Relaxed Plug-and-Play ADMM Image Deblurring

1st Wenzhu Xing
Computational Image Group
Tampere University, Finland
wenzhu.xing@tuni.fi

2nd Vladimir Katkovnik
Computational Image Group
Tampere University, Finland
vladimir.katkovnik@tuni.fi

3rd Karen Egiazarian
Computational Image Group
Tampere University, Finland
karen.eguiazarian@tuni.fi

***Abstract*—Plug-and-play (PnP) methods have become a popular approach for image reconstruction. Instead of explicitly modeling statistical priors, PnP leverages off-the-shelf image denoisers to iteratively refine image estimates. In this paper, we present a relaxed version of the Plug-and-Play (PnP) method for learning-based image deblurring, built upon the alternating direction method of multipliers (ADMM), which we call R-PnP-ADMM. In this approach, we introduce a relaxation parameter into the ADMM data-fidelity update step. Additionally, a lightweight HyperNet module is employed to optimize the hyperparameters of R-PnP-ADMM, significantly reducing the number of iterations. We validate the effectiveness of our algorithm through extensive experiments on image deblurring tasks. The results demonstrate that R-PnP-ADMM achieves stable convergence across a wide range of blur kernels and noise levels, with state-of-the-art (non-blind) deblurring performance.**

***Index Terms*—Plug-and-play (PnP), alternating direction method of multipliers (ADMM), non-blind image deblurring.**

I. Introduction

In image deblurring, the goal is to recover an original image x from its degraded (blurred and noisy) observation y

$$y = Hx + n. \tag{1}$$

Here H is a blur operator, $Hx = K * x$, where $*$ is the convolution operation, and n is an additive noise. The image reconstruction can be reformulated as the optimization problem:

$$\hat{x} = \arg\max_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|Hx - y\|^2 + \gamma g(x) \right\}, \tag{2}$$

where the first term - a data fidelity term measures the discrepancy between the observed y and true x images, $g(x)$ is the prior term representing regularization or prior knowledge about the clean image, and γ is a positive trade-off parameter.

There are variety of methods to solve (2), including gradient descent, conjugate gradients, proximal gradients, the Alternating Direction Method of Multipliers (ADMM), and Half Quadratic Splitting (HQS). These methods employ different regularization schemes and image priors to model the marginal distributions of $g(x)$, such as total variation (TV) regularization [9], sparse image priors [3], [7], natural image priors [6], [12], hyper-Laplacian priors [5], and Gaussian mixture model (GMM) priors [20]. However, these approaches often incur high computational costs and significant processing time to achieve high-quality results.

Plug-and-play image deblurring methods combine off-the-shelf denoisers with iterative optimization frameworks. These denoisers serve as priors, encoding natural image statistics like spatial coherence, sharpness, and texture [2], [14]. In [17], Zhang *et al*. proposed a PnP-HQS method involving deep convolutional neural network (CNN) denoisers. DPIR [16] applies a very deep CNN based denoiser (DRUNet) into the PnP HQS scheme achieving state-of-the-art performance on image deblurring.

To enhance the convergence and robustness of ADMM, relaxed ADMM methods are used [13] to balance the updates between the main and auxiliary (primal and dual) variables by incorporating a relaxation parameter. GS-PnP [4] method introduced a relaxation in the regularizer term, relaxing the output of the denoiser.

In this paper, we propose a novel relaxation method of PnP-ADMM algorithm for iterative image deblurring, where the relaxation is applied to the data-fidelity step.

Contributions of this paper are the following:

1) A relaxed PnP-ADMM image deblurring algorithm (R-PnP-ADMM) is proposed.
2) The convergence of R-PnP-ADMM based on assumptions on the criterion function $f(x)$ and PnP filter is proven.
3) A hyperparameter optimization HyperNet module $\mathcal{H}$ is integrated into the R-PnP-ADMM to allow efficient realization with a small number of iterations.

II. Method

A. PnP-ADMM framework

The augmented Lagrangian for the problem (2) is:

$$\mathcal{L}(x, z, u) = f(x) + \lambda g(z) + \langle u, x - z\rangle + \frac{1}{2}\|x - z\|_2^2 \tag{3}$$

Here $f(x) = \frac{1}{2}\|Hx - y\|_2^2$, $z \in \mathbb{R}^n$ is an auxiliary variable to split the optimization into simpler subproblems, $u \in \mathbb{R}^n$ is the Lagrange multiplier used to enforce consistency between x and z, The ADMM algorithm involves alternating updates for x, z, and u.

The ADMM applies the proximal operator defined as:

$$\mathrm{Prox}_{\tau h}(w) := \arg\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|x - w\|_2^2 + \tau h(x) \right\} \tag{4}$$

to arrive at the three-step updates of the algorithm:

$$x^k = \mathrm{Prox}_{\alpha f}(z^{k-1} - u^{k-1}) \tag{5}$$

$$z^k = \mathrm{Prox}_{\gamma g}(x^k + u^{k-1}) \tag{6}$$

$$u^k = u^{k-1} + x^k - z^k. \tag{7}$$

In the plug-and-play framework, the z-update step in (6) is replaced by a denoising step using a denoising algorithm $\mathcal{D}_\sigma$:

$$z^k = \mathcal{D}_\sigma(x^k + u^{k-1}), \tag{8}$$

where σ is the denoising parameter.

B. Plug-and-Play ADMM image deblurring with relaxation

The relaxed PnP ADMM is defined by modifying the x-update step (5) of PnP ADMM

$$x^k = \beta \text{Prox}_{\gamma f}(z^{k-1} - u^{k-1}) + (1-\beta)(z^{k-1} - u^{k-1}) \tag{9}$$

where $\beta \in (0, 2]$ is a relaxation parameter. This makes the PnP-ADMM the special case of R-PnP-ADMM for $\beta = 1$.

Implementing deblurring in the Fourier domain, (5) for x-update step can be written as

$$\begin{aligned} x^k = \beta \mathcal{F}^{-1}\Big(&\frac{\overline{\mathcal{F}(K)}\mathcal{F}(y) + \mu\mathcal{F}(z^{k-1} - u^{k-1})}{\overline{\mathcal{F}(K)}\mathcal{F}(K) + \mu}\Big) \\ &+ (1-\beta)((z^{k-1} - u^{k-1})), \end{aligned} \tag{10}$$

$\mathcal{F}(K)$ and $\mathcal{F}(y)$, are Fourier transforms of the blur kernel K and the observed image y, respectively, $\mathcal{F}^{-1}$ is the inverse Fourier operator, and μ is a regularization parameter.

The framework of R-PnP-ADMM-based image deblurring is presented in **Algorithm 1**.

Algorithm 1: Plug-and-play ADMM image deblurring with relaxation

```
Input: blur image $y$, kernel $K$, noise level $n$,
       pre-trained denoiser $\mathcal{D}$, pre-trained HyperNet $\mathcal{H}$.
Output: final prediction $\hat{x}$; relaxed prediction $\hat{z}$.
estimate hyper-parameters $\mu, \sigma, \beta = \mathcal{H}(n)$;
initialize $x_0 = y$ $u_0 = 0$;
for $k = 1, 2, 3, \cdots$ do
    update $x^k$ by (9);
    update $z^k$ by (8);
    update $u^k$ by (7);
Set $\hat{x} = x^k$, and $\hat{z} = z^k$.
return $\hat{x}, \hat{z}$
```

C. Hyperparameter network

There are three parameters to be selected/optimized in the Algorithm 1: the regularization parameter μ, the denoising parameter θ and the relaxation parameter β.

We apply the HyperNet module $\mathcal{H}$ to optimize the hyperparameters of the R-PnP-ADMM algorithm similar to the method in [15],. There are two branches in the HyperNet $\mathcal{H}$, one for μ and θ estimation, and another branch for the optimization of the relaxation parameter β. The structure of each branch is similar the module $\mathcal{H}$ in [15], but in the second branch we use a Sigmoid activation function for the final layer, ensuring that the relaxation parameter β stays within the range $[0, 2]$. Another difference of our HyperNet with one in [15], is that it is not used for training the HyperNet with the data module in an end-to-end manner like in [15], but we train the HyperNet separately. The experimental results demonstrate that the HyperNet $\mathcal{H}$ can efficiently optimize the hyperparameters for any type of blur kernels and noise levels. using only a single input, the noise level σ. Due to the compact structure of the HyperNet module, a small training set is sufficient for training.

III. Analysis

A. Convergence analysis

The relaxed plug-and-play ADMM updates follow the equations in (9), (8), and (7). We interpret R-PnP-ADMM as fixed-point iterations, where the fixed point (x^*, z^*, u^*) satisfies the following equations:

$$x^* = \beta \text{Prox}_{\alpha f}(z^* - u^*) + (1-\beta)(z^* - u^*), \tag{11}$$

$$z^* = D_\sigma(x^* + u^*), x^* = z^*. \tag{12}$$

In analyzing this algorithm, we focus on its convergence to a fixed point. This builds on the methodology and results from [10], [11], which examine the PnP-ADMM algorithm. We extend these results to the R-PnP-ADMM algorithm. The relaxation parameter β makes R-PnP-ADMM fundamentally different from PnP-ADMM, leading to differences in solutions, convergence conditions, and convergence rates. The main assumptions of our analysis are on the criterion function $f(x)$ and PnP filter D_σ. We assume that

Assumption 1. *$f(x)$ is strongly convex and differentiable,*

Assumption 2. *The operator D_σ: $R^d \longrightarrow R^d$ such that*

$$||(D_\sigma - I)(x) - (D_\sigma - I)(y)||^2 < \epsilon^2 ||x - y||^2 \tag{13}$$

for all $x, y \in R^d$ and for some $\varepsilon > 0$, where I is the identity operator and the norms are Euclidean. The parameter σ controls the strength of the filtering in PnP, we can expect D_σ to be close to the identity for small σ. If so, it is reasonable to assume the inequality (13).

Under these assumptions, the R-PnP-ADMM iterations are contractive: we can express the iterations as $x^{k+1} = T_\beta(x^k)$, where T_β: $R^d \longrightarrow R^d$

$$||T_\beta(x) - T_\beta(y)|| < \delta_\beta ||x - y|| \tag{14}$$

for all $x, y \in R^d$ and a contraction factor $\delta_\beta < 1$. If x satisfies $T_\beta(x^*) = x^*$, i.e., x^* is a fixed point, then x^k converges geometrically to x^* with the convergence rate δ_β.

Theorem III.1 (Convergence of R-PnP-ADMM)**.** *Let f be ψ -strongly convex and differentiable, and D_σ satisfy the Assumption 2.*

Then

$$\begin{aligned} T_\beta = \; & (1 - \tfrac{1}{2}\beta)I + \tfrac{1}{2}(2D_\sigma - I)(\beta I)(2\text{Prox}_{\alpha f} - I) \\ & + (D_\sigma(1-\beta) - (1-\beta)I) \end{aligned} \tag{15}$$

satisfies $||T_\beta(x) - T_\beta(y)|| < \delta_\beta ||x-y||$, *where*

$$\delta_\beta = (1+2\epsilon)(\frac{\beta + \alpha\psi - \beta\alpha\psi + 2\epsilon\beta\alpha\psi}{\beta + \alpha\psi + 2\epsilon\alpha\beta\psi} + (1-\beta)/2) - \epsilon. \quad (16)$$

The convergence condition:

$$(1+2\epsilon)(\frac{\beta + \alpha\psi - \beta\alpha\psi + 2\epsilon\beta\alpha\psi}{\beta + \alpha\psi + 2\epsilon\alpha\beta\psi} + (1-\beta)/2) - \epsilon < 1. \quad (17)$$

The convergence statement of the theorem is valid for the deblurring Algorithm 1 provided that $\alpha = 1/(2\mu)$, and $\psi > 0$ is the minimal eigenvalue of the operator $K^T K$. Here, μ is a regularization parameter from (10). It can be seen from (16) that smaller ψ results in a lower convergence rate.

If $\beta = 1$, δ_1 equals to the corresponding value δ obtained in [11] for the algorithm without relaxation.

B. Analysis of hyperparameters

Training settings of HyperNet. A small HyperNet module, $\mathcal{H}$, is trained independently to determine the set of 3 hyperparameters used in Algorithm 1. The loss function for training $\mathcal{H}$ is formulated as follows:

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i}^{N} \ell_1(x_i, \hat{x}_i) \quad (18)$$

$$\hat{x}_i = \text{Algorithm 1}(y_i, K, \mathcal{D}, \mathcal{H}(n)) \quad (19)$$

where $\{(y_i, x_i)\}_{i=1}^{N}$ denotes N blur-clean patch pairs, and $\mathcal{D}$ is pre-trained denoiser. To train $\mathcal{H}$, we crop the training images into 256×256 patches, use a batch size of 2, and train for 500 epochs with the Adam optimizer and a learning rate of 0.0001. The module is trained on an Nvidia Tesla V100 GPU with 32 GB of memory, taking approximately 1 hour to complete.

Adaptive hyperparameters or fixed hyperparameters to iterations. To investigate whether hyperparameters should be fixed or adaptive, we trained two variants of the $\mathcal{H}$ module to optimize the regularizer trade-off parameter μ and the denoiser parameter θ. One version of $\mathcal{H}$ optimizes fixed hyperparameters, while the other optimizes adaptive hyperparameters. The relaxation parameter β for R-PnP-ADMM is fixed at 0.8. In the fixed hyperparameter version, only two constants are estimated. In contrast, the adaptive version of $\mathcal{H}$ estimates parameters for each iteration, meaning the number of parameters to be optimized is twice the number of iterations.

To make the study more comprehensive, we also trained two types of HyperNet modules for the standard PnP-ADMM method. For this experiment, we set the number of iterations for R-PnP-ADMM and PnP-ADMM to 8 and 10, respectively.

In Table I, we evaluate two methods with different hyperparameter optimization schemes for deblurring on test set consisting of 6 commonly used grayscale images of size 256×256. The evaluation metric is the peak signal-to-noise ratio (PSNR). The results show that both methods with fixed hyperparameters outperform those with adaptive hyperparameters, achieving PSNR improvements of 0.14 dB and 0.15 dB, respectively. This indicates that PnP-ADMM-based methods tend to perform better with fixed hyperparameters rather than adaptive ones. Additionally, R-PnP-ADMM outperforms the standard PnP-ADMM method with fewer iterations (8 vs. 10).

Method	adaptive *hps*	fixed *hps*
PnP-ADMM	29.78	30.03
R-PnP-ADMM	30.30	30.44

TABLE I: The average PSNR(dB) results for PnP-ADMM and R-PnP-ADMM methods; gray-scale image deblurring. The test set is Set6. The blur kernel is uniform 9×9; noise level (the standard deviation of the additive noise) is 2, and *hps* stands for hyperparameters.

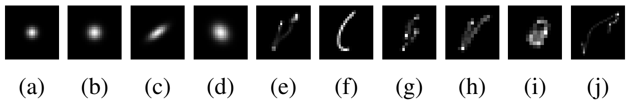


Fig. 1: The eight testing blur kernels for image deblurring: (a)-(b) isotropic Gaussian kernels; (c)-(d) anisotropic Gaussian kernels; (e)-(f) motion kernels; (g)-(h) real kernels.

IV. Evaluation

In this section, we compare the proposed R-PnP-ADMM with several recent PnP-based image restoration methods: DPIR [15], GS-PnP [4], and PnP-ADMM with hyperparameters optimized by $\mathcal{H}$, denoted as PnP-ADMM*. These methods are evaluated on a set of images distorted by various blur kernels and noise levels. We used 10 different blur kernels, shown in Fig. 1, including two isotropic Gaussian kernels with different widths (1.6 and 2.0), two anisotropic Gaussian kernels from [18], two motion blur kernels from [1], [8], and four real-world camera shake kernels from [8]. Additionally, we conducted experiments with two uniform blur kernels of sizes 9×9 and 17×17. The additive noise is assumed to be Gaussian, with one of three noise levels: $\sqrt{2}$, 4, or 25.

For training and testing, we generate blurred images by convolving an image with a blur kernel and adding Gaussian noise with a noise level of σ. For testing, we used six grayscale images and the McM set [19] for grayscale and color image deblurring, respectively. The proposed R-PnP-ADMM and PnP-ADMM* are set to 8 and 10 iterations, respectively. The DPIR method runs for 8 iterations, while GS-PnP stops when it converges, with a maximum of 400 iterations.

Numerical comparison. The PSNR results for grayscale and color image deblurring are presented in Tables II, respectively. From these tables, it is clear that the proposed method delivers the best performance, outperforming the other methods by up to 0.49 dB for grayscale images and up to 0.77 dB for color image deblurring.

Visual comparison. Figures 2 shows the deblurred grayscale and color images. For gray-scale deblurring, it is evident that DPIR and GS-PnP tend to smooth high-frequency details, such as the scarf pattern, while effectively removing noise. On the other hand, PnP-ADMM* successfully recovers the scarf pattern but retains noise artifacts. In contrast, the proposed method strikes a better balance between noise suppres-

Dataset	σ	Method	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
Set6	$\sqrt{2}$	DPIR	30.67	28.86	28.60	27.79	35.49	35.52	36.19	35.88	35.47	35.63	30.81	27.50
		GS-PnP	30.01	28.65	28.38	27.67	34.53	35.16	35.72	35.47	35.05	35.23	30.28	27.18
		PnP-ADMM*	30.43	28.81	28.59	27.78	35.30	35.19	36.26	35.96	35.11	35.74	30.84	27.85
		R-PnP-ADMM	**30.95**	**29.06**	**28.90**	**28.06**	**35.65**	**35.63**	**36.42**	**36.11**	**35.60**	**35.87**	**31.26**	**28.34**
	4	DPIR	28.99	27.95	27.68	26.93	31.99	31.89	32.36	32.01	31.96	31.84	28.48	25.48
		GS-PnP	28.96	27.92	27.73	27.01	31.73	31.66	32.07	31.74	31.69	31.60	28.38	25.59
		PnP-ADMM*	28.62	27.63	27.39	26.69	31.51	31.31	32.33	31.88	31.63	31.84	28.47	25.57
		R-PnP-ADMM	**29.17**	**28.10**	**27.90**	**27.16**	**32.12**	**31.97**	**32.56**	**32.19**	**31.99**	**32.03**	**28.87**	**26.10**
	25	DPIR	26.35	25.58	25.30	24.81	25.64	**25.60**	26.15	25.74	26.36	**25.42**	24.40	22.21
		GS-PnP	26.13	25.48	25.21	24.81	25.57	25.36	25.96	25.60	26.15	25.19	24.44	22.59
		PnP-ADMM*	25.64	23.58	24.72	24.36	23.87	23.54	26.11	23.66	25.73	23.56	24.13	22.35
		R-PnP-ADMM	**26.41**	**25.67**	**25.42**	**24.98**	**25.77**	**25.60**	**26.36**	**25.75**	**26.61**	**25.42**	**24.60**	**22.60**
McM	$\sqrt{2}$	DPIR	33.30	31.51	30.82	29.96	36.88	37.10	37.51	37.14	36.97	36.99	32.21	29.01
		GS-PnP	32.65	30.65	29.85	29.10	35.94	36.57	36.92	36.70	36.45	36.51	31.40	28.01
		PnP-ADMM*	33.01	31.44	31.00	30.02	36.04	36.67	37.63	36.89	36.77	36.48	32.34	28.53
		R-PnP-ADMM	**33.57**	**31.90**	**31.43**	**30.44**	**37.05**	**37.20**	**37.82**	**37.39**	**37.14**	**37.26**	**32.73**	**29.78**
	4	DPIR	31.75	30.22	29.67	28.85	33.50	**33.53**	33.85	33.54	33.64	33.34	29.87	26.82
		GS-PnP	31.78	30.27	29.71	28.90	33.36	33.41	33.60	33.31	33.53	33.07	30.00	26.92
		PnP-ADMM*	31.12	29.82	29.40	28.53	32.73	32.71	33.78	33.20	33.47	33.01	29.36	26.40
		R-PnP-ADMM	**31.98**	**30.53**	**30.09**	**29.24**	**33.57**	**33.53**	**34.03**	**33.65**	**33.71**	**33.46**	**30.32**	**27.58**
	25	DPIR	28.39	27.36	27.07	26.48	27.48	27.30	27.60	**27.46**	28.28	27.00	26.07	23.62
		GS-PnP	28.17	27.33	27.05	26.54	27.48	**27.43**	27.48	27.40	28.11	**27.05**	26.46	**24.29**
		PnP-ADMM*	27.50	26.65	26.41	24.24	25.58	25.08	27.30	25.24	27.54	25.83	24.93	23.25
		R-PnP-ADMM	**28.47**	**27.52**	**27.26**	**26.66**	**27.59**	27.38	**27.87**	27.36	**28.40**	26.90	**26.52**	24.12
Set3C	$\sqrt{2}$	DPIR	30.58	28.35	27.25	26.51	36.54	36.79	37.38	36.91	36.62	36.77	30.01	24.20
		GS-PnP	29.79	27.03	25.90	25.14	35.64	36.05	36.69	36.22	35.99	36.07	28.35	23.61
		PnP-ADMM*	30.40	28.51	27.75	26.82	35.97	36.09	37.55	36.67	36.13	36.48	30.55	23.17
		R-PnP-ADMM	**31.21**	**29.19**	**28.71**	**27.43**	**36.72**	**36.94**	**37.70**	**37.11**	**36.76**	**37.07**	**31.25**	**26.93**
	4	DPIR	28.90	27.03	25.99	25.31	32.44	32.53	33.16	32.65	32.45	32.50	26.83	21.54
		GS-PnP	29.42	27.22	26.33	25.37	32.36	32.45	32.90	32.42	32.49	32.23	27.39	23.02
		PnP-ADMM*	27.82	26.37	25.58	24.76	31.21	31.51	33.16	31.80	32.24	32.12	25.21	20.84
		R-PnP-ADMM	**29.44**	**27.66**	**27.09**	**26.03**	**32.64**	**32.63**	**33.40**	**32.80**	**32.66**	**32.69**	**28.00**	**23.93**
	25	DPIR	25.52	23.98	23.06	22.53	24.65	24.87	24.91	24.27	25.75	24.26	21.72	18.25
		GS-PnP	25.56	**24.19**	**23.37**	**22.93**	**25.24**	**25.25**	25.34	**25.01**	25.93	**24.82**	22.61	**19.59**
		PnP-ADMM*	24.04	22.64	21.93	19.83	20.45	20.78	24.87	19.98	24.58	21.74	20.10	17.75
		R-PnP-ADMM	**25.58**	24.16	23.35	22.72	25.18	25.07	**25.70**	24.56	**26.03**	24.38	**22.81**	19.13

TABLE II: Deblurring results for different methods; **gray-scale** test set Set6; **color** test set McM and Set3C; 12 blur kernels and 3 noise levels. The blur kernels (a)-(j) are plotted in 1, (k)-(l) are uniform kernels 9×9 and 17×17, respectively. Best and second-best results are displayed in bold and underlined.

sion and image reconstruction, yielding a more natural result. For the color image deblurring with a real kernel, the GS-PnP reconstruction exhibits noticeable ringing artifacts, while the PnP-ADMM* output contains visible noise artifacts. In comparison, the proposed method restores the image without introducing any artifacts.

Convergence analysis. We evaluated the proposed R-PnP-ADMM and the standard PnP-ADMM, utilizing optimized hyperparameters learned from the HyperNet module $\mathcal{H}$. PSNR values for both methods were computed over 50 iterations. To streamline the training process, the HyperNet modules for these methods were trained with their respective algorithms for only 8 and 10 iterations, respectively. In addition to these two methods, we also compare the convergence performance of the GS-PnP [4] method.

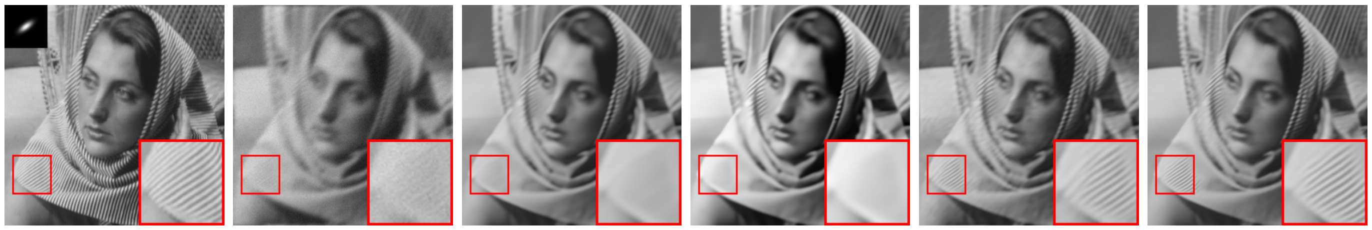
In Fig. 3, we plot the deblurring PSNR results of these three methods on the Barbara image (the first image in the Set6) with the 9×9 uniform kernel and the noise level $\sqrt{2}$. According to Fig.3, GS-PnP converges after more than 100 iterations, and standard PnP-ADMM and the proposed R-PnP-ADMM converge in less than 20 iterations. This demonstrates that optimizing the hyperparameters by the HyperNet yields faster convergence. In addition, the proposed R-PnP-ADMM surpasses the other two methods in only three iterations by at least 0.7 dB, which proves the importance of the relaxation.

V. Conclusion

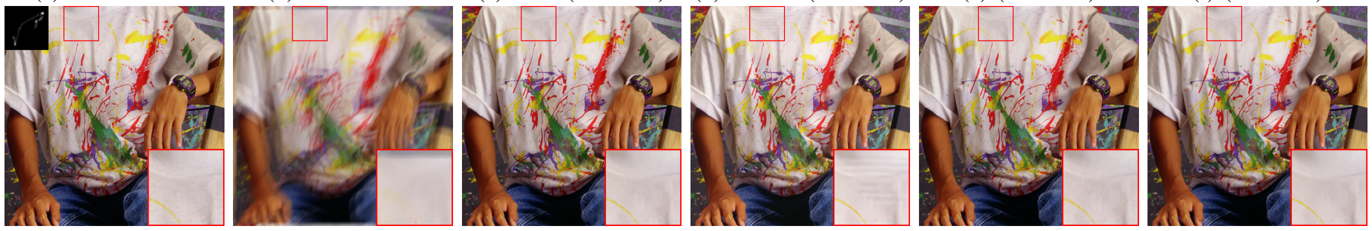
In this paper, we propose a relaxed Plug-and-Play ADMM (R-PnP-ADMM) algorithm applicable to a wide range of image reconstruction tasks. We thoroughly evaluate the algorithm in the context of non-blind image deblurring. Extensive experimental results demonstrate that R-PnP-ADMM effectively handles various blur kernel types, kernel sizes, and noise levels. Furthermore, we introduce a simple yet flexible HyperNet module for hyperparameter optimization and investigate the relationship between the optimized hyperparameters, kernel size, and noise level. Additionally, we establish the convergence of the R-PnP-ADMM algorithm. Our findings show that the proposed R-PnP-ADMM algorithm successfully recovers sharp, clear images across diverse conditions.

References

[1] Giacomo Boracchi and Alessandro Foi. Modeling the performance of image restoration from motion blur. *IEEE Transactions on Image Processing*, 21(8):3502–3517, 2012. 3



(a) Ground-truth (b) Observation (c) DPIR(24.28dB) (d) GS-PnP(24.29dB) (e) (24.36dB) (f) (24.45dB)



(g) Ground-truth (h) Observation (i) DPIR(36.38dB) (j) GS-PnP(35.90dB) (k) PnP-ADMM*(36.00dB) (l) R-PnP-ADMM(36.62dB)

Fig. 2: Image deblurring visual comparison. **Gray-scale**: anisotropic Gaussian kernel (c) and the noise level is 4.0. **Color**: real kernel (j) and the noise level is $\sqrt{2}$.

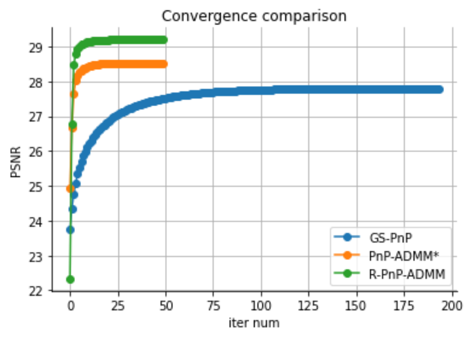


Fig. 3: Comparison of convergence rates for different methods; *barbara* test-image from Set6. The blur kernel is uniform 9×9; the noise level is $\sqrt{2}$.

[2] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. Bm3d frames and variational image deblurring. *IEEE Transactions on image processing*, 21(4):1715–1728, 2011. 1

[3] Rajshekhar Das, Anurag Bajpai, and Shankar M Venkatesan. Fast non-blind image deblurring with sparse priors. In *Proceedings of International Conference on Computer Vision and Image Processing: CVIP 2016, Volume 1*, pages 629–641. Springer, 2017. 1

[4] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient step denoiser for convergent plug-and-play. *arXiv preprint arXiv:2110.03220*, 2021. 1, 3, 4

[5] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in neural information processing systems*, 22, 2009. 1

[6] Anat Levin, Rob Fergus, Fredo Durand, and William T Freeman. Deconvolution using natural image priors. *Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory*, 3, 2007. 1

[7] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70–es, 2007. 1

[8] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1964–1971. IEEE, 2009. 3

[9] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 1

[10] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin, 2019. See Supplemental Material at http://proceedings.mlr.press/v97/ryu19a/ryu19a-supp.pdf for Plug-and-Play Methods Provably Converge with Properly Trained Denoisers. 2

[11] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557. PMLR, 2019. 2, 3

[12] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *Acm transactions on graphics (tog)*, 27(3):1–10, 2008. 1

[13] Jintao Song, Wenqi Lu, Yunwen Lei, Yuchao Tang, Zhenkuan Pan, and Jinming Duan. Optimizing admm and over-relaxed admm parameters for linear quadratic problems. In *Proc. of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. 1

[14] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE global conference on signal and information processing*, pages 945–948. IEEE, 2013. 1

[15] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3217–3226, 2020. 2, 3

[16] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021. 1

[17] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017. 1

[18] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3262–3271, 2018. 3

[19] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2):023016, 2011. 3

[20] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 international conference on computer vision*, pages 479–486. IEEE, 2011. 1