Rizwan Ahmad, Charles A. Bouman, Gregery T. Buzzard, Stanley Chan, Sizhuo Liu, Edward T. Reehorst, and Philip Schniter

Plug-and-Play Methods for Magnetic Resonance Imaging

Using denoisers for image recovery

©ISTOCKPHOTO.COM/GOODLIFESTUDIO

Digital Object Identifier 10.1109/MSP.2019.2949470
Date of current version: 15 January 2020

Magnetic resonance imaging (MRI) is a noninvasive diagnostic tool that provides excellent soft-tissue contrast without the use of ionizing radiation. Compared to other clinical imaging modalities (e.g., computed tomography or ultrasound), however, the data acquisition process for MRI is inherently slow, which motivates undersampling; thus, there is a need for accurate, efficient reconstruction methods from undersampled data sets. In this article, we describe the use of plug-and-play (PnP) algorithms for MRI image recovery. We first describe the linearly approximated inverse problem encountered in MRI. Then, we review several PnP methods for which the unifying commonality is to iteratively call a denoising subroutine as one step of a larger optimization-inspired algorithm. Next, we describe how the result of the PnP method can be interpreted as a solution to an equilibrium equation, allowing convergence analysis from this perspective. Finally, we present illustrative examples of PnP methods applied to MRI image recovery.

Introduction

MRI uses radio-frequency (RF) waves to noninvasively evaluate the structure, function, and morphology of soft tissues. It has become an indispensable imaging tool for diagnosing and evaluating a host of conditions and diseases. However, MRI suffers from slow data acquisition: a typical clinical MRI examination consists of multiple scans and can take more than an hour to complete. For each scan, the patient may be asked to stay still for several minutes, with slight motion potentially resulting in image artifacts. Furthermore, dynamic applications demand collecting a series of images in quick succession. Due to the limited time window in many dynamic applications (e.g., contrast-enhanced MR angiography), it is not feasible to collect fully sampled data sets. For these reasons, MRI data are often undersampled. Consequently, computationally efficient methods for recovering high-quality images from undersampled MRI data have been actively researched for the last two decades.

The combination of parallel (i.e., multicoil) imaging and compressive sensing (CS) has been shown to benefit a wide range of MRI applications [1], including dynamic applications, and has been included in the default image-processing frameworks offered by several major MRI vendors. More recently, learning-based techniques (e.g., [2]–[6]) have been shown to outperform CS methods. Some of these techniques learn the entire end-to-end mapping from undersampled *k*-space or aliased images to recovered images (e.g., [4]). Considering that the forward model in MRI changes from one data set to the next, such methods must be either trained over a large and diverse data corpus or limited to a specific application. Other methods train scan-specific convolutional neural networks (CNNs) on a fully sampled region of *k*-space and then use it to interpolate missing *k*-space samples [5]. These methods do not require separate training data but demand a fully sampled *k*-space region. Due to the large number of unknowns in CNNs, such methods require a fully sampled region that is larger than that typically acquired in parallel imaging, limiting the acceleration that can be achieved.

Other supervised learning methods are inspired by classic variational optimization methods and iterate between data-consistency enforcement and a trained CNN, which acts as a regularizer [3]. Such methods require a large number of fully sampled, multicoil *k*-space data sets, which may be difficult to obtain in many applications. Also, because CNN training occurs in the presence of data-set-specific forward models, generalization from training to test scenarios remains an open question [6]. Consequently, the integration of learning-based methods into physical inverse problems remains a fertile area of research. There are many directions for improvement, including recovery fidelity, computational and memory efficiency, robustness, interpretability, and ease of use.

This article focuses on PnP algorithms [7], which alternate image denoising with forward-model-based signal recovery. They facilitate the use of state-of-the-art image models through their manifestations as image denoisers, whether patch based (e.g., [8]) or deep neural network (DNN) based (e.g., [9]). The fact that PnP algorithms decouple image modeling from forward modeling has advantages in compressive MRI, where the forward model can change significantly among different scans due to variations in the coil sensitivity maps, sampling patterns, and image resolution. Furthermore, fully sampled *k*-space MRI data are not needed for PnP; the image denoiser can be learned from MRI image patches or, possibly, even magnitude-only patches. The objective of this article is twofold: 1) to review recent advances in PnP methods and 2) to discuss their application to compressive MRI image reconstruction. For an extended version of this article that contains additional references and more in-depth discussions on a variety of topics, see [10].

Image recovery in compressive MRI

In this section, we describe the standard linear inverse problem formulation in MRI. We acknowledge that more sophisticated formulations exist (see, e.g., [31] in this issue for a more careful modeling of physics effects). Briefly, the measurements are samples of the Fourier transform of the image, where the Fourier domain is often referred to as *k-space*. The transform can be taken across two or three spatial dimensions and includes an additional temporal dimension in dynamic applications. Furthermore, measurements are often collected in parallel from $C \geq 1$ receiver coils. In dynamic parallel MRI with Cartesian sampling, the *k*-space measurements from the *i*th coil at time *t* take the form

$$\boldsymbol{y}_i^{(t)} = \boldsymbol{P}^{(t)}\boldsymbol{F}\boldsymbol{S}_i\boldsymbol{x}^{(t)} + \boldsymbol{w}_i^{(t)}, \tag{1}$$

where $\boldsymbol{x}^{(t)} \in \mathbb{C}^N$ is the vectorized 2D or 3D image at discrete time t, $\boldsymbol{S}_i \in \mathbb{C}^{N\times N}$ is a diagonal matrix containing the sensitivity map for the ith coil, $\boldsymbol{F} \in \mathbb{C}^{N\times N}$ is the 2D or 3D discrete Fourier transform, the sampling matrix $\boldsymbol{P}^{(t)} \in \mathbb{R}^{M\times N}$ contains M rows of the $N \times N$ identity matrix, and $\boldsymbol{w}_i^{(t)} \in \mathbb{C}^M$ is additive white Gaussian noise (AWGN). Often, the sampling pattern changes across frames t. The MRI literature often refers to $R \triangleq N/M$ as the *acceleration rate*. The AWGN assumption, which does not hold for the measured parallel MRI data, is commonly enforced by using noise prewhitening filters, which yields the model (1) but with diagonal "virtual" coil maps $\boldsymbol{S}_i$ [11].

MRI measurements are acquired using a sequence of measurement trajectories through *k*-space that can be Cartesian or non-Cartesian in nature. Cartesian trajectories are, essentially, lines through *k*-space. In the Cartesian case, one *k*-space dimension (i.e., the frequency encoding) is fully sampled, while the other one or two dimensions (i.e., the phase encodings) are undersampled to reduce acquisition time. Typically, one line, or *readout*, is collected after every RF pulse, and the process is repeated several times to collect adequate samples of *k*-space. Non-Cartesian trajectories include radial or spiral curves, which have the effect of distributing the samples among all dimensions of *k*-space. Compared to Cartesian sampling, non-Cartesian sampling provides more efficient coverage of *k*-space and yields an "incoherent" forward operator that is more conducive to compressed-sensing reconstruction. However, Cartesian sampling remains the method of choice in clinical practice, due to its higher tolerance to system imperfections and an extensive record of success.

Because the sensitivity map, $\boldsymbol{S}_i$, is patient specific and varies with the location of the coil with respect to the imaging plane, both $\boldsymbol{S}_i$ and $\boldsymbol{x}^{(t)}$ are unknown in practice. Although calibration-free methods have been proposed to estimate $\boldsymbol{S}_i\boldsymbol{x}^{(t)}$ or to jointly estimate $\boldsymbol{S}_i$ and $\boldsymbol{x}^{(t)}$, it is more common to first estimate $\boldsymbol{S}_i$ through a calibration procedure and then treat $\boldsymbol{S}_i$ as known in (1). Stacking $\{\boldsymbol{y}_i^{(t)}\}$, $\{\boldsymbol{x}^{(t)}\}$, and $\{\boldsymbol{w}_i^{(t)}\}$ into vectors $\boldsymbol{y}, \boldsymbol{x}$, and $\boldsymbol{w}$, and packing $\{\boldsymbol{P}^{(t)}\boldsymbol{F}\boldsymbol{S}_i\}$ into a known block-diagonal matrix $\boldsymbol{A}$, we obtain the linear inverse problem of recovering $\boldsymbol{x}$ from

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}, \quad \boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}), \tag{2}$$

where $\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})$ denotes a circularly symmetric, complex-Gaussian random vector with mean $\mathbf{0}$ and covariance $\sigma^2\,\boldsymbol{I}$.

Signal recovery and denoising

The maximum likelihood (ML) estimate of $\boldsymbol{x}$ from $\boldsymbol{y}$ in (2) is $\widehat{\boldsymbol{x}}_{\text{ml}} \triangleq \operatorname{argmax}_x p(\boldsymbol{y}\,|\,\boldsymbol{x})$, where $p(\boldsymbol{y}\,|\,\boldsymbol{x})$, the probability density of $\boldsymbol{y}$ conditioned on $\boldsymbol{x}$, is known as the *likelihood function*. The ML estimate is often written in the equivalent form $\widehat{\boldsymbol{x}}_{\text{ml}} = \operatorname{argmin}_x\{-\ln p(\boldsymbol{y}\,|\,\boldsymbol{x})\}$. In the case of σ^2-variance AWGN $\boldsymbol{w}$, we have $-\ln p(\boldsymbol{y}\,|\,\boldsymbol{x}) = (1/2\sigma^2)\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \text{const}$ and, therefore, $\widehat{\boldsymbol{x}}_{\text{ml}} = \operatorname{argmin}_x \|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2$, which can be recognized as least-squares estimation. Although least-squares estimation can give reasonable performance when $\boldsymbol{A}$ is tall and well conditioned, this is rarely the case under moderate to high acceleration (i.e., $R>2$). With acceleration, it is critically important to exploit prior knowledge of signal structure.

The traditional approach to exploiting such prior knowledge is to formulate and solve an optimization problem of the form

$$\widehat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \phi(\boldsymbol{x})\right\}, \tag{3}$$

where the regularization term $\phi(\boldsymbol{x})$ encodes prior knowledge of $\boldsymbol{x}$. In fact, $\widehat{\boldsymbol{x}}$ in (3) can be recognized as the maximum a posteriori (MAP) estimate of $\boldsymbol{x}$ under the prior density model $p(\boldsymbol{x}) \propto \exp(-\phi(\boldsymbol{x}))$. To see why, recall that the MAP estimate maximizes the posterior distribution $p(\boldsymbol{x}\,|\,\boldsymbol{y})$. That is, $\widehat{\boldsymbol{x}}_{\text{map}} \triangleq \operatorname{argmax}_x p(\boldsymbol{x}\mid\boldsymbol{y}) = \operatorname{argmin}_x\{-\ln p(\boldsymbol{x}\mid\boldsymbol{y})\}$. Because Bayes' rule implies that $\ln p(\boldsymbol{x}\mid\boldsymbol{y}) = \ln p(\boldsymbol{y}\mid\boldsymbol{x}) + \ln p(\boldsymbol{x}) - \ln p(\boldsymbol{y})$, we have

$$\widehat{\boldsymbol{x}}_{\text{map}} = \operatorname*{argmin}_{\boldsymbol{x}}\{-\ln p(\boldsymbol{y}\,|\,\boldsymbol{x}) - \ln p(\boldsymbol{x})\}. \tag{4}$$

Recalling that the first term in (3) (i.e., the "loss" term) was observed to be $-\ln p(\boldsymbol{y}\,|\,\boldsymbol{x})$ (plus a constant) under AWGN noise, the second term in (3) must obey $\phi(\boldsymbol{x}) = -\ln p(\boldsymbol{x}) + \text{const}$. We will find this MAP interpretation useful in the sequel.

It is not easy to design good regularizers ϕ for use in (3). They must not only mimic the negative log of the prior density but also enable tractable optimization. One common approach is to use $\phi(\boldsymbol{x}) = \lambda\|\boldsymbol{\Psi}\boldsymbol{x}\|_1$, where $\boldsymbol{\Psi}^{\mathsf{H}}$ is a tight frame (e.g., a wavelet transform) and $\lambda>0$ is a tuning parameter [12]. Such regularizers are convex, and the ℓ_1 norm rewards sparsity in the transform outputs $\boldsymbol{\Psi}\boldsymbol{x}$ when used with the quadratic loss.

Particular insight comes from considering the special case of $\boldsymbol{A}=\boldsymbol{I}$, where the measurement vector in (2) reduces to an AWGN-corrupted version of the image $\boldsymbol{x}$:

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{w}, \quad \boldsymbol{w} \sim \mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I}). \tag{5}$$

The problem of recovering $\boldsymbol{x}$ from noisy $\boldsymbol{z}$, known as *denoising*, has been intensely researched for decades. Although it is possible to perform denoising by solving a regularized optimization problem of the form (3) with $\boldsymbol{A}=\boldsymbol{I}$, today's state-of-the-art approaches are either algorithmic (e.g., [8]) or DNN based (e.g., [9]). This begs an important question: can these state-of-the-art denoisers be leveraged for MRI signal reconstruction by exploiting the connections between the denoising problem and (3)? As we shall see, this is precisely what the PnP methods do.

PnP methods

In this section, we review several approaches to PnP signal reconstruction. What these approaches have in common is that they recover $\boldsymbol{x}$ from measurements $\boldsymbol{y}$ of the form (2) by iteratively calling a sophisticated denoiser within a larger optimization or inference algorithm.

Prox-based PnP

To start, let us imagine how the optimization in (3) might be solved. Through what is known as *variable splitting*, we could introduce a new variable, $\boldsymbol{v}$, to decouple the regularizer $\phi(\boldsymbol{x})$ from the data fidelity term $(1/2\sigma^2)\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2$. The variables $\boldsymbol{x}$ and $\boldsymbol{v}$ could then be equated using an external constraint, leading to the constrained minimization problem

$$\widehat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{C}^N}\min_{\boldsymbol{v}\in\mathbb{C}^N}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \phi(\boldsymbol{v})\right\} \quad \text{subject to} \quad \boldsymbol{x}=\boldsymbol{v}. \tag{6}$$

Equation (6) suggests an algorithmic solution that alternates between separately estimating $\boldsymbol{x}$ and estimating $\boldsymbol{v}$, with an additional mechanism to asymptotically enforce the constraint $\boldsymbol{x}=\boldsymbol{v}$.

The original PnP method [7] is based on the alternating direction method of multipliers (ADMM) [13]. For ADMM, (6) is first reformulated as the "augmented Lagrangian":

$$\min_{\boldsymbol{x},\boldsymbol{v}}\max_{\boldsymbol{\lambda}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \phi(\boldsymbol{v}) + \operatorname{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}(\boldsymbol{x}-\boldsymbol{v})\} + \frac{1}{2\eta}\|\boldsymbol{x}-\boldsymbol{v}\|^2\right\}, \tag{7}$$

where $\boldsymbol{\lambda}$ are Lagrange multipliers and $\eta>0$ is a penalty parameter that affects the convergence speed of the algorithm but not the final solution. With $\boldsymbol{u} \triangleq \eta\boldsymbol{\lambda}$, (7) can be rewritten as

$$\min_{\boldsymbol{x},\boldsymbol{v}}\max_{\boldsymbol{u}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \phi(\boldsymbol{v}) + \frac{1}{2\eta}\|\boldsymbol{x}-\boldsymbol{v}+\boldsymbol{u}\|^2 - \frac{1}{2\eta}\|\boldsymbol{u}\|^2\right\}. \tag{8}$$

ADMM solves (8) by alternating the optimization of $\boldsymbol{x}$ and $\boldsymbol{v}$ with gradient ascent of $\boldsymbol{u}$; i.e.,

$$\boldsymbol{x}_k = \boldsymbol{h}(\boldsymbol{v}_{k-1} - \boldsymbol{u}_{k-1};\eta) \tag{9a}$$
$$\boldsymbol{v}_k = \operatorname{prox}_{\phi}(\boldsymbol{x}_k + \boldsymbol{u}_{k-1};\eta) \tag{9b}$$
$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + (\boldsymbol{x}_k - \boldsymbol{v}_k), \tag{9c}$$

where $\boldsymbol{h}(z;\eta)$ and $\operatorname{prox}_{\phi}(z;\eta)$, known as *proximal maps*, are defined as

$$\text{prox}_{\phi}(z;\eta) \triangleq \underset{x}{\text{argmin}}\left\{\phi(\boldsymbol{x}) + \frac{1}{2\eta}\|\boldsymbol{x}-z\|^2\right\} \tag{10}$$

$$\begin{aligned}\boldsymbol{h}(z;\eta) &\triangleq \underset{x}{\text{argmin}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 + \frac{1}{2\eta}\|\boldsymbol{x}-z\|^2\right\} \\ &= \text{prox}_{\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2/(2\sigma^2)}(z;\eta) \qquad (11)\\ &= \left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} + \frac{\sigma^2}{\eta}\boldsymbol{I}\right)^{-1}\left(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{y} + \frac{\sigma^2}{\eta}z\right). \qquad (12)\end{aligned}$$

Under some weak technical constraints, it can be proven [13] that when ϕ is convex, the ADMM iteration (9) converges to $\widehat{\boldsymbol{x}}$, the global minimum of (3) and (6).

From the discussion in the "Signal Recovery and Denoising" section, we immediately recognize $\text{prox}_{\phi}(z;\eta)$ in (10) as the MAP denoiser of z under AWGN variance η and signal prior $p(\boldsymbol{x}) \propto \exp(-\phi(\boldsymbol{x}))$. The key idea behind the original PnP work [7] was, in the ADMM recursion (9), to plug in a powerful image denoising algorithm, such as block-matching and 3D filtering (BM3D) [8], in place of the proximal denoiser $\text{prox}_{\phi}(\boldsymbol{x};\eta)$ from (10). If the plug-in denoiser is denoted by "$\boldsymbol{f}$," then the PnP ADMM algorithm becomes

$$\boldsymbol{x}_k = \boldsymbol{h}(\boldsymbol{v}_{k-1} - \boldsymbol{u}_{k-1};\eta) \tag{13a}$$

$$\boldsymbol{v}_k = \boldsymbol{f}(\boldsymbol{x}_k + \boldsymbol{u}_{k-1}) \tag{13b}$$

$$\boldsymbol{u}_k = \boldsymbol{u}_{k-1} + (\boldsymbol{x}_k - \boldsymbol{v}_k). \tag{13c}$$

A wide variety of empirical results (e.g., [7], [14], and [15]) have demonstrated that, when $\boldsymbol{f}$ is a powerful denoising algorithm, such as BM3D, the PnP algorithm (13) produces far better recoveries than the regularization-based approach (9). Although the value of η does not change the fixed point of the standard ADMM algorithm (9), it affects the fixed point of the PnP ADMM algorithm (13) through the ratio σ^2/η in (12).

The success of PnP methods raises important theoretical questions. Because $\boldsymbol{f}$ is not, in general, the proximal map of any regularizer ϕ, the iterations (13) may not minimize a cost function of the form in (3), and (13) may not be an implementation of ADMM. It is, then, unclear if the iterations (13) will converge, and, if they do converge, it is uncertain what they converge to. The consensus equilibrium (CE) framework, which we discuss in the "Understanding PnP Through CE" section, aims to provide answers to these questions.

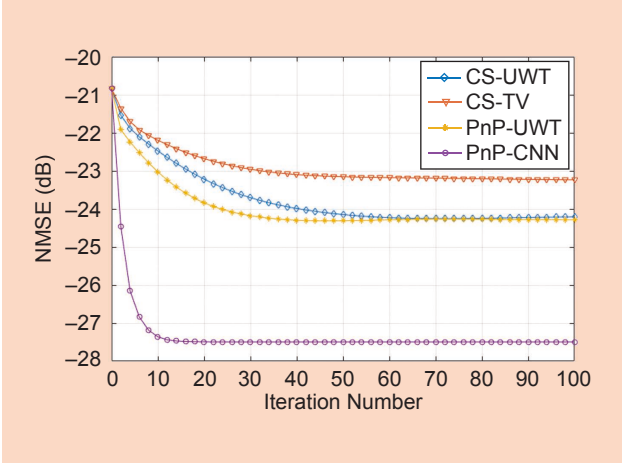

FIGURE 1. The normalized mean-squared error (NMSE) versus iteration for two PnP and two CS algorithms on the cardiac cine recovery data set 3 at $R = 10$. UWT: undecimated wavelet transform.

The use of a generic denoiser in place of a proximal denoiser can be translated to non-ADMM algorithms, such as the fast iterative shrinkage and thresholding algorithm (FISTA), primal–dual splitting (PDS), and others, as in [16]–[18]. Instead of optimizing $\boldsymbol{x}$ as in (13), PnP FISTA [16] uses the iterative update

$$\boldsymbol{z}_k = \boldsymbol{s}_{k-1} - \frac{\eta}{\sigma^2}\boldsymbol{A}^{\mathsf{H}}(\boldsymbol{A}\boldsymbol{s}_{k-1} - \boldsymbol{y}) \tag{14a}$$

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{z}_k) \tag{14b}$$

$$\boldsymbol{s}_k = \boldsymbol{x}_k + \frac{q_{k-1}-1}{q_k}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}), \tag{14c}$$

where (14a) is a gradient descent (GD) step on the negative log-likelihood $(1/2\sigma^2)\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2$ at $\boldsymbol{x} = \boldsymbol{s}_{k-1}$ with step-size $\eta \in (0, \sigma^2\|\boldsymbol{A}\|_2^{-2})$, (14b) is the plug-in replacement of the usual proximal denoising step in FISTA, and (14c) is an acceleration step, where it is typical to use $q_k = (1+\sqrt{1+4q_{k-1}^2})/2$ and $q_0 = 1$.

Comparing PnP ADMM (13) to PnP FISTA (3), one can see that they differ in how the data fidelity term $(1/2\sigma^2)\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2$ is handled: PnP ADMM uses the proximal update (12), whereas PnP FISTA and PnP PDS use the GD step (14a). In most cases, solving the proximal update (12) is much more computationally costly than taking a GD step (14a). Thus, with ADMM, it is common to approximate the proximal update (12) using, for example, several iterations of the conjugate gradient (CG) algorithm or GD, which should reduce the per-iteration complexity of (13) but may increase the number of iterations. However, even with these approximations of (12), PnP ADMM is usually close to "convergence" after 10–50 iterations (e.g., see Figure 1).

An important difference between the aforementioned flavors of PnP is that the step size η is constrained in FISTA but not in ADMM or PDS. Thus, PnP FISTA restricts the range of reachable fixed points relative to PnP ADMM and PnP PDS.

The balanced FISTA approach

In the "Signal Recovery and Denoising" section, when discussing the optimization problem (3), the regularizer $\phi(\boldsymbol{x}) = \lambda\|\boldsymbol{\Psi}\boldsymbol{x}\|_1$ was mentioned as a popular option, where $\boldsymbol{\Psi}$ is often a wavelet transform. The resulting optimization problem,

$$\widehat{\boldsymbol{x}} = \underset{x}{\text{argmin}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}x\|_2^2 + \lambda\|\boldsymbol{\Psi}\boldsymbol{x}\|_1\right\}, \tag{15}$$

is said to be stated in "analysis" form (see [19] in this issue). The proximal denoiser associated with (15) has the form

$$\text{prox}_{\phi}(z;\eta) = \underset{x}{\text{argmin}}\left\{\lambda\|\boldsymbol{\Psi}\boldsymbol{x}\|_1 + \frac{1}{2\eta}\|\boldsymbol{x}-z\|^2\right\}. \quad (16)$$

When $\boldsymbol{\Psi}$ is orthogonal, it is well known that $\text{prox}_{\phi}(z;\eta) = \boldsymbol{f}_{\text{tdt}}(z;\lambda\eta)$, where

$$\boldsymbol{f}_{\text{tdt}}(z;\tau) \triangleq \boldsymbol{\Psi}^{\mathsf{H}}\text{soft-thresh}(\boldsymbol{\Psi}z;\tau) \quad (17)$$

is the "transform-domain thresholding" denoiser with $[\text{soft-thresh}(\boldsymbol{u},\tau)]_n \triangleq \max\{0,(|u_n|-\tau)/|u_n|\}u_n$. The denoiser (17) is very efficient to implement because it amounts to little more than computing forward and reverse transforms.

In practice, (15) yields much better results with nonorthogonal $\boldsymbol{\Psi}$, such as when $\boldsymbol{\Psi}^{\mathsf{H}}$ is a tight frame (see, e.g., the references in [20]). In the latter case, $\boldsymbol{\Psi}^{\mathsf{H}}\boldsymbol{\Psi} = \boldsymbol{I}$ with tall $\boldsymbol{\Psi}$. However, for general tight frames $\boldsymbol{\Psi}^{\mathsf{H}}$, the proximal denoiser (16) has no closed-form solution. What if we simply plugged the transform-domain thresholding denoiser (17) into an algorithm, such as ADMM or FISTA? How can we interpret the resulting approach? Interestingly, as we will describe, if (17) is used in PnP FISTA, then it does solve a convex optimization problem, although one with a different form than (3). This approach was independently proposed in [12] and [20]; in the latter, it was referred to as *balanced FISTA* (*bFISTA*) and applied to parallel cardiac MRI. Notably, bFISTA was proposed before the advent of PnP FISTA. More details are provided later in the article.

The optimization problem (15) can be stated in constrained "synthesis" form as

$$\widehat{\boldsymbol{x}} = \boldsymbol{\Psi}^{\mathsf{H}}\widehat{\boldsymbol{\alpha}} \text{ for } \widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}\in\text{range}(\boldsymbol{\Psi})}{\text{argmin}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\Psi}^{\mathsf{H}}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1\right\}, \quad (18)$$

where $\boldsymbol{\alpha}$ are transform coefficients. Then, as $\beta\to\infty$, (18) can be expressed in the unconstrained form

$$\widehat{\boldsymbol{x}} = \boldsymbol{\Psi}^{\mathsf{H}}\widehat{\boldsymbol{\alpha}} \text{ for}$$
$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\text{argmin}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{\Psi}^{\mathsf{H}}\boldsymbol{\alpha}\|_2^2 + \frac{\beta}{2}\|P_{\boldsymbol{\Psi}}^{\perp}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1\right\}, \quad (19)$$

with projection matrix $\boldsymbol{P}_{\boldsymbol{\Psi}}^{\perp} \triangleq \boldsymbol{I} - \boldsymbol{\Psi}\boldsymbol{\Psi}^{\mathsf{H}}$. In practice, it is not possible to take $\beta\to\infty$ and, for finite values of β, the problems (18) and (19) are not equivalent. However, (19) under finite β is interesting to consider in its own right, and it is sometimes referred to as the *balanced approach*. If we solve (19) using FISTA with step size $\eta > 0$ [recall (14a)] and choose the particular value $\beta = 1/\eta$, then, remarkably, the resulting algorithm takes the form of PnP FISTA (14) with $\boldsymbol{f}(z) = \boldsymbol{f}_{\text{tdt}}(z;\lambda)$. This particular choice of β is motivated by computational efficiency (because it leads to the use of $\boldsymbol{f}_{\text{tdt}}$) rather than recovery performance. Still, as we demonstrate in the "Demonstration of PnP in MRI" section, it performs relatively well.

Regularization by denoising

Another PnP approach, proposed by Romano, Elad, and Milanfar in [21], recovers $\boldsymbol{x}$ from measurements $\boldsymbol{y}$ in (2) by finding the $\widehat{\boldsymbol{x}}$ that solves the optimality condition

$$\boldsymbol{0} = \frac{1}{\sigma^2}\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\widehat{\boldsymbol{x}}-\boldsymbol{y}) + \frac{1}{\eta}(\widehat{\boldsymbol{x}} - \boldsymbol{f}(\widehat{\boldsymbol{x}})), \quad (20)$$

where $\boldsymbol{f}$ is an arbitrary (i.e., "plug in") image denoiser and $\eta > 0$ is a tuning parameter. In [21], several algorithms were proposed to solve (20). Numerical experiments in [21] suggest that, when $\boldsymbol{f}$ is a sophisticated denoiser (such as BM3D) and η is well tuned, the solutions $\widehat{\boldsymbol{x}}$ to (20) are state of the art, similar to those of PnP ADMM. As in [21] and [22], we first focus on the real-valued case, but later, we consider the complex-valued case of interest in MRI.

The approach in (20) was termed *regularization by denoising (RED)* in [21] because, under certain conditions, the $\widehat{\boldsymbol{x}}$ that solve (20) are the solutions to the regularized least-squares problem

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\text{argmin}}\left\{\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 + \phi_{\text{red}}(\boldsymbol{x})\right\} \quad (21a)$$

with

$$\phi_{\text{red}}(\boldsymbol{x}) \triangleq \frac{1}{2\eta}\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{x}-\boldsymbol{f}(\boldsymbol{x})), \quad (21b)$$

where the regularizer ϕ_{red} is explicitly constructed from the plug-in denoiser $\boldsymbol{f}$. What are these conditions? Assuming that $\boldsymbol{f}$ is differentiable almost everywhere, it was shown in [22] that the solutions of (20) correspond to those of (21) when 1) $\boldsymbol{f}$ is locally homogeneous, which means that $(1+\epsilon)\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}((1+\epsilon)\boldsymbol{x})$ for all $\boldsymbol{x}$ and sufficiently small nonzero ϵ, and 2) $\boldsymbol{f}$ has a symmetric Jacobian matrix (i.e., $[\boldsymbol{J}\boldsymbol{f}(\boldsymbol{x})]^{\mathsf{T}} = \boldsymbol{J}\boldsymbol{f}(\boldsymbol{x})\ \forall\boldsymbol{x}$). However, it was demonstrated in [22] that these properties are not satisfied by popular image denoisers, such as the median filter, transform-domain thresholding, nonlocal means, BM3D, trainable nonlinear reaction diffusion, and denoising convolutional neural network (DnCNN). Furthermore, it was proven in [22] that if the Jacobian of $\boldsymbol{f}$ is nonsymmetric, then there does not exist any regularizer ϕ under which the solutions of (20) minimize a regularized loss of the form in (3).

One may then wonder how to justify (20). In [22], Reehorst and Schniter proposed an explanation for (20) based on "score matching," which we now summarize. Suppose we are given a large corpus of training images $\{\boldsymbol{x}_t\}_{t=1}^{T}$, from which we could build the empirical prior model

$$\widehat{p}_{\text{X}}(\boldsymbol{x}) \triangleq \frac{1}{T}\sum_{t=1}^{T}\delta(\boldsymbol{x}-\boldsymbol{x}_t),$$

where δ denotes the Dirac delta. Because images are known to exist outside $\{\boldsymbol{x}_t\}_{t=1}^{T}$, it is possible to build an improved prior model $\tilde{p}_{\text{X}}$ using kernel density estimation (KDE), that is,

$$\tilde{p}_{\mathsf{x}}(\boldsymbol{x};\eta) \triangleq \frac{1}{T}\sum_{t=1}^{T} \mathcal{N}(\boldsymbol{x};\boldsymbol{x}_t,\eta\boldsymbol{I}), \tag{22}$$

where $\eta > 0$ is a tuning parameter. If we adopt $\tilde{p}_{\mathsf{x}}$ as the prior model for $\boldsymbol{x}$, then the MAP estimate of $\boldsymbol{x}$ [recall (4)] becomes

$$\widehat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 - \ln \tilde{p}_{\mathsf{x}}(\boldsymbol{x};\eta) \right\}. \tag{23}$$

Because $\ln \tilde{p}_{\mathsf{x}}$ is differentiable, the solutions to (23) must obey

$$\boldsymbol{0} = \frac{1}{\sigma^2}\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\widehat{\boldsymbol{x}}-\boldsymbol{y}) - \nabla \ln \tilde{p}_{\mathsf{x}}(\widehat{\boldsymbol{x}};\eta). \tag{24}$$

A classical result known as *Tweedie's formula* says that

$$\nabla \ln \tilde{p}_{\mathsf{x}}(\boldsymbol{z};\eta) = \frac{1}{\eta}(\boldsymbol{f}_{\text{mmse}}(\boldsymbol{z};\eta)-\boldsymbol{z}), \tag{25}$$

where $\boldsymbol{f}_{\text{mmse}}(\cdot;\eta)$ is the minimum mean-squared error (MMSE) denoiser under the prior $\boldsymbol{x} \sim \widehat{p}_{\mathsf{x}}$ and η-variance AWGN. That is, $\boldsymbol{f}_{\text{mmse}}(\boldsymbol{z}) = \mathrm{E}\{\boldsymbol{x}|\boldsymbol{z}\}$, where $\boldsymbol{z} = \boldsymbol{x} + \mathcal{N}(\boldsymbol{0},\eta\boldsymbol{I})$ and $\boldsymbol{x} \sim \widehat{p}_{\mathsf{x}}$. Applying (25) to (24), the MAP estimate $\widehat{\boldsymbol{x}}$ under the KDE prior $\tilde{p}_{\mathsf{x}}$ obeys

$$\boldsymbol{0} = \frac{1}{\sigma^2}\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\widehat{\boldsymbol{x}}-\boldsymbol{y}) + \frac{1}{\eta}(\widehat{\boldsymbol{x}} - \boldsymbol{f}_{\text{mmse}}(\widehat{\boldsymbol{x}};\eta)), \tag{26}$$

which matches the RED condition (20) when $\boldsymbol{f} = \boldsymbol{f}_{\text{mmse}}(\cdot;\eta)$. Thus, if we could implement the MMSE denoiser $\boldsymbol{f}_{\text{mmse}}$ for a given training corpus $\{\boldsymbol{x}_t\}_{t=1}^{T}$, then RED provides a way to compute the MAP estimate of $\boldsymbol{x}$ under the KDE prior $\tilde{p}_{\mathsf{x}}$.

Although the MMSE denoiser $\boldsymbol{f}_{\text{mmse}}$ can be expressed in closed form (see [22, eq. 67]), it is not practical to implement for large T. Thus, the question remains: can the RED approach (20) also be justified for non-MMSE denoisers $\boldsymbol{f}$, especially those that are not locally homogeneous or Jacobian symmetric? As shown in [22], the answer is yes. Consider a practical denoiser $\boldsymbol{f}_{\boldsymbol{\theta}}$ parameterized by tunable weights $\boldsymbol{\theta}$ (e.g., a DNN). A typical strategy is to choose $\boldsymbol{\theta}$ to minimize the MSE on $\{\boldsymbol{x}_t\}_{t=1}^{T}$, that is, to set $\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathrm{E}\{\|\boldsymbol{x}-\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2\}$, where the expectation is taken over $\boldsymbol{x} \sim \widehat{p}_{\mathsf{x}}$ and $\boldsymbol{z} = \boldsymbol{x} + \mathcal{N}(\boldsymbol{0},\eta\boldsymbol{I})$. By the MMSE orthogonality principle, we have

$$\begin{aligned}\mathrm{E}\{\|\boldsymbol{x}-\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2\} &= \mathrm{E}\{\|\boldsymbol{x}-\boldsymbol{f}_{\text{mmse}}(\boldsymbol{z};\eta)\|^2\} \\ &\quad + \mathrm{E}\{\|\boldsymbol{f}_{\text{mmse}}(\boldsymbol{z};\eta)-\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2\},\end{aligned} \tag{27}$$

and so we can write

$$\widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathrm{E}\{\|\boldsymbol{f}_{\text{mmse}}(\boldsymbol{z};\eta)-\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2\} \tag{28}$$

$$= \operatorname*{argmin}_{\boldsymbol{\theta}} \mathrm{E}\left\{\left\| \nabla \ln \tilde{p}_{\mathsf{x}}(\boldsymbol{z};\eta) - \frac{1}{\eta}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})-\boldsymbol{z})\right\|^2\right\}, \tag{29}$$

where (29) follows from (25). Equation (29) says that choosing $\boldsymbol{\theta}$ to minimize the MSE is equivalent to choosing $\boldsymbol{\theta}$ so that $(1/\eta)(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{z})-\boldsymbol{z})$ best matches the "score" $\nabla \ln \tilde{p}_{\mathsf{x}}(\boldsymbol{z};\eta)$.

In summary, the RED approach (20) approximates the KDE-MAP approach (24)–(26) by using a plug-in denoiser $\boldsymbol{f}$ to approximate the MMSE denoiser $\boldsymbol{f}_{\text{mmse}}$. When $\boldsymbol{f} = \boldsymbol{f}_{\text{mmse}}$, RED exactly implements MAP-KDE, but with a practical $\boldsymbol{f}$, RED implements a score-matching approximation of MAP-KDE. Thus, a more appropriate title for RED might be "score matching by denoising."

Comparing the RED approach from this section to the prox-based PnP approach from the "Prox-Based PnP" section, we see that RED starts with the KDE-based MAP estimation problem (23) and replaces the $\tilde{p}_{\mathsf{x}}$-based MMSE denoiser $\boldsymbol{f}_{\text{mmse}}$ with a plug-in denoiser $\boldsymbol{f}$, whereas PnP ADMM starts with the ϕ-based MAP estimation problem (3) and replaces the ϕ-based MAP denoiser prox_{ϕ} from (10) with a plug-in denoiser $\boldsymbol{f}$. It has recently been demonstrated that, when the prior is constructed from image examples, MAP recovery often leads to sharper, more natural-looking image recoveries than MMSE recovery [23]. Thus, it is interesting that RED offers an approach to MAP-based recovery using MMSE denoising, which is much easier to implement than MAP denoising [23].

Further insight into the difference between RED and prox-based PnP can be obtained by considering the case of symmetric linear denoisers, that is, $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{W}\boldsymbol{z}$ with $\boldsymbol{W} = \boldsymbol{W}^{\mathsf{T}}$, where we will also assume that $\boldsymbol{W}$ is invertible. Although such denoisers are far from state of the art, they are useful for interpretation. It is easy to show [24] that $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{W}\boldsymbol{z}$ is the proximal map of $\phi(\boldsymbol{x}) = (1/2\eta)\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{W}^{-1}-\boldsymbol{I})\boldsymbol{x}$, that is, that $\text{prox}_{\phi}(\boldsymbol{z};\eta) = \boldsymbol{W}\boldsymbol{z}$, recalling (10). With this proximal denoiser, we know that the prox-based PnP algorithm solves the optimization problem

$$\widehat{\boldsymbol{x}}_{\text{pnp}} = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 + \frac{1}{2\eta}\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{W}^{-1}-\boldsymbol{I})\boldsymbol{x} \right\}. \tag{30}$$

Meanwhile, because $\boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{W}\boldsymbol{z}$ is both locally homogeneous and Jacobian symmetric, we know from (21) that the RED under this $\boldsymbol{f}$ solves the optimization problem

$$\widehat{\boldsymbol{x}}_{\text{red}} = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2 + \frac{1}{2\eta}\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{I}-\boldsymbol{W})\boldsymbol{x} \right\}. \tag{31}$$

By comparing (30) and (31), we see a clear difference between RED and prox-based PnP. The "CE for RED" section compares RED to prox-based PnP from yet another perspective: CE.

So far, we have described RED as solving for $\widehat{\boldsymbol{x}}$ in (20), but how, exactly, is this accomplished? In the original RED article [21], three algorithms were proposed to solve (20): GD, inexact ADMM, and a "fixed-point" heuristic that was later recognized [22] as a special case of the proximal gradient (PG) algorithm. Generalizations of PG RED were proposed in [22]. The fastest among them is the accelerated-PG RED algorithm, which uses the iterative update

$$\boldsymbol{x}_k = \boldsymbol{h}(\boldsymbol{v}_{k-1};\eta/L) \tag{32a}$$

$$\boldsymbol{z}_k = \boldsymbol{x}_k + \frac{q_{k-1}-1}{q_k}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \tag{32b}$$

$$\boldsymbol{v}_k = \frac{1}{L}\boldsymbol{f}(\boldsymbol{z}_k) + \left(1-\frac{1}{L}\right)\boldsymbol{z}_k, \tag{32c}$$

where $\boldsymbol{h}$ was defined in (12), (32b) uses the same acceleration as PnP FISTA (14b), and $L > 0$ is a design parameter that can be related to the Lipschitz constant of $\phi_{\text{red}}(\cdot)$ from (21) (see Section V-C in [22]). The RED equations (32) and (33) may be used with complex-valued quantities. When $L = 1$ and $q_k = 1\ \forall k$, (32) reduces to the "fixed-point" heuristic from [21]. To reduce the implementation complexity of $\boldsymbol{h}$, one could replace (32a) with the GD step

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} - \frac{\eta}{L\sigma^2}\boldsymbol{A}^{\mathsf{H}}(\boldsymbol{A}\boldsymbol{v}_{k-1} - \boldsymbol{y}), \tag{33}$$

which achieves a similar complexity reduction as when going from PnP ADMM to PnP FISTA (as discussed in the "Prox-Based PnP" section). The result would be an "accelerated GD" form of RED. Convergence of the RED algorithms will be discussed in the "CE for RED" section.

Understanding PnP through CE

The success of the PnP methods described raises important theoretical questions. For example, in the case of PnP ADMM, if the plug-in denoiser $\boldsymbol{f}$ is not the proximal map of any regularizer ϕ, then it is not clear what cost function is being minimized (if any) or whether the algorithm will even converge. Similarly, in the case of RED, if the plug-in denoiser $\boldsymbol{f}$ is not the MMSE denoiser $\boldsymbol{f}_{\text{mmse}}$, then RED no longer solves the MAP-KDE problem, and it is not clear what RED does solve or whether a given RED algorithm will even converge. In this section, we show that many of these questions can be answered through the consensus equilibrium (CE) framework [18], [22], [24], [25]. We start by discussing CE for the PnP approaches from the "Prox-Based PnP" section and follow with a discussion of CE for the RED approaches from the "Regularization by Denoising" section.

CE for prox-based PnP

Let us start by considering the PnP ADMM algorithm (13). Rather than viewing (13) as minimizing some cost function, we can view it as seeking a solution, $(\widehat{\boldsymbol{x}}_{\text{pnp}}, \widehat{\boldsymbol{u}}_{\text{pnp}})$, to

$$\widehat{\boldsymbol{x}}_{\text{pnp}} = \boldsymbol{h}(\widehat{\boldsymbol{x}}_{\text{pnp}} - \widehat{\boldsymbol{u}}_{\text{pnp}}; \eta) \tag{34a}$$

$$\widehat{\boldsymbol{x}}_{\text{pnp}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{\text{pnp}} + \widehat{\boldsymbol{u}}_{\text{pnp}}), \tag{34b}$$

which, by inspection, must hold when (13) is at a fixed point. Not surprisingly, by setting $\boldsymbol{x}_k = \boldsymbol{x}_{k-1}$ in the PnP FISTA algorithm (14), it is straightforward to show that it, too, seeks a solution to (34). It is easy to show that the PnP PDS algorithm [17] seeks the same solution. With (34), the goal of the prox-based PnP algorithms becomes well defined! The pair (34) reaches a consensus in that the denoiser $\boldsymbol{f}$ and the data-fitting operator $\boldsymbol{h}$ agree on the output $\widehat{\boldsymbol{x}}_{\text{pnp}}$. The equilibrium comes from the opposing signs on the correction term $\widehat{\boldsymbol{u}}_{\text{pnp}}$: the data-fitting subtracts it, while the denoiser adds it.

By viewing the goal of prox-based PnP as solving the equilibrium problem (34), it becomes clear that other solvers beyond ADMM, FISTA, and PDS can be used. For example, it was shown in [25] that the PnP CE condition (34) can be achieved by finding a fixed point of the system

$$\underline{z} = (2\mathcal{G} - \boldsymbol{I})(2\mathcal{F} - \boldsymbol{I})\underline{z} \tag{35}$$

$$\underline{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad \mathcal{F}(\underline{z}) = \begin{bmatrix} \boldsymbol{h}(z_1; \eta) \\ \boldsymbol{f}(z_2) \end{bmatrix}, \quad \text{and} \quad \mathcal{G}(\underline{z}) = \begin{bmatrix} \frac{1}{2}(z_1 + z_2) \\ \frac{1}{2}(z_1 + z_2) \end{bmatrix}. \tag{36}$$

The paper [25] actually considers the CE among $N > 1$ agents, whereas here, we consider the simple case of $N = 2$ agents. There exist many algorithms to solve (35). For example, one could use the Mann iteration

$$\underline{z}^{(k+1)} = (1 - \gamma)\underline{z}^k + \gamma(2\mathcal{G} - \boldsymbol{I})(2\mathcal{F} - \boldsymbol{I})\underline{z}^{(k)}, \quad \text{with } \gamma \in (0, 1), \tag{37}$$

when $\mathcal{F}$ is nonexpansive. [25] also shows that this fixed point is equivalent to the solution of $\mathcal{F}(\underline{z}) = \mathcal{G}(\underline{z})$, in which case Newton's method or other root-finding methods could be applied.

The CE viewpoint also provides a path to proving the convergence of the PnP ADMM algorithm. Sreehari et al. [14] used a classical result from convex analysis to show that sufficient conditions for convergence are that 1) $\boldsymbol{f}$ is nonexpansive, that is, $\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y})\| \le \|\boldsymbol{x} - \boldsymbol{y}\|$ for any $\boldsymbol{x}$ and $\boldsymbol{y}$, and 2) $\boldsymbol{f}(\boldsymbol{x})$ is a subgradient of some convex function, that is, there exists φ such that $\boldsymbol{f}(\boldsymbol{x}) \in \partial\varphi(\boldsymbol{x})$. If these two conditions are met, then PnP ADMM (13) will converge to a global solution. Similar observations were made in other recent studies, such as [24]. That said, Chan et al. [15] showed that many practical denoisers do not satisfy these conditions, and so they designed a variant of PnP ADMM in which η is decreased at every iteration. Under appropriate conditions on $\boldsymbol{f}$ and the rate of decrease, this latter method also guarantees convergence, although not exactly to a fixed point of (34) because η is no longer fixed.

Similar techniques can be used to prove the convergence of other prox-based PnP algorithms. For example, under certain technical conditions, including nonexpansiveness of $\boldsymbol{f}$, it was established [18] that PnP FISTA converges to the same fixed point as PnP ADMM.

CE for RED

Just as the prox-based PnP algorithms can be viewed as seeking the CE of (34), it was shown in [22] that the proximal gradient and ADMM-based RED algorithms seek the CE $(\widehat{\boldsymbol{x}}_{\text{red}}, \widehat{\boldsymbol{u}}_{\text{red}})$ of

$$\widehat{\boldsymbol{x}}_{\text{red}} = \boldsymbol{h}(\widehat{\boldsymbol{x}}_{\text{red}} - \widehat{\boldsymbol{u}}_{\text{red}}; \eta) \tag{38a}$$

$$\widehat{\boldsymbol{x}}_{\text{red}} = \left(\left(1 + \frac{1}{L}\right)\boldsymbol{I} - \frac{1}{L}\boldsymbol{f}\right)^{-1}(\widehat{\boldsymbol{x}}_{\text{red}} + \widehat{\boldsymbol{u}}_{\text{red}}), \tag{38b}$$

where $\boldsymbol{h}$ was defined in (12), and L is the algorithmic parameter that appears in (32). (The parameter L also manifests in ADMM RED, as discussed in [22].) Because (38) takes the

same form as (34), we can directly compare the CE conditions of RED and prox-based PnP.

Perhaps a more intuitive way to compare the CE conditions of RED and prox-based PnP follows from rewriting (38b) as $\widehat{\boldsymbol{x}}_{\text{red}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{\text{red}}) + L\widehat{\boldsymbol{u}}_{\text{red}}$, after which the RED CE condition becomes

$$\widehat{\boldsymbol{x}}_{\text{red}} = \boldsymbol{h}(\widehat{\boldsymbol{x}}_{\text{red}} - \widehat{\boldsymbol{u}}_{\text{red}}; \eta) \tag{39a}$$

$$\widehat{\boldsymbol{x}}_{\text{red}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{\text{red}}) + L\widehat{\boldsymbol{u}}_{\text{red}}, \tag{39b}$$

which involves no inverse operations. In the typical case of $L = 1$, we see that (39) matches (34), except that the correction $\widehat{\boldsymbol{u}}_{\text{red}}$ is added after denoising in (39b) and before denoising in (34b).

Yet another way to compare the CE conditions of RED and prox-based PnP is to eliminate the $\widehat{\boldsymbol{u}}_{\text{red}}$ variable. Solving (39a) for $\widehat{\boldsymbol{u}}_{\text{red}}$ gives

$$\widehat{\boldsymbol{u}}_{\text{red}} = \frac{\eta}{\sigma^2}\boldsymbol{A}^{\mathsf{H}}(\boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}_{\text{red}}), \tag{40}$$

which mirrors the expression for $\widehat{\boldsymbol{u}}_{\text{pnp}}$. Then, plugging $\widehat{\boldsymbol{u}}_{\text{red}}$ back into (39b) and rearranging, we obtain the fixed-point equation

$$\widehat{\boldsymbol{x}}_{\text{red}} = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{\text{red}}) + \frac{L\eta}{\sigma^2}\boldsymbol{A}^{\mathsf{H}}(\boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}_{\text{red}}), \tag{41}$$

or, equivalently,

$$\frac{L\eta}{\sigma^2}\boldsymbol{A}^{\mathsf{H}}(\boldsymbol{A}\widehat{\boldsymbol{x}}_{\text{red}} - \boldsymbol{y}) = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{\text{red}}) - \widehat{\boldsymbol{x}}_{\text{red}}, \tag{42}$$

which says that the data-fitting correction [i.e., the left side of (42)] must balance the denoiser correction [i.e., the right side of (42)].

The CE framework also facilitates the convergence analysis of RED algorithms. For example, using the Mann iteration, it was proven in [22] that, when $\boldsymbol{f}$ is nonexpansive and $L > 1$, the PG RED algorithm converges to a fixed point.

Demonstration of PnP in MRI

Parallel cardiac MRI

We now demonstrate the application of PnP methods to parallel cardiac MRI. Because the signal $\boldsymbol{x}$ is a cine (i.e., a video) rather than a still image, there are relatively few options available for sophisticated denoisers. Although algorithmic denoisers, such as BM4D, have been proposed, they tend to run very slowly, especially relative to the linear operators $\boldsymbol{A}$ and $\boldsymbol{A}^{\mathsf{H}}$. For this reason, we first trained an application-specific CNN denoiser for use in the PnP framework. The architecture of the CNN denoiser, implemented and trained in PyTorch, is shown in Figure 2.

For training, we acquired 50 fully sampled cine data sets with high signal-to-noise ratio (SNR) from eight healthy volunteers. Thirty-three of those were collected on a 3-T scanner, and the remaining 17 were collected on a 1.5-T scanner. (The 3-T scanner was a Magnetom Prisma Fit, and the 1.5-T scanner was a Magnetom Avanto, both from Siemens Healthineers in Erlangen, Germany.) Of the 50 data sets, 28, seven, seven, and eight were collected in the short-axis, two-, three-, and four-chamber views, respectively. The spatial and temporal resolutions of the images ranged from 1.8 to 2.5 mm and from 34 to 52 ms, respectively. The image sizes ranged from 160×130 to 256×208 pixels, and the number of frames ranged from 15 to 27. For each of the 50 data sets, the reference image series was estimated as the least-squares solution to (1), with the sensitivity maps $\boldsymbol{S}_i$ estimated from the time-averaged data using ESPIRiT.

We added zero-mean, complex-valued independent and identically distributed Gaussian noise to these "noise-free" reference images to simulate noisy images with an SNR of 24 dB. Using a fixed stride of $30 \times 30 \times 10$ pixels, we decomposed the images into patches of size $55 \times 55 \times 15$ pixels. The noise-free and corresponding noisy patches were assigned as output and input to the CNN denoiser, with the real and imaginary parts processed as two separate channels. All 3D convolutions were performed using $3 \times 3 \times 3$ kernels. There were 64 filters of size $3 \times 3 \times 3 \times 2$ in the first layer, 64 filters of size $3 \times 3 \times 3 \times 64$ in the second through fourth layers, and two filters of size $3 \times 3 \times 3 \times 64$ in the last layer. We set the minibatch size to four and used the Adam optimizer with a learning rate of 1×10^{-4} over 400 epochs. The training process was completed in 12 h on a workstation equipped with a single NVIDIA graphic processing unit (GPU), a GeForce RTX 2080 Ti.

For testing, we acquired four fully sampled cine data sets from two different healthy volunteers, with two image series in the short-axis view, one image series in the two-chamber

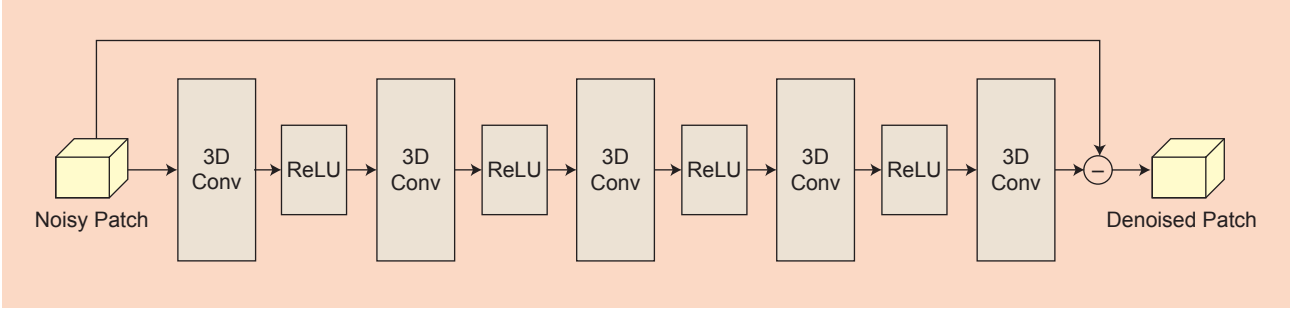

FIGURE 2. The architecture of the CNN-based cardiac cine denoiser operating on spatiotemporal volumetric patches. ReLU: rectified linear unit; Conv.: convolution.

view, and one image series in the four-chamber view. The spatial and temporal resolutions of the images ranged from 1.9 to 2 mm and from 37 to 45 ms, respectively. For the four data sets, the space-time signal vector, $\boldsymbol{x}$, in (2) had dimensions of $192 \times 144 \times 25$, $192 \times 144 \times 25$, $192 \times 166 \times 16$, and $192 \times 166 \times 16$, respectively, with the first two dimensions representing the number of pixels and the last dimension representing the number of frames. The data sets were retrospectively downsampled at acceleration rates, R, of 6, 8, and 10 by using pseudorandom sampling [26]. A representative sampling pattern used to undersample one of the data sets is shown in Figure 3. The data were compressed to $C = 12$ virtual coils for faster computation. The measurements were modeled as described in (1), with the sensitivity maps, $\boldsymbol{S}_i$, estimated from the time-averaged data using ESPIRiT.

The success of PnP methods raises important theoretical questions. Many of these questions can be answered through the consensus equilibrium framework.

For compressive MRI recovery, we used PnP ADMM from (2) with $\boldsymbol{f}$ as the CNN-based denoiser described previously; we will refer to the combination as *PnP-CNN*. We employed a total of 100 ADMM iterations, and in each ADMM iteration, we performed four steps of CG to approximate (12), for which we used $\sigma^2 = 1 = \eta$. We compared this PnP method to three CS-based methods: CS–undecimated wave transform (UWT), CS-total variation (TV) (note that sometimes UWT and TV are combined [1]), and low-rank plus sparse (L + S) as well as PnP-UWT and the transform-learning (see the overview [32] in this issue) method called *low-rank and adaptive sparse signal* (*LASSI*) [27].

For PnP-UWT, we used PnP FISTA from (14) with $\boldsymbol{f}$ implemented as $\boldsymbol{f}_{\text{tdt}}$ given in (17), that is, bFISTA. A 3D, single-level Haar undecimated wavelet transform was used as $\boldsymbol{\Psi}$ in (17). For CS-TV, we used a 3D finite-difference operator for $\boldsymbol{\Psi}$ in the regularizer $\phi(\boldsymbol{x}) = \lambda \|\boldsymbol{\Psi x}\|_1$, and for CS-UWT, we used the aforementioned UWT instead. For both CS-TV and CS-UWT, we used monotone FISTA [28] to solve the resulting convex optimization problem (3). For L + S, the low-rank plus sparse method by Otazo et al. [29] was used. The regularization weights for CS-UWT, PnP-UWT, CS-TV, and L + S were manually tuned to maximize the reconstruction SNR (rSNR) (defined as $\|\boldsymbol{x}\|^2 / \|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|^2$, where $\boldsymbol{x}$ is the true image and $\widehat{\boldsymbol{x}}$ is the estimate) for data set 3 at $R = 10$. For LASSI, we used the authors' implementation at https://gitlab.com/ravsa19/lassi, and we did our best to manually tune all available parameters.

The rSNR values are summarized in Table 1. For all four data sets and three acceleration rates, PnP-CNN exhibited the highest rSNR with respect to the fully sampled reference. Also, compared to the CS methods and PnP-UWT, which uses a more traditional denoiser based on soft-thresholding of UWT coefficients, PnP-CNN was better at preserving anatomical details of the heart (Figure 4). The performance of PnP-UWT was similar to that of CS-UWT. Figure 1 plots normalized mean-squared error (NMSE) as a function of the number of iterations for the CS and PnP methods. Because the CS methods were implemented with CPU computation and the PnP methods were implemented with GPU computation, a direct runtime comparison was not possible. We did, however, compare the per-iteration runtime of PnP ADMM for two different denoisers: the CNN and UWT-based $\boldsymbol{f}_{\text{tdt}}$ described previously in this section.

When the CNN denoiser was replaced with the UWT-based $\boldsymbol{f}_{\text{tdt}}$, the per-iteration runtime changed from 2.05 to 2.1 s, implying that the two approaches have very similar computational

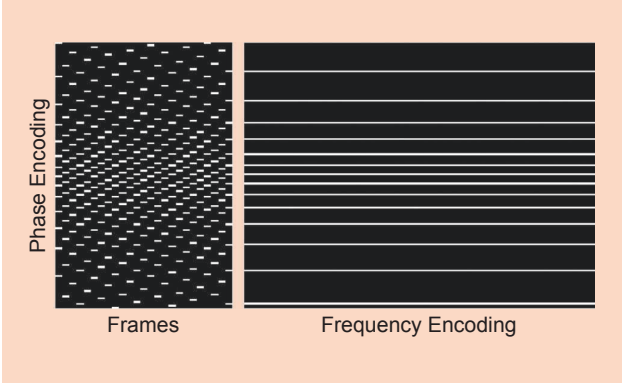

FIGURE 3. Two different views of the 3D sampling pattern used to retrospectively undersample one of the four test data sets at $R = 10$. The undersampling was performed only in the phase-encoding direction, and the pattern was varied across frames. In this example, the number of frequency encoding steps, phase encoding steps, and frames are 192, 144, and 25, respectively.

Table 1. The rSNR (dB) of MRI cardiac cine recovery from four test data sets.

Acceleration	CS-UWT	CS-TV	L + S	LASSI	PnP-UWT	PnP-CNN
Data Set 1 (Short Axis)						
$R = 6$	30.10	29.03	30.97	27.09	30.18	**31.82**
$R = 8$	28.50	27.35	29.65	25.91	28.60	**31.25**
$R = 10$	26.94	25.78	28.29	24.98	27.06	**30.46**
Data Set 2 (Short Axis)						
$R = 6$	29.23	28.27	29.73	25.87	29.29	**30.81**
$R = 8$	27.67	26.65	28.23	24.54	27.75	**30.17**
$R = 10$	26.12	25.11	26.89	23.61	26.22	**29.21**
Data Set 3 (Two Chamber)						
$R = 6$	27.33	26.38	27.83	24.97	27.38	**29.36**
$R = 8$	25.63	24.63	26.30	23.52	25.69	**28.50**
$R = 10$	24.22	23.24	24.93	22.51	24.28	**27.49**
Data Set 4 (Four Chamber)						
$R = 6$	30.41	29.63	30.62	27.62	30.60	**32.19**
$R = 8$	28.68	27.76	29.00	26.33	28.94	**31.42**
$R = 10$	27.09	26.18	27.60	25.24	27.37	**30.01**

Bold indicates the winning method.

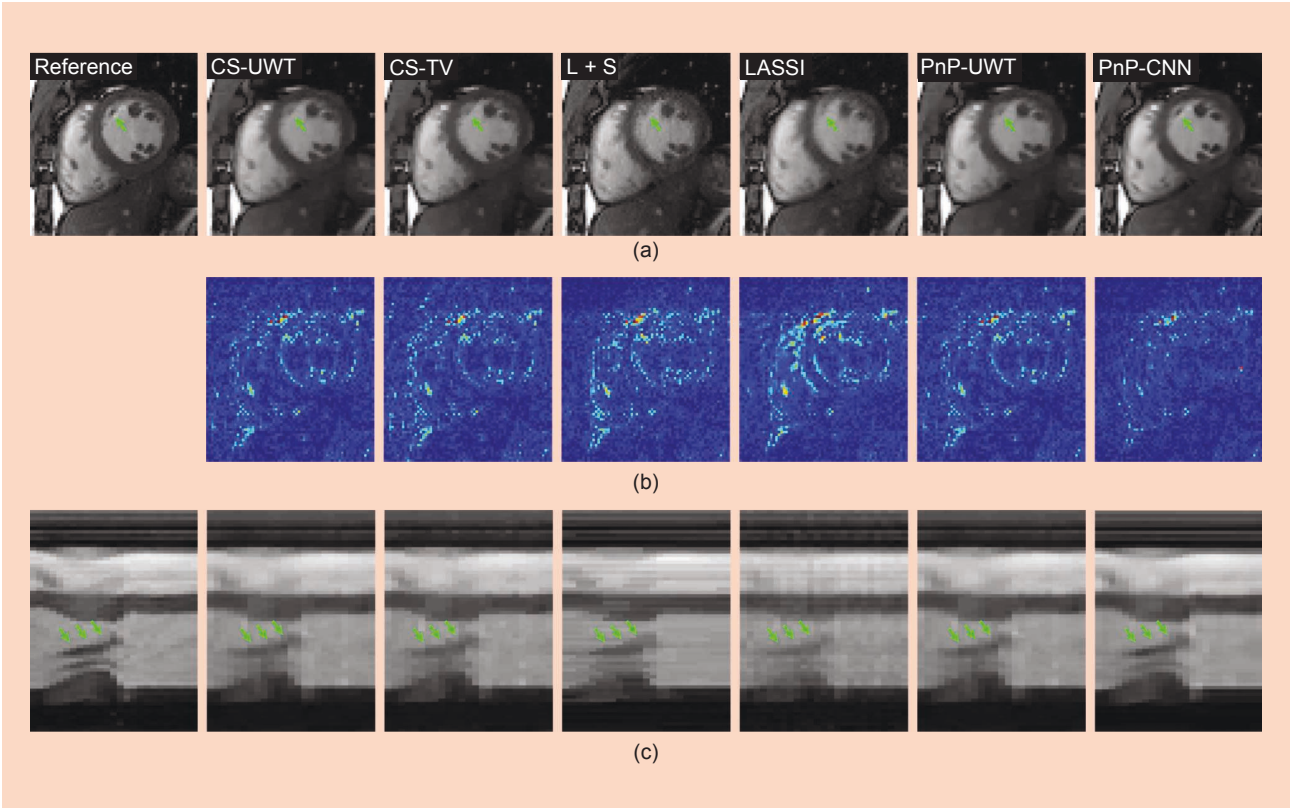

FIGURE 4. Results from the cardiac cine data set 1 at $R = 10$. (a) A representative frame from the fully sampled reference and various recovery methods. The green arrow points to an image feature that is preserved only by PnP-CNN and not by other methods. (b) The error map $\times 6$. (c) The temporal frame showing the line drawn horizontally through the middle of the image in (a), with the time dimension along the horizontal axis. The arrows point to the movement of the papillary muscles, which are more well defined in PnP-CNN.

costs. The extended version of this article [10] shows the results of experiments that investigate the effect of σ^2/η on the final NMSE and the convergence rate. Overall, final NMSE varies less than 0.5 dB for $\sigma^2/\eta \in [0.5, 2]$ for all four data sets and all three acceleration rates, and the convergence rate is nearly the same. The extended version also explores the use of CG versus GD when solving (12) in PnP ADMM. The results suggest that one to four inner iterations of either method are optimal; more inner iterations slows the overall convergence time. The results in this section, although preliminary, highlight the potential of PnP methods for MRI recovery of cardiac cines. By optimizing the denoiser architecture, the performance of PnP-CNN may be further improved.

What PnP approaches have in common is that they recover the signal by iteratively calling a sophisticated denoiser within a larger optimization or inference algorithm.

Single-coil fastMRI knee data

In this section, we investigate recovery of 2D knee images from the single-coil fastMRI data set [30]. This data set contains fully sampled *k*-space data that are partitioned into 34,742 training slices and 7,135 testing slices. The Cartesian sampling patterns from [30] were used to achieve acceleration rate $R = 4$.

We evaluated PnP using the ADMM algorithm with a learned DnCNN [9] denoiser. To accommodate complex-valued images, DnCNN was configured with two input and two output channels. The denoiser was then trained by using only the central slices of the 3-T scans without fat suppression from the training set, comprising a total of 267 slices (i.e., $<1\%$ of the total training data). The training-noise variance and the PnP ADMM tuning parameter σ^2/η were manually adjusted in an attempt to maximize rSNR.

PnP was then compared to the TV and U-Net baseline methods described and configured in [30]. For example, 128 channels were used for the U-Net's first layer, as recommended in [30]. We then trained three versions of the U-Net. The first version was trained on the full fastMRI training set with random sampling masks. (The full fastMRI training set includes 1.5-T and 3-T scans, with and without fat suppression, and an average of 36 slices per volume.) The second U-Net was trained on the full fastMRI training set, but with a fixed sampling mask. The third U-Net was trained with only the central slices of the 3-T scans without fat suppression (i.e., the same data used to train the DnCNN denoiser) and with a fixed sampling mask.

Table 2. The rSNR and SSIM for fastMRI single-coil test data with $R = 4$.

	Random Testing Masks		Fixed Testing Masks	
	rSNR (dB)	SSIM	rSNR (dB)	SSIM
CS-TV	17.56	0.647	18.16	0.654
U-Net: Random training masks, full training data	20.76	**0.772**	20.72	0.768
U-Net: Fixed training masks, full training data	19.63	0.756	20.82	**0.770**
U-Net: Fixed training masks, smaller training data	18.90	0.732	19.67	0.742
PnP-CNN	**21.16**	0.758	**21.14**	0.754

Bold indicates the winning method.

To evaluate performance, we used the central slices of the nonfat-suppressed 3-T scans from the validation set, comprising a total of 49 slices. The evaluation considered both random sampling masks and the same fixed mask used for training. The resulting average rSNR and structured similarity index (SSIM) scores are summarized in Table 2, which shows that PnP-CNN performed similarly to the U-Nets and significantly better than TV. In particular, PnP-CNN achieved the highest rSNR score with both random and fixed testing masks, and the U-Net gave slightly higher SSIM scores in both tests. Among the U-Nets, the version trained with a fixed sampling mask and full data gave the best rSNR and SSIM performance when testing with the same mask, but its performance dropped considerably when testing with random masks. Meanwhile, the U-Net trained with the smaller data performed significantly worse than the other U-Nets with either fixed or random testing masks. Although this latter U-Net used exactly the same training data as the PnP-CNN method, it was not competitive with PnP-CNN. Although preliminary, these results suggest that 1) PnP methods are much less sensitive to deviations in the forward model between training and testing and 2) PnP methods are effective with relatively small training data sets.

PnP was able to significantly outperform traditional CS methods and to perform on par with modern deep-learning methods but with considerably less training data.

Conclusions

PnP methods present an attractive avenue for compressive MRI recovery. In contrast to traditional CS methods, PnP methods can exploit richer image structure by using state-of-the-art denoisers. To demonstrate the potential of such methods for MRI reconstruction, we used PnP to recover cardiac cines and knee images from highly undersampled data sets. With application-specific CNN-based denoisers, PnP was able to significantly outperform traditional CS methods and to perform on par with modern deep-learning methods but with considerably less training data. The time is ripe to investigate the potential of PnP methods for a variety of MRI applications.

Authors

Rizwan Ahmad (ahmad.46@osu.edu) received his B.S. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2000. He received his M.S. and Ph.D. degrees from The Ohio State University, Columbus, in 2004 and 2007, respectively. He held postdoctoral and research scientist positions at The Ohio State University Medical Center. In 2017, he joined the Department of Biomedical Engineering at The Ohio State University, where he is currently an assistant professor. His research interests include signal processing, medical imaging, and cardiac magnetic resonance imaging.

Charles A. Bouman (bouman@purdue.edu) received his B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1981 and his M.S. degree from the University of California at Berkeley in 1982. From 1982 to 1985, he was a full staff member at the Massachusetts Institute of Technology Lincoln Laboratory, and in 1989, he received his Ph.D. degree in electrical engineering from Princeton University. He joined the faculty of Purdue University, West Lafayette, Indiana, where he is currently the Showalter Professor of Electrical and Computer Engineering and Biomedical Engineering. He also is a founding codirector of Purdue's Magnetic Resonance Imaging Facility. He is a Fellow of the IEEE.

Gregery T. Buzzard (buzzard@purdue.edu) received degrees in violin performance, computer science, and mathematics from Michigan State University, East Lansing, and received his Ph.D. degree in mathematics from the University of Michigan in 1995. He held postdoctoral positions at both Indiana University, Bloomington, and Cornell University, Ithaca, New York, before joining the mathematics faculty at Purdue University, West Lafayette, Indiana, in 2002, where he currently serves as head of the Department of Mathematics. His research interests include dynamical systems, experiment design, and uncertainty quantification. He is a Member of the IEEE.

Stanley Chan (stanchan@purdue.edu) received his B. Eng. degree in electrical engineering from the University of Hong Kong in 2007, his M.A. degree in mathematics from the University of California (UC) San Diego in 2009, and his Ph.D. degree in electrical engineering from UC San Diego in 2011. In 2012–2014, he was a postdoctoral research fellow at Harvard, Cambridge, Massachusetts. He is currently an assistant professor in the School of Electrical and Computer Engineering and the Department of Statistics at Purdue University, West Lafayette, Indiana. He currently serves on the IEEE Computational Imaging Technical Committee. He is a Senior Member of the IEEE.

Sizhuo Liu (liu.6221@osu.edu) joined The Ohio State University via a B.S./M.S. program after completing three years of her B.S. degree at the University of Electronic Science and Technology of China. She received her M.S. degree in electrical and computer engineering from The Ohio State University, Columbus, in 2018. She is currently a Ph.D. student in biomedical engineering at The Ohio State University. Her research interests include magnetic resonance imaging, signal processing, and machine learning.

Edward T. Reehorst (reehorst.3@osu.edu) received his B.S. degree in electrical and computer engineering from The Ohio State University, Columbus, in 2016. He has completed several internship experiences at NASA Glenn Research Center in Cleveland, Ohio, between 2012 and 2016, including the NASA Academy and Space Communication and Navigation Internship Program. He is currently a Ph.D. student in electrical and computer engineering at The Ohio State University. His research interests include imaging and machine learning.

Philip Schniter (schniter.1@osu.edu) received his B.S. and M.S. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, in 1992 and 1993, respectively, and his Ph.D. degree in electrical engineering from Cornell University, Ithaca, New York, in 2000. He is currently a professor in the Department of Electrical and Computer Engineering at The Ohio State University, Columbus. He received the IEEE Signal Processing Society Best Paper Award in 2016 and currently serves on the IEEE Computational Imaging Technical Committee. He is a Fellow of the IEEE.

References

[1] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reason. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007. doi: 10.1002/mrm.21391.

[2] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo, "Deep learning for undersampled MRI reconstruction," *Phys. Med. Biol.*, vol. 63, no. 13, pp. 135007, 2018. doi: 10.1088/1361-6560/aac71a.

[3] H. K. Aggarwal, M. P. Mani, and M. Jacob, "Model based image reconstruction using deep learned priors (MoDL)," in *Proc. IEEE Int. Symp. Biomedical Imaging*, 2018, pp. 671–674.

[4] A. Hauptmann, S. Arridge, F. Lucka, V. Muthurangu, and J. A. Steeden, "Real-time cardiovascular MR with spatio-temporal artifact suppression using deep learning—Proof of concept in congenital heart disease," *Magn. Reason. Med.*, vol. 81, no. 2, pp. 1143–1156, 2019. doi: 10.1002/mrm.27480.

[5] M. Akçakaya, S. Moeller, S. Weingärtner, and K. Uğurbil, "Scan-specific robust artificial-neural-networks for k-space interpolation (RAKI) reconstruction: Database-free deep learning for fast imaging," *Magn. Reason. Med.*, vol. 81, no. 1, pp. 439–453, 2019. doi: 10.1002/mrm.27420.

[6] F. Knoll, K. Hammernik, E. Kobler, T. Pock, M. P. Recht, and D. K. Sodickson, "Assessment of the generalization of learned image reconstruction and the potential for transfer learning," *Magn. Reason. Med.*, vol. 81, no. 1, pp. 116–128, 2019. doi: 10.1002/mrm.27355.

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

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007. doi: 10.1109/TIP.2007.901238.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017. doi: 10.1109/TIP.2017.2662206.

[10] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, Plug-and-play methods for magnetic resonance imaging (long version). 2019. [Online]. Available: https://arxiv.org/pdf/1903.08616.pdf

[11] M. S. Hansen and P. Kellman, "Image reconstruction: An overview for clinicians," *J. Magn. Reason. Imaging*, vol. 41, no. 3, pp. 573–585, 2015. doi: 10.1002/jmri.24687.

[12] Y. Liu, Z. Zhan, J.-F. Cai, D. Guo, Z. Chen, and X. Qu, "Projected iterative soft-thresholding algorithm for tight frames in compressed sensing magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 35, no. 9, pp. 2130–2140, 2016. doi: 10.1109/TMI.2016.2550080.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers* (*Now Foundations and Trends*, vol. 3, no. 1). Boston, MA: Now, 2011. doi: 10.1561/2200000016.

[14] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comp. Imag.*, vol. 2, no. 4, pp. 408–423, 2016.

[15] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comp. Imag.*, vol. 3, no. 1, pp. 84–98, 2017. doi: 10.1109/TCI.2016.2629286.

[16] U. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1872–1876, May 2017. doi: 10.1109/LSP.2017.2763583.

[17] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1108–1112, 2017. doi: 10.1109/LSP.2017.2710233.

[18] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Comp. Imag.*, vol. 5, no. 3, pp. 395–408, 2019. doi: 10.1109/TCI.2019.2893568.

[19] J. Fessler, "Optimization methods for magnetic resonance image reconstruction: Key models and optimization algorithms," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 33–40, Jan. 2020. doi: 10.1109/MSP.2019.2943645.

[20] S. T. Ting, R. Ahmad, N. Jin, J. Craft, J. Serafim da Silverira, H. Xue, and O. P. Simonetti, "Fast implementation for compressive recovery of highly accelerated cardiac cine MRI using the balanced sparse model," *Magn. Reason. Med.*, vol. 77, no. 4, pp. 1505–1515, Apr. 2017. doi: 10.1002/mrm.26224.

[21] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imaging Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017. doi: 10.1137/16M1102884.

[22] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. Comp. Imag.*, vol. 5, no. 1, pp. 52–67, Mar. 2019. doi: 10.1109/TCI.2018.2880326.

[23] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, Amortised MAP inference for image super-resolution. 2016. [Online]. Available: https://arXiv.org/pdf/1610.04490v1.pdf

[24] S. H. Chan, "Performance analysis of plug-and-play ADMM: A graph signal processing perspective," *IEEE Trans. Comp. Imag.*, vol. 5, no. 2, pp. 274–286, 2019. doi: 10.1109/TCI.2019.2892123.

[25] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM J. Imag. Sci.*, vol. 11, no. 3, pp. 2001–2020, 2018. doi: 10.1137/17M1122451.

[26] R. Ahmad, H. Xue, S. Giri, Y. Ding, J. Craft, and O. P. Simonetti, "Variable density incoherent spatiotemporal acquisition (VISTA) for highly accelerated cardiac MRI," *Magn. Reason. Med.*, vol. 74, no. 5, pp. 1266–1278, 2015. doi: 10.1002/mrm.25507.

[27] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, "Low-rank and adaptive sparse signal (LASSI) models for highly accelerated dynamic imaging," *IEEE Trans. Med. Imag.*, vol. 36, no. 5, pp. 1116–1128, 2017. doi: 10.1109/TMI.2017.2650960.

[28] Z. Tan, Y. C. Eldar, A. Beck, and A. Nehorai, "Smoothing and decomposition for analysis sparse recovery," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1762–1774, Apr. 2014.

[29] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magn. Reson. Med.*, vol. 73, no. 3, pp. 1125–1136, 2015. doi: 10.1002/mrm.25240.

[30] J. Zbontar, F. Knoll, A. Sriram, M. J. Muckley, M. Bruno, A. Defazio, M. Parente, K. J. Geras, et al. "fastMRI: An open dataset and benchmarks for accelerated MRI," 2018. [Online]. Available: https://arxiv.org/pdf/1811.08839.pdf

[31] M. Doneva, "Mathematical models for magnetic resonance imaging reconstruction: An overview of the approaches, problems, and future research areas," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 24–32, Jan. 2020. doi: 10.1109/MSP.2019.2936964.

[32] B. Wen, S. Ravishankar, L. Pfister, and Y. Bresler, "Transform learning for magnetic resonance image reconstruction," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 41–53, Jan. 2020.

SP