

Algorithm-Induced Prior for Image Restoration

Stanley H. Chan, *Member, IEEE*

Abstract—This paper studies a type of image priors that are constructed implicitly through the alternating direction method of multiplier (ADMM) algorithm, called the algorithm-induced prior. Different from classical image priors which are defined before running the reconstruction algorithm, algorithm-induced priors are defined by the denoising procedure used to replace one of the two modules in the ADMM algorithm. Since such prior is not explicitly defined, analyzing the performance has been difficult in the past.

Focusing on the class of symmetric smoothing filters, this paper presents an explicit expression of the prior induced by the ADMM algorithm. The new prior is reminiscent to the conventional graph Laplacian but with stronger reconstruction performance. It can also be shown that the overall reconstruction has an efficient closed-form implementation if the associated symmetric smoothing filter is low rank. The results are validated with experiments on image inpainting.

Index Terms—Image reconstruction, image denoising, graph Laplacian, ADMM, symmetric smoothing filters.

I. INTRODUCTION

A. Background, Scope, and Related Work

Alternating direction methods of multipliers (ADMM) is perhaps the most popular algorithm for solving linear inverse problems in recent years, particularly for image restoration [1], [2]. Despite different perspectives of the algorithm, (e.g., operator splitting [3], proximal methods [4], split Bregman [5], to name a few,) the common principle behind ADMM is to convert the original minimization of the form

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + \lambda s(\mathbf{x}) \quad (1)$$

into an equivalent constrained problem

$$\begin{aligned} &\underset{\mathbf{x} \in \mathbb{R}^n, \mathbf{v} \in \mathbb{R}^n}{\text{minimize}} && f(\mathbf{x}) + \lambda s(\mathbf{v}) \\ &\text{subject to} && \mathbf{x} = \mathbf{v}, \end{aligned} \quad (2)$$

and solve the constrained problem by alternately minimizing the augmented Lagrangian function. Under mild conditions, e.g., when $f(\cdot)$ is strongly convex and $s(\cdot)$ is convex, the convergence of the algorithm is typically guaranteed [6].

In setting up the optimization problem in (1), the objective function $f(\cdot)$ and the regularization function $s(\cdot)$ are almost always fixed *before* running the algorithm. For example, when solving a non-blind deblurring problem using a total variation regularization [7], the functions $f(\cdot)$ and $s(\cdot)$ are

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2, \quad s(\mathbf{x}) = \|\mathbf{x}\|_{TV}, \quad (3)$$

where \mathbf{A} is the blur operator, \mathbf{y} is the observed image, and $\|\mathbf{x}\|_{TV}$ is the total variation norm of the image \mathbf{x} .

For most image restoration problems, $f(\mathbf{x})$ is chosen according to the forward imaging model, and is fixed as long

as we agree with the forward model. But $s(\mathbf{x})$ is the user's subjective belief of how the solution should look like, a.k.a. the *prior*. In literature, apart from the total variation prior mentioned in [3], there are enormous number of priors we can use. However, there is one thing in common, which is that $s(\mathbf{x})$ has to be defined before using the ADMM algorithm.

In this paper, I present an ADMM algorithm where the regularization function $s(\mathbf{x})$ is *unknown a-priori*. At a first glance, this might seem unnatural because if $s(\mathbf{x})$ is unknown, then it is unclear about what we are trying to optimize in (1). However, as will be discussed shortly, the ADMM algorithm can generally be written as two *modules* – *an inverse module*, and *a denoising module*. The idea is to *replace the denoising module by some off-the-shelf image denoising algorithm*, e.g., non-local means [8], [9] or BM3D [10]. In other words, we do not explicitly define $s(\mathbf{x})$ before running the algorithm, but use a denoising algorithm to perform the role of $s(\mathbf{x})$.

Replacing the denoising module of the ADMM algorithm by an off-the-shelf denoising algorithm was first proposed by Bouman and colleagues [11], to the best of my knowledge. Perhaps of the heuristic nature of the method, they call it the “Plug-and-Play” algorithm to stress that one can plug in any denoising algorithm and get the ADMM algorithm running. Under appropriate conditions on the denoising algorithm, one can prove the convergence of the plug-and-play [12].

In the context of *compressive sensing* [13]–[15], a similar version of the plug-and-play is also being studied. In [13], Baranuik and colleagues considered an approximate message passing (AMP) algorithm for recovering images. Recognizing that AMP also has a “inverse module” and a “denoising module”, they *replace the shrinkage step in the conventional AMP with an off-the-shelf denoising algorithm* (BM3D in their paper). Again, under appropriate conditions of the denoising algorithm, they proved that the AMP converges.

B. Contributions

The focus of this paper is not to find weaker conditions under which “plug-and-play” converges. Rather, I like to address another equally important question: *What is the original prior $s(\mathbf{x})$ if we choose a particular denoising algorithm?* Answering this problem is essential to understand this type of algorithms in general. To make the discussion concrete, I will focus on the class of symmetric smoothing filters [16]–[18] which is broad enough to include many denoising methods such as bilateral filter, non-local means [8] and LARK [19], but at the same time also allows us to exploit matrix structures, e.g., the graph Laplacian [20]. I call the new prior as an *algorithm-induced prior* to reflect the algorithmic nature of the prior.

The rest of the paper is organized as follows. I will first setup the problem in Section II. Then, in Section III, I will

address the question about the original prior of the ADMM-induced algorithm, and discuss linkages with the conventional graph Laplacian prior. Experimental results are presented in Section IV, and a conclusion is given in Section V.

II. CONCEPT OF ALGORITHM-INDUCED PRIOR

A. ADMM Algorithm

To begin the discussion I will first briefly introduce the ADMM algorithm. Interested readers can read [6] for additional technical details.

Given the constrained minimization (2), the ADMM algorithm defines the augmented Lagrangian function as

$$\mathcal{L}(\mathbf{x}, \mathbf{v}, \mathbf{u}) = f(\mathbf{x}) + \lambda s(\mathbf{v}) + \mathbf{u}^T(\mathbf{x} - \mathbf{v}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{v}\|^2. \quad (4)$$

where $\mathbf{u} \in \mathbb{R}^n$ is the Lagrange multiplier, and $\rho > 0$ is the half quadratic penalty parameter. The algorithm then proceeds to update each variable as follows

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \mathcal{L}(\mathbf{x}, \mathbf{v}^{(k)}, \mathbf{u}^{(k)}), \quad (5)$$

$$\mathbf{v}^{(k+1)} = \underset{\mathbf{v} \in \mathbb{R}^n}{\operatorname{argmin}} \mathcal{L}(\mathbf{x}^{(k+1)}, \mathbf{v}, \mathbf{u}^{(k)}), \quad (6)$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \rho(\mathbf{x}^{(k+1)} - \mathbf{v}^{(k+1)}). \quad (7)$$

The minimizations in (5) and (6) are known as the primal updates, whereas the descent step in (7) is the dual update. If both $f(\cdot)$ and $s(\cdot)$ are closed, proper and convex, and if $\mathcal{L}(\cdot)$ has a saddle point, then one can prove convergence of the ADMM algorithm in terms of primal residue, primal objective and dual variable [6]. In case when (5) and (6) are solved simultaneously instead of sequentially as presented above, then one will obtain the augmented Lagrangian method (ALM).

With some manipulations and rearrangement of terms we can show the following.

Proposition 1: The iterations (5)-(7) are equivalent to

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} f(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{x} - \tilde{\mathbf{x}}^{(k)}\|^2, \quad (8)$$

$$\mathbf{v}^{(k+1)} = \underset{\mathbf{v} \in \mathbb{R}^n}{\operatorname{argmin}} \lambda s(\mathbf{v}) + \frac{\rho}{2} \|\mathbf{v} - \tilde{\mathbf{v}}^{(k)}\|^2, \quad (9)$$

$$\bar{\mathbf{u}}^{(k+1)} = \bar{\mathbf{u}}^{(k)} + (\mathbf{x}^{(k+1)} - \mathbf{v}^{(k+1)}), \quad (10)$$

where $\bar{\mathbf{u}}^{(k)} \stackrel{\text{def}}{=} \frac{1}{\rho} \mathbf{u}^{(k)}$ is the scaled multiplier, $\tilde{\mathbf{x}}^{(k)} \stackrel{\text{def}}{=} \mathbf{x}^{(k)} - \bar{\mathbf{u}}^{(k)}$ and $\tilde{\mathbf{v}}^{(k)} \stackrel{\text{def}}{=} \mathbf{x}^{(k+1)} + \bar{\mathbf{u}}^{(k)}$.

The proof is skipped because it is essentially completing squares. The reason of rewriting the ADMM as above is to demonstrate the *modular* structure of the ADMM algorithm which we shall discuss shortly. As a side remark, the iterations (8)-(10) can be defined as a *proximal algorithm* [4].

To gain more insights into the modular structure presented in Proposition 1, let us consider the following example.

Example 1: If we use $f(\cdot)$ and $s(\cdot)$ given in (3), we observe that (8) and (9) become

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \frac{\rho}{2} \|\mathbf{x} - \tilde{\mathbf{x}}^{(k)}\|^2,$$

$$\mathbf{v}^{(k+1)} = \underset{\mathbf{v}}{\operatorname{argmin}} \lambda \|\mathbf{x}\|_{TV} + \frac{\rho}{2} \|\mathbf{v} - \tilde{\mathbf{v}}^{(k)}\|^2,$$

which is a reconstruction problem with a quadratic regularization, and a denoising problem (to denoise $\tilde{\mathbf{v}}^{(k)}$) with a total variation regularization, respectively.

B. Algorithm-Induced Prior

Recognizing the “denoising” module in the ADMM algorithm, we replace the \mathbf{v} -subproblem by a denoising algorithm. Formally, if we denote \mathcal{D}_h as the denoising algorithm, i.e.,

$$\mathcal{D}_h(\tilde{\mathbf{v}}^{(k)}) \stackrel{\text{def}}{=} \underset{\mathbf{v} \in \mathbb{R}^n}{\operatorname{argmin}} \lambda s(\mathbf{v}) + \frac{\rho}{2} \|\mathbf{v} - \tilde{\mathbf{v}}^{(k)}\|^2, \quad (11)$$

where the subscript $h > 0$ specifies the internal parameter of the denoising algorithm, then the \mathbf{v} -subproblem becomes

$$\mathbf{v}^{(k+1)} = \mathcal{D}_h(\tilde{\mathbf{v}}^{(k)}). \quad (12)$$

For example, if we choose a symmetric smoothing filter, then the denoising algorithm $\mathcal{D}_h : \mathbb{R}^n \rightarrow \mathbb{R}^n$ takes the form

$$\mathcal{D}_h(\tilde{\mathbf{v}}^{(k)}) = \mathbf{W}_h^{(k)} \tilde{\mathbf{v}}^{(k)}, \quad (13)$$

where $\mathbf{W}_h^{(k)} \in \mathbb{R}^{n \times n}$ is a doubly stochastic matrix. An explicit example of $\mathbf{W}_h^{(k)}$ using the non-local means is illustrated in the following example.

Example 2: Consider the non-local means [8] as an example. One can first construct a kernel matrix $\mathbf{K}_h^{(k)}$ with the (i, j) -th entry

$$[\mathbf{K}_h^{(k)}]_{ij} \stackrel{\text{def}}{=} \exp \left\{ -\|\tilde{\mathbf{v}}_i^{(k)} - \tilde{\mathbf{v}}_j^{(k)}\|^2 / (2h^2) \right\},$$

where $\tilde{\mathbf{v}}_i^{(k)}$ denotes the i -th patch of the input $\tilde{\mathbf{v}}^{(k)}$. Then, by applying Sinkhorn-Knopp balancing algorithm [21] one can determine a pair of diagonal matrices $\mathbf{R}^{(k)}$ and $\mathbf{C}^{(k)}$ such that

$$\mathbf{W}_h^{(k)} \stackrel{\text{def}}{=} \mathbf{R}^{(k)} \mathbf{K}_h^{(k)} \mathbf{C}^{(k)}$$

is a doubly stochastic matrix.

III. ANALYSIS OF ALGORITHM-INDUCED PRIOR

In this section I will address the question: What is the original prior $s(\mathbf{x})$ if we choose \mathcal{D}_h as a symmetric smoothing filter? For notational simplicity I will drop the scripts $(\cdot)^{(k)}$ and $(\cdot)_h$.

A. Original $s(\mathbf{x})$

The first main result is stated in the following proposition, which provides an explicit formula for the regularization $s(\mathbf{x})$ when symmetric smoothing filters are used.

Proposition 2: Let \mathbf{W} be a symmetric smoothing filter with $\operatorname{rank}(\mathbf{W}) = m$, and let \mathbf{W}^+ be the pseudo-inverse of \mathbf{W} , i.e., $\mathbf{W}^+ = \mathbf{U} \Sigma_+^{-1} \mathbf{U}^T$ where \mathbf{U} and $\Sigma = \operatorname{diag}\{s_1, \dots, s_m, 0, \dots, 0\}$ are the eigenvectors and eigenvalues of \mathbf{W} respectively, and $\Sigma_+^{-1} = \operatorname{diag}\{1/s_1, \dots, 1/s_m, 0, \dots, 0\}$. For a fixed \mathbf{W} , if

$$s(\mathbf{v}) = \frac{\rho}{2\lambda} \mathbf{v}^T (\mathbf{I} - \mathbf{W}) \mathbf{W}^+ \mathbf{v}, \quad (14)$$

then

$$\underset{\mathbf{v}}{\operatorname{argmin}} \lambda s(\mathbf{v}) + \frac{\rho}{2} \|\mathbf{v} - \tilde{\mathbf{v}}\|^2 = \mathbf{W} \tilde{\mathbf{v}} \stackrel{\text{def}}{=} \hat{\mathbf{v}}. \quad (15)$$

Proof: There are two ways of proving this proposition. The first way is a “reverse engineering” approach. By plugging (14) into (15) and setting the first order derivative to zero we can show that $\hat{\mathbf{v}} = \mathbf{W}\tilde{\mathbf{v}}$.

The alternative proof is a constructive one. First, we observe that in order to obtain $\mathbf{W}\tilde{\mathbf{v}}$ on the right hand side of (15), we must have $s(\mathbf{v})$ being quadratic. Therefore, we let

$$s(\mathbf{v}) = \alpha \mathbf{v}^T \mathbf{C} \mathbf{v},$$

for some symmetric matrix \mathbf{C} and constant α . Taking the first order derivative of the resulting function yields

$$\frac{d}{d\mathbf{v}} \left(\lambda s(\mathbf{v}) + \frac{\rho}{2} \|\mathbf{v} - \tilde{\mathbf{v}}\|^2 \right) = 2\lambda\alpha \mathbf{C} \mathbf{v} + \rho(\mathbf{v} - \tilde{\mathbf{v}}) = 0.$$

Rearranging the terms, we obtain a linear equation

$$\left(\frac{2\lambda}{\rho} \alpha \mathbf{C} + \mathbf{I} \right) \mathbf{v} = \tilde{\mathbf{v}}.$$

Since α can be arbitrary, we set $\alpha = \rho/(2\lambda)$. Consequently, we have $(\mathbf{C} + \mathbf{I})\mathbf{v} = \tilde{\mathbf{v}}$. Multiplying both sides with \mathbf{W} yields $\mathbf{W}(\mathbf{C} + \mathbf{I})\mathbf{v} = \mathbf{W}\tilde{\mathbf{v}}$. Thus, in order to obtain $\mathbf{v} = \mathbf{W}\tilde{\mathbf{v}}$, \mathbf{C} must be chosen such that

$$\mathbf{W}(\mathbf{C} + \mathbf{I}) = \mathbf{I},$$

which gives $\mathbf{C} = (\mathbf{I} - \mathbf{W})\mathbf{W}^+$. ■

There are a few important properties of the regularization $s(\mathbf{v})$ shown in (14). First, for any fixed \mathbf{W} , the matrix $(\mathbf{I} - \mathbf{W})\mathbf{W}^+$ is symmetric positive semidefinite. In fact, since the eigenvalues of a symmetric smoothing filter \mathbf{W} is always bounded between 0 and 1, i.e., $0 \preceq \Sigma \preceq \mathbf{I}$, it holds that the eigenvalues of $(\mathbf{I} - \mathbf{W})\mathbf{W}^+$ is also bounded between 0 and 1. Therefore, if \mathbf{W} is pre-defined before running the ADMM algorithm, then $s(\mathbf{x})$ is convex and hence the overall optimization is also convex.

Second, if we compare (14) with the conventional graph Laplacian regularization $s(\mathbf{v}) = \mathbf{v}^T \mathbf{L} \mathbf{v}$ in the literature [22], where $\mathbf{L} \stackrel{\text{def}}{=} \mathbf{I} - \mathbf{W}$, we observe that (14) has an additional term \mathbf{W}^+ . Using a graph signal processing terminology, we can view \mathbf{W} as a lowpass filter and \mathbf{L} is a highpass filter. \mathbf{W}^+ is a bandpass filter because of the truncation property of the pseudo-inverse. Therefore, the regularization $\mathbf{v}^T (\mathbf{I} - \mathbf{W}) \mathbf{W}^+ \mathbf{v}$ penalizes a smaller (but more focused) set of graph frequencies than the conventional regularization $\mathbf{v}^T \mathbf{L} \mathbf{v}$. In Section IV we will compare the performance.

B. Closed-form Solution

Proposition 2 suggests a new prior which deserves a closer look. First of all, assume, for simplicity, that the matrix \mathbf{W} is *fixed* throughout the ADMM iteration. This can be done either in an oracle setting (i.e., find \mathbf{W} from the ground truth solution), or in a pre-filtering setting (i.e., find \mathbf{W} from some initial guess of the solution). Both ways are common in image restoration [23], [24].

Substituting $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2$ and the specific prior $s(\mathbf{x})$ given by (14) into the original optimization (1), the problem

becomes

$$\underset{\mathbf{x} \in \mathbb{R}^n}{\text{minimize}} \quad \varphi(\mathbf{x}) \stackrel{\text{def}}{=} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2 + \frac{\rho}{2} \mathbf{x}^T (\mathbf{I} - \mathbf{W}) \mathbf{W}^+ \mathbf{x}, \quad (16)$$

which is a quadratic optimization. Closed-form solution of (16) exists, and is given by solving the normal equation

$$\left(\mathbf{A}^T \mathbf{A} + \rho (\mathbf{I} - \mathbf{W}) \mathbf{W}^+ \right) \mathbf{x} = \mathbf{A}^T \mathbf{y}. \quad (17)$$

Since closed-form solution exists, it is possible to bypass the ADMM iterations and obtain the solution efficiently. However, from a computational perspective, there are two issues of (17) which we need to overcome. First, (17) involves inverting an $n \times n$ matrix which is computationally prohibitive for large n . Second, if \mathbf{W} has a full rank but with some very small eigenvalues, \mathbf{W}^+ will cause numerical instability, depending on the numerical threshold for truncating the eigenvalues. Therefore, if we want to use the closed form solution in (17), one possible approach is to bypass the pseudo-inverse \mathbf{W}^+ . This can be done using the following algebraic trick.

Proposition 3: The solution of (16) is given by

$$\mathbf{x} = \mathbf{U} \Sigma \left(\Sigma \mathbf{U}^T \mathbf{A}^T \mathbf{A} \mathbf{U} \Sigma + \rho \Sigma (\mathbf{I} - \Sigma) \right)^+ \Sigma \mathbf{U}^T \mathbf{A}^T \mathbf{y}, \quad (18)$$

where $\mathbf{W} = \mathbf{U} \Sigma \mathbf{U}^T$ is the eigen-decomposition of \mathbf{W} .

Proof: Define $\mathbf{p} \stackrel{\text{def}}{=} \mathbf{W}^+ \mathbf{x}$ (or, equivalently, $\mathbf{x} = \mathbf{W} \mathbf{p}$). Then it holds that

$$\varphi(\mathbf{p}) = \frac{1}{2} \|\mathbf{A} \mathbf{W} \mathbf{p} - \mathbf{y}\|^2 + \frac{\rho}{2} \mathbf{p}^T \mathbf{W} (\mathbf{I} - \mathbf{W}) \mathbf{p},$$

because $\mathbf{W} = \mathbf{W}^T$. Consider the eigen-decomposition $\mathbf{W} = \mathbf{U} \Sigma \mathbf{U}^T$, and let $\mathbf{q} = \mathbf{U}^T \mathbf{p}$, it follows that

$$\varphi(\mathbf{q}) = \frac{1}{2} \|\mathbf{A} \mathbf{U} \Sigma \mathbf{q} - \mathbf{y}\|^2 + \frac{\rho}{2} \mathbf{q}^T \Sigma (\mathbf{I} - \Sigma) \mathbf{q}.$$

The minimizer of this quadratic function is given by the solution of the normal equation

$$\left(\Sigma \mathbf{U}^T \mathbf{A}^T \mathbf{A} \mathbf{U} \Sigma + \rho \Sigma (\mathbf{I} - \Sigma) \right) \mathbf{q} = \Sigma \mathbf{U}^T \mathbf{A}^T \mathbf{y}.$$

Since $\mathbf{U} \mathbf{U}^T = \mathbf{I}$, it holds that $\mathbf{p} = \mathbf{U} \mathbf{q}$ and hence the solution is $\mathbf{x} = \mathbf{W} \mathbf{p} = \mathbf{W} \mathbf{U} \mathbf{q} = \mathbf{U} \Sigma \mathbf{q}$. ■

The importance of Proposition 3 is that (18) only involves one pseudo-inverse whereas (17) requires two pseudo-inverses. Computing (18) is numerically easy. If $\text{rank}(\mathbf{W}) = m$, we decompose $\mathbf{W} = \mathbf{V} \mathbf{S} \mathbf{V}^T$ where $\mathbf{V} \in \mathbb{R}^{n \times m}$ is the truncated eigenvector, and $\mathbf{S} \in \mathbb{R}^{m \times m}$ is the truncated eigenvalue. The matrices \mathbf{V} and \mathbf{S} can be computed using the Nyström approximation [23]. Consequently, (18) becomes

$$\mathbf{x} = \mathbf{V} \mathbf{S} \left(\mathbf{S} \mathbf{V}^T \mathbf{A}^T \mathbf{A} \mathbf{V} \mathbf{S} + \rho \mathbf{S} (\mathbf{I} - \mathbf{S}) \right)^{-1} \mathbf{S} \mathbf{V}^T \mathbf{A}^T \mathbf{y}. \quad (19)$$

Inspecting (19), we observe that the matrix inversion only involves an $m \times m$ matrix, which is significantly smaller than the $n \times n$ matrix in (17). The matrix $\mathbf{A} \mathbf{V}$ are usually not difficult to evaluate. Below are two examples for image inpainting and deblurring.

Example 3: For image inpainting, the matrix \mathbf{A} is a binary diagonal matrix. Hence, $\mathbf{A} \mathbf{V}$ involves picking the non-zero columns of \mathbf{V} .

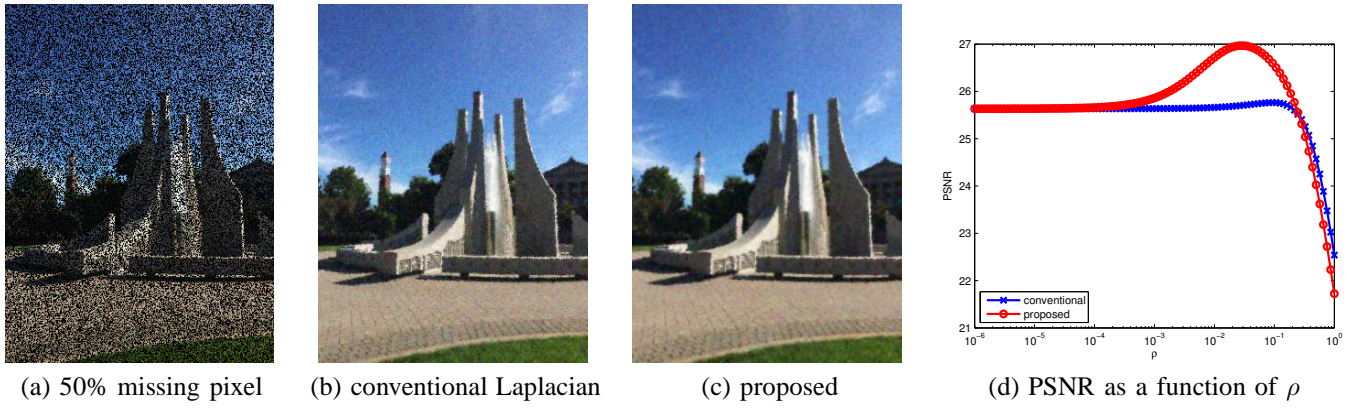


Fig. 1: Image inpainting using (b) the conventional graph Laplacian prior $s(\mathbf{x}) = \frac{\rho}{2}\mathbf{x}^T(\mathbf{I} - \mathbf{W})\mathbf{x}$ and (c) the proposed algorithm-induced prior $s(\mathbf{x}) = \frac{\rho}{2}\mathbf{x}^T(\mathbf{I} - \mathbf{W})\mathbf{W}^+\mathbf{x}$. For both methods, an optimal ρ is selected for the best PSNR as shown in (d). The matrix \mathbf{W} is the non-local mean weight computed from an initial solution using Shepard's interpolation [25]. In this example, the RGB color channels are processed independently.

Example 4: For image deblurring, the matrix \mathbf{A} is a convolution. Hence, the multiplication of \mathbf{A} and \mathbf{v}_i , the i th column of \mathbf{V} , is a blurring operation on \mathbf{v}_i .

Remark 1: In practice, symmetric smoothing filters sometimes have very narrow spatial support and hence \mathbf{W} is a banded diagonal matrix. A banded diagonal \mathbf{W} has a significantly higher rank, making the eigen-decomposition difficult. However, the good news is that such \mathbf{W} is often easy to compute. In this case, the closed form should be replaced by the ADMM iteration.

IV. EXPERIMENTAL RESULTS

To demonstrate the usefulness of the algorithm-induced prior, I use image inpainting as an example. In this experiment, let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a binary diagonal matrix, where $A_{ii} \sim \text{Bernoulli}(\xi)$ for a sampling ratio $0 \leq \xi \leq 1$. Here, $\xi = 1$ means that all pixels are acquired with probability 1, whereas $\xi = 0$ means a probability 0. The observed image is

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$ is an additive iid Gaussian noise. In this experiment, $\sigma = 0.05$.

There are two choices of the filter \mathbf{W} . The first choice is to compute \mathbf{W} from the ground truth solution. This is called the oracle setting, and is the best possible setting we can use under our framework. The second choice is to compute \mathbf{W} from some initial estimate of the solution. In this problem, the initial estimate is performed using the classical Shepard's interpolation method [25]. If a more sophisticated initial estimator is used, it is likely that the performance will be improved.

We compare two graph Laplacian priors, namely

$$\begin{aligned} \text{conventional:} \quad & s_{\mathbf{L}}(\mathbf{x}) = \frac{\rho}{2}\mathbf{x}^T\mathbf{L}\mathbf{x}, \\ \text{proposed:} \quad & s_{\mathbf{C}}(\mathbf{x}) = \frac{\rho}{2}\mathbf{x}^T\mathbf{C}\mathbf{x}, \end{aligned} \quad (20)$$

where $\mathbf{L} \stackrel{\text{def}}{=} \mathbf{I} - \mathbf{W}$ is the classical graph Laplacian, and

% missing	estimated \mathbf{W}		oracle \mathbf{W}	
	conventional	proposed	conventional	proposed
20%	26.24dB	27.59dB	44.48dB	47.22dB
40%	24.87dB	25.82dB	43.36dB	46.44dB
60%	23.72dB	24.32dB	41.04dB	44.97dB
80%	21.56dB	21.85dB	37.36dB	42.57dB

TABLE I: PSNR of “cameraman” with optimized ρ .

$\mathbf{C} \stackrel{\text{def}}{=} (\mathbf{I} - \mathbf{W})\mathbf{W}^+$ is the proposed algorithm-induced prior. The results are shown in Table I. It is evident from the result that $s_{\mathbf{C}}(\mathbf{x})$ performs consistently better than $s_{\mathbf{L}}(\mathbf{x})$ for both the estimated \mathbf{W} and the oracle \mathbf{W} . The gap is especially prominent if we look at the oracle case at 80% missing.

Figure 1 shows a visual comparison of an image captured by an i-Phone 6 camera with 50% missing pixel generated by MATLAB simulation. It should be reminded that in all experiments the parameter ρ are adjusted accordingly for $s_{\mathbf{L}}(\mathbf{x})$ and $s_{\mathbf{C}}(\mathbf{x})$. Figure 1(d) illustrates such dependence: The optimal ρ are different for different priors. However, the best PSNR of $s_{\mathbf{C}}(\mathbf{x})$ is significantly higher than that of $s_{\mathbf{L}}(\mathbf{x})$.

V. CONCLUSION

Algorithm-induced prior is a strong performing but intriguing prior that we have little understanding about. Therefore, being able to explicitly write down the formula of the algorithm-induced prior is an important step which allows us to analyze the performance of such prior. In this paper, I demonstrated the case of symmetric smoothing filters and drew connections with the conventional graph Laplacian prior. On a set of image inpainting experiments, algorithm-induced prior offers consistently better results than the conventional graph Laplacian. As we progress along this direction, I believe that the interplay between the objective function and the denoising procedure should be studied in greater details.

VI. ACKNOWLEDGEMENT

I like to thank Charles Bouman for pointing me to the problem, and Suhas Sreehari for fruitful discussions. I also

thank Dror Baron for telling me his work on AMP using BM3D.

REFERENCES

- [1] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM J. on Sci. Comput.*, vol. 31, no. 4, pp. 2842–2865, Jul. 2009.
- [2] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Aug. 2010.
- [3] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.
- [4] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.
- [5] E. Esser, "Applications of Lagrangian-based alternating direction methods and connections to split Bregman," Tech. Rep., University of California, Los Angeles, 2009, Available online: <ftp://ftp.math.ucla.edu/pub/camreport/cam09-31.pdf>.
- [6] 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. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
- [7] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3097–3111, Nov. 2011.
- [8] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *SIAM Multiscale Model and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [9] S. H. Chan, T. Zickler, and Y. M. Lu, "Monte-Carlo non-local means: Random sampling for large-scale image filtering," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3711–3725, Aug. 2014.
- [10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
- [11] S. Venkatakrishnan, C. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conference on Signal and Information Processing*, 2013, pp. 945–948.
- [12] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," Available online: <http://arxiv.org/abs/1512.07331>, Dec. 2015.
- [13] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," Available online: <http://arxiv.org/abs/1406.4175>, Jul. 2014.
- [14] Y. Ma, J. Zhu, and D. Baron, "Approximate message passing with universal denoising," Available online: <http://arxiv.org/abs/1506.02693>, Jun. 2015.
- [15] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, "Postprocessing of compressed images via sequential denoising," Available online: <http://arxiv.org/pdf/1510.09041v1.pdf>, Oct. 2015.
- [16] P. Milanfar, "Symmetrizing smoothing filters," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 263–284, 2013.
- [17] S. H. Chan, T. Zickler, and Y. M. Lu, "Understanding symmetric smoothing filters via Gaussian mixtures," in *Proc. IEEE Intl. Conf. Image Process.*, Sep 2015.
- [18] S. H. Chan, T. Zickler, and Y. M. Lu, "Demystifying symmetric smoothing filters," Available online: <http://arxiv.org/abs/1601.00088>, Jan. 2016.
- [19] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Processing Magazine*, vol. 30, pp. 106–128, Jan. 2013.
- [20] F. Meyer and X. Shen, "Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising," *Applied and Computational Harmonic Analysis*, 2013, In press. Available online at <http://arxiv.org/abs/1202.6666>.
- [21] R. Sinkhorn and P. Knopp, "Concerning non-negative matrices and doubly-stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, pp. 343 – 348, 1967.
- [22] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 1053–5888, Apr. 2013.
- [23] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 755–768, Feb. 2014.
- [24] H. Talebi, X. Zhu, and P. Milanfar, "How to SAIF-ly boost denoising performance," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1470–1485, Apr. 2013.
- [25] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proc. ACM National Conference*, 1968, pp. 517–524.