Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications

Stanley H. Chan, *Member, IEEE*, Xiran Wang, *Student Member, IEEE*, and Omar A. Elgendy, *Student Member, IEEE*

Abstract—**Alternating direction method of multiplier (ADMM) is a widely used algorithm for solving constrained optimization problems in image restoration. Among many useful features, one critical feature of the ADMM algorithm is its modular structure, which allows one to plug in any off-the-shelf image denoising algorithm for a subproblem in the ADMM algorithm. Because of the plug-in nature, this type of ADMM algorithms is coined the name "Plug-and-Play ADMM." Plug-and-Play ADMM has demonstrated promising empirical results in a number of recent papers. However, it is unclear under what conditions and by using what denoising algorithms would it guarantee convergence. Also, since Plug-and-Play ADMM uses a specific way to split the variables, it is unclear if fast implementation can be made for common Gaussian and Poissonian image restoration problems. In this paper, we propose a Plug-and-Play ADMM algorithm with provable fixed-point convergence. We show that for any denoising algorithm satisfying an asymptotic criteria, called bounded denoisers, Plug-and-Play ADMM converges to a fixed point under a continuation scheme. We also present fast implementations for two image restoration problems on superresolution and single-photon imaging. We compare Plug-and-Play ADMM with state-of-the-art algorithms in each problem type and demonstrate promising experimental results of the algorithm.**

Index Terms—**ADMM, deblurring, denoising, image restoration, inpainting, Plug-and-Play, Poisson noise, single photon imaging, super-resolution.**

I. INTRODUCTION

A. MAP and ADMM

MANY image restoration tasks can be posted as the following inverse problem: Given an observed image $\boldsymbol{y} \in \mathbb{R}^n$ corrupted according to some forward model and noise, find the underlying image $\boldsymbol{x} \in \mathbb{R}^n$ which "best explains" the observation. In estimation, we often formulate this problem as a maximum-a-posteriori (MAP) estimation [1], where the goal is to maximize the posterior probability:

$$\widehat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} \quad p(\boldsymbol{x} \mid \boldsymbol{y})$$
$$= \operatorname*{argmin}_{\boldsymbol{x}} \quad -\log p(\boldsymbol{y} \mid \boldsymbol{x}) - \log p(\boldsymbol{x}), \qquad (1)$$

for some conditional probability $p(\boldsymbol{y} \mid \boldsymbol{x})$ defining the forward imaging model, and a prior distribution $p(\boldsymbol{x})$ defining the probability distribution of the latent image. Because of the explicit use of the forward and the prior models, MAP estimation is also a model-based image reconstruction (MBIR) method [2] which has many important applications in deblurring [3]–[5], interpolation [6]–[8], super-resolution [9]–[12] and computed tomography [13], to name a few.

It is not difficult to see that solving the MAP problem in (1) is equivalent to solving an optimization problem

$$\widehat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}), \qquad (2)$$

with $f(\boldsymbol{x}) \stackrel{\text{def}}{=} -\log p(\boldsymbol{y} \mid \boldsymbol{x})$ and $g(\boldsymbol{x}) \stackrel{\text{def}}{=} -(1/\lambda) \log p(\boldsymbol{x})$. The optimization in (2) is a generic unconstrained optimization. Thus, standard optimization algorithms can be used to solve the problem. In this paper, we focus on the alternating direction method of multiplier (ADMM) [14], which has become the workhorse for a variety of problems in the form of (2).

The idea of ADMM is to convert (2), an unconstrained optimization, into a constrained problem

$$(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{v}}) = \operatorname*{argmin}_{\boldsymbol{x}, \boldsymbol{v}} \quad f(\boldsymbol{x}) + \lambda g(\boldsymbol{v}), \quad \text{subject to} \quad \boldsymbol{x} = \boldsymbol{v}, \quad (3)$$

and consider its augmented Lagrangian function:

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

The minimizer of (3) is then the saddle point of $\mathcal{L}$, which can be found by solving a sequence of subproblems

$$\boldsymbol{x}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^n} \quad f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}^{(k)}\|^2, \qquad (5)$$

$$\boldsymbol{v}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{v} \in \mathbb{R}^n} \quad \lambda g(\boldsymbol{v}) + \frac{\rho}{2}\|\boldsymbol{v} - \widetilde{\boldsymbol{v}}^{(k)}\|^2, \qquad (6)$$

$$\bar{\boldsymbol{u}}^{(k+1)} = \bar{\boldsymbol{u}}^{(k)} + (\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)}), \qquad (7)$$

S. H. Chan is with the School of Electrical and Computer Engineering and the Department of Statistics, Purdue University, West Lafayette, IN 47907 USA (e-mail: stanchan@purdue.edu).

X. Wang and O. A. Elgendy are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: wang470@purdue.edu; oelgendy@purdue.edu).

where $\bar{\boldsymbol{u}}^{(k)} \stackrel{\text{def}}{=} (1/\rho)\boldsymbol{u}^{(k)}$ is the scaled Lagrange multiplier, $\widetilde{\boldsymbol{x}}^{(k)} \stackrel{\text{def}}{=} \boldsymbol{v}^{(k)} - \bar{\boldsymbol{u}}^{(k)}$ and $\widetilde{\boldsymbol{v}}^{(k)} \stackrel{\text{def}}{=} \boldsymbol{x}^{(k+1)} + \bar{\boldsymbol{u}}^{(k)}$. Under mild conditions, e.g., when both f and g are closed, proper and convex, and if a saddle point of $\mathcal{L}$ exists, one can show that the iterates (5)–(7) converge to the solution of (3) (See [14] for details).

B. Plug-and-Play ADMM

An important feature of the ADMM iterations (5)–(7) is its modular structure. In particular, (5) can be regarded as an inversion step as it involves the forward imaging model $f(\boldsymbol{x})$, whereas (6) can be regarded as a denoising step as it involves the prior $g(\boldsymbol{v})$. To see the latter, if we define $\sigma = \sqrt{\lambda/\rho}$, it is not difficult to show that (6) is

$$\boldsymbol{v}^{(k+1)} = \underset{\boldsymbol{v} \in \mathbb{R}^n}{\text{argmin}} \ \ g(\boldsymbol{v}) + \frac{1}{2\sigma^2}\|\boldsymbol{v} - \widetilde{\boldsymbol{v}}^{(k)}\|^2. \qquad (8)$$

Treating $\widetilde{\boldsymbol{v}}^{(k)}$ as the "noisy" image, (8) minimizes the residue between $\widetilde{\boldsymbol{v}}^{(k)}$ and the "clean" image $\boldsymbol{v}$ using the prior $g(\boldsymbol{v})$. For example, if $g(\boldsymbol{x}) = \|\boldsymbol{x}\|_{\text{TV}}$ (the total variation norm), then (8) is the standard total variation denoising problem.

Building upon this intuition, Venkatakrishnan *et al.* [15] proposed a variant of the ADMM algorithm by suggesting that one does not need to specify g before running the ADMM. Instead, they replace (6) by using an off-the-shelf image denoising algorithm, denoted by $\mathcal{D}_\sigma$, to yield

$$\boldsymbol{v}^{(k+1)} = \mathcal{D}_\sigma\left(\widetilde{\boldsymbol{v}}^{(k)}\right). \qquad (9)$$

Because of the heuristic nature of the method, they called the resulting algorithm as the Plug-and-Play ADMM. An interesting observation they found in [15] is that although Plug-and-Play ADMM appears ad-hoc, for a number of image reconstruction problems the algorithm indeed performs better than some state-of-the-art methods. A few recent reports have concurred similar observations [13], [16]–[18].

C. Challenges of Plug-and-Play ADMM

From a theoretical point of view, the main challenge of analyzing Plug-and-Play ADMM is the denoiser $\mathcal{D}_\sigma$. Since $\mathcal{D}_\sigma$ is often nonlinear and does not have closed form expressions, the analysis has been very difficult. Specifically, the following three questions remain open:

1) *Convergence of the Algorithm:* Classical results of ADMM require g to be closed, proper and convex in order to ensure convergence [14]. While newer results have extended ADMM for nonconvex problems [19], there is little work addressing the case when g is defined implicitly through $\mathcal{D}_\sigma$. To the best of our knowledge, the only existing convergence analysis, to date, is the one by Sreehari *et al.* [13] for the case when $\mathcal{D}_\sigma$ is a symmetric smoothing filter [20], [21]. However, for general $\mathcal{D}_\sigma$ the convergence is not known.

2) *Original Prior:* Since $\mathcal{D}_\sigma$ is an off-the-shelf image denoising algorithm, it is unclear what prior g does it correspond to. In [22], Chan addresses this question by explicitly deriving the original prior g when $\mathcal{D}_\sigma$ is a symmetric smoothing filter [22]. In this case, the author shows that g is a modified graph Laplacian prior, with better restoration performance compared to the conventional graph Laplacian [23]. However, beyond symmetric smoothing filters it becomes unclear if we can find the corresponding g.

3) *Implementation:* The usage of Plug-and-Play ADMM has been reported in a few scattered occasions, with some work in electron tomography [13], compressive sensing [16], and some very recent applications in Poisson recovery [17] and super-resolution [18]. However, the common challenge underpinning these applications is whether one can obtain a fast solver for the inversion step in (5). This has not been a problem for conventional ADMM, because in many cases we can use another variable splitting strategy to replace $\boldsymbol{v} = \boldsymbol{x}$ in (3), e.g., using $\boldsymbol{v} = \boldsymbol{B}\boldsymbol{x}$ when $g(\boldsymbol{x}) = \|\boldsymbol{B}\boldsymbol{x}\|_1$ [3].

D. Related Works

Plug-and-Play ADMM was first reported in 2013. Around the same period of time there is an independent series of studies using denoisers for approximate message passing (AMP) algorithms [24]–[27]. The idea was to replace the shrinkage step of the standard AMP algorithm with any off-the-shelf algorithm in the class of "proper denoisers"—denoisers which ensure that the noise variance is sufficiently suppressed. (See Section II-C for more discussions.) However, this type of denoise-AMP algorithms rely heavily on the Gaussian statistics of the random measurement matrix $\boldsymbol{A}$ in a specific forward model $f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2$. Thus, if $f(\boldsymbol{x})$ departs from quadratic or if $\boldsymbol{A}$ is not random, then the behavior of the denoise-AMP becomes unclear.

Using denoisers as building blocks of an image restoration algorithm can be traced back further, e.g., wavelet denoiser for signal deconvolution [28]. Of particular relevance to Plug-and-Play ADMM is the work of Danielyan *et al.* [29], where they proposed a variational method for deblurring using BM3D as a prior. The idea was later extended by Zhang *et al.* to other restoration problems [30]. However, these algorithms are customized for the specific denoiser BM3D. In contrast, the proposed Plug-and-Play ADMM supports any denoiser satisfying appropriate assumptions. Another difference is that when BM3D is used in [29] and [30], the grouping of the image patches are fixed throughout the iterations. Plug-and-Play ADMM allows re-calculation of the grouping at every iteration. In this aspect, the Plug-and-Play ADMM is more general than these algorithms.

A large number of denoisers we use nowadays are patch-based denoising algorithms. All these methods can be considered as variations in the class of universal denoisers [31], [32] which are asymptotically optimal and do not assume external knowledge of the latent image (e.g., prior distribution). Asymptotic optimality of patch-based denoisers has been recognized empirically by Levin *et al.* [33], [34], who showed that non-local means [35] approaches the MMSE estimate as the number of patches grows to infinity. Recently, Ma *et al.* [36] made attempts to integrate universal denoisers with approximate message passing algorithms.

E. Contributions

The objective of this paper is to address the first and the third issue mentioned in Section I-C. The contributions of this paper are as follows:

First, we modify the original Plug-and-Play ADMM by incorporating a continuation scheme. We show that the new algorithm is guaranteed to converge for a broader class of denoisers known as the *bounded denoisers*. Bounded denoisers are asymptotically invariant in the sense that the denoiser approaches an identity operator as the denoising parameter vanishes. Bounded denoisers are weaker than the non-expansive denoisers presented in [13]. However, for weaker denoisers we should also expect a weaker form of convergence. We prove that the new Plug-and-Play ADMM has a *fixed point* convergence, which complements the global convergence results presented in [13].

Second, we discuss fast implementation techniques for image super-resolution and single photon imaging problems. For the super-resolution problem, conventional ADMM requires multiple variable splits or an inner conjugate gradient solver to solve the subproblem. We propose a polyphase decomposition based method which gives us closed-form solutions. For the single photon imaging problem, existing ADMM algorithm are limited to explicit priors such as total variation. We demonstrate how Plug-and-Play ADMM can be used and we present a fast implementation by exploiting the separable feature of the problem.

The rest of the paper is organized as follows. We first discuss the Plug-and-Play ADMM algorithm and the convergence properties in Section II. We then discuss the applications in Section III. Experimental results are presented in Section IV.

II. PLUG-AND-PLAY ADMM AND CONVERGENCE

In this section we present the proposed Plug-and-Play ADMM and discuss its convergence property. Throughout this paper, we assume that the unknown image $\boldsymbol{x}$ is bounded in an interval $[x_{\min}, x_{\max}]$ where the upper and lower limits can be obtained from experiment or from prior knowledge. Thus, without loss of generality we assume $\boldsymbol{x} \in [0, 1]^n$.

A. Plug-and-Play ADMM

The proposed Plug-and-Play ADMM algorithm is a modification of the conventional ADMM algorithm in (5)–(7). Instead of choosing a constant ρ, we increase ρ by $\rho_{k+1} = \gamma_k \rho_k$ for $\gamma_k \geq 1$. In optimization literature, this is known as a continuation scheme [37] and has been used in various problems, e.g., [38], [39]. Incorporating this idea into the ADMM algorithm, we obtain the following iteration:

$$\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\arg\min} \ f(\boldsymbol{x}) + (\rho_k/2)\|\boldsymbol{x} - (\boldsymbol{v}^{(k)} - \boldsymbol{u}^{(k)})\|^2$$
$$(10)$$

$$\boldsymbol{v}^{(k+1)} = \mathcal{D}_{\sigma_k}(\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)}) \tag{11}$$

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + (\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)}) \tag{12}$$

$$\rho_{k+1} = \gamma_k \rho_k, \tag{13}$$

where $\mathcal{D}_{\sigma_k}$ is a denoising algorithm (called a "denoiser" for short), and $\sigma_k \overset{\text{def}}{=} \sqrt{\lambda/\rho_k}$ is a parameter controlling the strength of the denoiser.

Algorithm 1: Plug-and-Play ADMM.

Input: $\rho_0, \lambda, \eta < 1, \gamma > 1$.
while Not Converge **do**
  $\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\arg\min} \ f(\boldsymbol{x}) + (\rho_k/2)\|\boldsymbol{x} - (\boldsymbol{v}^{(k)} - \boldsymbol{u}^{(k)})\|^2$
  $\boldsymbol{v}^{(k+1)} = \mathcal{D}_{\sigma_k}(\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)})$, where $\sigma_k = \sqrt{\lambda/\rho_k}$
  $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + (\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)})$
  if $\Delta_{k+1} \geq \eta\Delta_k$ **then**
    $\rho_{k+1} = \gamma\rho_k$
  else
    $\rho_{k+1} = \rho_k$
  end if
  $k = k + 1$.
end while

There are different options in setting the update rule for ρ_k. In this paper we present two options. The first one is a *monotone update* rule which defines

$$\rho_{k+1} = \gamma\rho_k, \qquad \text{for all } k \tag{14}$$

for a constant $\gamma > 1$. The second option is an *adaptive update* rule by considering the relative residue:

$$\Delta_{k+1} \overset{\text{def}}{=} \frac{1}{\sqrt{n}} \Big(\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2 + \|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2$$
$$+ \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2 \Big). \tag{15}$$

For any $\eta \in [0, 1)$ and let $\gamma > 1$ be a constant, we conditionally update ρ_k according to the followings:
1) If $\Delta_{k+1} \geq \eta\Delta_k$, then $\rho_{k+1} = \gamma\rho_k$.
2) If $\Delta_{k+1} < \eta\Delta_k$, then $\rho_{k+1} = \rho_k$.

The adaptive update scheme is inspired from [40], which was originally used to accelerate ADMM algorithms for convex problems. It is different from the residual balancing technique commonly used in ADMM, e.g., [14], as Δ_{k+1} sums of all primal and dual residues instead of treating them individually. Our experience shows that the proposed scheme is more robust than residual balancing because the denoiser could potentially generate nonlinear effects to the residuals. Algorithm 1 shows the overall Plug-and-Play ADMM.

Remark 1 (Comparison with [15]): In the original Plug-and-Play ADMM by Venkatakrishnan *et al.* [15], the update scheme is $\rho_k = \rho$ for some constant ρ. This is valid when the denoiser $\mathcal{D}_\sigma$ is non-expansive and has symmetric gradient. However, for general denoisers which could be expansive, the update scheme for ρ_k becomes crucial to the convergence. (See discussion about non-expansiveness in Section II-B.)

Remark 2 (Role of σ_k): Many denoising algorithms nowadays such as BM3D and non-local means require one major parameter,[1] typically an estimate of the noise level, to control

---
[1] A denoising algorithm often involves many other "internal" parameters. However, as these internal parameters do not have direct interaction with the

the strength of the denoiser. In our algorithm, the parameter σ_k in (11) is reminiscent to the noise level. However, unlike BM3D and non-local means where σ_k is directly linked to the standard deviation of the i.i.d. Gaussian noise, in Plug-and-Play ADMM we treat σ_k simply as a tunable knob to control the amount of denoising because the residue $(\boldsymbol{v} - \widetilde{\boldsymbol{v}}^{(k)})$ at the kth iterate is not exactly Gaussian. The adoption of the Gaussian denoiser $\mathcal{D}_{\sigma_k}$ is purely based on the formal equivalence between (8) and a Gaussian denoising problem.

Remark 3 (Role of λ): In this paper, we assume that the parameter λ is pre-defined by the user and is fixed. Its role is similar to the regularization parameter in the conventional ADMM problem. Tuning λ can be done using external tools such as cross validation [41] or SURE [42].

B. Global and Fixed Point Convergence

Before we discuss the convergence behavior, we clarify two types of convergence.

We refer to the type of convergence in the conventional ADMM as *global convergence*, i.e., convergence in primal residue, primal objective and dual variables. To ensure global convergence, one sufficient condition is that g is convex, proper and closed [14]. For Plug-and-Play ADMM, a sufficient condition is that $\mathcal{D}_\sigma$ has symmetric gradient and is non-expansive [13]. In this case, g exists due to a proximal mapping theorem of Moreau [43]. However, proving non-expansive denoisers could be difficult as it requires

$$\|\mathcal{D}_\sigma(\boldsymbol{x}) - \mathcal{D}_\sigma(\boldsymbol{y})\|^2 \le \kappa \|\boldsymbol{x} - \boldsymbol{y}\|^2$$

for *any* $\boldsymbol{x}$ and $\boldsymbol{y}$, with $\kappa \le 1$. Even for algorithms as simple as non-local means, one can verify numerically that there exists pairs $(\boldsymbol{x}, \boldsymbol{y})$ that would cause $\kappa > 1$. In the Appendix we demonstrate a counter example.

Since $\mathcal{D}_\sigma$ can be arbitrary and we do not even know the existence of g, we consider fixed point convergence instead. Fixed point convergence guarantees that a nonlinear algorithm can enter into a steady state asymptotically. In nonlinear dynamical systems, these limit points are referred to as the stable-fixed-points. For any initial guess lying in a region called the basin of attraction the algorithm will converge [44]. For Plug-and-Play ADMM, we conjecture that fixed point convergence is the best we can ask for unless further assumptions are made on the denoisers.

C. Convergence Analysis of Plug-and-Play ADMM

We define the class of bounded denoisers.

Definition 1 (Bounded Denoiser): A *bounded denoiser* with a parameter σ is a function $\mathcal{D}_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ such that for any input $\boldsymbol{x} \in \mathbb{R}^n$,

$$\|\mathcal{D}_\sigma(\boldsymbol{x}) - \boldsymbol{x}\|^2 / n \le \sigma^2 C, \tag{16}$$

for some universal constant C independent of n and σ.

Bounded denoisers are asymptotically invariant in the sense that it ensures $\mathcal{D}_\sigma \to \mathcal{I}$ (i.e., the identity operator) as $\sigma \to 0$. It is a weak condition which we expect most denoisers to have. The

asymptotic invariant property of a bounded denoiser prevents trivial mappings from being considered, e.g., $\mathcal{D}_\sigma(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$.

Remark 4: It would be useful to compare a bounded denoiser with a "proper denoiser" defined in [24]. A proper denoiser $\widetilde{\mathcal{D}}_\sigma$ is a mapping that denoises a noisy input $\boldsymbol{x} + \sigma\boldsymbol{\epsilon}$ with the property that

$$\mathbb{E}\left[\|\widetilde{\mathcal{D}}_\sigma(\boldsymbol{x} + \sigma\boldsymbol{\epsilon}) - \boldsymbol{x}\|^2 / n\right] \le \kappa \sigma^2, \tag{17}$$

for any $\kappa < 1$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ is the i.i.d. Gaussian noise. Note that in (17), we require the input to be a deterministic signal $\boldsymbol{x}$ plus an i.i.d. Gaussian noise. Moreover, the parameter must match with the noise level. In contrast, a bounded denoiser can take any input and any parameter.

Besides the conditions on $\mathcal{D}_\sigma$ we also assume that the negative log-likelihood function f has bounded gradients:

Assumption 1: We assume that $f : [0, 1]^n \to \mathbb{R}$ has bounded gradients. That is, for any $\boldsymbol{x} \in [0, 1]^n$, there exists $L < \infty$ such that $\|\nabla f(\boldsymbol{x})\|_2 / \sqrt{n} \le L$.

Example 1: Let $f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ for $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with eigenvalues bounded between 0 and 1. The gradient of f is $\nabla f(\boldsymbol{x}) = 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$ and

$$\|\nabla f(\boldsymbol{x})\|_2 / \sqrt{n} \le 2\lambda_{\max}(\boldsymbol{A})(\|\boldsymbol{x}\|_2 + \|\boldsymbol{y}\|_2) / \sqrt{n}.$$

The main convergence result of this paper is as follows.

Theorem 1 (Fixed Point Convergence of Plug-and-Play ADMM): Under Assumption 1 and for any bounded denoiser $\mathcal{D}_\sigma$, the iterates of the Plug-and-Play ADMM defined in Algorithm 1 demonstrates a fixed-point convergence. That is, there exists $(\boldsymbol{x}^*, \boldsymbol{v}^*, \boldsymbol{u}^*)$ such that $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_2 \to 0$, $\|\boldsymbol{v}^{(k)} - \boldsymbol{v}^*\|_2 \to 0$ and $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2 \to 0$ as $k \to \infty$.

Proof: See Appendix B.

Intuitively, what Theorem 1 states is that as $k \to \infty$, the continuation scheme forces $\rho_k \to \infty$. Therefore, the inversion in (10) and the denoising in (11) have reducing influence as ρ_k grows. Hence, the algorithm converges to a fixed point. Theorem 1 also ensures that $\boldsymbol{x}^{(k)} \to \boldsymbol{v}^{(k)}$ which is an important property of the original Plug-and-Play ADMM algorithm [13]. The convergence of $\boldsymbol{x}^{(k)} \to \boldsymbol{v}^{(k)}$ holds because $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + (\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)})$ converges. In practice, experimentally we observe that if the algorithm is terminated early to reduce the runtime, then $\boldsymbol{v}^{(k)}$ tends to provide a slightly better solution.

D. Stopping Criteria

Since we are seeking for fixed point convergence, a natural stopping criteria is to determine if $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2$, $\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2$ and $\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2$ are sufficiently small. Following the definition of Δ_{k+1} in (15), we choose to terminate the iteration when

$$\Delta_{k+1} \stackrel{\text{def}}{=} \frac{1}{\sqrt{n}}\Big(\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2 + \|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2$$

$$+ \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2\Big) \le \texttt{tol} \tag{18}$$

ADMM algorithm, in this paper we keep all internal parameters in their default settings to simplify the analysis.
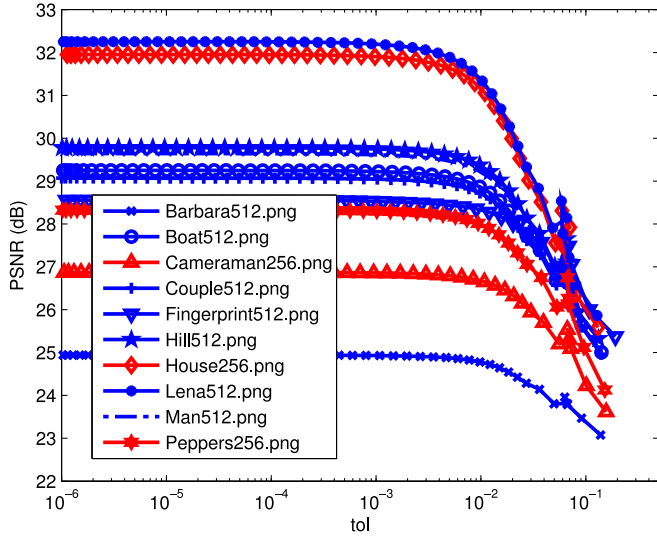
Fig. 1.   Stopping criteria. The PSNR drops as the tolerance level increases. However, regardless of the size of the images, the PSNR becomes steady when `tol` $\approx 10^{-3}$.
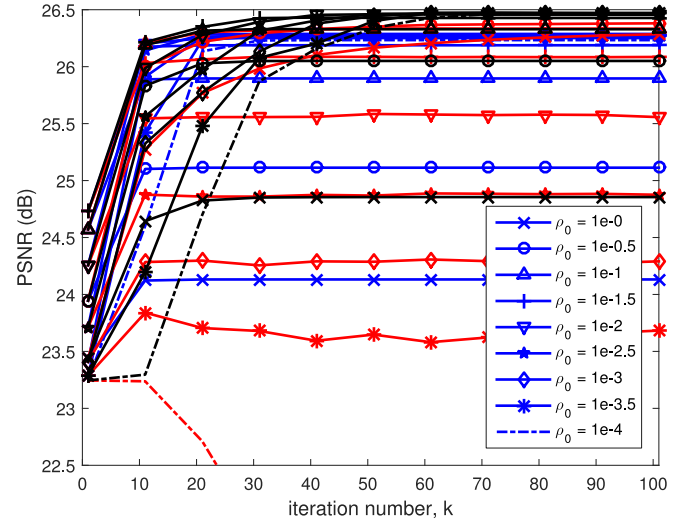

Fig. 2.   Influence of ρ_0. Red curves represent the original method in [13]; Blue curves represent the monotone update rule; Black curves represent the adaptive update rule. Note the diversified behavior of the red curve, which implies that a precise ρ_0 is required. The blue and black curves are more robust.

for some tolerance level `tol`. Alternatively, we can also terminate the algorithm when

$$\max\left\{\epsilon_1, \epsilon_2, \epsilon_3\right\} \leq \texttt{tol}/3,$$

where $\epsilon_1 = \|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\|_2/\sqrt{n}$, $\epsilon_2 = \|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\|_2/\sqrt{n}$ and $\epsilon_3 = \|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\|_2/\sqrt{n}$.

In practice, the tolerance level does not need to be extremely small in order to achieve good reconstruction quality. In fact, for many images we have tested, setting `tol` $\approx 10^{-3}$ is often sufficient. Fig. 1 provides a justification. In this experiment, we tested an image super-resolution problem for 10 testing images (See Configuration 3 in Section IV-A for details). It can be observed that the PSNR becomes steady when `tol` drops below 10^{-3}. Moreover, size of the image does not seem to be an influencing factor. Smaller images such as `Cameraman256`, `House256` and `Peppers256` shows similar characteristics as bigger images. The more influencing factor is the combination of the update ratio γ and the initial value ρ_0. However, unless γ is close to 1 and ρ_0 is extremely small (which does not yield good reconstruction anyway), our experience is that setting `tol` at 10^{-3} is usually valid for $\gamma \in (1, 2)$ and $\rho_0 \in (10^{-5}, 10^{-2})$.

E. Initial Parameter ρ_0

The choice of the initial parameter ρ_0 requires some tuning but is typically good for $\rho_0 \in (10^{-5}, 10^{-2})$. Fig. 2 shows the behavior of the algorithm for different values of ρ_0, ranging from 10^0 to 10^{-4}. We compare the original Plug-and-Play ADMM (i.e., with constant $\rho_k = \rho_0$, the red lines), monotone update rule (i.e., $\rho_{k+1} = \gamma\rho_k$, the blue lines), and the adaptive update rule (the black lines). We make two observations regarding the difference between the proposed algorithm and the original Plug-and-Play ADMM [13]:

1) *Stability*: The original Plug-and-Play ADMM [13] requires a highly precise ρ_0. For example, in Fig. 2 the best PSNR is achieved when $\rho_0 = 1$; When ρ_0 is less

than 10^{-2}, the PSNR becomes very poor. The proposed algorithm works for a much wider range of ρ_0.

2) *Final PSNR*: The proposed Plug-and-Play ADMM is a generalization of the original Plug-and-Play ADMM. The added degrees of freedom are the new parameters (ρ_0, γ, η). The original Plug-and-Play ADMM is a special case when $\gamma = 1$. Therefore, for optimally tuned parameters, the proposed Plug-and-Play ADMM is always better than or equal to the original Plug-and-Play ADMM. This is verified in Fig. 2, which shows that the best PSNR is attained by the proposed method.

F. Initial Guesses

The initial guesses $\boldsymbol{x}^{(0)}$, $\boldsymbol{v}^{(0)}$ and $\boldsymbol{u}^{(0)}$ have less impact to the final PSNR. This can be seen from Fig. 3. In this experiment, we randomly draw 100 initial guesses $\boldsymbol{x}^{(0)}$ from a uniform distribution in $[0, 1]^n$. The auxiliary variable is set as $\boldsymbol{v}^{(0)} = \boldsymbol{x}^{(0)}$, and the Lagrange multiplier $\boldsymbol{u}^{(0)}$ is 0. As shown in Fig. 3, the initial guesses do not cause significant difference in terms of PSNR at the limit. The standard deviation at the limit is 0.0059 dB, implying that with 99.7% probability (3 standard deviations) the PSNR will stay within ± 0.0176 dB from its average.

III. Applications

As we discussed in the introduction, Plug-and-Play ADMM algorithm has a wide range of applications. However, in order to enable the denoising step, Plug-and-Play ADMM uses a specific variable splitting strategy. The challenge it brings, therefore, is whether we can solve the subsequent subproblems efficiently. The purpose of this section is to address this issue by presenting two applications where fast implementation can be achieved.

A. Application 1: Image Super-Resolution

Image super-resolution can be described by a linear forward model with two operations: an anti-aliasing filter and a subsam-
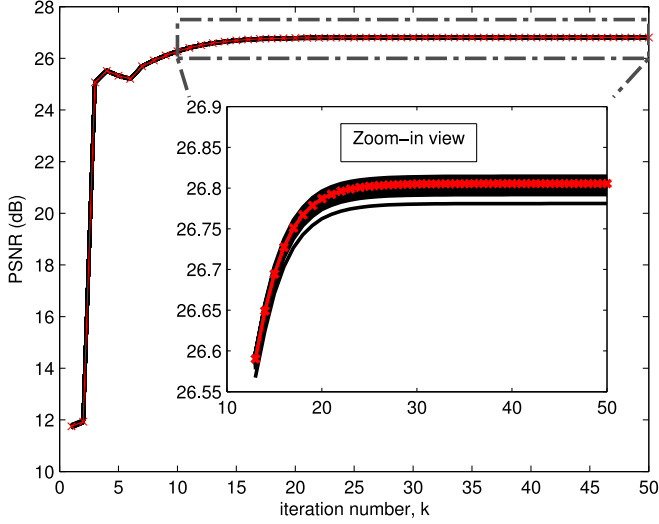
Fig. 3. Influence of the initial point $\boldsymbol{x}^{(0)}$. We start the algorithm with 100 different initial guesses $\boldsymbol{x}^{(0)}$ where each is a uniformly random vector drawn from $[0, 1]^n$. Over these 100 random realizations we plot the average (red line). Note the small fluctuation of the PSNR at the limit.

pling process. The function $f(\boldsymbol{x})$ is quadratic in the form

$$f(\boldsymbol{x}) = \|\boldsymbol{S}\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|^2, \qquad (19)$$

where the matrix $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ is a circulant matrix representing the convolution for the anti-aliasing filter. The matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ is a binary sampling matrix, where the rows are subsets of the identity matrix. By defining $\boldsymbol{G} \overset{\text{def}}{=} \boldsymbol{S}\boldsymbol{H}$ we recognize that when substituting (19) into (5), the f-subproblem becomes (we dropped the iteration number k to simplify notation)

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\operatorname{argmin}} \quad \|\boldsymbol{G}\boldsymbol{x} - \boldsymbol{y}\|^2 + \frac{\rho}{2}\|\boldsymbol{x} - \widetilde{\boldsymbol{x}}\|^2. \qquad (20)$$

Consequently, the solution is the pseudo-inverse

$$\widehat{\boldsymbol{x}} = (\boldsymbol{G}^T\boldsymbol{G} + \rho\boldsymbol{I})^{-1}(\boldsymbol{G}^T\boldsymbol{y} + \rho\widetilde{\boldsymbol{x}}). \qquad (21)$$

For special cases of $\boldsymbol{H}$ and $\boldsymbol{S}$, (21) has known efficient implementation as follows.

Example 2 (Non-blind deblurring [3], [4], [39]): Non-blind deblurring is a special case when $\boldsymbol{S} = \boldsymbol{I}$. In this case, since $\boldsymbol{H}$ is circulant which is diagonalizable by the discrete Fourier transform matrices, (21) can be efficiently implemented by

$$\widehat{\boldsymbol{x}} = \mathcal{F}^{-1} \left\{ \frac{\overline{\mathcal{F}(h)}\mathcal{F}(\boldsymbol{y}) + \rho\mathcal{F}(\widetilde{\boldsymbol{x}})}{|\mathcal{F}(h)|^2 + \rho} \right\}, \qquad (22)$$

where $\mathcal{F}(\cdot)$ is the Fourier transform operator, h is the finite impulse response filter representing the blur kernel, $\overline{(\cdot)}$ is the complex conjugate, and the multiplication/division are element-wise operations.

Example 3 (Interpolation [6]–[8], [45]): Image interpolation is a special case when $\boldsymbol{H} = \boldsymbol{I}$. In this case, since $\boldsymbol{S}^T\boldsymbol{S}$ is a diagonal matrix with binary entries, (21) can be efficiently implemented using an element-wise division:

$$\widehat{\boldsymbol{x}} = (\boldsymbol{S}^T\boldsymbol{y} + \rho\widetilde{\boldsymbol{x}})./(\boldsymbol{s} + \rho), \qquad (23)$$

where $\boldsymbol{s} = \operatorname{diag}\{\boldsymbol{S}^T\boldsymbol{S}\}$.

B. Polyphase Implementation for Image Super-Resolution

When $\boldsymbol{G} = \boldsymbol{S}\boldsymbol{H}$, solving the f-subproblem becomes non-trivial because $\boldsymbol{H}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{H}$ is neither diagonal nor diagonalizable by the Fourier transform. In literature, the two most common approaches are to introduce multi-variable split to bypass (21) (e.g., [3], [46]) or use an inner conjugate gradient to solve (21) (e.g., [18]). However, multi-variable splitting requires additional Lagrange multipliers and internal parameters. It also generally leads to slower convergence than single-variable split. Inner conjugate gradient is computationally expensive as it requires an iterative solver. In what follows, we show that when $\boldsymbol{S}$ is the standard K-fold downsampler (i.e., sub-sample the spatial grid uniformly with a factor K along horizontal and vertical directions), and when $\boldsymbol{H}$ is a circular convolution, it is possible to derive a closed-form solution.[2]

Our closed form solution begins by considering the Sherman-Morrison-Woodbury identity, which allows us to rewrite (21) as

$$\widehat{\boldsymbol{x}} = \rho^{-1}\boldsymbol{b} - \rho^{-1}\boldsymbol{G}^T(\rho\boldsymbol{I} + \boldsymbol{G}\boldsymbol{G}^T)^{-1}\boldsymbol{G}\boldsymbol{b}, \qquad (24)$$

where $\boldsymbol{b} \overset{\text{def}}{=} \boldsymbol{G}^T\boldsymbol{y} + \rho\widetilde{\boldsymbol{x}}$. Note that if $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ with $m < n$, then (24) only involves a $m \times m$ inverse, which is smaller than the $n \times n$ inverse in (21).

The more critical step is the following observation. We note that the matrix $\boldsymbol{G}\boldsymbol{G}^T$ is given by

$$\boldsymbol{G}\boldsymbol{G}^T = \boldsymbol{S}\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{S}^T.$$

Since $\boldsymbol{S}$ is a K-fold downsampling operator, $\boldsymbol{S}^T$ is a K-fold upsampling operator. Defining $\widetilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{H}^T$, which can be implemented as a convolution between the blur kernel h and its time-reversal, we observe that $\boldsymbol{S}\widetilde{\boldsymbol{H}}\boldsymbol{S}^T$ is a "upsample-filter-downsample" sequence. This idea is illustrated in Fig. 4.

We next study the polyphase decomposition [47] of Fig. 4. Polyphase decomposition allows us to write

$$\widetilde{H}(z) = \sum_{k=0}^{K-1} z^{-k}\widetilde{H}_k(z^K), \qquad (25)$$

where $\widetilde{H}(z)$ is the z-transform representation of the blur matrix $\widetilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{H}^T$, and $\widetilde{H}_k(z^K)$ is the kth polyphase component of $\widetilde{H}(z)$. Illustrating (25) using a block diagram, we show in Fig. 5 the decomposed structure of Fig. 4. Then, using Noble identity [47], the block diagram on the left hand side of Fig. 5 becomes the one shown on the right hand side. Since for any $k > 1$, placing a delay z^{-k} between an upsampling and a downsampling operator leads to a zero, the overall system simplifies to a finite impulse response filter $\widetilde{H}_0(z)$, which can be pre-computed. We summarize this by the following proposition.

Proposition 1: The operation of $\boldsymbol{S}\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{S}^T$ is equivalent to applying a finite impulse response filter $\widetilde{H}_0(z)$, which is the 0th polyphase component of the filter $\boldsymbol{H}\boldsymbol{H}^T$.

To implement the 0th polyphase component, we observe that it can be done by downsampling the convolved

---

[2]We assume the boundaries are circularly padded. In case of other types boundary conditions or unknown boundary conditions, we can pre-process the image by padding the boundaries circularly. Then, after the super-resolution algorithm we crop the center region. The alternative approach is to consider multiple variable split as discussed in [46].
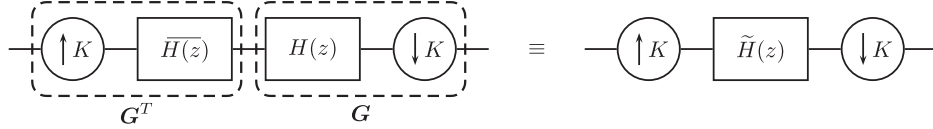
Fig. 4.   [Left] Block diagram of the operation $\boldsymbol{G}\boldsymbol{G}^T$. [Right] The equivalent system, where $\widetilde{H}(z) \stackrel{\text{def}}{=} \overline{H(z)}H(z)$.
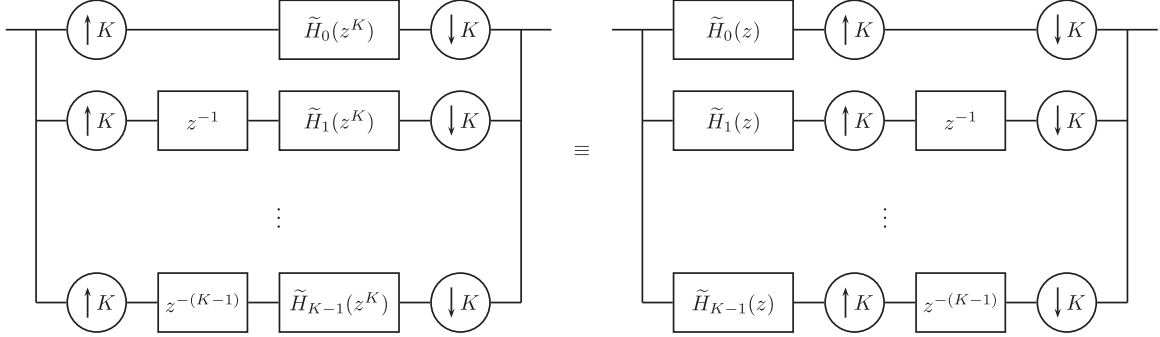

Fig. 5.   [Left] Polyphase decomposition of $\widetilde{H}(z)$. [Right] Equivalent representation.

Algorithm 2: Compute the 0th polyphase component.

Input: h: the blur kernel, and K: downsampling factor

Let $\widetilde{h} = \mathcal{F}^{-1}(\mathcal{F}(h)\overline{\mathcal{F}(h)})$ be the convolved filter.

Output: $\widetilde{h}_0 = (\downarrow_K)(\widetilde{h})$.

---

filter $\widetilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{H}^T$. This leads to the procedure illustrated in Algorithm 2.

The implication of Proposition 1 is that since $\boldsymbol{G}\boldsymbol{G}^T$ is equivalent to a finite impulse response filter $\widetilde{h}_0$, (24) can be implemented in closed-form using the Fourier transform:

$$\boldsymbol{x} = \rho^{-1}\boldsymbol{b} - \rho^{-1}\boldsymbol{G}^T\left(\mathcal{F}^{-1}\left\{\frac{\mathcal{F}(\boldsymbol{G}\boldsymbol{b})}{|\mathcal{F}(\widetilde{h}_0)|^2 + \rho}\right\}\right), \quad (26)$$

where we recall that $\boldsymbol{b} = \boldsymbol{G}^T\boldsymbol{y} + \rho\widetilde{\boldsymbol{x}}$.

The effectiveness of the proposed closed-form solution can be seen from Fig. 6. In this figure, we compare with a brute force conjugate gradient method presented in [18]. When $\boldsymbol{H}$ satisfies periodic boundary conditions, the closed-form solution is *exact*. If the boundaries are not periodic, alternative solutions can be considered, e.g., [48].

C. Application 2: Single Photon Imaging

The second application is a single photon imaging problem using quanta image sensors (QIS) [49]. Using ADMM for QIS was previously reported in [50], [51]. Here, we show how the Plug-and-Play ADMM can be used for the problem.

QIS is a spatial oversampling device. A QIS is composed of many tiny single photon detectors called jots. In each unit space, K jots are used to acquire light corresponding to a pixel in the usual sense (e.g., a pixel in a CMOS sensor). Therefore, for an image of n pixels, a total number of nK jots are required. By assuming homogeneous distribution of the light within each pixel, we consider a simplified QIS imaging model which relates
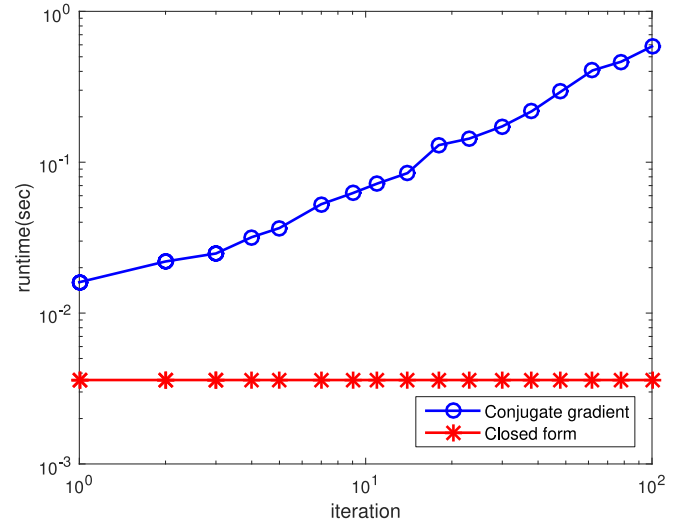

Fig. 6.   Runtime of conjugate gradient for solving a $\boldsymbol{x}$-subproblem. Note the non-iterative nature of the closed-form solution.

the underlying image $\boldsymbol{x} \in \mathbb{R}^n$ and the actual photon arrival rate at the jots $\boldsymbol{s} \in \mathbb{R}^{nK}$ as

$$\boldsymbol{s} = \alpha\boldsymbol{G}\boldsymbol{x},$$

where the matrix $\boldsymbol{G} \in \mathbb{R}^{nK \times n}$ is

$$\boldsymbol{G} = \frac{1}{K}\begin{bmatrix} \mathbf{1}_{K\times 1} & \mathbf{0}_{K\times 1} & \cdots & \mathbf{0}_{K\times 1} \\ \mathbf{0}_{K\times 1} & \mathbf{1}_{K\times 1} & \cdots & \mathbf{0}_{K\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{K\times 1} & \mathbf{0}_{K\times 1} & \cdots & \mathbf{1}_{K\times 1} \end{bmatrix}, \quad (27)$$

and α is a sensor gain. Given $\boldsymbol{s}$, the photons arriving at the sensors follow a Poisson distribution with a rate given by $\boldsymbol{s}$. Let Z_i be the random variable denoting the number of photons at

jot i, we have

$$p(z_i) = \frac{s_i^{-z_i} e^{-s_i}}{z_i!}, \quad i = 1, \dots, nK. \tag{28}$$

The final QIS output, Y_i, is a binary bit resulted from truncating Z_i using a threshold q. That is,

$$Y_i = \begin{cases} 1, & \text{if } Z_i \geq q, \\ 0, & \text{if } Z_i < q. \end{cases}$$

When $q = 1$, the probability of observing $Y_i = y_i$ given s_i is

$$p(y_i \mid s_i) = \begin{cases} e^{-s_i}, & \text{if } y_i = 0, \\ 1 - e^{-s_i}, & \text{if } y_i = 1. \end{cases}$$

The recovery goal is to estimate $\boldsymbol{x}$ from the observed binary bits $\boldsymbol{y}$. Taking the negative log and summing over all pixels, the function f is defined as

$$\begin{aligned} f(\boldsymbol{x}) &\stackrel{\text{def}}{=} p(\boldsymbol{y} \mid \boldsymbol{s}) = \sum_{i=1}^{nK} - \log \, p(y_i \mid s_i) \\ &= \sum_{i=1}^{nK} - \log \left((1 - y_i) e^{-s_i} + y_i \left(1 - e^{-s_i}\right) \right) \\ &= \sum_{j=1}^{n} - K_j^0 \log(e^{-\frac{\alpha x_j}{K}}) - K_j^1 \log(1 - e^{-\frac{\alpha x_j}{K}}), \quad (29) \end{aligned}$$

where $K_j^1 = \sum_{i=1}^{K} y_{(j-1)K+i}$ is the number of ones in the jth unit pixel, and $K_j^0 = \sum_{i=1}^{K} (1 - y_{(j-1)K+i})$ is the number of zeros in the jth unit pixel. (Note that for any j, $K_j^1 + K_j^0 = K$.) Consequently, substituting (29) into (10) yields the f-subproblem

$$\begin{aligned} \min_{\boldsymbol{x}} \sum_{j=1}^{n} &-K_j^0 \log(e^{-\frac{\alpha x_j}{K}}) - K_j^1 \log(1 - e^{-\frac{\alpha x_j}{K}}) \\ &+ \frac{\rho}{2} (x_j - \widetilde{x}_j)^2. \end{aligned} \tag{30}$$

Since this optimization is separable, we can solve each individual variable x_j independently. Thus, for every j, we solve a single-variable optimization by taking derivative with respect to x_j and setting to zero, yielding

$$K e^{-\frac{\alpha x_j}{K}} (\alpha + \rho(x_j - \widetilde{x}_j)) = \alpha K_j^0 + \rho K (x_j - \widetilde{x}_j),$$

which is a one-dimensional root finding problem. By constructing an offline lookup table in terms of K_0, ρ and $\widetilde{x}_j$, we can solve (30) efficiently.

IV. EXPERIMENTAL RESULTS

In this section we present the experimental results. For consistency we use BM3D in all experiments, although other bounded denoisers will also work. We shall not compare Plug-and-Play ADMM using different denoisers as it is not the focus of the paper.

Fig. 7.   10 testing images for the experiment.

TABLE I
CONFIGURATIONS AND PARAMETERS

Config	Description
1	$K = 2$, $\boldsymbol{H} =$ bicubic, Noise $= 0$
2	$K = 4$, $\boldsymbol{H} =$ bicubic, Noise $= 0$
	$\rho_0 = 1.3 \times 10^{-5}, \gamma = 2.5, \lambda = 10^{-5}$
3	$K = 2$, $\boldsymbol{H} =$ Gaussian of std 1, Noise $= 5/255$
4	$K = 4$, $\boldsymbol{H} =$ Gaussian of std 1, Noise $= 5/255$
	$\rho_0 = 1 \times 10^{-5}, \gamma = 1.2, \lambda = 10^{-4}$

A. Image Super-Resolution

We consider a set of 10 standard test images for this experiment as shown in Fig. 7. All images are gray-scaled, with sizes between 256×256 and 512×512. Four sets of experimental configurations are studied, and are shown in Table I.

We compared the proposed algorithm with several existing super-resolution algorithms. These methods include the deep convolutional neural network method (DCNN) by Dong *et al.* [9], the statistical patch-based sparse representation method (SPSR) by Peleg and Elad [10], the transformed self-exemplar method (TSE) by Huang *et al.* [52], the classical sparse representation method for super-resolution (SR) by Yang *et al.* [12], and the Gaussian process regression method (GPR) by He and Siu [11]. Among these methods, we note that TSE and GPR are single image methods whereas DCNN, SPSR and SR require training using external databases.

The results of the experiment are shown in Table II. For the proposed algorithm, we present the two update rules as Ours-M (for monotone update rule), and Ours-A (for adaptive update rule). When noise is present in the simulation, we conduct a Monte-Carlo simulation over 5 random realizations. In this case, the reported PSNR values are the average over the random realizations. The per image standard deviation is reported in the last column of Table II (if applicable).

For configurations 3 and 4 when we use a Gaussian anti-aliasing filter, we observe that not all existing methods can handle such case as the training part of those algorithms was performed on a bicubic model. Therefore, for fair comparison, we present two versions of the proposed algorithm. The first version Ours-M assumes the correct knowledge about the Gaussian filter, whereas the second version Ours-

TABLE II
IMAGE SUPER RESOLUTION RESULTS

Size	Images										Dataset Avg	Avg STD per image
	1 512^2	2 512^2	3 256^2	4 512^2	5 512^2	6 512^2	7 256^2	8 512^2	9 512^2	10 256^2		
Factor: $\times 2$; Anti-aliasing Filter: Bicubic; Noise: 0												
DCNN [9]	25.71	31.83	28.79	31.13	32.73	**32.60**	35.11	36.34	**33.10**	33.05	32.04	–
SR [12]	**25.87**	31.51	27.92	30.94	33.02	32.46	34.79	36.14	32.80	32.67	31.81	–
SPSR [10]	25.71	31.49	27.85	31.00	33.30	32.35	34.37	36.18	32.67	32.66	31.76	–
TSE [52]	25.66	31.64	28.17	31.01	32.88	32.45	34.78	36.22	32.88	**33.29**	31.90	–
GPR [11]	24.99	29.99	26.44	29.50	30.36	31.33	33.08	34.19	31.09	30.98	30.20	–
Ours - M	25.87	**31.85**	**28.81**	**31.42**	**33.63**	32.56	**35.30**	**36.45**	32.86	33.06	**32.18**	–
Ours - A	25.74	31.68	28.44	31.27	33.49	32.39	35.20	36.02	32.65	32.65	31.95	–
Factor: $\times 4$; Anti-aliasing Filter: Bicubic; Noise: 0												
DCNN [9]	23.93	26.84	23.76	26.08	24.18	28.32	29.48	30.45	28.17	27.88	26.91	–
SR [12]	23.91	26.39	23.43	26.02	24.44	28.17	29.15	30.19	27.94	27.42	26.71	–
SPSR [10]	23.90	26.49	23.42	26.02	25.11	28.14	29.22	30.24	27.85	27.62	26.80	–
TSE [52]	23.90	26.62	23.83	26.10	24.70	28.06	30.03	30.29	28.03	27.97	26.95	–
GPR [11]	23.55	25.47	22.54	25.27	22.33	27.79	27.61	28.74	26.76	25.79	25.58	–
Ours - M	**23.99**	26.87	23.82	**26.32**	25.54	**28.35**	30.16	**30.74**	**28.17**	**28.26**	27.22	–
Ours - A	23.99	**26.87**	**23.83**	26.33	**25.58**	28.29	**30.48**	30.62	28.12	28.22	**27.23**	–
Factor: $\times 2$; Anti-aliasing Filter: Gaussian 9×9, $\sigma = 1$; Noise: $5/255$												
DCNN [9]	23.61	26.30	23.75	26.21	23.79	27.50	27.84	28.15	27.05	26.07	26.03	0.0219
SR [12]	23.61	26.25	23.71	26.15	23.80	27.41	27.71	28.07	26.99	26.10	25.98	0.0221
SPSR [10]	23.75	26.57	23.88	26.47	23.91	27.80	28.19	28.58	27.33	26.39	26.29	0.0208
TSE [52]	23.57	26.22	23.65	26.12	23.79	27.34	27.55	28.00	26.93	26.11	25.93	0.0238
GPR [11]	23.82	26.81	23.91	26.63	24.05	28.38	29.16	29.54	27.78	26.76	26.68	0.0170
Ours - M	**24.64**	**29.41**	**26.73**	**29.22**	**28.82**	**29.82**	**32.65**	**32.76**	**29.66**	**30.10**	**29.38**	0.0267
Ours - M*	24.01	27.09	24.17	27.00	25.03	28.46	29.32	29.78	27.85	26.99	26.97	0.0178
Factor: $\times 4$; Anti-aliasing Filter: Gaussian 9×9, $\sigma = 1$; Noise: $5/255$												
DCNN [9]	20.72	21.30	18.91	21.68	16.10	23.39	22.33	22.99	22.46	20.23	21.01	0.0232
SR [12]	20.67	21.30	18.86	21.51	16.37	23.15	22.19	22.85	22.26	20.33	20.95	0.0212
SPSR [10]	20.85	21.58	19.18	21.85	16.59	23.52	22.42	23.05	22.53	20.50	21.21	0.0217
TSE [52]	20.59	21.24	18.80	21.49	16.40	23.14	22.21	22.78	22.21	20.30	20.92	0.0252
GPR [11]	21.55	22.68	19.90	22.77	17.70	24.57	23.51	24.37	23.63	21.35	22.20	0.0313
Ours - M	**23.62**	**25.75**	**23.06**	**25.30**	**24.48**	**27.17**	**29.14**	**29.42**	**26.86**	**26.86**	**26.17**	0.0223
Ours - M*	21.21	22.12	19.43	22.43	16.90	24.37	23.13	23.95	23.39	21.13	21.81	0.0253

When noise is present, the PSNR values are averaged over 5 random realizations of the noise pattern. Gray color rows represent external database methods.

Fig. 8. Image Super Resolution Results. [Top] (from left to right). Ground truth; The low resolution input; Bicubic interpolation; DCNN [9] (24.19 dB). [Bottom] (from left to right). SPSR [10] (22.44 dB); TSE [52] (22.80 dB); GPR [11] (20.81 dB); Ours-M (23.49 dB).

M* ignores such assumption and use the bicubic model for reconstruction.

From the PSNR results shown in Table II, we observe very similar performance of the competing methods. For configurations 1 and 2, the proposed algorithm shows the best performance overall, although in some occasions the deep neural network [9] is better. For configurations 3 and 4, we observe a significant gap between Ours-M and the competing methods. This is caused by the model mismatch of the competing methods as the implementations provided by the authors only support the bicubic model. For fairness, we consider Ours-M* by pretending that the anti-aliasing filter is bicubic. In this case, Ours-M* still performs better than the others for configuration 3, but slightly worse than GPR [11] for configuration 4.

For visual comparison we conduct a color image experiment. In this experiment, we simulate a low resolution image by downsampling the color image by a factor 4 using a bicubic anti-aliasing filter. Then, we apply the proposed algorithm to the three color channels individually to recover the image. The result is shown in Fig. 8. As seen, the proposed method produces better results than SPSR [10], TSE [52] and GPR [11], with slightly sharper edges and less halo artifacts. We also observe that the deep neural network [9] shows better results than that in Table II. One possibility is that the training data used to train the neural network are natural images that have better correlation to Fig. 8. However, considering the training-free nature of the proposed Plug-and-Play algorithm, losing to a well-trained neural network is not surprising.

B. Single Photon Imaging

We next consider the single-photon imaging problem. In this experiment, we consider four sets of experiments for $K = 4, 6, 8, 10$ (along horizontal and vertical directions). The sensor gain is set as $\alpha = K^2$. For comparison, we choose the two existing algorithms. The first one is the maximum likelihood estimation (MLE) method by Yang *et al.* [53]. For our specific choice of G in (27), the MLE solution has a closed-form expression. The second method is a total variation method by Chan and Lu [50]. This method utilizes the ADMM algorithm when solving the problem. We are aware of other existing Poisson denoising methods such as [17], [54]. However, none of these methods are directly applicable to the quantized Poisson problem.

Since for this problem the observed binary pattern is a truncated Poisson random variable, we perform a Monte-Carlo simulation by repeating each case for 8 independent trials. We then report the average and the standard deviation of these 8 independent trials. As shown in Table III, the standard deviation is indeed insignificant compared to the average PSNR. Here, we report the dataset average over the 10 images to ensure sufficient variability of the test. To visually compare the performance, in Fig. 9 we show the result of a color image. In this experiment, we process the 3 color channels individually. For each channel, we simulate the photon arrivals by assuming $K = 8$. Then, we reconstruct the image using different algorithms. The result in Fig. 9 shows that visually the proposed algorithm produces images with less noise.

V. Conclusion

We presented a continuation scheme for Plug-and-Play ADMM. We showed that for any *bounded denoisers* (denoisers that asymptotically converges to the identity operator), the new Plug-and-Play ADMM has a provable fixed point convergence. We demonstrated two applications of the new algorithm for single image super-resolution and the single photon imaging problem. For the single image super-resolution problem, we presented a closed-form approach to solve one of the two subproblems in the ADMM algorithm. The closed-form result allows significantly faster implementation than iterative methods. Experimentally, we found that Plug-and-Play ADMM performs better than several existing methods.

Appendix A
Counter Example of Non-Expansive Denoiser

As mentioned in Section II-B, showing non-expansiveness of a denoiser could be difficult. Here we provide a counter example for the non-local means [35].

To show that non-local means are expansive, we only need to find a pair $(\boldsymbol{x}, \boldsymbol{y})$ such that

$$\kappa = \|\mathcal{D}_\sigma(\boldsymbol{x}) - \mathcal{D}_\sigma(\boldsymbol{y})\|^2 / \|\boldsymbol{x} - \boldsymbol{y}\|^2 > 1.$$

To construct such example, we show in Fig. 10 a pair of $(\boldsymbol{x}, \boldsymbol{y})$ obtained through an inpainting problem using Plug-and-Play ADMM with constant ρ, i.e., $\gamma = 1$. (In fact, it does not matter how we obtain this pair of $(\boldsymbol{x}, \boldsymbol{y})$. All we need to show is that there exists $(\boldsymbol{x}, \boldsymbol{y})$ which makes $\kappa > 1$.)

The non-local means is a weighted average operation with weights

$$W_{\mathrm{ij}} = \exp\big\{ -\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 / (2\sigma^2)\big\},$$

where $\boldsymbol{x}_i$ is the ith patch of the image $\boldsymbol{x}$. To further ensure that $\boldsymbol{W}$ is doubly stochastic so that its eigenvalues are bounded between 0 and 1, we apply Sinkhorn-Knopp [55] to $\boldsymbol{W}$ until convergence. Define this doubly stochastic matrix as $\widetilde{\boldsymbol{W}}$. Then, the denoised output is given by

$$\widetilde{\boldsymbol{x}} = \mathcal{D}_\sigma(\boldsymbol{x}) \stackrel{\text{def}}{=} \widetilde{\boldsymbol{W}}\boldsymbol{x}.$$

Therefore, the ratio we need to check is

$$\kappa = \|\widetilde{\boldsymbol{W}}_{\boldsymbol{x}}(\boldsymbol{x}) - \widetilde{\boldsymbol{W}}_{\boldsymbol{y}}(\boldsymbol{y})\|^2 / \|\boldsymbol{x} - \boldsymbol{y}\|^2,$$

where the subscript $(\cdot)_{\boldsymbol{x}}$ specifies the dependency of $\widetilde{\boldsymbol{W}}$ on $\boldsymbol{x}$ (or $\boldsymbol{y}$). The denoised results are shown in Fig. 10(c) and (d). Although it may look subtle, one can verify that $\kappa = 1.1775$ which violates the requirement of non-expansiveness. This happens because $\widetilde{\boldsymbol{W}}_{\boldsymbol{x}} \neq \widetilde{\boldsymbol{W}}_{\boldsymbol{y}}$ for $\boldsymbol{x} \neq \boldsymbol{y}$. The dependency on $\boldsymbol{x}$ and $\boldsymbol{y}$ makes the operators nonlinear, and hence makes non-expansiveness difficult to validate.

Readers at this point may wonder why the proposed Plug-and-Play ADMM can alleviate the expansive issue. In a nutshell, the reason is that we force $\rho \rightarrow \infty$ so that $\sigma \rightarrow 0$. Consequently, the weight $\boldsymbol{W} \rightarrow \boldsymbol{I}$ as $\rho \rightarrow \infty$. For the original Plug-and-Play ADMM in [15], $\boldsymbol{W} \nrightarrow \boldsymbol{I}$ because ρ is fixed.

TABLE III
SINGLE PHOTON IMAGING RESULTS

Size	Images 1 512^2	2 512^2	3 256^2	4 512^2	5 512^2	6 512^2	7 256^2	8 512^2	9 512^2	10 256^2	Dataset Avg	Avg STD per image
					$K = 4$							
Yang *et al.* [53]	14.80	14.18	14.68	14.39	14.28	14.90	14.21	14.48	14.78	14.59	14.53	0.0157
Chan-Lu [50]	22.59	24.76	23.63	24.74	21.47	25.65	25.38	26.42	25.35	24.78	24.48	0.0425
Ours-M	25.99	25.72	25.58	26.03	23.54	26.60	28.15	28.17	26.17	26.10	26.20	0.0821
Ours-A	**26.06**	**25.75**	**25.64**	**26.10**	**23.59**	**26.66**	**28.27**	**28.27**	**26.18**	**26.16**	**26.27**	0.0801
					$K = 6$							
Yang *et al.* [53]	17.94	17.27	17.67	17.61	17.29	18.22	17.22	17.62	18.00	17.65	17.65	0.0147
Chan-Lu [50]	23.97	26.25	25.53	26.26	24.47	26.66	26.75	27.32	26.58	26.40	26.02	0.0339
Ours-M	**28.34**	**27.76**	27.60	**27.91**	**25.66**	**28.30**	**30.34**	**30.06**	**27.75**	28.15	**28.19**	0.0451
Ours-A	**28.34**	27.72	**27.61**	27.84	25.62	28.25	30.28	29.86	27.71	**28.16**	28.14	0.0472
					$K = 8$							
Yang *et al.* [53]	20.28	19.68	20.00	20.05	19.53	20.64	19.49	19.99	20.42	19.97	20.01	0.0183
Chan-Lu [50]	25.14	27.09	26.56	27.24	25.75	27.55	27.17	27.89	27.49	27.07	26.90	0.0325
Ours-M	**29.79**	**29.07**	**29.14**	**29.25**	**27.19**	**29.55**	**31.70**	**31.43**	**28.99**	**29.52**	**29.56**	0.0527
Ours-A	29.74	29.00	29.09	29.16	27.14	29.51	31.54	31.35	28.94	29.43	29.49	0.0520
					$K = 10$							
Yang *et al.* [53]	22.14	21.60	21.89	21.98	21.32	22.51	21.32	21.86	22.35	21.83	21.88	0.0198
Chan-Lu [50]	26.20	27.57	27.26	27.70	26.41	27.99	27.64	28.16	27.95	27.66	27.46	0.0264
Ours-M	**30.88**	**30.19**	**30.34**	**30.31**	**28.29**	**30.48**	**32.68**	**32.29**	**29.97**	**30.56**	**30.60**	0.0386
Ours-A	30.81	30.12	30.31	30.22	28.22	30.41	32.51	32.17	29.90	30.47	30.51	0.0397

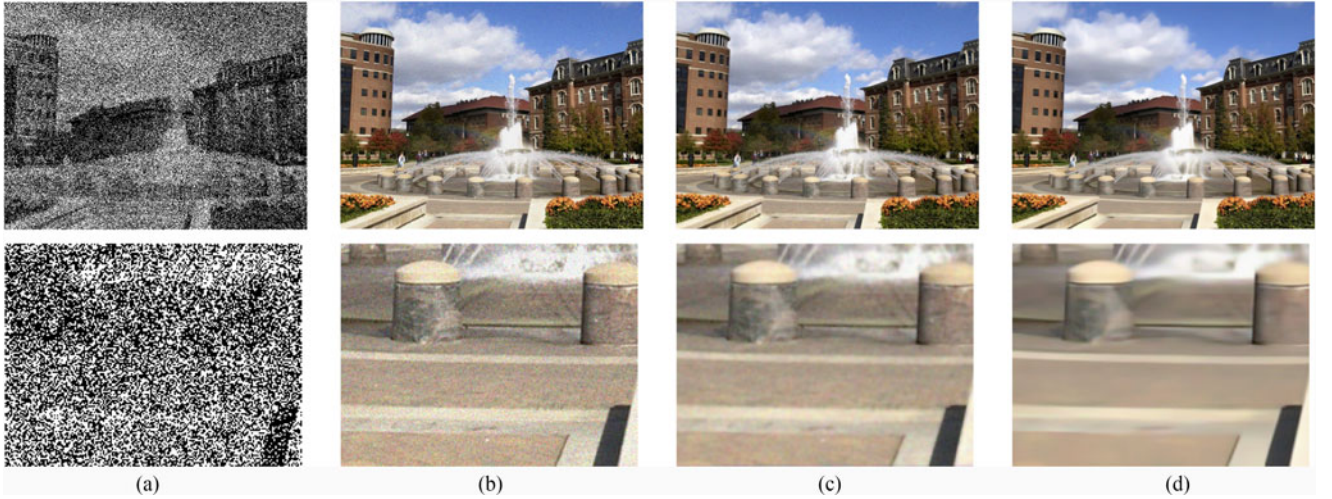The PSNR values are averaged over 8 random realizations of the photon arrivals.

Fig. 9. Single photon imaging results. The bottom row is a zoomed-in figure of the top row (a) Binary input (b) Yang *et al.* [53] 20.16 dB (c) Chan and Lu [50] 26.74 dB (d) Ours-C 28.81 dB.
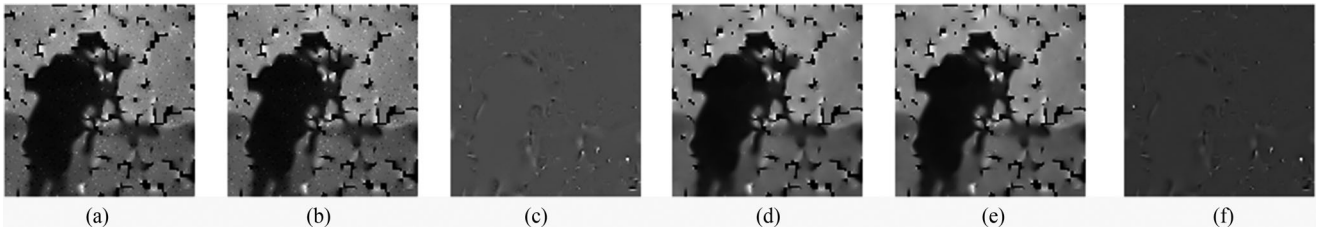

Fig. 10. Counter example showing non-local means is expansive. $\kappa = \|\mathcal{D}_\sigma(\boldsymbol{x}) - \mathcal{D}_\sigma(\boldsymbol{y})\|^2 / \|\boldsymbol{x} - \boldsymbol{y}\|^2 = 1.1775$ (a) x (b) y (c) $x - y$ (d) $\mathcal{D}_\sigma(x)$ (e) $\mathcal{D}_\sigma(y)$ (f) $\mathcal{D}_\sigma(x) - \mathcal{D}_\sigma(Y)$.

APPENDIX B
PROOF OF THEOREM 1

To simplify the notations we first define a triplet $\boldsymbol{\theta}^{(k)} \stackrel{\text{def}}{=} (\boldsymbol{x}^{(k)}, \boldsymbol{v}^{(k)}, \boldsymbol{u}^{(k)})$. Let $\boldsymbol{\Theta}$ be the domain of $\boldsymbol{\theta}^{(k)}$ for all k. On $\boldsymbol{\Theta}$ we define a distance function $D : \boldsymbol{\Theta} \times \boldsymbol{\Theta} \to \mathbb{R}$ such that

$$D(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(j)}) = \frac{1}{\sqrt{n}} \Big(\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(j)}\|_2 + \|\boldsymbol{v}^{(k)} - \boldsymbol{v}^{(j)}\|_2$$
$$+ \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(j)}\|_2 \Big).$$

It then follows that $\Delta_{k+1} = D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)})$. Since $\boldsymbol{\Theta} \subseteq \mathbb{R}^{3n}$ and $\mathbb{R}^{3n}$ is a complete metric space, as long as we can show that $\{\boldsymbol{\theta}^{(k)}\}_{k=1}^{\infty}$ is a Cauchy sequence in $\boldsymbol{\Theta}$ with the distance function D, then $\boldsymbol{\theta}^{(k)}$ should converge.

The Plug-and-Play ADMM involves two cases of the parameter update:

1) Case 1: If $\Delta_{k+1} > \eta \Delta_k$, then $\rho_{k+1} = \gamma \rho_k$.
2) Case 2: If $\Delta_{k+1} \leq \eta \Delta_k$, then $\rho_{k+1} = \rho_k$.

At iteration k, if Case 1 holds, then by Lemma 1, $\boldsymbol{\theta}^{(k+1)}$ satisfies

$$D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \leq \frac{C'}{\sqrt{\rho_k}}, \tag{31}$$

for some universal constant $C' > 0$ independent of k. On the other hand, if Case 2 holds, then since $\Delta_{k+1} = D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)})$ we have

$$D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \leq \eta D(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^{(k-1)}). \tag{32}$$

As $k \to \infty$, one of the following situations will happen:

S1): Case 1 occurs infinitely many times but Case 2 occurs finitely many times;
S2): Case 2 occurs infinitely many times but Case 1 occurs finitely many times;
S3): Both Case 1 and Case 2 occur infinitely many times.

These three cases can be analyzed as follows. When (S_1) happens, there must exists a K_1 such that for $k \geq K_1$ only Case 1 will be visited. Thus,

$$D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \leq \frac{C'}{\sqrt{\rho_{K_1-1}} \sqrt{\gamma}^{k-K_1}}.$$

When (S_2) happens, there must exists a K_2 such that for $k \geq K_2$ only Case 2 will be visited. Thus, we have

$$D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \leq \eta^{k-K_2} D\Big(\boldsymbol{\theta}^{(K_2)}, \boldsymbol{\theta}^{(K_2-1)} \Big)$$
$$\leq \eta^{k-K_2} \frac{C'}{\rho_{K_2-1}}.$$

(S_3) is a union o the (S_1) and (S_2). Therefore, as long as we can show under (S_1) and (S_2) the sequence $\{\boldsymbol{\theta}^{(k)}\}_{k=1}^{\infty}$ converges, the sequence will also converge under (S_3). To summarize, we show in Lemma 2 that regardless which of (S_1)–(S_3), for any k we have

$$D(\boldsymbol{\theta}^{k+1}, \boldsymbol{\theta}^{(k)}) \leq C'' \delta^k,$$

for some constants C'' and $0 < \delta < 1$. Therefore,

$$D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \to 0, \tag{33}$$

as $k \to \infty$.

To prove $\{\boldsymbol{\theta}^{(k)}\}_{k=1}^{\infty}$ is a Cauchy sequence, we need to show

$$D(\boldsymbol{\theta}^{(m)}, \boldsymbol{\theta}^{(k)}) \to 0, \tag{34}$$

for all integers $m > k$ and $k \to \infty$. This result holds because for any finite m and k,

$$D(\boldsymbol{\theta}^{(m)}, \boldsymbol{\theta}^{(k)}) \leq \sum_{n=k+1}^{m} C'' \delta^n$$
$$= \sum_{\ell=1}^{m-k} C'' \delta^{\ell+k}$$
$$= C'' \delta^k \frac{1 - \delta^{m-k+1}}{1 - \delta}.$$

Therefore, as $k \to \infty$, $D(\boldsymbol{\theta}^{(m)}, \boldsymbol{\theta}^{(k)}) \to 0$. Hence, $\{\boldsymbol{\theta}^{(k)}\}_{k=1}^{\infty}$ is a Cauchy sequence. Since a Cauchy sequence in $\mathbb{R}^{3n}$ always converges, there must exists $\boldsymbol{\theta}^* = (\boldsymbol{x}^*, \boldsymbol{v}^*, \boldsymbol{u}^*)$ such that

$$D(\boldsymbol{\theta}^{(k)}, \boldsymbol{\theta}^*) \to 0. \tag{35}$$

Consequently, we have $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|_2 \to 0$, $\|\boldsymbol{v}^{(k)} - \boldsymbol{v}^*\|_2 \to 0$ and $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^*\|_2 \to 0$. This completes the proof.

Lemma 1: At iteration k, if Case 1 holds, then

$$D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \leq \frac{C'}{\sqrt{\rho_k}}, \tag{36}$$

for some universal constant $C' > 0$ independent of k.

Proof: Following the definition of $D(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{\theta}^{(k)})$, it is sufficient to show that

$$\frac{1}{\sqrt{n}} \left\| \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)} \right\|_2 \leq \frac{C_1}{\sqrt{\rho_k}},$$
$$\frac{1}{\sqrt{n}} \left\| \boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)} \right\|_2 \leq \frac{C_2}{\sqrt{\rho_k}},$$
$$\frac{1}{\sqrt{n}} \left\| \boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)} \right\|_2 \leq \frac{C_3}{\sqrt{\rho_k}},$$

for some universal constants C_1, C_2 and C_3.

Let us consider

$$\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ f(\boldsymbol{x}) + \frac{\rho_k}{2} \|\boldsymbol{x} - (\boldsymbol{v}^{(k)} - \boldsymbol{u}^{(k)})\|^2.$$

The first order optimality implies that

$$\boldsymbol{x} - (\boldsymbol{v}^{(k)} - \boldsymbol{u}^{(k)}) = -\frac{1}{\rho_k} \nabla f(\boldsymbol{x}).$$

Since the minimizer is $\boldsymbol{x} = \boldsymbol{x}^{(k+1)}$, substituting $\boldsymbol{x} = \boldsymbol{x}^{(k+1)}$ and using the fact that ∇f is bounded yields

$$\frac{1}{\sqrt{n}} \left\| \boldsymbol{x}^{(k+1)} - (\boldsymbol{v}^{(k)} - \boldsymbol{u}^{(k)}) \right\|_2 = \frac{\|\nabla f(\boldsymbol{x})\|_2}{\rho_k \sqrt{n}} \leq \frac{L}{\rho_k}. \tag{37}$$

Next, let $\widetilde{\boldsymbol{v}}^{(k)} = \boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)}$ and $\sigma_k = \sqrt{\lambda/\rho_k}$. Define

$$\boldsymbol{v}^{(k+1)} = \mathcal{D}_{\sigma_k}(\widetilde{\boldsymbol{v}}^{(k)}).$$

Since $\mathcal{D}_{\sigma_k}$ is a bounded denoiser, we have that

$$\frac{1}{\sqrt{n}}\left\|\boldsymbol{v}^{(k+1)} - (\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)})\right\|_2 = \frac{1}{\sqrt{n}}\left\|\boldsymbol{v}^{(k+1)} - \widetilde{\boldsymbol{v}}^{(k)}\right\|_2$$

$$= \frac{1}{\sqrt{n}}\left\|\mathcal{D}_{\sigma_k}(\widetilde{\boldsymbol{v}}^{(k)}) - \widetilde{\boldsymbol{v}}^{(k)}\right\|_2 \le \sigma_k\sqrt{C} = \frac{\sqrt{\lambda}\sqrt{C}}{\sqrt{\rho_k}}. \quad (38)$$

We can now bound $\left\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\right\|_2$ as follows.

$$\frac{1}{\sqrt{n}}\left\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\right\|_2 \le \frac{1}{\sqrt{n}}\left\|\boldsymbol{v}^{(k+1)} - (\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)})\right\|_2$$

$$+ \frac{1}{\sqrt{n}}\left\|(\boldsymbol{x}^{(k+1)} + \boldsymbol{u}^{(k)}) - \boldsymbol{v}^{(k)}\right\|_2.$$

Using (37) and (38), we have

$$\frac{1}{\sqrt{n}}\left\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\right\|_2 \le \frac{\sqrt{\lambda}\sqrt{C}}{\sqrt{\rho_k}} + \frac{L}{\rho_k}$$

$$= \frac{1}{\sqrt{\rho_k}}\left(\sqrt{\lambda}\sqrt{C} + \frac{L}{\sqrt{\rho_k}}\right) \le \frac{1}{\sqrt{\rho_k}}\left(\sqrt{\lambda}\sqrt{C} + \frac{L}{\sqrt{\rho_0}}\right). \quad (39)$$

Similarly, we can show that

$$\frac{1}{\sqrt{n}}\left\|\boldsymbol{u}^{(k+1)}\right\|_2 = \frac{1}{\sqrt{n}}\left\|\boldsymbol{u}^{(k)} + (\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)})\right\|_2$$

$$= \frac{1}{\sqrt{n}}\left\|\boldsymbol{u}^{(k)} + \boldsymbol{x}^{(k+1)} - \mathcal{D}_{\sigma_k}(\widetilde{\boldsymbol{v}}^{(k)})\right\|_2$$

$$= \frac{1}{\sqrt{n}}\left\|\boldsymbol{u}^{(k)} + \boldsymbol{x}^{(k+1)} - (\mathcal{D}_{\sigma_k}(\widetilde{\boldsymbol{v}}^{(k)}) - \widetilde{\boldsymbol{v}}^{(k)}) - \widetilde{\boldsymbol{v}}^{(k)}\right\|_2$$

$$\stackrel{(a)}{=} \frac{1}{\sqrt{n}}\left\|\mathcal{D}_{\sigma_k}(\widetilde{\boldsymbol{v}}^{(k)}) - \widetilde{\boldsymbol{v}}^{(k)}\right\|_2 \le \frac{\sqrt{\lambda}\sqrt{C}}{\sqrt{\rho_k}}, \quad (40)$$

where (a) holds because $\widetilde{\boldsymbol{v}}^{(k)} = \boldsymbol{u}^{(k)} + \boldsymbol{x}^{(k+1)}$. Thus,

$$\frac{1}{\sqrt{n}}\left\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\right\|_2 \le \frac{1}{\sqrt{n}}\left(\left\|\boldsymbol{u}^{(k+1)}\right\|_2 + \left\|\boldsymbol{u}^{(k)}\right\|_2\right)$$

$$\le \frac{2\sqrt{\lambda}\sqrt{C}}{\sqrt{\rho_k}}.$$

Finally, since $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + (\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)})$, we have

$$\frac{1}{\sqrt{n}}\left\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right\|_2 = \frac{1}{\sqrt{n}}\left\|\left(\boldsymbol{u}^{(k+1)}\right.\right.$$

$$\left.\left. - \boldsymbol{u}^{(k)} + \boldsymbol{v}^{(k+1)}\right) - \left(\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(k-1)} + \boldsymbol{v}^{(k)}\right)\right\|_2$$

$$\le \frac{1}{\sqrt{n}}\left(\left\|\boldsymbol{u}^{(k+1)} - \boldsymbol{u}^{(k)}\right\|_2 + \left\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{(k-1)}\right\|_2\right.$$

$$\left. + \left\|\boldsymbol{v}^{(k+1)} - \boldsymbol{v}^{(k)}\right\|_2\right)$$

$$\le \frac{2\sqrt{\lambda}\sqrt{C}}{\sqrt{\rho_k}} + \frac{2\sqrt{\lambda}\sqrt{C}}{\sqrt{\rho_{k-1}}} + \frac{1}{\sqrt{\rho_k}}\left(\sqrt{\lambda}\sqrt{C} + \frac{L}{\sqrt{\rho_k}}\right)$$

$$\le \left((3 + 2\sqrt{\gamma})\sqrt{\lambda}\sqrt{C} + \frac{L}{\sqrt{\rho_0}}\right)\frac{1}{\sqrt{\rho_k}}. \quad (41)$$

■

Lemma 2: The sequence $\{\boldsymbol{\theta}^{(k)}\}_{k=1}^{\infty}$ always satisfies

$$D(\boldsymbol{\theta}^{k+1}, \boldsymbol{\theta}^{(k)}) \le C''\delta^k, \quad (42)$$

for some constants C'' and $0 < \delta < 1$.

Proof: At any iteration k, it holds that

$$D(\boldsymbol{\theta}^{k+1}, \boldsymbol{\theta}^{(k)}) \le \max\left(\frac{C'}{\sqrt{\rho_{K_1-1}}\sqrt{\gamma}^{k-K_1}}, \eta^{k-K_2}\frac{C'}{\rho_{K_2-1}}\right)$$

$$\le \max\left(C_1'\left(\frac{1}{\sqrt{\gamma}}\right)^k, C_2'\eta^k\right),$$

where $C_1' = C'\sqrt{\frac{\gamma^{K_1}}{\rho_{K_1-1}}}$ and $C_2' = \frac{C'\eta^{-K_2}}{\rho_{K_2-1}}$. Therefore, by letting

$$C'' = \max(C_1', C_2'), \quad \text{and} \quad \delta = \max(1/\sqrt{\gamma}, \eta),$$

we obtain the desired result, as $\gamma > 1$. ■

ACKNOWLEDGMENT

REFERENCES

[1] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed. New York, NY, USA: Springer, 1998.

[2] C. A. Bouman, *Model-based image processing*. 2015. [Online]. Available: https://engineering.purdue.edu/bouman/publications/pdf/MBIP-book.pdf

[3] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Apr. 2010.

[4] 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, May 2011.

[5] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM J. Sci. Comput.*, vol. 31, no. 4, pp. 2842–2865, Jul. 2009.

[6] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algorithms*, vol. 53, no. 1, pp. 67–92, Jan. 2010.

[7] D. Garcia, "Robust smoothing of gridded data in one and higher dimensions with missing values," *Comput. Statist. Data Anal.*, vol. 54, no. 4, pp. 1167–1178, Apr. 2010.

[8] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley, "Non-parametric Bayesian dictionary learning for sparse image representations," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2009, vol. 22, pp. 2295–2303.

[9] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vision*, Sep. 2014, pp. 184–199.

[10] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2569–2582, Jun. 2014.

[11] H. He and W. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2011, pp. 449–456.

[12] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2008, pp. 1–8.

[13] S. Sreehari *et al.*, "Plug-and-play priors for bright field electron tomography and sparse interpolation," Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.07331

[14] 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. [Online]. Available: http://dx.doi.org/10.1561/2200000016

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

[16] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, "Postprocessing of compressed images via sequential denoising," Oct. 2015. [Online]. Available: http://arxiv.org/abs/1510.09041

[17] A. Rond, R. Giryes, and M. Elad, "Poisson inverse problems by the plug-and-play scheme," Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.02500

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

[19] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," Nov. 2015. [Online]. Available: http://arxiv.org/abs/1410.1390

[20] P. Milanfar, "Symmetrizing smoothing filters," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 263–284, 2013.

[21] S. H. Chan, T. Zickler, and Y. M. Lu, "Understanding symmetric smoothing filters via Gaussian mixtures," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 2500–2504.

[22] S. H. Chan, "Algorithm-induced prior for image restoration," Feb. 2016. [Online]. Available: http://arxiv.org/abs/1602.00715

[23] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Process. Mag.*, vol. 30, pp. 106–128, Jan. 2013. [Online]. Available: http://cs.ucsc.edu/milanfar/publications/journal/ModernTour_FinalSubmission.pdf

[24] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," Jul. 2014. [Online]. Available: http://arxiv.org/abs/1406.4175

[25] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "BM3D-AMP: A new image recovery algorithm based on BM3D denoising," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 3116–3120.

[26] J. Tan, Y. Ma, and D. Baron, "Compressive imaging via approximate message passing with wavelet-based image denoising," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2014, pp. 424–428.

[27] J. Tan, Y. Ma, and D. Baron, "Compressive imaging via approximate message passing with image denoising," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2085–2092, Apr. 2015.

[28] R. Neelamani, H. Choi, and R. Baraniuk, "ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 418–433, Feb. 2004.

[29] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, Aug. 2012.

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

[31] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 5–28, Jan. 2005.

[32] K. Sivaramakrishnan and T. Weissman, "A context quantization approach to universal denoising," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2110–2129, Jun. 2009.

[33] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2011, pp. 2833–2840.

[34] A. Levin, B. Nadler, F. Durand, and W. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *Proc. 12th Eur. Conf. Comput. Vision*, Oct. 2012, vol. 7576, pp. 73–86.

[35] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *SIAM Multiscale Model Simul.*, vol. 4, no. 2, pp. 490–530, 2005. [Online]. Available: http://epubs.siam.org/doi/pdf/10.1137/040616024

[36] Y. Ma, J. Zhu, and D. Baron, "Approximate message passing with universal denoising and Gaussian mixture learning," *IEEE Trans. Image Process.*, vol. 64, no. 21, pp. 5611–5622, Nov. 2016.

[37] K.-M. Ng, "A continuation approach for solving nonlinear optimization problems with discrete variables," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2002.

[38] Z. T. Harmany, R. F. Marcia, and R. M. Willet, "This is SPIRAL-TAP: Sparse Poisson intensity reconstruction algorithms: Theory and practice," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1084–1096, Sep. 2011.

[39] Y. Wang, Y. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248–272, Aug. 2008.

[40] T. Goldstein, B. O. Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," , Univ. California, Los Angeles, Los Angeles, CA, USA, *Tech. Rep.*, 2012. [Online]. Available: ftp://ftp.math.ucla.edu/pub/camreport/cam12-35.pdf

[41] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1299–1308, Sep. 2001.

[42] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.

[43] J.-J. Moreau, "Proximité et dualité dans un espace Hibertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[44] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. New York, NY, USA: Springer-Verlag, 1990.

[45] L. Liu, S. H. Chan, and T. Q. Nguyen, "Depth reconstruction from sparse samples: Representation, algorithm, and sampling," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1983–1996, Jun. 2015.

[46] M. S. C. Almeida and M. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3074–3086, Aug. 2013.

[47] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.

[48] A. Matakos, S. Ramani, and J. A. Fessler, "Accelerated edge-preserving image restoration without boundary artifacts," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 2019–2029, Jan. 2013.

[49] E. R. Fossum, "The quanta image sensor (QIS): Concepts and challenges," in *Proc. OSA Top. Meeting Comput. Opt. Sens. Imag.*, Jul. 2011, Paper JTuE1. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=ISA-2011-JTuE1

[50] S. H. Chan and Y. M. Lu, "Efficient image reconstruction for gigapixel quantum image sensors," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2014, pp. 312–316.

[51] O. A. Elgendy and S. H. Chan, "Image reconstruction and threshold design for quanta image sensors," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 978–982.

[52] J. Huang, A. Singh, and N. Ahuia, "Single image super-resolution using transformed self-exemplars," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2015, pp. 5197–5206.

[53] F. Yang, Y. M. Lu, L. Sbaiz, and M. Vetterli, "Bits from photons: Oversampled image acquisition using binary poisson statistics," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1421–1436, Apr. 2012.

[54] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–3145, Dec. 2010.

[55] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Ann. Math. Statist.*, vol. 35, pp. 876–879, 1964.
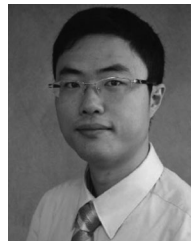
Stanley H. Chan (S'06–M'12) received the B.Eng. degree (first class honors) in electrical engineering from the University of Hong Kong, Hong Kong, in 2007, and the M.A. degree in mathematics and the Ph.D. degree in electrical engineering from the University of California at San Diego, CA, USA, in 2009 and 2011, respectively.

He is currently an Assistant Professor with the School of Electrical and Computer Engineering and the Department of Statistics, Purdue University, West Lafayette, IN, USA. Prior to joining Purdue University, he was a Postdoctoral Research Fellow with Harvard John A. Paulson School of Engineering and Applied Sciences from 2012 to 2014. His research interests include signal and image processing, applied statistics, and large-scale numerical optimization.

Dr. Chan received the Best Paper Award of the 2016 IEEE International Conference on Image Processing, the IEEE Signal Processing Cup 2016 Second Prize (as a faculty mentor), the Purdue COE Outstanding Graduate Mentor Award 2016, and the Eta Kappa Nu (Beta Chapter) Outstanding Teaching Award 2015. He also received the Croucher Foundation Fellowship for Postdoctoral Research 2012–2013 and the Croucher Foundation Scholarship for Ph.D. Studies 2008–2010.
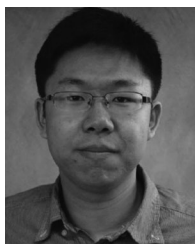
Xiran Wang (S'12) received the B.Sc. degree (*cum laude*) in electrical engineering from Rice University, Houston, TX, USA, in 2013. He is currently working toward the Ph.D. degree with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His research interests include image and video restoration, large-scale optimization, and computational photography.

Mr. Wang received the Ross fellowship of Purdue University in 2013.

Omar A. Elgendy (S'15) received the B.Sc. degree in electronics and communications engineering (distinction with honors) in 2010 and the M.Sc. degree in engineering mathematics in 2015 from the Faculty of Engineering, Cairo University, Giza, Egypt. He is currently working toward the Ph.D. degree with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His research interests include statistical signal processing, single-photon imaging, pattern classification, and large-scale optimization.

Mr. Elgendy received the Best Paper Award of the 2016 IEEE International Conference on Image Processing.