PROVABLY CONVERGENT PLUG & PLAY LINEARIZED ADMM, APPLIED TO DEBLURRING SPATIALLY VARYING KERNELS

Under review

ABSTRACT

Plug & Play methods combine proximal algorithms with denoiser priors to solve inverse problems. These methods rely on the computability of the proximal operator of the data fidelity term. In this paper, we propose a Plug & Play framework based on linearized ADMM that allows us to bypass the computation of intractable proximal operators. We demonstrate the convergence of the algorithm and provide results on restoration tasks such as super-resolution and deblurring with non-uniform blur.

***Index Terms*—** Plug & Play, Image resoration, Deblurring, Optimization

1. INTRODUCTION

Many image restoration tasks can be formulated as inverse problems:

$$y = Hx + \varepsilon \tag{1}$$

with $y \in \mathbb{R}^p$ the degraded image, $x \in \mathbb{R}^n$ the unknown clean image, $H \in \mathbb{R}^{p*n}$ the degradation matrix and ε is the measurement noise. Such tasks include denoising, deblurring, super-resolution, compressed sensing and so on. The reconstructed image x can be obtained by maximizing the posterior $p(x|y) \propto p(y|x)p(x)$. Equivalently the posterior maximization or MAP estimator can be expressed as

$$x_{MAP} = \arg\min_x \underbrace{h(Hx) + \lambda f(x)}_{E(x)} \tag{2}$$

where $h(x) = -\log(p(y|x))$ is known as the data fitting term or negative log-likelihood and $\lambda f(x) = -\log(p(x))$ is the regularization term or negative log-prior. Classical approaches used convex regularization terms such as Tikhonov [1, 2], Total-Variation [3, 4] or wavelet-ℓ_1 [5] for example. More recently, [6] introduced *Plug & Play (PnP)* algorithms that enable the use of pretrained neural networks as implicit regularizers. PnP algorithms use a proximal splitting algorithm to solve the optimisation problem (2), and then substitute the regularization subproblem by a pretrained denoiser. The focus of this work is a variation of the alternating direction method of multipliers (ADMM) algorithm [7], but the same idea has been extended to other splitting schemes including primal-dual [8, 9], fast iterative shrinkage [10], and gradient descent [11, 12].

Proximal-based PnP algorithms like ADMM or Primal-Dual involve the computation at each iteration of the proximal operator of the data fitting term

$$\operatorname{prox}_{\alpha h(H\cdot)}(x) = \arg\min_z \frac{1}{2\alpha}\|x - z\|_2^2 + h(Hz). \tag{3}$$

This computation admits a fast closed form solution for many inverse problems like super-resolution [13] or deconvolution [14]. For more complex tasks like deblurring with spatially-varying blur for example, the exact solution of (3) is computationally intractable, and even approximate solutions can be computationally expensive.
A common solution in such cases is to use ISTA [15], RED [11] or SGD [12] schemes where the more computationally friendly gradient $\nabla h(H\cdot)$ is computed instead of the intractable proximal operator $\operatorname{prox}_{\alpha h(H\cdot)}$. Nevertheless this solution employing the gradient is not ideal because PnP-ADMM usually converges in far fewer iterations and is more robust to initial conditions than its gradient-based counterparts [16].

We propose in Section 2 a linearized version of PnP-ADMM which preserves the benefits of PnP-ADMM while avoiding the costly proximal computation. We also show that the proposed method converges to a critical point of $E(x) = h(Hx) + \lambda f(x)$ under less restrictive conditions than in previous works on PnP-ADMM, which require the denoiser residual to be Lipschitz continuous [17, 18], and impose constraints on the regularization parameter λ [17]. Such constraints on λ mean that we need to choose between convergence guarantees and optimal regularization. As for the denoiser constraints, several techniques exist to train a Lipschitz denoiser, but at the cost of degraded denoising performance [18]. The Linearized PnP-ADMM that we introduce in the next section does not require the denoiser to be Lipschitz continuous nor does it impose any constraints on λ.

[1] Work mostly done while Matias Tassano was at GoPro France.

[2] We thank Pauline Tan, Arthur Leclaire, and Samuel Hurault for fruitful discussions. This work has been partially funded by the French National Research and Technology Agency (ANRT) and GoPro France.

2. MODEL

In this section, we introduce our Plug & Play linearized-ADMM algorithm (PnP LADMM). We first describe the main difference between ADMM and linearized-ADMM before discussing the convergence of linearized-ADMM in the case of Plug & Play.

2.1. Linearized-ADMM (LADMM)

In order to solve MAP estimation problems like (2) ADMM starts from the augmented Lagrangian

$$\begin{aligned}\mathcal{L}_\beta(x,z,w) =& h(z) + \lambda f(x) + \langle w, Hx - z\rangle \\ &+ \frac{\beta}{2}\|Hx - z\|^2. \end{aligned} \tag{4}$$

Note that in our case we used the splitting variable $Hx = z$, instead of the more common choice $x = z$ [17, 18], which leads to the potentially expensive computation of $\mathrm{prox}_{\alpha h(H\cdot)}$. ADMM is based on a alternate minimization on the three variables of the Lagrangian (4), namely

$$x_{k+1} = \arg\min_x \mathcal{L}_\beta(x, z_k, w_k) \tag{5}$$

$$z_{k+1} = \arg\min_z \mathcal{L}_\beta(x_{k+1}, z, w_k) \tag{6}$$

$$w_{k+1} = w_k + \beta(Hx_{k+1} - z_{k+1}). \tag{7}$$

Now the z-update only requires the simpler computation of $\mathrm{prox}_{\alpha h}$, but the x-update is intractable because it involves both f and H. The main idea of linearized-ADMM is to replace the minimization of the Lagrangian in the x-update by the minimization of an approximate or "linearized" Lagrangian where the quadratic term $\frac{\beta}{2}\|z - Hx\|^2$ is replaced by an isotropic majorizer with curvature $L_x \geq \beta\|H\|^2$:

$$\begin{aligned}\tilde{\mathcal{L}}^k_\beta(x,z,w) =& h(z) + \lambda f(x) + \langle w, Hx - z\rangle + \frac{L_x}{2}\|x - x_k\|_2^2 \\ &+ \frac{\beta}{2}\langle x - x_k, 2H^T(Hx_k - z)\rangle. \end{aligned} \tag{8}$$

Using this notation, we can express linearized-ADMM as:

$$x_{k+1} = \arg\min_x \tilde{\mathcal{L}}^k_\beta(x, z_k, w_k) \tag{9}$$

$$z_{k+1} = \arg\min_z \mathcal{L}_\beta(x_{k+1}, z, w_k) \tag{10}$$

$$w_{k+1} = w_k + \beta(Hx_{k+1} - z_{k+1}). \tag{11}$$

2.2. Convergence

Despite the approximation we can show that LADMM converges to the expected critical point under mild assumptions.

Assumption 1.
- *$h(z) + \lambda f(x)$ is lower bounded on the set $\{(z,x) \in (\mathbb{R}^{n*p})^2 | z = Hx\}$.*
- *h is strongly convex and L_h-Lipschitz differentiable*

Theorem 1. *Under Assumption 1, for linearized-ADMM with hyper parameters such that:*

$$\beta \geq L_h \tag{12}$$

$$L_x \geq \beta\|H\|^2 \tag{13}$$

then the sequence $\{\mathcal{L}_\beta(x_k, z_k, w_k)\}$ is convergent and the primal residues $\|x_{k+1} - x_k\|$, $\|z_{k+1} - z_k\|$ and the dual residue $\|w_{k+1} - w_k\|$ converge to 0 as k approaches infinity. We also have that the sequence (x_k, z_k, w_k) satisfies

$$\lim_{k\to\infty} \nabla_w \mathcal{L}_\beta(x_k, z_k, w_k) = \lim_{k\to\infty} \nabla_z \mathcal{L}_\beta(x_k, z_k, w_k) = 0 \tag{14}$$

and that there exists

$$d^k \in \partial_x \mathcal{L}_\beta(x_k, z_k, w_k) \quad \textit{s.t} \quad \lim_{k\to\infty} d^k = 0. \tag{15}$$

If in addition f is differentiable then $\lim_{k\to\infty} \nabla E(x_k) = 0$.[3]

2.3. Plug & Play linearized-ADMM (PnP-LADMM)

Using the change of variable $u_k = \frac{w_k}{\beta}$ and re-aranging the terms in the optimization steps from equation (9-11), we can obtain the proximal version of linearized ADMM:

$$x_{k+1} = \mathrm{prox}_{\frac{\lambda}{L_x} f}(x_k - \frac{\beta}{L_x} H^T(Hx_k - z_k + u_k)) \tag{16}$$

$$z_{k+1} = \mathrm{prox}_{\frac{1}{\beta} h}(Hx_{k+1} + u_k) \tag{17}$$

$$u_{k+1} = u_k + (Hx_{k+1} - z_{k+1}). \tag{18}$$

The proximal operator in (16) can be seen as a denoising problem with regularization function f and noise level $\sigma_d^2 = \frac{\lambda}{L_x}$. In the spirit of Plug & Play approaches, this proximal operator can be replaced by an off-the-shelf denoiser $\mathcal{D}_{\sigma_d}$ (see Algorithm 1). In comparison to PnP-ADMM [17, 18] which requires the denoiser residual $\mathcal{D}_{\sigma_d} - Id$ to be non-expansive to ensure convergence, the proposed PnP-LADMM converges for a larger family of denoisers:

Proposition 1. *If $\mathcal{D}_{\sigma_d}$ is any MMSE denoiser or the Proximal Gradient Step denoiser in [18], then there exists a lower bounded function f such that $\mathcal{D}_{\sigma_d} = \mathrm{prox}_{\sigma_d^2 f}$.*[3]

As a consequence Theorem 1 ensures convergence of PnP-LADMM for the gradient step denoiser or any MMSE denoiser. So we can use any state-of-the-art denoising architecture trained with quadratic loss for $\mathcal{D}_\sigma$. In practice, we adopt the widely used DRUNet denoiser that was introduced in [20]. The computational efficiency of the method relies both on the use of the splitting variable $Hx = z$ and the linearization. Using the splitting variable $Hx = z$ leads to a z-update that is very easy to compute since it corresponds

[3] A proof of Theorem 1 (adapted from [19]) and Proposition 1 is provided in Appendix

Algorithm 1 PnP Linearized ADMM algorithm
Solves $x = \arg\min_x h(Hx) + \lambda f(x)$

```
Require: $x_0$, $z_0$, $u_0$, $\beta$, $L_x$, $\mathcal{D}_{\sigma_d} = \text{prox}_{\sigma_d^2 f}$
  for $k \in [0, N-1]$ do
    $x_{k+1} = \mathcal{D}_{\sqrt{\frac{\lambda}{L_x}}}(x_k - \frac{\beta}{L_x} H^T(Hx_k - z_k + u_k))$
    $z_{k+1} = \frac{y + \sigma^2 \beta (Hx_{k+1} + u_k)}{1 + \beta\sigma^2}$
    $u_{k+1} = u_k + \beta(z_{k+1} - Hx_{k+1})$
```

to the proximal operator of a quadratic norm which does not involve the degradation operator H. On the other hand, the linearization leads to an x-update that bypasses the inversion of the degradation operator H. Our PnP-LADMM algorithm only requires that we can efficiently compute the quantities Hx and $H^T x$ at each iteration. The forward and adjoint of the degradation operator can be efficiently computed for a wide diversity of tasks such as super-resolution, spatially-varying blur, inpainting, compressed sensing, etc. The whole iterative process with the closed form formulation is summarized in Algorithm 1.

PnP-LADMM has 4 different hyperparameters, λ, σ_d, β and L_x. The parameters λ and σ_d are model parameters of the MAP estimator, they will be responsible for the quality of the output and control the balance between data fidelity and regularization. On the other hand, β and L_x are parameters of the optimization algorithm, they control the convergence speed. Since these 4 parameters are linked to each other via the constraint $\sigma_d = \sqrt{\lambda / L_x}$, there are only 3 degrees of freedom for our algorithm. For a Gaussian data fitting term, we have $h(x) = \frac{1}{2\sigma^2}\|x - y\|_2^2$ so $L_h = 1/\sigma^2$. The condition of Theorem 1 implies that:

$$L_x \geq \beta \|H\|^2 \geq L_h \|H\|^2 \tag{19}$$

$$\Leftrightarrow \lambda \geq \sigma_d^2 \beta \|H\|^2 \geq \frac{\sigma_d^2}{\sigma^2}\|H\|^2 \tag{20}$$

This means that we can choose any non-negative regularization parameter $\lambda > 0$ as long as we decrease σ_d accordingly for very small values of λ.

3. EXPERIMENTS

In this section, we evaluate the performance of our approach on deblurring images with spatially-varying blur. All the code used in our experiments can be found in the GitHub page of the project.

3.1. Datasets

We test our approach on deblurring images with non-uniform blur. Non-blind deblurring algorithms usually suppose the blur to be uniform since it leads to easier computations both for the generation of synthetic data and for the deblurring. However, the uniform blur assumptions does not hold for many real-world applications, such as motion blur or defocus blur. To highlight the performance of our algorithm on deblurring spatially-varying blur, we degrade our images using the dataset introduced in [21] which uses the O'Leary [22] model. In particular, we suppose that the blur H in the inverse problem (1) is decomposed as a linear combination

$$H = \sum_{i=1}^{P} U_i K_i,\ \varepsilon \sim \mathcal{N}(0, \sigma^2) \tag{21}$$

of uniform blur (convolution) operators K_i with spatially varying mixing coefficients, *i.e.* diagonal matrices U_i such that $\sum_{i=1}^{P} U_i = Id$, $U_i \geq 0$. Please note that even with this decomposition, the proximal operator $\text{prox}_{h(H.)}$ from Equation (3) cannot be easily and efficiently computed. The advantage of the O'Leary model is that its forward and transpose operators can be computed very efficiently using convolution and masking operations. Also, this formulation can model a large diversity of spatially varying blurs. In our experiments, we apply this degradation process on the COCO dataset [23], using the segmentation masks as the U_i's and we build random Gaussian and motion blur kernels for the K_i's. Figure 1 shows the low-resolution obtained.

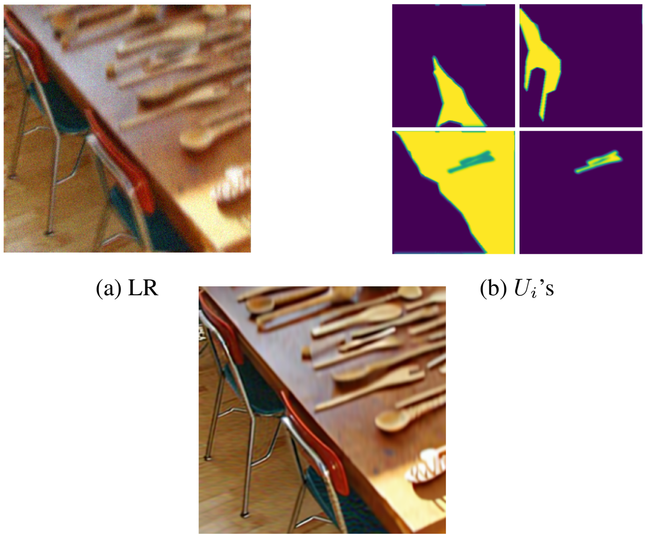
(a) LR
(b) U_i's
(c) Deblurred

Fig. 1: Example of sample from the testset with restoration result.

3.2. Compared methods

We compare our approach to the Richardson-Lucy algorithm [24, 25], Plug & Play ADMM with splitting variable $x = z$ [17] where the proximal operator is approximated using conjugate gradient algorithm (PnP-ADMM + CG) and

Model	Runtime	σ	Metrics
			(PSNR↑ , SSIM↑ , LPIPS↓)
Richardson-Lucy	10sec	1	(23.4, 0.74, 0.27)
		10	(20.9, 0.43, 0.55)
		20	(18.8, 0.25, 0.64)
		40	(15.4, 0.13, 0.72)
PnP-ISTA	247sec	1	(23.4, 0.71, 0.34)
		10	(23.3, 0.71, 0.33)
		20	(22.7, **0.67**, 0.38)
		40	(21.7, **0.61**, 0.43)
PnP-ADMM + CG	286sec	1	(**25.8**, **0.82**, 0.26)
		10	(**23.7**, **0.72**, **0.32**)
		20	(**22.9**, **0.67**, **0.37**)
		40	(**21.7**, 0.60, **0.43**)
PnP-LADMM	124sec	1	(25.6, 0.81, **0.22**)
		10	(**23.7**, **0.72**, **0.32**)
		20	(22.8, 0.66, 0.38)
		40	(**21.7**, **0.61**, **0.43**)

Table 1: Performance of the different models, PnP-ADMM + CG refers to PnP-ADMM where the proximal operator of the data term is computed using conjugate gradient algorithm. Best results are in **bold**.

Plug & Play ISTA [26] (PnP-ISTA). These algorithms are common for deblurring with spatially varying kernels; see [27] for PnP-ADMM+CG and [28] for Richardson-Lucy. For our experiments, we use the same DRUNet denoiser pretrained on Gaussian noise with ℓ^2 loss for all the methods for a fair comparison. This choice ensures that the denoiser is a good approximation of an MMSE estimator, which is sufficient to guarantee convergence for our PnP-LADMM algorithm, and for PnP-ISTA as well [15]. In order to ensure convergence of PnP-ADMM we could have used a modified denoiser with non-expansive residual, but this constraint degrades its performance and it is most often not necessary to ensure convergence in practice. We observed that all methods do not behave the same for similar denoiser noise level σ_d and regularization parameter λ. Following this observation, we decided to separately tune these parameters for each method and dataset noise level to find their respective optima in terms of restoration quality on a small training dataset. Results are then evaluated on a different test dataset of 40 images with the optimal parameters found on the training set.

3.3. Results

We evaluate the performance of the models using classical metrics such as PSNR, SSIM and LPIPS. The overall performance results are summarized in Table 1. In terms of performances, the baseline Richardson-Lucy fails to deblur the noisy images. PnP-LADMM and PnP-ADMM + CG have similar performances with a maximum difference of 0.3dB in favor of the latter. PnP-ISTA has slightly lower performances than the two ADMM based algorithms. It seems that the margin is decreasing when the noise level increases. In terms of runtime, similar observation are found in Table 1 and Figure 2. Richardson-Lucy quickly reaches its optimum since it does not involve expensive operations. PnP-ISTA is twice slower than PnP-LADMM. As discussed in the introduction, gradient-based methods are often slow to converge. In particular here, PnP-ISTA requires around 200 iterations to converge in comparison to 100 for PnP-LADMM. Since one PnP-ISTA iteration takes as much time as one PnP-LADMM iteration (we apply one denoiser forward, one $H^T x$ and one Hx operation), it results in an algorithm that is slower. PnP-ADMM + CG is the slowest algorithm despite needing only 40 iterations to converge. This slowness is due to the fact that the conjugate gradient is computed at each step in order to approximate the proximal operator. Finally, PnP-LADMM is the fastest PnP algorithm. In fact, even though the linearization is causing the algorithm to converge in more iterations, the benefit of bypassing the computation of the proximal operator is greater resulting in a faster algorithm. These results highlight the fact that for complex degradation operators, PnP-LADMM achieves the best performance/ratio trade-off.

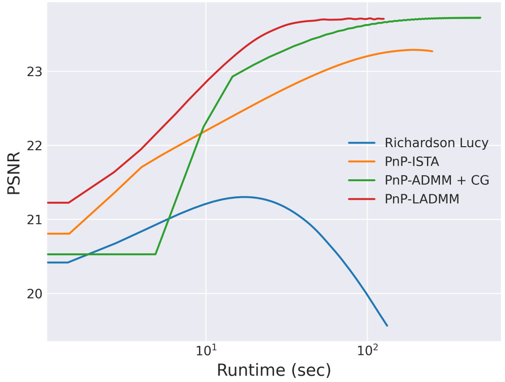

Fig. 2: Convergence speed of the different methods, we use 40 images with spatially-varying blur and Gaussian noise with $\sigma = 10/255$.

4. CONCLUSION

In this article, we present a novel Plug & Play approach based on linearized-ADMM to solve inverse problems with complex degradation operators such as non-uniform blur. The linearized version of ADMM allows to bypass the computation of intractable proximal operators. We demonstrate the efficiency of our method on the problem of deblurring images with O'Leary spatially-varying blur. We found that PnP-LADMM obtains the best performance/runtime trade-off compared to the gradient-based method PnP-ISTA or PnP-ADMM combined with conjugate gradient to compute the proximal operator. In addition, the proposed algorithm provides convergence guarantees under less restrictive conditions than previous PnP-ADMM results.

5. REFERENCES

[1] Peyman Milanfar, Ed., *Super-Resolution Imaging*, CRC Press, dec 2011.

[2] Michal Šorel, Filip Šroubek, and Jan Flusser, "Towards Super-Resolution in the Presence of Spatially Varying Blur," in *Super-Resolution Imaging*, chapter 7, pp. 187–218. CRC Press, dec 2017.

[3] F Malgouyres and F Guichard, "Edge Direction Preserving Image Zooming: A Mathematical and Numerical Analysis," *SIAM Journal on Numerical Analysis*, vol. 39, no. 1, pp. 1–37, jan 2001.

[4] A Almansa, S Durand, and B Rougé, "Measuring and Improving Image Resolution by Adaptation of the Reciprocal Cell," *JMIV*, vol. 21, no. 3, pp. 235–279, nov 2004.

[5] P Escande, P Weiss, and F Malgouyres, "Image restoration using sparse approximations of spatially varying blur operators in the wavelet domain," *Journal of Physics: Conference Series*, vol. 464, no. 1, oct 2013.

[6] S.V. Venkatakrishnan, C.A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," *IEEE GlobalSIP*, 2013.

[7] Neal Parikh and Stephen Boyd, "Proximal Algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[8] F Heide, M Steinberger, Y Tsai, M Rouf, D Pajak, D Reddy, O Gallo, J Liu, W Heidrich, K Egiazarian, J Kautz, and K Pulli, "FlexISP: A Flexible Camera Image Processing Framework," *ACM ToG*, vol. 33, 2014.

[9] T Meinhardt, M Moeller, C Hazirbas, and D Cremers, "Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems," in *ICCV*. oct 2017, pp. 1799–1808, IEEE.

[10] U Kamilov, H Mansour, and B Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Processing Letters*, 2017.

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

[12] R Laumont, V de Bortoli, A Almansa, J Delon, A Durmus, and M Pereyra, "On Maximum-a-Posteriori estimation with Plug & Play priors and stochastic gradient descent," *arXiv:2201.06133*, jan 2022.

[13] N Zhao, Q Wei, A Basarab, N Dobigeon, D Kouamé, and J Tourneret, "Fast Single Image Super-Resolution Using a New Analytical Solution for l2–l2 Problems," *IEEE TIP*, vol. 25, no. 8, pp. 3683–3697, Aug. 2016.

[14] S.H. Chan, X. Wang, and O.A. Elgendy, "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications," *IEEE TCI*, 2017.

[15] Xiaojian Xu, Yu Sun, Jiaming Liu, Brendt Wohlberg, and Ulugbek S. Kamilov, "Provable convergence of plug-and-play priors with mmse denoisers," *IEEE Signal Processing Letters*, vol. 27, pp. 1280–1284, 2020.

[16] R Ahmad, C Bouman, G Buzzard, S Chan, S Liu, E Reehorst, and P Schniter, "Plug-and-Play Methods for Magnetic Resonance Imaging: Using Denoisers for Image Recovery," *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 105–116, jan 2020.

[17] E Ryu, J Liu, S Wang, X Chen, Z Wang, and W Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *36th ICML*, 09–15 Jun 2019, vol. 97, pp. 5546–5557.

[18] S Hurault, A Leclaire, and N Papadakis, "Proximal Denoiser for Convergent Plug-and-Play Optimization with Nonconvex Regularization," in *ICML*, jul 2022, `https://proceedings.mlr.press/v162/hurault22a.html`.

[19] Q Liu, X Shen, and Y Gu, "Linearized ADMM for Nonconvex Nonsmooth Optimization With Convergence Analysis," *IEEE Access*, vol. 7, pp. 76131–76144, 2019.

[20] K Zhang, Y Li, W Zuo, L Zhang, L Van Gool, and R Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE TPAMI*, 2021.

[21] C Laroche, A Almansa, and M Tassano, "Deep Model-Based Super-Resolution with Non-uniform Blur," in *WACV 2023*, jan 2023, to appear, arXiv:2204.10109.

[22] J Nagy and D O'Leary, "Restoring images degraded by spatially variant blur," *SIAM Journal on Scientific Computing*, vol. 19, no. 4, pp. 1063–1082, 1998.

[23] T Lin, M Maire, S Belongie, L Bourdev, R Girshick, J Hays, P Perona, D Ramanan, L Zitnick, and P Dollár, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2015.

[24] W H Richardson, "Bayesian-based iterative method of image restoration∗," *J. Opt. Soc. Am.*, vol. 62, no. 1, pp. 55–59, Jan 1972.

[25] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astronomical Journal*, vol. 79, pp. 745, June 1974.

[26] R Gavaskar and K Chaudhury, "Plug-and-play ista converges with kernel denoisers," *IEEE Signal Processing Letters*, vol. 27, pp. 610–614, 2020.

[27] Valentin Debarnot and Pierre Weiss, “Deep-blur : Blind identification and deblurring with convolutional neural networks,” hal-03687822, 2022.

[28] Guillermo Carbajal, Patricia Vitoria, Mauricio Delbracio, Pablo Musé, and José Lezama, “Non-uniform blur kernel estimation via adaptive basis decomposition,” *arXiv:2102.01026*, 2021.

[29] R Gribonval, “Should penalized least squares regression be interpreted as maximum a posteriori estimation?,” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2405–2410, 2011.

A. CONVERGENCE OF LINEARIZED-ADMM

In this section, we prove Theorem 1 and Proposition 1. The proof of Theorem 1 is an adaptation of a similar result in [19]. The main difference is that in [19] the ADMM is linearized in both proximal descent steps, whereas in our case we are interested in linearizing only one of them.

In the sequel, we suppose (without loss of generality) that $\lambda = 1$ in our MAP estimator defined in (2).

Lemma 2. *Under Assumption 1, the following inequality holds for the x-update:*

$$\mathcal{L}_\beta(x_k, z_k, w_k) - \mathcal{L}_\beta(x_{k+1}, z_k, w_k) \geq \frac{L_x - \beta\|H\|^2}{2}\|x_k - x_{k+1}\|^2$$

with $\|H\|^2$ the largest singular value of $H^T H$.

Proof. Using the notation of equation (8), we define $\overline{f}^k(x) = \tilde{\mathcal{L}}_\beta^k(x, z_k, w_k)$ and by definition of the x-update, we have:

$$\begin{aligned} &\overline{f}^k(x_k) \geq \overline{f}^k(x_{k+1}) && (22)\\ \Leftrightarrow &\langle x_k - x_{k+1}, H^T w_k + \beta H^T(Hx_k - z_k)\rangle \\ &+ f(x_k) - f(x_{k+1}) \geq \frac{L_x}{2}\|x_{k+1} - x_k\|^2 && (23)\end{aligned}$$

We also have that:

$$\begin{aligned} &\mathcal{L}_\beta(x_k, z_k, w_k) - \mathcal{L}_\beta(x_{k+1}, z_k, w_k) \\ &= f(x_k) - f(x_{k+1}) + \langle w_k, H(x_k - x_{k+1})\rangle \\ &+ \frac{\beta}{2}\|Hx_k - z_k\|^2 - \frac{\beta}{2}\|Hx_{k+1} - z_k\|^2 && (24)\\ &= f(x_k) - f(x_{k+1}) - \frac{\beta}{2}\|H(x_{k+1} - x_k)\|^2 \\ &+ \langle x_k - x_{k+1}, H^T w_k + \beta H^T(Hx_k - z_k)\rangle && (25)\\ &\geq \frac{L_x}{2}\|x_{k+1} - x_k\|^2 - \frac{\beta}{2}\|H(x_{k+1} - x_k)\|^2 && (26)\\ &\geq \frac{L_x - \beta\|H\|^2}{2}\|x_{k+1} - x_k\|^2 && (27)\end{aligned}$$

where the inequality (26) is obtained using (23). □

Lemma 3. $\mathcal{L}_\beta(x_{k+1}, z_k, w_k) - \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_k) \geq m\|z_k - z_{k+1}\|^2$

Proof. From Assumption 1, we have that $\mathcal{L}_\beta$ is strongly convex in z with parameter m. The strong convexity of $\mathcal{L}_\beta$ implies that:

$$\begin{aligned} &\mathcal{L}_\beta(x, z_k, w) - \mathcal{L}_\beta(x, z_{k+1}, w) && (28)\\ &\geq \nabla_z \mathcal{L}_\beta(x, z_{k+1}, w)(z_k - z_{k+1}) + m\|z_k - z_{k+1}\|^2 && (29)\end{aligned}$$

However, the z-update of Algorithm 1 is such that

$$\nabla_z \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_k) = 0 \tag{30}$$

which leads to the results. □

Lemma 4. *Under Assumption 1, the following equality holds:*

$$w_k = \nabla_z h(z_k) \tag{31}$$

Proof. From the definition of the Lagrangian:

$$\nabla_z \mathcal{L}_\beta(x, z, w) = \nabla h(z) - w - \beta(Hx - z)$$

Using the fact that

$$w_{k+1} = w_k + \beta(Hx_{k+1} - z_{k+1}) \quad \text{and} \quad \nabla_z \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_k) = 0 \tag{32}$$

We have:

$$\begin{aligned} 0 &= \nabla h(z_{k+1}) - w_k - \beta(Hx_{k+1} - z_{k+1}) \qquad (33) \\ \Leftrightarrow & \nabla h(z_{k+1}) = w_{k+1} \qquad (34) \end{aligned}$$

□

Lemma 5. *Under assumption 1,*

$$\begin{aligned} &\mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_{k+1}) - \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_k) \qquad (35) \\ &= \frac{1}{\beta}\|w_{k+1} - w_k\|^2 \le C_1 \|z_{k+1} - z_k\|^2 \qquad (36) \end{aligned}$$

with $C_1 = L_h^2/\beta$.

Proof. By definition of the augmented Lagrangian we have that:

$$\begin{aligned} \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_{k+1}) &- \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_k) \\ &= \langle w_{k+1} - w_k, Hx_{k+1} - z_{k+1} \rangle \\ &= \frac{1}{\beta}\|w_{k+1} - w_k\|^2 \\ &= \frac{1}{\beta}\|\nabla_z h(z_{k+1}) - \nabla_z h(z_k)\|^2 \quad \text{from Lemma 4} \\ &\le \frac{L_h^2}{\beta}\|z_{k+1} - z_k\|^2 \;\; \text{from Assumption 1} \end{aligned}$$

□

Lemma 6. *If g is L_g-Lipschitz differentiable then:*

$$g(y_2) - g(y_1) \ge \nabla g(s)(y_2 - y_1) - \frac{L_g}{2}\|y_2 - y_1\|^2 \tag{37}$$

where s denotes y_1 or y_2

Proof.

$$\begin{aligned} g(y_2) - g(y_1) &= \int_0^1 \nabla g(ty_2 + (1-t)y_1) \cdot (y_2 - y_1)\mathrm{d}t \qquad (38) \\ = \int_0^1 \nabla g(s) \cdot (y_2 - y_1)\mathrm{d}t &+ \int_0^1 (\nabla g(y_2 + (1-t)y_1) - \nabla g(s)) \cdot (y_2 - y_1)\mathrm{d}t, \qquad (39) \end{aligned}$$

where $\nabla g(\cdot)$ defines the gradient of $g(\cdot)$. If we take $s = y_1$, then by inequality

$$\|\nabla g(ty_2 + (1-t)y_1) - \nabla g(y_1)\| \le L_g \|t(y_2 - y_1)\| \tag{40}$$

we have

$$\begin{aligned} &\int_0^1 \nabla g(y_1) \cdot (y_2 - y_1)\mathrm{d}t + \int_0^1 (\nabla g(ty_2 + (1-t)y_1) - \nabla g(y_1)) \cdot (y_2 - y_1)\mathrm{d}t \qquad (41) \\ \ge & \nabla g(y_1) \cdot (y_2 - y_1) - \int_0^1 L_g t \|y_2 - y_1\|^2 \, \mathrm{d}t \qquad (42) \\ = & \nabla g(y_1) \cdot (y_2 - y_1) - \frac{L_g}{2}\|y_2 - y_1\|^2. \qquad (43) \end{aligned}$$

Therefore, we get

$$g(y_2) - g(y_1) \geq \nabla g(y_1) \cdot (y_2 - y_1) - \frac{L_g}{2}\|y_2 - y_1\|^2. \tag{44}$$

Similarly, if we take $s = y_2$, we can get

$$g(y_2) - g(y_1) \geq \nabla g(y_2) \cdot (y_2 - y_1) - \frac{L_g}{2}\|y_2 - y_1\|^2. \tag{45}$$

□

Lemma 7. *Under Assumption 1, if we choose the hyper-parameters β and L_x satisfying (12) and* (13)*, then the sequence $\{m_k\}$ defined by*

$$m_k = \mathcal{L}_\beta(x_k, z_k, w_k) \tag{46}$$

is convergent.

Proof. 1) **Monotonicity:** By using Lemma 2, Lemma 3 and Lemma 5 we have:

$$\begin{aligned}
m_k - m_{k+1} &= \mathcal{L}_\beta(x_k, z_k, w_k) - \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_{k+1}) && (47)\\
&\geq \mathcal{L}_\beta(x_{k+1}, z_k, w_k) - \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_{k+1}) && (48)\\
&\quad + \frac{L_x - \beta\|H\|^2}{2}\|x_k - x_{k+1}\|^2 && (49)\\
&\geq \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_k) - \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_{k+1}) && (50)\\
&\quad + \frac{L_x - \beta\|H\|^2}{2}\|x_k - x_{k+1}\|^2 + m\|z_k - z_{k+1}\|^2 && (51)\\
&\geq \frac{L_x - \beta\|H\|^2}{2}\|x_k - x_{k+1}\|^2 + (m + \frac{L_h^2}{\beta})\|z_k - z_{k+1}\|^2 && (52)
\end{aligned}$$

Since we chose L_x such that:

$$\begin{aligned}
&L_x \geq \beta\|H\|^2 && (53)\\
\Leftrightarrow\quad &\frac{L_x - \beta\|H\|^2}{2} > 0 && (54)
\end{aligned}$$

we obtain the monotonocity of $\{m_k\}$.

2) **Lower bound**:

$$m_k = h(z_k) + f(x_k) + \langle w_k, Hx_k - z_k\rangle + \frac{\beta}{2}\|Hx_k - z_k\|^2 \tag{55}$$

Let $z'_k = Hx_k$, from Lemma 4 we have:

$$\begin{aligned}
\langle w_k, Hx_k - z_k\rangle &= \langle w_k, z'_k - z_k\rangle && (56)\\
&= \langle \nabla h(z_k), z'_k - z_k\rangle && (57)
\end{aligned}$$

so we can rewrite:

$$\begin{aligned}
m_k &= h(z_k) + f(x_k) + \langle \nabla h(z_k), z'_k - z_k\rangle + \frac{\beta}{2}\|z'_k - z_k\|^2 && (58)\\
& && (59)
\end{aligned}$$

We chose β such that $\beta \geq L_h$ so:

$$\begin{aligned}
m_k &\geq h(z_k) + f(x_k) - \langle \nabla h(z_k), z_k - z'_k\rangle + \frac{L_h}{2}\|z_k - z'_k\|^2 && (60)\\
&\geq h(z'_k) + f(x_k) \;\text{ from Lemma 6.} && (61)
\end{aligned}$$

Following Assumption 1, $h(z'_k) + f(x_k)$ is lower bounded so m_k is lower bounded. m_k is monotonically decreasing and lower bounded which ensures the convergence. □

Lemma 8. *Suppose we have a differentiable function f_1, a possibly non differentiable function f_2, and a point x. If there exist $d_2 \in \partial f_2(x)$, then we have:*

$$d = d_2 - \nabla f_1(x) \in \partial(f_2(x) - f_1(x))$$

Proof. From the subgradient definition we have that:

$$f_2(y) \geq f_2(x) + \langle d_2, y - x \rangle + o(\|y - x\|) \tag{62}$$

From the fact that f_1 is differentiable we have that:

$$-f_1(y) = -f_1(x) - \langle \nabla f_1(x), y - x \rangle + o(\|y - x\|) \tag{63}$$

Combining the two leads to:

$$f_2(y) - f_1(y) \geq f_2(x) - f_1(x) + \langle d_2 - \nabla f_1(x), y - x \rangle + o(\|y - x\|) \tag{64}$$

□

Proof of Theorem 1: We divide the proof in three parts:

a) Convergence of the residuals:

From Lemma 7 and its proof we have that:

$$m_{k+1} - m_k \geq a\|x_k - x_{k+1}\|^2 + \left(m + \frac{L_h^2}{\beta}\right)\|z_{k-1} - z_k\|^2 \geq 0 \tag{65}$$

with $(m + \frac{L_h^2}{\beta}) > 0$, $a = \frac{L_x - \beta\|H\|^2}{2} > 0$ (according to Assumption 1) and that m_k converges. This implies that $\|x_k - x_{k+1}\|^2$ and $\|y_k - y_{k+1}\|^2$ converge to 0 as k approaches infinity. Lemma 5 ensure the convergence of $\|w_k - w_{k+1}\|^2$ to 0. The convergence of m_k directly implies the convergence of $\mathcal{L}_\beta(x_k, z_k, w_k)$.

b) Convergence of the gradients:

For the convergence of $\lim_{k\to\infty} \nabla_w \mathcal{L}_\beta(x_k, z_k, w_k)$, we have that:

$$\lim_{k\to\infty} \nabla_w \mathcal{L}_\beta(x_k, z_k, w_k) = \lim_{k\to\infty} Hx_k - z_k = \lim_{k\to\infty} \frac{1}{\beta}(w_{k+1} - w_k) = 0. \tag{66}$$

On the other side, we have using Lemma 4 that:

$$\nabla_z \mathcal{L}_\beta(x_k, z_k, w_k) = \nabla h(z_k) - w_k - \beta(Hx_k - z_k) \tag{67}$$

$$= w_k - w_k - (w_{k+1} - w_k) = -(w_{k+1} - w_k) \to 0 \tag{68}$$

Finally, we want to show that there exists

$$d^k \in \partial_x \mathcal{L}_\beta(x_k, z_k, w_k) \quad \text{s.t} \quad \lim_{k\to\infty} d^k = 0. \tag{69}$$

Since x^{k+1} is the minimum point of $\tilde{\mathcal{L}}_\beta^k(x, z_k, w_k)$, we have that $0 \in \partial\tilde{\mathcal{L}}_\beta^k(x, z_k, w_k)$. Using Lemma 8 and the definition of $\tilde{\mathcal{L}}_\beta^k$ we have:

$$\exists d_{k+1} \in \partial f(x_{k+1}) \tag{70}$$

$$s.t \quad H^T w_k + L_x(x_{k+1} - x_k) + \beta H^T(Hx_k - z_k) + d_{k+1} = 0 \tag{71}$$

Lets us define:

$$\tilde{d}_{k+1} = H^T w_{k+1} + \beta H^T(Hx_{k+1} - z_{k+1}) + d_{k+1} \tag{72}$$

we can easily verify that $\tilde{d}_{k+1} \in \partial_x \mathcal{L}_\beta(x_{k+1}, z_{k+1}, w_{k+1})$.

We already showed that the primal residues $\|x_{k+1} - x_k\|$, $\|z_{k+1} - z_k\|$, $\|w_{k+1} - w_k\|$ converge to 0 as k approaches infinity, therefore:

$$\lim_{k\to\infty} \tilde{d}_{k+1} = \lim_{k\to\infty} H^T w_{k+1} + \beta H^T(Hx_{k+1} - z_{k+1}) + d_{k+1} \tag{73}$$

$$= \lim_{k\to\infty} H^T w_k + L_x(x_{k+1} - x_k) + \beta H^T(Hx_k - z_k) + d_{k+1} = 0 \tag{74}$$

where the last equality is obtained using 71.

c) Convergence to a critical point of $h(H\cdot) + \lambda f(\cdot)$:
Since we are optimizing

$$E(x) = h(Hx) + f(x) \tag{75}$$

we would like to show that

$$\lim_{k\to\infty} \nabla E(x_k) = 0 \tag{76}$$

Indeed, from the chain rule

$$\nabla E(x_k) = H^* \nabla h(Hx) + \nabla f(x). \tag{77}$$

From part b of this proof we have that

$$\lim_{k\to\infty} \nabla_w \mathcal{L}_\beta(x_k, z_k, w_k) = \lim_{k\to\infty} z_k - Hx_k \qquad = 0 \tag{78}$$

$$\lim_{k\to\infty} \nabla_z \mathcal{L}_\beta(x_k, z_k, w_k) = \lim_{k\to\infty} w_k + \beta(z_k - Hx_k) + \nabla h(z_k) \qquad = 0 \tag{79}$$

$$\lim_{k\to\infty} \nabla_x \mathcal{L}_\beta(x_k, z_k, w_k) = \lim_{k\to\infty} \nabla f(x_k) - H^* w_k + \beta H^*(Hx_k - z_k) \qquad = 0 \tag{80}$$

where in the last equation we used the additional hypothesis that f is differentiable. Using equation (78) in equations (79) and (80):

$$\lim_{k\to\infty} z_k - Hx_k = 0 \tag{81}$$

$$\lim_{k\to\infty} \nabla h(z_k) + w_k = 0 \tag{82}$$

$$\lim_{k\to\infty} \nabla f(x_k) - H^* w_k = 0 \tag{83}$$

Since ∇g is continuous, using equation (81) we get that

$$\lim_{k\to\infty} \nabla h(Hx_k) + w_k = \lim_{k\to\infty} \nabla h(z_k) + w_k = 0 \tag{84}$$

Rearranging the terms

$$\begin{aligned} \nabla E(x_k) &= H^* \nabla h(Hx_k) + \nabla f(x_k) \\ &= H^* \nabla h(Hx_k) + H^* w_k + \nabla f(x_k) - H^* w_k \\ &= H^*(\nabla h(Hx_k) + w_k) + (\nabla f(x_k) - H^* w_k) \end{aligned}$$

Finally using equations (83) and (84) in the previous result we obtain

$$\lim_{k\to\infty} \nabla E(x_k) = H^* 0 + 0 = 0$$

This shows that with the additional hypothesis of f being differentiable, the Linearized ADMM converges to a critical point of the original objective $E(x)$. □

B. APPLICATION TO PNP-LADMM

B.1. Proof of Proposition 1

B.1.1. Proximal Gradient Step Denoiser.

Proof. Let $\mathcal{D}_{\sigma_d}$ be the proximal gradient step denoiser defined in [18] as $\mathcal{D}_{\sigma_d} := Id - \nabla g_{\sigma_d}$ where $g_{\sigma_d}(x) = \frac{1}{2}\|x - N_{\sigma_d}\|^2$ and N_{σ_d} is a neural network.

According to [18, Proposition 3.1] there exists ϕ_{σ_d} such that $\mathcal{D}_{\sigma_d} = \operatorname{prox}_{\phi_{\sigma_d}}$.

In addition [18, Equation (26)] states that $\phi_{\sigma_d} \geq g_{\sigma_d}$, and by definition $g_{\sigma_d} \geq 0$.

Hence, $f = \phi_{\sigma_d}/\sigma_d^2$ is lower bounded by 0 and $\mathcal{D}_{\sigma_d} = \operatorname{prox}_{\sigma_d^2 f}$ as indicated by Proposition 1. □

B.1.2. MMSE Denoiser.

Proof. Let $\mathcal{D}_{\sigma_d}(y) = E[X|Y = y]$ be an MMSE denoiser, where $Y = X + \sigma_d N$ and $N \sim \mathcal{N}(0, \sigma_d^2 Id)$, and $X \sim p_X$, p_X being a probability measure.

We want to show that there exists a lower bounded ϕ_{σ_d} such that $\mathcal{D}_{\sigma_d}(x) = \operatorname{prox}_{\phi_{\sigma_d}}(x)$.

For $\sigma_d = 1$ according to [29] there exists $f(x) \geq -\log p_Y(x)$, such that $\mathcal{D}_1 = \operatorname{prox}_f$. f is lower bounded because the noisy density $p_Y(x) = (p_X * g_1)(x) \leq 1/\sqrt{2\pi}$ is upper-bounded by the maximum value of g_1 (the gaussian pdf with identity covariance matrix).

For $\sigma_d \neq 1$ the problem can be reduced to the previous case via the following scaling: Consider $\mathcal{P}(x) = \frac{1}{\sigma_d}\mathcal{D}_{\sigma_d}(\sigma_d x)$. Then $\mathcal{P}(y) = E[\tilde{X}|\tilde{Y} = y]$ is an MMSE denoiser with variance 1 with $\tilde{X} = X/\sigma_d$ and $\tilde{Y} = \tilde{X} + N$. So we can find (according to the previous argument for $\sigma_d = 1$) f such that $\mathcal{P} = \operatorname{prox}_f$. Applying a change of variables in the proximal operator we obtain

$$\mathcal{D}_{\sigma_d}(y) = \sigma_d \mathcal{P}(y/\sigma_d) = \operatorname{prox}_{\phi_{\sigma_d}}(y)$$

where

$$\phi_{\sigma_d}(x) = \sigma_d^2 f(x/\sigma_d)$$

Finally, since f is lower-bounded ϕ_{σ_d} is lower bounded too.

□