mathbf{D}^T\mathbf{U}+\rho\,\mathbf{I})^{-1} = \mathbf{U}^T(|\mathbf{D}|^2+\rho\,\mathbf{I})^{-1}\mathbf{U}.$$

The inversion of the diagonal matrix $|\mathbf{D}|^2+\rho\mathbf{I}$ has linear cost, and the multiplications by $\mathbf{U}$ and $\mathbf{U}^H$ can be done via the FFT algorithm, with cost $O(n\log(n))$.

2) Compressive Imaging: In compressive imaging [39], [40], $\mathbf{H}\in\mathbb{R}^{m\times n}$ is a Gaussian measurement matrix, with $m<n$. The solution to (2) is again given by the $n\times n$ inversion in (11), which may be too expensive to compute. Resorting to the matrix inversion lemma, it is possible to reduce the cost of the inversion by a factor of roughly $(n/m)^3$ (assuming that inverting a general $n\times n$ matrix costs $O(n^3)$):

$$(\mathbf{H}^T\mathbf{H}+\rho\mathbf{I})^{-1}=\frac{1}{\rho}\left(I-\mathbf{H}^T(\mathbf{H}\mathbf{H}^T+\rho\mathbf{I})^{-1}\mathbf{H}\right), \tag{12}$$

where the required inversion is now of size $m\times m$. Note, also, that this inversion is done only once, and can be precomputed, stored, and used at each iteration of the algorithm.

Another way of tackling this problem in an efficient manner is to consider that the Gaussian measurement matrix is applied to non-overlapping blocks of the image. With this, we are able to keep the same measurement ratio, reducing substantially the size of the matrix that needs to be inverted [41], [42].

3) Super-resolution: In image super-resolution $\mathbf{H}=\mathbf{SB}$, where $\mathbf{S}\in\mathbb{R}^{m\times n}$ is a sub-sampling operator or down-sampling matrix with factor d (with $n=d^2m$) along the columns and rows of the image, and $\mathbf{B}\in\mathbb{R}^{n\times n}$ is a periodic blurring operator as in the deblurring case. Once again, since $\mathbf{B}$ is block circulant with circulant blocks, it can be factored into $\mathbf{B}=\mathbf{U}^T\mathbf{D}\mathbf{U}$. Moreover, since operator $\mathbf{S}$ down-samples the image, $\mathbf{S}^T$ corresponds to up-sampling and filling the missing samples with zeros; consequently, $\mathbf{S}\mathbf{S}^T=\mathbf{I}_m\in\mathbb{R}^{m\times m}$ (identity matrix) and $\mathbf{S}^T\mathbf{S}\in\mathbb{R}^{n\times n}$ is a binary diagonal matrix with ones at the observed positions and zeros elsewhere. Following [43], [44], the inversion in (11) can be computed in closed form as

$$\begin{aligned}\left(\mathbf{H}^T\mathbf{H}+\rho\,\mathbf{I}\right)^{-1}&=\left(\mathbf{B}^T\mathbf{S}^T\mathbf{S}\mathbf{B}+\rho\,\mathbf{I}\right)^{-1},\\&=\left(\mathbf{U}^T\mathbf{D}^T\mathbf{U}\mathbf{S}^T\mathbf{S}\mathbf{U}^T\mathbf{D}\mathbf{U}+\rho\,\mathbf{I}\right)^{-1},\\&=\mathbf{U}^T\left(\mathbf{D}^T\mathbf{U}\mathbf{S}^T\mathbf{S}\mathbf{U}^T\mathbf{D}+\rho\,\mathbf{I}\right)^{-1}\mathbf{U},\\&=\mathbf{U}^T\left(\mathbf{D}^T\left(\frac{1}{d^2}\mathbf{J}_{d^2}\otimes\mathbf{I}_m\right)\mathbf{D}+\rho\,\mathbf{I}\right)^{-1}\mathbf{U},\end{aligned} \tag{13}$$

where $\mathbf{J}_{d^2}\in\mathbb{R}^{d^2\times d^2}$ is a matrix of ones and $\otimes$ denotes the Kronecker product. However, in (13), the matrix to be inverted is of size $n\times n$, which may be prohibitive. Further simplification can be achieved by first exploiting properties of the Kronecker product (with $\mathbf{1}_a$ denoting an $a-$dimensional vector filled with ones),

$$\begin{aligned}\mathbf{D}^T\left(\frac{1}{d^2}\mathbf{J}_{d^2}\otimes\mathbf{I}_m\right)\mathbf{D}&=\frac{1}{d^2}\mathbf{D}^T\left(\left(\mathbf{1}_{d^2}\mathbf{1}_{d^2}^T\right)\otimes\left(\mathbf{I}_m\mathbf{I}_m\right)\right)\mathbf{D}\\&=\frac{1}{d^2}\mathbf{D}^T\left(\mathbf{1}_{d^2}\otimes\mathbf{I}_m\right)\left(\mathbf{1}_{d^2}^T\otimes\mathbf{I}_m\right)\mathbf{D}\\&=\frac{1}{d^2}\left(\mathbf{D}^T[\mathbf{I}_m,\cdots,\mathbf{I}_m]^T\right)\left([\mathbf{I}_m,\cdots,\mathbf{I}_m]\,\mathbf{D}\right)\\&=\frac{1}{d^2}\left(\underline{\mathbf{D}}^T\underline{\mathbf{D}}\right)\end{aligned} \tag{14}$$

(where $\underline{\mathbf{D}}=[\mathbf{I}_m,\cdots,\mathbf{I}_m]\,\mathbf{D}$) and then applying the matrix inversion lemma to obtain

$$\left(\mathbf{H}^T\mathbf{H}+\rho\,\mathbf{I}\right)^{-1}=\frac{1}{\rho}\left(\mathbf{I}_n-\underline{\mathbf{D}}^T\left(d^2\rho\,\mathbf{I}_m+\underline{\mathbf{D}}\underline{\mathbf{D}}^T\right)^{-1}\underline{\mathbf{D}}\right), \tag{15}$$

where the inversion is now of size $m\times m$, which reduces the computational cost by a factor of $\left(d^2\right)^3=d^6$ [43].

III. Proposed Method

A. Plug-and-Play

The *plug-and-play* (PnP) framework was motivated by noticing that equation (10) can be interpreted as the MAP estimate (recall (8)) in a pure denoising problem (with unit noise variance), *i.e.*, with $\mathbf{H}=\mathbf{I}$, function ψ acting as regulariser. The rationale behind PnP is that, instead of designing a convex regulariser for which it is possible to efficiently compute the proximity operator, this proximity operator (which is essentially a denoiser) can be replaced by some state-of-the-art denoiser. In other words, the denoiser is *plugged* into the iterative algorithm, playing the role of the proximity operator of the regularizer [11]. We stress that iterative algorithms other than ADMM can also be used in a PnP fashion. For example, [15] uses a primal-dual splitting, which can include hard constraints, while the work in [16] uses FISTA [13] and has the advantage of being able to handle non-linear observation operators. In this work, we stick to the original PnP framework, which uses ADMM, for the following reasons: the inverse problems described above include only linear observation operators; the inversion in Eq. (11) can be efficiently computed in closed form, as explained in the previous section.

Amongst the many state-of-the-art denoising methods, we opted for a GMM-based, patch-based denoiser. The reasons for this choice are threefold: (i) it has been shown by several authors that a GMM is a good prior for clean image patches [4], [5], [45], [46]; (ii) a GMM prior can be efficiently learned from an external dataset of clean images [4], [45] or directly from noisy patches [5]; (iii) by learning the GMM prior from an external dataset of clean images, we open the door to developing class-specific priors/denoisers [21]. In principle, class adaptation should yield better results because, by training on images of a particular class allows the prior to better capture the characteristics of that class.

The GMM denoiser used in this paper was proposed in [5], but the training of the model is done using an external dataset, as in [45], instead of the noisy patches. After extracting patches from a dataset of clean images, the GMM is learned using the well-known *expectation-maximization* (EM) algorithm. Then, this model is used as a prior to denoise the observed patches which, in this particular situation, can be done by computing the *minimum mean squared error* (MMSE) estimate in closed form [5]. For the sake of completeness, the main aspects of the patch-based GMM denoiser are repeated here. Consider the additive Gaussian noise model for each image patch[1] $\mathbf{y}=\mathbf{x}+\mathbf{n}$, with $\mathbf{n}\sim\mathcal{N}(0,\sigma^2\mathbf{I})$, and a GMM

[1]Notice that whereas in the rest of the paper $\mathbf{x}$ and $\mathbf{y}$ denote full images, here they refer to an arbitrary individual patch.

prior

$$p(\mathbf{x}) = \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{x}; \mu_m, \mathbf{C}_m), \tag{16}$$

where $\alpha_1, ..., \alpha_K$ are the mixing weights and $\mathcal{N}(\cdot; \mu, \mathbf{C})$ denotes a Gaussian density of mean μ and covariance matrix $\mathbf{C}$. Without loss of generality, assume zero-mean components, that is $\mu_j = 0$, for $j = 1, \ldots, K$. Because the sum of two zero-mean Gaussians is a zero-mean Gaussian with covariance equal to the sum of the covariances, the marginal $p(\mathbf{y})$ is

$$p(\mathbf{y}) = \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{x}; 0, \mathbf{C}_m + \sigma^2 \mathbf{I}). \tag{17}$$

By Bayes' law, $p(\mathbf{x}|\mathbf{y}) = p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})/p(\mathbf{y})$, *i.e.*,

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{p(\mathbf{y})} \sum_{m=1}^{K} \alpha_m p(\mathbf{y}|\mathbf{x})\, \mathcal{N}(\mathbf{x}; 0, \mathbf{C}_m) \tag{18}$$

$$= \frac{1}{p(\mathbf{y})} \sum_{m=1}^{K} \alpha_m p_m(\mathbf{y})\, \frac{p(\mathbf{y}|\mathbf{x})\, \mathcal{N}(\mathbf{x}; 0, \mathbf{C}_m)}{p_m(\mathbf{y})}, \tag{19}$$

where $p_m(\mathbf{y}) = \mathcal{N}(\mathbf{y}; 0, \mathbf{C}_m + \sigma^2 \mathbf{I})$. The MMSE estimator of $\mathbf{x}$ is the posterior expectation, $\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{y}] = \int \mathbf{x}\, p(\mathbf{x}|\mathbf{y}) d\mathbf{x}$, thus

$$\hat{\mathbf{x}} = \frac{1}{p(\mathbf{y})} \sum_{m=1}^{K} \alpha_m p_m(\mathbf{y}) \underbrace{\int \mathbf{x}\, \frac{p(\mathbf{y}|\mathbf{x})\, \mathcal{N}(\mathbf{x}; 0, \mathbf{C}_m)}{p_m(\mathbf{y})}\, d\mathbf{x}}_{\hat{\mathbf{x}}_m}. \tag{20}$$

Each $\hat{\mathbf{x}}_m$ is the MMSE estimate of $\mathbf{x}$ from $\mathbf{y}$, using a single Gaussian prior (the m-th component of the GMM). This is well-known to be $\hat{\mathbf{x}}_m = \mathbf{C}_m (\mathbf{C}_m + \sigma^2)^{-1} \mathbf{y}$, finally yielding

$$\hat{\mathbf{x}} = \frac{\sum_{m=1}^{K} \alpha_m\, \mathcal{N}(\mathbf{y}; 0, \mathbf{C}_m + \sigma^2 \mathbf{I})\, \hat{\mathbf{x}}_m}{\sum_{m=1}^{K} \alpha_m\, \mathcal{N}(\mathbf{y}; 0, \mathbf{C}_m + \sigma^2 \mathbf{I})}, \tag{21}$$

or, using compact notation,

$$\hat{\mathbf{x}} = \sum_{m=1}^{K} \beta_m(\mathbf{y})\, \mathbf{v}_m(\mathbf{y}), \tag{22}$$

where

$$\mathbf{v}_m(\mathbf{y}) = \mathbf{C}_m \Big(\mathbf{C}_m + \sigma^2 \mathbf{I} \Big)^{-1} \mathbf{y}, \tag{23}$$

and

$$\beta_m(\mathbf{y}) = \frac{\alpha_m\, \mathcal{N}(\mathbf{y}; 0, \mathbf{C}_m + \sigma^2 \mathbf{I})}{\sum_{m=1}^{K} \alpha_m\, \mathcal{N}(\mathbf{y}; 0, \mathbf{C}_m + \sigma^2 \mathbf{I})}. \tag{24}$$

Notice that $\beta_m(\mathbf{y})$ is the posterior probability that (noisy) patch $\mathbf{y}$ belongs to the m-th component of the GMM and $\mathbf{v}_m(\mathbf{y})$ is the MMSE estimate of that patch, if we knew it had been generated by the m-th component of the GMM. Notice that both β_m and $\mathbf{v}_m$ depend on the noisy $\mathbf{y}$; consequently, although $\mathbf{v}_m(\mathbf{y})$ is a linear (affine, to be precise if $\mu_m \neq 0$) function of $\mathbf{y}$, the MMSE estimate $\hat{\mathbf{x}}$ is a nonlinear function thereof.

After computing the MMSE patch estimates, $\hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_N$, they are returned to their location and combined by averaging. The simplest choice is to use a straight average, which corresponds to solving the optimization problem (see, *e.g.*, [47])

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \quad \sum_{i=1}^{N} \|\hat{\mathbf{x}}_i - \mathbf{P}_i \mathbf{x}\|_2^2, \tag{25}$$

where $\mathbf{P}_i \in \{0,1\}^{p \times n}$ is a binary matrix that extracts the i-th patch (assumed to have p pixels) from the image (thus $\mathbf{P}_i^T$ puts the patch back into its place) and N is the number of patches. The solution to (25) is

$$\hat{\mathbf{x}} = \Big(\sum_{i=1}^{N} \mathbf{P}_i^T \mathbf{P}_i \Big)^{-1} \sum_{i=1}^{N} \mathbf{P}_i^T \hat{\mathbf{x}}_i. \tag{26}$$

Assuming that the patches are extracted with unit stride and periodic boundary conditions, every pixel belongs to exactly p patches and $\sum_{i=1}^{N} \mathbf{P}_i^T \mathbf{P}_i = p\mathbf{I}$, thus its inverse is $p^{-1}\mathbf{I}$ and (26) corresponds simply to averaging the p estimates of each pixel (one for each of the p patches that contain it).

B. Targeted Priors

By *targeted* (or *class-adapted*) priors (equivalently, denoisers), we mean those that are learned from images of a particular class, such as faces or text. In this paper, we distinguish two ways of leveraging this type of image priors. First, we consider the case where it is known that the image in hand belongs to a certain class (*i.e.*, that class is predominant in the image). Although this hypothesis is not satisfied in generic images, there are many applications where it is, namely in text and face image restoration, or in specific medical imaging modalities, where a known organ is being imaged.

Subsequently, we consider the scenario where several classes may be present in the image. In this case, rather than considering a single class-adapted GMM prior, we consider C different models, each targeted to one of a set of C classes. In other words, we have C different priors, each given by

$$p(\mathbf{x}|c) = \sum_{m=1}^{K^{(c)}} \alpha_m^{(c)}\, \mathcal{N}(\mathbf{x}; \mu_m^{(c)}, \mathbf{C}_m^{(c)}), \tag{27}$$

where $c \in \{1, ..., C\}$ is the class label. To estimate the i-th patch, we begin by classifying it into one of the classes, and then use the corresponding GMM to obtain an MMSE estimate of that patch (given by (22)), conditioned on its noisy version.

In this paper, we use two different ways to classify each patch. The simplest approach consists in classifying each patch independently, using the *maximum likelihood* (ML) criterion,

$$\hat{c}_i = \arg \max_{c \in \{1, ...C\}} p(\mathbf{x}_i|c), \tag{28}$$

where $\hat{c}_i$ denotes the class estimate of the i-th patch. A more sophisticated approach, which provides a more spatially coherent segmentation of the image, is to consider a *Markov random field* (MRF [48]) prior $p(\mathbf{c})$ (where $\mathbf{c}$ denotes the field of all the patch class labels), more specifically a Potts prior:

$$\hat{\mathbf{c}} = \arg \max_{\mathbf{c} \in \{1, ..., C\}^N} \log p(\mathbf{c}) + \sum_i \log p(\mathbf{x}_i|c_i). \tag{29}$$

We solve (29) using the α-expansion graph-cut algorithm proposed in [49]; for more details about MRF, in particular Potts priors for image segmentation, see [48]–[50]. We stress that our goal is to leverage the classification to provide a better restoration or reconstruction, not to obtain a state-of-the-art segmentation. By plugging this joint segmentation-denoising procedure in the iterations of ADMM, we take advantage of the synergy between these two problems, in the sense that

segmentation improves the restoration results and, at the same time, a better restoration contributes to a better segmentation.

C. Proposed PnP-GMM Algorithm

Finally, the complete proposed algorithm is as shown in Algorithm 1. In the case of a single known class, the classification step is omitted. The algorithm assumes that the parameters of each of the classes present in the image have been previously estimated from a collection of clean images belonging to those classes. We denote the set of parameters of each class as $\theta^{(c)} = \{\mu_1^{(c)}, ..., \mu_{K^{(c)}}^{(c)}, \mathbf{C}_1^{(c)}, ..., \mathbf{C}_{K^{(c)}}^{(c)}, \alpha_1^{(c)}, ..., \alpha_{K^{(c)}}^{(c)}\}$, for $c \in \{1, ..., C\}$, and the complete collection of parameters as $\theta = \{\theta^{(1)}, ..., \theta^{(C)}\}$.

Algorithm 1: Class-Adapted PnP-GMM

Data: $\mathbf{y}$, $\mathbf{H}$, θ
Result: Estimate $\widehat{\mathbf{x}}$
1 Set ADMM parameter ρ;
2 Initialization: $k = 0$, $\mathbf{u}^{(k)} = 0$, $\mathbf{v}^{(k)} = 0$;
3 **repeat**
4 Compute $\mathbf{x}^{(k+1)}$ via (11);
5 Compute $\mathbf{s} = \mathbf{x}^{(k+1)} + \mathbf{u}^{(k)}$;
6 Estimate $\gamma = noiseVariance(\mathbf{s})$;
7 Classify the patches of $\mathbf{s}$ via (28) or (29);
8 **for** $i \in$ *set of patches* **do**
9 $\hat{\mathbf{v}}_i = \text{denoise}\left(\mathbf{s}_i, \gamma, \theta^{(\hat{c}_i)}\right)$
10 **end**
11 $\mathbf{v}^{(k+1)} = (1/p)\sum_{i \in \text{patches}} \mathbf{P}_i^T \hat{\mathbf{v}}_i$;
12 $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{v}^{(k+1)}$;
13 $k \leftarrow k + 1$;
14 **until** *convergence*;

A comment about line 6 of the algorithm is in order. To maintain the formal equivalence between the denoising operation and the proximity operator of some regularizer, the noise variance assumed by the denoiser should be set to σ^2/ρ (see (9)). It turns out that, as shown by preliminary experiments (not reported here), better results are obtained by estimating the variance of the noise present in the image being denoised at each iteration, $\mathbf{s}$, rather than keeping it fixed at the formal choice σ^2/ρ.

D. A Note About Convergence

The convergence of the ADMM algorithm using a GMM-denoiser was studied in detail in [31], [32]. To summarize, the convergence proof reduces to showing that the GMM-denoiser can be modified in order to ensure that it corresponds to a proximity operator, thus the standard convergence guarantees of ADMM apply [51]. In [32], the modification results naturally from the fact that, in the problem therein addressed, the GMM prior and the image being estimated are from the same scene, thus we keep the posterior weights (see (24)) fixed, which makes the MMSE estimate an affine function.

In this paper, although we train each model with images from specific classes, none of them depicts the same scene that we are restoring. Thus, keeping the weights β_m fixed from the start, as in [32], does not make sense. To circumvent the resulting non-linearity, we assume that, after a number of iterations, the weights stop changing significantly, and freeze them, thus ensuring convergence from that point on. The first few iterations can be seen as a transient stage which, in practice, only changes the initial estimate of the target image. Furthermore, fixing the weights implies that the segmentation no longer changes. In the next section, we illustrate with an example in image deblurring that, in practice, convergence is still observed (see Figure 4, in Section IV-B).

IV. Experimental Results

This section presents results obtained with the methods described above. In order to simplify the analysis, in all experiment, we use GMMs with 20 components per class and patches of size 8×8.

A. Image Denoising

We begin by comparing the performance of the GMM-based denoiser with that of the well-known state-of-the-art methods BM3D [3], NCSR [52], and a recent convolutional neural network approach (termed DnCNN) [53]; the results are reported in Table I. It should be noted that in the case of image denoising, although we could tackle it via an instance of ADMM with $\mathbf{H} = \mathbf{I}$, we use the simpler approach of applying the denoiser directly. The denoisers in the columns labeled BM3D, NCSR, and *Internal*, operate on the noisy image without resorting to any external data. The method in the column labeled *Generic* corresponds to using a GMM learned from a dataset of clean generic images, in particular a random subset of the Berkeley image segmentation dataset [54]. The methods labeled *Text* and *Face* use GMMs targeted to text, trained on images made available in [21], and to faces, using the Gore dataset [55]. Finally, in the method labeled as *Multi-class*, the algorithm starts by using the generic GMM prior, but then it decides at each iteration which prior is best for each patch, as described above. Besides the priors mentioned above, we also learned GMMs for fingerprint images[2] and magnetic resonance images of the brain[3]. Concerning the DnCNN, we used the *flexible* model provided together with the code, which handles different noise levels; that model was trained using 128×3000 patches of size 50×50, from 400 images (180×180) of the Berkeley segmentation dataset. The complex architecture coupled with such a large training set lead to the best results on most of the test settings presented, at the expense of a very high computational cost during training.

For training the several external GMMs, we used about 5×10^5 clean patches, extracted from randomly selected images of each database. The mean of the patches is subtracted in the beginning of the training phase, leading to zero-mean models, and the estimated covariance matrix is full. The EM algorithm is run for 100 iterations or until the relative improvement of the log-likelihood on two consecutive iterations drops below a threshold (set to 10^{-5} in all the experiments).

[2]Available at: http://bias.csr.unibo.it/fvc2004/
[3]Available at: http://www.medinfo.cs.ucy.ac.cy/index.php/downloads/datasets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2019.2914773, IEEE Transactions on Computational Imaging

Fig. 1: Top row: 12 benchmark (generic) images; Bottom row: samples for text, face, fingerprints, and examples comprising multiple classes.

Concerning GMM-based denoisers, the results in Table I suggest that methods leveraging the input noisy image (internal) perform better than purely external methods, on generic images. This is to be expected, because the prior is learned from the noisy image itself, and thus we may say it is adapted to that specific image. Moreover, on images belonging to a specific class or containing several classes, using a combination of internal model learning together with class-adapted models provides a more robust and higher-quality solution. In Table I, the shaded cell emphasizes the best result using the GMM-denoiser. An exception to this observation is found for the face images and can be explained by the relatively small image size (only 80×60), which may not contain enough patches to train an accurate model from the observed noisy input.

In the remaining experiments with the external priors, the approach that selected the most adequate prior locally (labeled as *Multi*, for multiple classes) is more robust. In fact, when the input image belongs to a particular class, using multiple priors that are automatically selected leads, in the worst case scenario, to only a minor loss of performance, whereas using the wrong prior, for example a text prior when the input image is actually a face, leads to a significant performance drop.

The last row of Table I reports an experiment with real data, which includes 11 photographs of static scenes taken in a controlled environment [56]. In particular, the noisy images were taken with three different cameras, on high ISO settings and the authors also provide the ground truth images which were obtained by averaging 500 samples of the same scene. We used the 15 smaller images (cropped to 512×512), and the GMM-based denoiser achieves the highest PSNR. Moreover, the results also give supporting evidence to the hypothesis that using locally selected priors leads to slightly better performance, even in the case of generic images, due to the higher flexibility of the overall model.

Finally, Figure 2 illustrates the difference between choosing the simple maximum likelihood criterion (28) or the computationally more expensive Potts prior (29). From a PSNR point of view, it is not obvious which type of segmentation will produce better results which is ultimately our objective. However, since the latter provides a much smoother segmentation of the image and broadens the range of possible applications, we will use it in the experiments that follow, disregarding the additional computational cost.

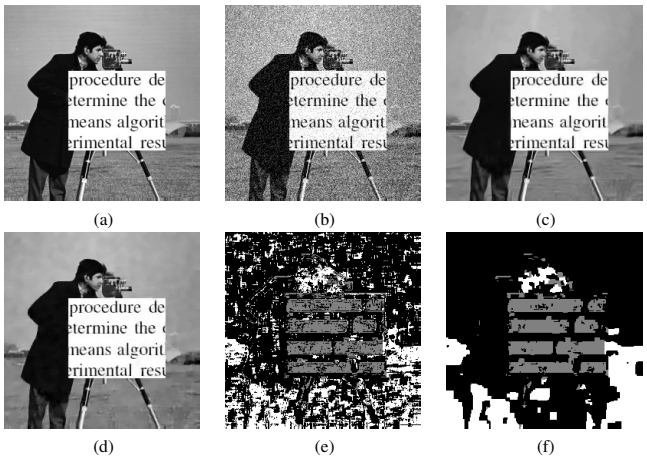

Fig. 2: Denoising: (a) original image; (b) noisy image ($\sigma = 30$); (c) denoised with BM3D [3] (PSNR = 27.80dB); (d) denoised with *Multi-class* GMM; (e) ML patch classification (PSNR = 27.94dB) (f) patch classification with Potts prior (PSNR = 27.92dB). Labels: black - generic prior; grey - text prior; white - face prior.

B. Image Deblurring

We assess the performance of the proposed PnP algorithm with different GMM priors, namely a prior for generic images, a prior targeted to specific classes of images, and a combination of both, in comparison with the state-of-the-art non-blind deblurring algorithm IDD-BM3D [7], on a total of six experiments (*i.e.*, different blur kernels and noise levels; see Table II). Moreover, for comparison purposes, we also use PnP with out-of-the-box BM3D and DnCNN denoisers.

For each experiment we run 200 iterations of ADMM on 12 standard generic benchmark images, 10 images of text, face, and fingerprints, and three montages combining different classes, namely generic with text, and a document including text and a face, and an image with all the five classes. As seen in the denoising results, learning the model from the noisy input patches usually leads to better performance thus, halfway through ADMM, we replace the general-purpose prior by another one, learned from the current image estimate. The underlying assumption is that, by then, we already have a good enough estimate of the final image. Besides the parameters of the GMM priors described above, one more parameter needs to be selected: the penalty parameter of ADMM, ρ, which was hand-tuned for best results over a grid of up to 20 possible values between 10^{-3} and 10. Regarding the convergence of the algorithm, with an adaptive noise level, we can not claim convergence to a global minimum, as in [32], as changing the parameter also changes the minimum itself.

Table III summarizes the results of deblurring experiments. We start by noticing that, as a general-purpose method, the proposed PnP algorithm does not achieve state-of-the-art per-

TABLE I: Comparison (PSNR) of several versions of GMM-based denoising versus BM3D [3], NCSR [52], and [53], for several values of noise standard deviation σ. *Internal:* the GMM is learned from the noisy image (as in [5]); *Generic:* the GMM is learned from a collection of generic images; *Text:* the GMM is learned from a collection of text images; *Face:* the GMM is learned from a collection of face images; *Multi:* generic or internal (respectively, *Gen-* or *Int-*) GMM denoiser, together with locally selected external prior with 4 classes (face, text, fingerprints, brain MRI); patch classification step using ML criterion or Potts prior (denoted by *-P*). The best result for each experiment is shown in bold and the second best in italic. Shaded cells show the best result using GMM-denoiser.

Image/Dataset	σ	BM3D	NCSR	DnCNN	*Internal*	*Generic*	*Text*	*Face*	*Gen-Multi*	*Gen-Multi-P*	*Int-Multi-P*
Set12	5	38.07	38.12	**38.26**	*38.16*	37.78	36.63	35.40	37.83	37.84	*38.16*
	15	32.38	*32.43*	**32.86**	32.33	31.92	30.97	30.75	32.01	32.02	32.33
	30	*29.14*	29.02	**29.64**	28.94	28.58	28.08	28.02	28.67	28.67	28.94
	50	*26.73*	26.54	**27.28**	26.60	26.29	26.11	26.04	26.37	26.36	26.57
Text (avg.)	5	39.82	39.03	39.56	*40.59*	39.63	39.82	28.40	40.43	40.22	**40.73**
	15	31.08	31.05	**32.61**	32.01	30.34	31.35	23.32	31.69	31.60	*32.14*
	30	25.64	26.10	*27.70*	26.87	24.93	26.35	21.07	26.47	26.43	*26.92*
	50	20.83	22.29	**23.79**	*23.29*	21.34	22.85	19.35	22.78	22.75	23.19
Face (avg.)	5	39.86	39.63	40.01	39.65	39.67	37.53	**40.33**	*40.29*	*40.29*	39.68
	15	33.53	33.02	**33.97**	32.50	33.10	31.57	*33.90*	33.82	33.85	32.55
	30	29.54	28.77	**30.01**	28.31	29.11	28.43	*29.84*	29.64	29.74	28.26
	50	26.85	25.68	**27.12**	25.25	26.26	25.92	*26.91*	26.68	26.81	25.17
Fingerprint (avg.)	5	39.56	*39.59*	39.26	**39.62**	39.21	37.77	39.37	39.57	39.56	**39.62**
	15	**34.75**	34.53	34.46	34.44	34.02	33.29	34.37	*34.57*	34.54	34.47
	30	**31.44**	*31.17*	31.05	30.91	30.58	30.28	30.89	30.96	31.02	30.97
	50	**28.99**	28.62	28.22	28.47	28.42	28.30	28.56	28.59	*28.68*	28.52
Cameraman + Text	5	38.46	38.21	38.48	*38.59*	38.36	37.99	32.11	38.56	38.58	**38.73**
	15	31.58	31.52	**32.34**	31.74	31.30	31.09	27.27	31.74	31.77	*31.95*
	30	27.80	27.74	**28.75**	27.83	27.33	27.79	24.94	27.93	27.96	*28.10*
	50	24.52	25.00	**26.11**	25.17	24.58	25.25	23.24	25.26	25.27	*25.35*
Face + Text	5	35.38	*37.07*	**38.22**	36.79	36.49	35.99	25.58	36.38	36.34	36.78
	15	28.78	29.39	**30.78**	29.64	28.37	28.89	21.13	29.16	29.12	*29.67*
	30	23.82	24.74	**25.83**	24.87	23.03	24.18	19.23	24.27	24.26	*24.93*
	50	19.96	21.38	**22.17**	21.66	19.78	21.03	17.96	20.96	21.04	*21.67*
5 Classes	5	38.00	*38.33*	**38.59**	38.20	37.83	36.77	32.12	38.03	38.03	38.24
	15	31.96	*31.99*	**32.65**	31.81	31.28	30.91	27.26	31.79	31.78	31.92
	30	27.87	27.86	**28.84**	27.79	27.03	27.34	24.85	27.75	27.76	*27.94*
	50	24.64	24.72	**25.96**	25.02	24.24	24.81	23.18	24.94	24.99	*25.11*
Real	-	36.29	36.39	36.36	37.11	37.16	-	-	**37.22**	*37.19*	37.12

TABLE II: Deblurring: Kernel point-spread function and additive noise variance for each experiment.

Experiment	**Kernel PSF**	σ^2
1	$1/(1+x_1^2+x_2^2)$, $x_1,x_2=-7,\ldots,7$	2
2	$1/(1+x_1^2+x_2^2)$, $x_1,x_2=-7,\ldots,7$	8
3	Uniform, 9×9	≈ 0.3
4	$[1\,4\,6\,4\,1]^T\,[1\,4\,6\,4\,1]/256$	49
5	Gaussian, $std=1.6$	4
6	Gaussian, $std=0.4$	64

formance. On the other hand, when dealing with images of particular classes, the PnP algorithm exhibits better results. Yet, the most important conclusion we can draw from these experiments has to do with the use of class-adapted versus generic priors. When the input image belongs to a single and known class, using a prior targeted to that class yields the best results, as would be expected (see the results for the *text* and *face* images, in the *PnP-GMM (Targ)* row). Considering several possible classes and letting the algorithm decide which prior to use locally is a more computationally expensive approach but it is more robust and provides results that are just as good in most of the experiments. Additionally, it provides a straightforward way to segment the image, which may be useful in some applications, such as document analysis [57]. In any case, either of these approaches outperform general-purpose methods, when the targeted priors cover the input class, and the multi-class approach outperforms the generic one every time, as the prior is broader.

Figure 3 illustrates how we can leverage the synergy between reconstruction and segmentation to improve the final results: on the one hand, as we deblur the input image we are able to achieve a better segmentation; on the other hand, taking into account the segmentation we can apply targeted priors locally, and thus improve the restoration. In this example, once again, we used the MRF-based classification criterion (29), as it provides a smoother segmentation of the image; white represents patches that are best represented with the prior for faces, grey for text, and black for the other priors, including the generic, fingerprints, and brain MRI.

Finally, Table IV presents results obtained on four real blurred images. Once again, the images were carefully obtained on a controlled environment, and thus the ground truths

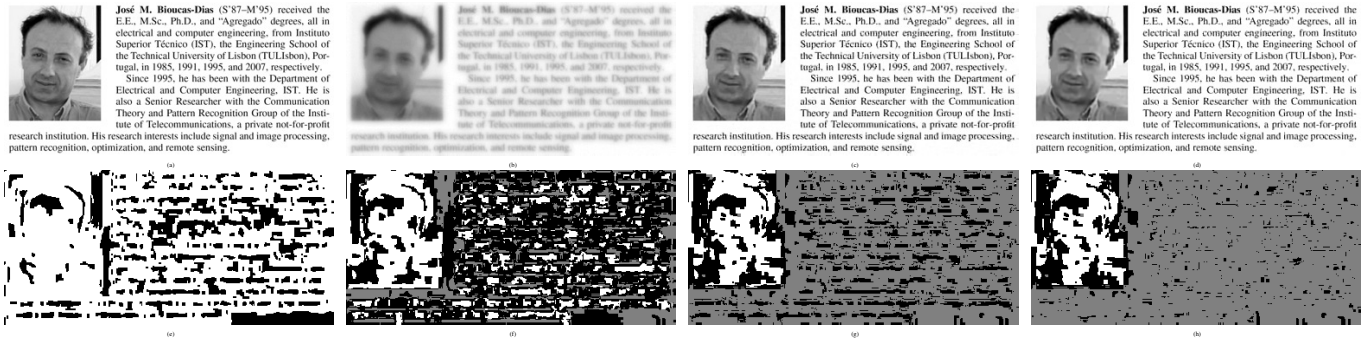

Fig. 3: Deblurring: (a) input image; (b) blurred image (Experiment 1); (c) deblurred with IDD-BM3D [7] (ISNR = 9.71dB); (d) deblurred with PnP-GMM with α-expansion (ISNR = 10.53dB); (e) α-expansion (1st iteration); (f) α-expansion (10th iteration); (g) α-expansion (50th iteration); (h) α-expansion (100th iteration). Labels: black - generic / brain MRI / fingerprints prior; grey - text prior; white - face prior.

of the eight real kernels are also available, allowing us to test our non-blind deblurring method (see [58]). For each blurring kernel, we average the results over the four images, using the same methods as before, namely the proposed PnP-GMM method with the generic prior and with locally selected multiple priors (same classes as before), PnP-BM3D, and PnP-DnCNN. We do not compare with IDD-BM3D because we could not achieve any reasonable results for this testing scenario; a possible reason for this is that IDD-BM3D requires a very good initialisation, provided by another method of the same authors, called BM3D-DEB [59], and the latter fails to do so. All the methods relying on ADMM, used the same penalty parameter, on all test settings, that is, $\rho = 0.05$, which is a value that produces reasonable overall results on the previous experiments. This was done to approximate a real scenario where the ground truth image is not available and thus we can not perform grid search for the best possible outcome. Furthermore, all the experiments were conducted with 40db input blurred signal-to-noise ratio (BSNR). Whereas in the denoising experiment we observe that DnCNN produces the best results, in image deblurring this is not the case. We conjecture that this is due to the network being trained and highly tuned to handle Gaussian noise, leading otherwise to suboptimal performance. Moreover, due to the absence of any of the classes considered in the targeted GMMs, the generic methods perform best.

Figure 4 illustrates the typical behaviour of the proposed PnP-GMM algorithm using a single model. As mentioned in Section III-D, although we do not guarantee the convergence of the method, in practice, we observe that the primal and dual residuals, computed as described in [12], vanish as the number of iterations increases, and the ISNR also stabilizes. The discontinuities halfway through the iterations are due to the replacement of the generic prior by the prior trained on the current image estimate, which shows that this approach contributes to improving the final result.

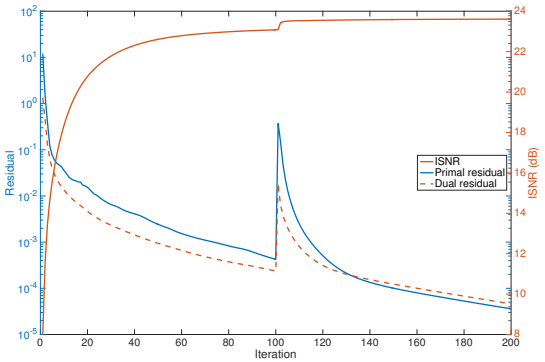

Fig. 4: Convergence of PnP-GMM.

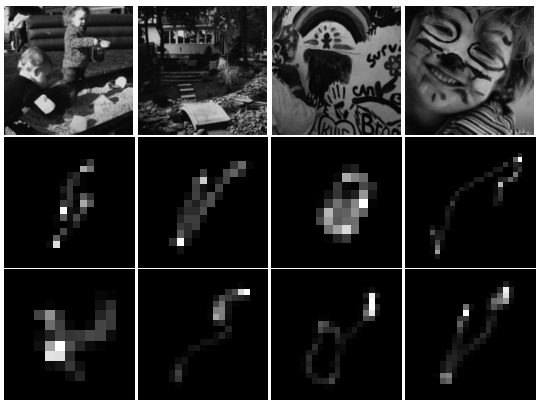

Fig. 5: Top row: dataset of 4 benchmark (generic) images; Middle and bottom rows: real blurring kernels from [58].

C. Compressive Imaging

In this Section, we repeat the analysis that was done for the image deblurring problem, this time letting the algorithm run for 300 iterations and replacing the generic prior by a learned model every 100 iterations, which showed improved performance. As before, the penalty parameter of ADMM was hand-tuned for best results. We compare the results of the proposed PnP algorithm against the method proposed in [42], which is based on a sparse representation of groups of image patches, for two different compression ratios. In compressive imaging with random Gaussian measurements, the spatial structure of the original image is totally absent from the observations, hence it is a reasonable approach to use the generic prior. Alternatively, we can use the generic prior together with the other class-adapted models, to account for the situations where the image is known to be, at least partly, from one of the considered classes. Note that we chose

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCI.2019.2914773, IEEE Transactions on Computational Imaging

TABLE III: ISNR results on image deblurring experiments. Methods: IDD-BM3D [7]; PnP with off-the-shelf BM3D and DnCNN denoisers; PnP with generic GMM prior (*Gen*); PnP with targeted GMM prior (*Targ*); PnP with multiple GMM priors and patch classification (*Multi*). The best result for each experiment is shown in bold and the second best in italics.

Image	Test images: Set12					
Experiment	1	2	3	4	5	6
IDD-BM3D	**8.76**	**6.87**	**10.11**	**4.08**	**4.59**	*5.02*
PnP-BM3D	7.41	*6.07*	9.40	*3.06*	3.69	4.84
PnP-DnCNN	7.14	3.94	*9.69*	1.43	2.31	**5.23**
PnP-GMM (*Gen*)	*8.08*	4.99	9.51	3.01	*3.84*	4.91
PnP-GMM (*Multi*)	8.01	5.84	9.27	2.96	3.78	4.90
Image	Test images: Text (avg.)					
Experiment	1	2	3	4	5	6
IDD-BM3D	12.49	9.45	16.94	5.92	6.53	5.70
PnP-BM3D	13.73	10.19	19.27	7.03	8.78	3.97
PnP-DnCNN	*15.21*	11.65	20.44	8.35	9.81	*6.00*
PnP-GMM (*Gen*)	14.50	11.08	20.34	8.02	9.85	5.35
PnP-GMM (*Targ*)	*15.21*	*11.98*	*20.57*	*9.02*	**10.70**	5.78
PnP-GMM (*Multi*)	**15.29**	**12.00**	**20.71**	**9.05**	*10.68*	**6.05**
Image	Test images: Face (avg.)					
Experiment	1	2	3	4	5	6
IDD-BM3D	13.01	10.54	14.40	6.86	9,76	6.32
PnP-BM3D	14.45	*12.09*	16.02	8.19	10.87	6.47
PnP-DnCNN	11.53	4.98	14.60	1.85	2.68	6.87
PnP-GMM (*Gen*)	**13.96**	*11.53*	16.13	7.68	10.01	*6.95*
PnP-GMM (*Targ*)	*13.94*	**11.67**	**16.50**	**7.96**	**10.62**	**7.15**
PnP-GMM (*Multi*)	13.82	11.49	*16.47*	*7.86*	*10.54*	6.86
Image	Test images: Fingerprint (avg.)					
Experiment	1	2	3	4	5	6
IDD-BM3D	**9.66**	8.22	11.75	5.92	6.77	6.90
PnP-BM3D	*9.57*	8.12	12.32	**6.25**	6.81	*7.12*
PnP-DnCNN	8.96	5.69	10.84	3.17	4.95	7.04
PnP-GMM (*Gen*)	9.45	8.27	12.28	6.03	*6.92*	7.05
PnP-GMM (*Targ*)	9.55	**8.42**	*12.44*	*6.20*	**6.95**	**7.13**
PnP-GMM (*Multi*)	9.39	*8.35*	**12.54**	5.98	6.72	*7.12*
Image	Test image: Cameraman + Text					
Experiment	1	2	3	4	5	6
IDD-BM3D	10.83	8.71	12.39	5.40	6.27	**4.89**
PnP-BM3D	11.13	8.64	12.66	5.15	7.12	*3.72*
PnP-DnCNN	10.70	7.52	12.56	4.75	6.46	3.66
PnP-GMM (*Gen*)	*11.28*	*8.95*	*12.71*	*5.49*	*7.50*	4.19
PnP-GMM (*Multi*)	**11.58**	**9.34**	**13.25**	**6.25**	**7.75**	*4.20*
Image	Test image: Face + Text					
Experiment	1	2	3	4	5	6
IDD-BM3D	9.71	6.57	11.24	2.43	1.79	**4.28**
PnP-BM3D	*10.42*	7.13	*12.02*	3.73	2.63	1.31
PnP-DnCNN	10.28	**7.64**	11.90	**4.47**	2.69	2.46
PnP-GMM (*Gen*)	10.39	7.01	11.26	*4.06*	*2.84*	*2.98*
PnP-GMM (*Multi*)	**10.53**	*7.39*	**12.14**	3.90	**2.89**	2.84
Image	Test images: 5 classes					
Experiment	1	2	3	4	5	6
IDD-BM3D	10.38	7.53	10.49	3.33	3.08	5.33
PnP-BM3D	10.04	6.66	10.63	2.66	3.29	4.92
PnP-DnCNN	9.19	6.01	10.42	2.23	2.39	**6.02**
PnP-GMM (*Gen*)	**10.98**	*7.93*	*11.33*	*3.38*	*3.69*	5.58
PnP-GMM (*Multi*)	*10.94*	**7.95**	**11.41**	**3.64**	**3.94**	*5.62*

GSR [42] as baseline because an implementation of the code is available, thus guaranteeing that same input image and random matrix is utilized on both methods. The results are summarized on Table V and, once again, using multiple priors achieves better results than a single generic prior on most cases.

TABLE IV: Deblurring: ISNR results on *semi*-real image deblurring experiments. Methods: PnP with BM3D; PnP with DnCNN; PnP with generic GMM prior (*Gen*); PnP with multiple GMM priors and patch classification (*Multi*). The best result for each experiment is shown in bold and the second best in italics.

Method	Kernel 1	Kernel 2	Kernel 3	Kernel 4	Kernel 5	Kernel 6	Kernel 7	Kernel 8
PnP-BM3D	19.89	*20.55*	18.97	**25.81**	20.10	27.30	**24.61**	**24.39**
PnP-DnCNN	19.16	19.19	18.18	21.93	20.06	**27.69**	23.13	21.39
PnP-GMM (Gen)	**20.15**	**20.72**	**19.22**	*25.75*	**20.42**	*27.49*	*24.53*	*24.29*
PnP-GMM (Multi)	*20.02*	20.52	*19.09*	25.17	*20.39*	27.43	24.43	23.95

D. Image Super-Resolution

This section reports results of image super-resolution experiments. Unlike in the previous experiments, we empirically observe that 50 iterations are enough to reach a plateau, with further iterations bringing only marginal changes and, furthermore, replacing the generic prior by a trained model halfway through the iterations also brings no considerable improvement. The inversion of the observation operator, see Eq. (15), is done resorting to the authors implementation[4], from [43], but replacing the total-variation (TV) denoiser with our patch-based (class-adapted) GMM denoiser. Regarding the input images, since all of them are in grayscale, there is no need to convert to luminance/chrominance (YCbCr) color space as is typically reported in super-resolution experiments.

Interestingly, we compare the performance of our method with another PnP approach [60], where the authors show that the plugging the denoiser proposed in [52] into ADMM could actually improve the results over generalizing the same denoiser to tackle super-resolution, as the authors had done in [52]. Once again, we also include our PnP approach with plugged-in BM3D and DnCNN denoiser for comparison. Table VI illustrates the results on two different experiments: the first one, for a down-sampling factor of 4 and no noise, that is, $d = 4$ and $\sigma = 0$, and the second with down-sampling factor $d = 3$, and standard deviation of the input noise $\sigma = 10$. All the methods seem to be on the same ballpark; however, it is worth mentioning that our PnP-GMM approach handles every situation the same way, whereas PnP-NCSR [60] tackles the noiseless case differently. In the same line of thought, which argues that class-adapted models should perform better than general-purpose ones, methods tuned to specific situations should also be better.

V. Conclusions

In this paper, we leveraged the flexible PnP framework to tackle a variety of inverse imaging problems, under a GMM patch-based image prior. In particular, this choice of prior has two important features: (i) we can learn the model from external datasets of clean images [45], opening the door to class-adapted models, which have recently been shown to yield better performance than the traditional general-purpose priors [21], [22]; (ii) being a probabilistic model allows classifying each patch in a set of possible classes, jointly performing restoration and segmentation [27].

Experimental results presented in this paper show that the GMM prior is a good choice as competitive results can

[4]Available at http://zhao.perso.enseeiht.fr/

TABLE V: PSNR on compressive imaging with Gaussian measurement matrix. Methods: GSR [42]; PnP with off-the-shelf BM3D and DnCNN denoisers; PnP with generic GMM prior (*Gen*); PnP with multiple GMM priors and patch classification (*Multi*). The best result for each experiment is shown in bold and the second best in italics.

Image class:	Set12		Text (Avg.)		Face (Avg.)		Fingerprints (Avg.)		Cameraman + Text		Face + Text		5 Classes	
Compression ratio:	10%	30%	10%	30%	10%	30%	10%	30%	10%	30%	10%	30%	10%	30%
GSR [42]	**26.58**	*32.92*	14.75	26.03	31.85	41.23	**35.51**	41.25	18.73	26.65	13.33	22.42	22.08	31.42
PnP-BM3D	23.89	31.75	16.40	25.75	30.56	42.19	30.41	40.87	19.61	26.73	15.31	20.83	21.90	28.61
PnP-DnCNN	*26.18*	**33.32**	18.10	**33.09**	27.73	40.33	29.68	39.70	20.99	31.24	13.32	25.07	21.52	31.70
PnP-GMM (Gen)	24.75	31.98	*18.92*	30.41	**32.99**	*42.42*	*33.78*	**41.56**	*20.04*	*28.98*	*15.51*	*24.37*	*22.18*	*31.92*
PnP-GMM (Multi)	24.27	32.55	**19.10**	*31.48*	*32.13*	**42.88**	32.27	*41.39*	**20.75**	**30.15**	**15.79**	**25.28**	**22.19**	**32.10**

TABLE VI: PSNR on image super-resolution experiments. Methods: PnP with NCSR [60]; PnP with off-the-shelf BM3D denoiser; PnP with generic GMM prior (*Gen*); PnP with multiple GMM priors and patch classification (*Multi*). The best result for each experiment is shown in bold and the second best in italics.

Image class:	Set12		Text (Avg.)		Face (Avg.)		Cameraman + Text		Face + Text		5 Classes	
Compression ratio:	1	2	1	2	1	2	1	2	1	2	1	2
PnP-NCSR	24.66	23.92	14.23	16.75	**28.61**	*27.25*	19.09	**20.53**	16.63	*16.67*	21.85	22.38
PnP-BM3D	25.16	*25.06*	*14.67*	*16.92*	*28.42*	27.14	*19.35*	*20.39*	16.60	16.31	21.99	21.89
PnP-GMM (Gen)	*25.57*	*25.06*	14.54	16.42	28.27	27.20	**19.36**	20.30	**16.84**	**16.69**	*22.01*	*22.61*
PnP-GMM (Multi)	**25.71**	**25.20**	**14.82**	**16.97**	28.36	**27.28**	19.34	20.38	*16.79*	*16.67*	**22.09**	**22.71**

be achieved in a variety of tasks, always using the same models. Furthermore, the results corroborate the hypothesis that targeted priors lead to better performance, not only in denoising, but in other inverse imaging problems as well.

REFERENCES

[1] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," *IEEE Conf. Computer Vision and Pattern Recognition*, 2011.

[2] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 60–65.

[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, pp. 2080–2095, 2007.

[4] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: from Gaussian mixture models to structured sparsity," *IEEE Trans. Image Proc.*, vol. 21, pp. 2481–2499, 2012.

[5] A. Teodoro, M. Almeida, and M. Figueiredo, "Single-frame image denoising and inpainting using Gaussian mixtures," in *4th International Conference on Pattern Recognition Applications and Methods*, 2015.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[7] A. Danielyan, V. Katkovnik, , and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Transactions on Image Processing*, vol. 21, pp. 1715–1728, 2012.

[8] M. Zhao, W. Zhang, Z. Wang, and F. Wang, "Spatially adaptive image deblurring based on nonlocal means," in *Int. Cong. Image and Signal Proc.*, Oct 2010, vol. 2, pp. 853–858.

[9] J. Xu, C. Qi, and Z. Chang, "Coupled K-SVD dictionary training for super-resolution," in *IEEE Int. Conf. Image Proc.*, Oct 2014, pp. 3910–3914.

[10] M. Mignotte, "A non-local regularization strategy for image deconvolution," *Patt. Rec. Lett.*, vol. 29, pp. 2206–2212, 2008.

[11] S. Venkatakrishnan, C. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," *IEEE Global Conf. Signal Inf. Proc.*, pp. 945–948, 2013.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learning*, vol. 3, pp. 1–122, 2011.

[13] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[14] J.M. Bioucas-Dias and M.A.T. Figueiredo, "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, pp. 2992–3004, 2007.

[15] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Proc. Letters*, vol. 24, no. 8, pp. 1108–1112, Aug 2017.

[16] U. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Proc. Letters*, vol. 24, no. 12, pp. 1872–1876, Dec 2017.

[17] C. Metzler, A. Maleki, and R. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Information Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[18] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[19] Y. Sun, B. Wohlberg, and U. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Computational Imaging*, pp. 1–1, 2019.

[20] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Trans. Image Proc.*, vol. 28, no. 3, pp. 1220–1234, March 2019.

[21] E. Luo, S. Chan, and T. Nguyen, "Adaptive image denoising by targeted databases," *IEEE Trans. Image Proc.*, vol. 24, pp. 2167–2181, 2015.

[22] A. Teodoro, J. Bioucas-Dias, and M. Figueiredo, "Image restoration and reconstruction using variable splitting and class-adapted image priors," in *IEEE International Conference on Image Processing*, 2016.

[23] S. Anwar, F. Porikli, and C. Huynh, "Category-specific object image denoising," *IEEE Trans. Image Proc.*, vol. 26, no. 11, pp. 5506–5518, Nov 2017.

[24] M. Niknejad, J. Bioucas-Dias, and M. Figueiredo, "Class-specific image denoising using importance sampling," in *IEEE Int. Conf. Image Proc.*, Sep. 2017, pp. 1242–1246.

[25] M. Ljubenović and M. Figueiredo, "Blind image deblurring using class-adapted image priors," in *IEEE Int. Conf. Image Proc.*, Sep. 2017, pp. 490–494.

[26] T. Remez, O. Litany, R. Giryes, and A. Bronstein, "Class-aware fully convolutional Gaussian and Poisson denoising," *IEEE Trans. Image Proc.*, vol. 27, no. 11, pp. 5707–5722, Nov 2018.

[27] A. Teodoro, J. Bioucas-Dias, and M. Figueiredo, "Image restoration with locally selected class-adapted models," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2016.

[28] G. Paul, J. Cardinale, and I. Sbalzarini, "Coupling image restoration and segmentation: A generalized linear model/Bregman perspective," *Int. Journ. Comp. Vision*, vol. 104, pp. 69–93, 2013.

[29] M. Kallel, R. Aboulaich, A. Habbal, and M. Moakher, "A Nash-game approach to joint image restoration and segmentation," *Journ. Applied Mathematical Modelling*, vol. 38, 01 2011.

[30] D. Liu, B. Wen, X. Liu, and T. Huang, "When image denoising meets high-level vision tasks: A deep learning approach," in *Proc. Int. Joint Conf. Artificial Intelligence*, 2018.

[31] A. Teodoro, J. Bioucas-Dias, and M. Figueiredo, "Scene-adapted plug-and-play with convergence guarantees," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2017.

[32] A. Teodoro, J. Bioucas-Dias, and M. Figueiredo, "A convergent image fusion algorithm using scene-adapted Gaussian-mixture-based denoising," *IEEE Trans. Image Proc.*, vol. 28, no. 1, pp. 451–463, Jan 2019.

[33] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Comp. and Math. Appl.*, vol. 2, pp. 17–40, 1976.

[34] M. Figueiredo and J. Bioucas-Dias, "Algorithms for imaging inverse problems under sparsity regularization," *Int. Workshop Cognitive Inf. Proc.*, pp. 1–6, 2012.

[35] Z. Xu, M. Figueiredo, and T. Goldstein, "Adaptive ADMM with spectral penalty parameter selection," in *Proc. Intern. Conf. Artificial Intelligence and Statistics*, 2017, pp. 718–727.

[36] H. Bauschke and P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, 2011.

[37] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Proc.*, vol. 19, pp. 2345–2356, 2010.

[38] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Proc.*, vol. 20, pp. 681–695, 2011.

[39] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Proc. Mag.*, vol. 25, pp. 14–20, 2008.

[40] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a markov-tree prior," *IEEE Trans. Signal Proc.*, vol. 60, pp. 3439–3448, 2012.

[41] L. Gan, "Block compressed sensing of natural images," *IEEE Int. Conf. on Digital Signal Proc.*, 2007.

[42] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Proc.*, vol. 23, no. 8, pp. 3336–3351, Aug 2014.

[43] Q. Wei, N. Dobigeon, and J.-Y. Tourneret, "Fast fusion of multi-band images based on solving a Sylvester equation," *IEEE Trans. Image Proc.*, vol. 24, pp. 4109–4121, 2015.

[44] N. Zhao, Q. Wei, A. Basarab, D. Kouame, and J. -Y. Tourneret, "Fast single image super-resolution," *CoRR*, vol. abs/1510.00143, 2015.

[45] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE Int. Conf. Comp. Vision*, 2011, pp. 479–486.

[46] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *International Conference on Computer Vision*, 2011, pp. 479–486.

[47] Y. Romano and M. Elad, "SOS boosting of image denoising algorithms," *Computing Research Repository*, vol. abs/1502.06220, 2015.

[48] S. Li, *Markov Random Field Modeling in Computer Vision*, Springer-Verlag, 1995.

[49] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 23, pp. 1222–1239, 2001.

[50] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *IEEE Conf. Computer Vision and Pattern Recognition*, 1998.

[51] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Prog.*, vol. 5, pp. 293–318, 1992.

[52] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Proc.*, vol. 22, pp. 1620–1630, 2013.

[53] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Proc.*, vol. 26, no. 7, pp. 3142–3155, July 2017.

[54] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001.

[55] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," in *IEEE Conf. Computer Vision and Pattern Recognition*, June 2010, pp. 763–770.

[56] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *IEEE Conf. Computer Vision and Pattern Recognition*, June 2016, pp. 1683–1691.

[57] V. Shapiro, G. Gluhchev, and V. Sgurev, "Handwritten document image segmentation and analysis," *Patt. Rec. Letters*, vol. 14, no. 1, pp. 71 – 78, 1993.

[58] A. Levin, Y. Weiss, F. Durand, and W. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *IEEE Conf. Computer Vision and Pattern Recognition*, June 2009, pp. 1964–1971.

[59] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," 2008, vol. 6812.

[60] A. Brifman, Y. Romano, and M. Elad, "Turning a denoiser into a super-resolver using plug and play priors," *IEEE Int. Conf. on Image Proc.*, pp. 1404–1408, 2016.