Deep Model-Based Super-Resolution with Non-Uniform Blur

Charles Laroche, Andrés Almansa, Eva Coupeté, Matias Tassano

Outline

1. Introduction
2. Plug & Play Linearized-ADMM
3. Deep unfolding
4. Conclusion and futur work

Motivations

- Small sensors (action cameras, mobile phones):
  - Blur
  - Noise

- No optical Zoom -> Super resolution

- Real-world blur is spatially-variant (defocus, motion blur…)

Defocus blur from iPhone 11

Motivations

- Small sensors (action cameras, mobile phones):
    - Blur
    - Noise

- No optical Zoom -> Super resolution

- Real-world blur is spatially-variant (defocus, motion blur…)

Motion blur from a GoPro camera

Maximum a-posteriori

- Super resolution as an inverse problem:

High-resolution

Low-resolution $\longrightarrow$ $y = Hx + \epsilon,$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$

Down-sampling Blur operator Noise

Maximum a-posteriori

- Super resolution as an inverse problem:

High-resolution

Low-resolution $\longrightarrow$ $y = Hx + \epsilon, \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$

Down-sampling Blur operator Noise

- Usually solved using maximum a-posteriori (MAP) or learning-based approaches:

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2} \|Hx - y\|_2^2 + \lambda \Phi(x)$$

Regularization function

Deep Plug & Play

- Usually solved using **maximum a-posteriori** (MAP) or learning-based approaches:

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2} \|Hx - y\|_2^2 + \lambda \Phi(x)$$

Regularization function

- [Gribonval 2011]:

$$prox_{\Phi}(i) = argmin_x \Phi_{MMSE}(x) + \frac{1}{2\beta^2} \|i - x\|_2^2 = E[X | I = i] \approx \mathcal{P}_{\beta}(i)$$

$$I = X + \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \beta Id) \quad X \sim P_X$$

- Plug & Play:

We replace all equation of the form $\arg\min_x \frac{1}{2\beta^2} \|x - y\|_2^2 + \Phi(x)$ with a denoiser $\mathcal{P}_{\beta}(y)$

We use DRUNet [Zhang, 2021] denoiser.

[Gribonval 2011]: Rémi Gribonval, Should penalized least squares regression be interpreted as Maximum A Posteriori estimation? 2011.
[Zhang, 2021]: Zhang Kai, Li Yawei, Zuo Wangmeng, Zhang Lei, Van Gool Luc, Timofte Radu, Plug-and-Play Image Restoration with Deep Denoiser Prior. TPAMI 2021.

Optimization

- Usually solved using maximum a-posteriori (MAP) or learning-based approaches:

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2}\|Hx - y\|_2^2 + \lambda\Phi(x)$$

Regularization function

PnP- ADMM [Venkatakrishnan 2013] :

$$x_{k+1} = \mathcal{P}_{\sqrt{\lambda/\mu}}(v_k - u_k)$$

$$v_{k+1} = \arg\min_v \frac{\mu}{2}\|v - (x_k + u_k)\|_2^2 + \frac{1}{2\sigma^2}\|Hv - y\|_2^2$$

$$u_{k+1} = u_k + (x_{k+1} - v_{k+1}).$$

It requires the **inversion of the forward** model

$$(\sigma^2\mu Id + H^T H)^{-1}(\mu\sigma^2(x_k + u_k) + H^T y)$$

PnP-ISTA [Kamilov 2017]:

$$x_{k+1} = \mathcal{P}_{\sqrt{\lambda\gamma}}(x_k - 2\gamma H^T(Hx_k - y))$$

It requires **more iterations** to converge

[Venkatakrishnan 2013]: .V. Venkatakrishnan, C.A. Bouman, and B. Wohlberg, Plug-and-play priors for model based reconstruction, IEEE GlobalSIP, 2013.
[Kamilov 2017]: U. S. Kamilov, H. Mansour, and B. Wohlberg, “A Plug-and-Play Priors Approach for Solving Nonlinear Imaging Inverse Problems,” IEEE Signal Process. 2017.

ADMM

- Usually solved using **maximum a-posteriori** (MAP) or learning-based approaches:

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2}\|Hx - y\|_2^2 + \lambda\Phi(x)$$

Regularization function

- ADMM with splitting variable **Hx=z** to solve the optimization problem:

$$x_{k+1} = \arg\min_x \frac{\mu}{2}\|Hx - (z_k - u_k)\|_2^2 + \lambda\Phi(x)$$

$$z_{k+1} = \arg\min_z \frac{\mu}{2}\|z - (Hx_k + u_k)\|_2^2 + \frac{1}{2\sigma^2}\| \; z - y\|_2^2$$

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

Linearized ADMM

- Usually solved using **maximum a-posteriori** (MAP) or learning-based approaches:

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2}\|Hx - y\|_2^2 + \lambda\Phi(x)$$

Regularization function

- ADMM with splitting variable **Hx=z** to solve the optimization problem:

$$x_{k+1} = \arg\min_x \cancel{\frac{\mu}{2}\|Hx - (z_k - u_k)\|_2^2} + \lambda\Phi(x) + \frac{\mu}{2}\langle x - x_k, 2H^T(Hx_k - (z_k - u_k))\rangle + \frac{\rho}{2}\|x_k - x\|_2^2$$

$$z_{k+1} = \arg\min_z \frac{\mu}{2}\|z - (Hx_k + u_k)\|_2^2 + \frac{1}{2\sigma^2}\| \; z - y\|_2^2$$

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

Linearization

Extra regularization

Linearized ADMM

- Usually solved using **maximum a-posteriori** (MAP) or learning-based approaches:

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2} \|Hx - y\|_2^2 + \lambda\Phi(x)$$

Regularization function

- Linearized-ADMM [Liu 2019] to solve the optimization problem:

$$x_{k+1} = \arg\min_x \frac{\rho}{2} \|x - (x_k - (\mu/\rho)H^T(Hx_k - z_k + u_k))\|_2^2 + \lambda\Phi(x)$$

$$z_{k+1} = \arg\min_z \frac{1}{2\sigma^2} \|z - y\|_2^2 + \frac{\mu}{2} \|z - (Hx_{k+1} + u_k)\|_2^2$$

$$u_{k+1} = u_k + Hx_{k+1} - z_{k+1}$$

[Liu 2019]: Qinghua Liu, Xinyue Shen, Yuantao Gu. Linearized ADMM for Non-convex Non-smooth Optimization with Convergence Analysis, 2019

Linearized ADMM

- Usually solved using **maximum a-posteriori** (MAP) or learning-based approaches:

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2} \|Hx - y\|_2^2 + \lambda\Phi(x)$$

Regularization function

- Linearized-ADMM to solve the optimization problem:

$$\begin{aligned} x_{k+1} &= \mathcal{P}_\beta(x_k - \gamma H^T(Hx_k - z_k + u_k)) \\ z_{k+1} &= \frac{y + \alpha(Hx_{k+1} + u_k)}{1 + \alpha} \\ u_{k+1} &= u_k + Hx_{k+1} - z_{k+1} \end{aligned}$$

with

$$\beta^2 = \frac{\lambda}{\rho}$$

$$\alpha = \sigma^2\mu$$

$$\gamma = \frac{\mu}{\rho}$$

Convergence

Recall

$$x_{MAP} = argmin_x \frac{1}{2\sigma^2}\|Hx - y\|_2^2 + \lambda\Phi(x)$$
$$= argmin_x g(Hx) + \lambda\Phi(x) = E(x)$$

Assumption 1

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

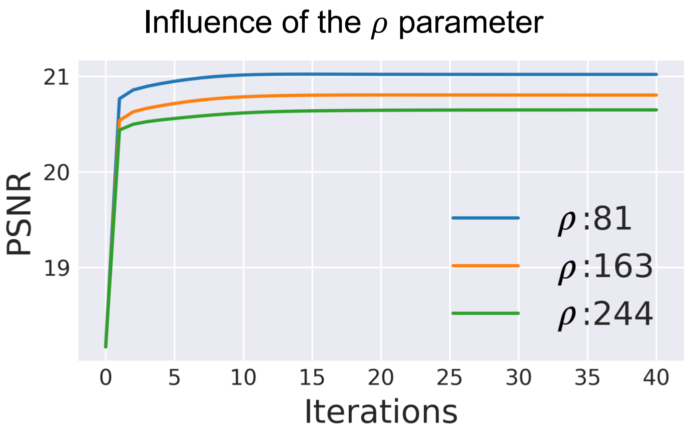
Theorem 1 *Under Assumption 1, for linearized-ADMM with hyper parameters such that:*

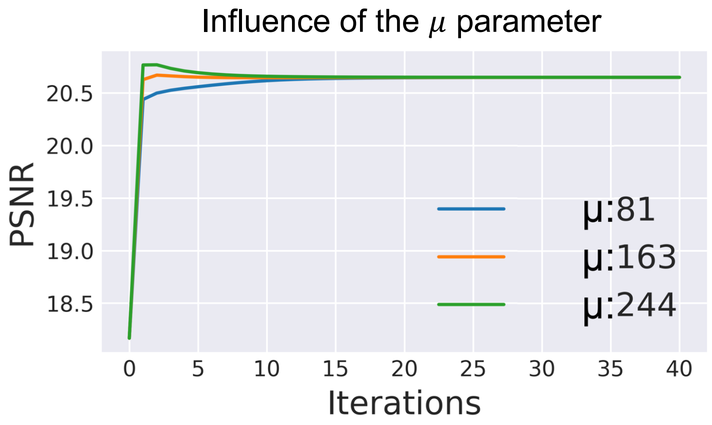
$$\mu \geq L_h$$
$$\rho \geq \mu\|H\|^2$$

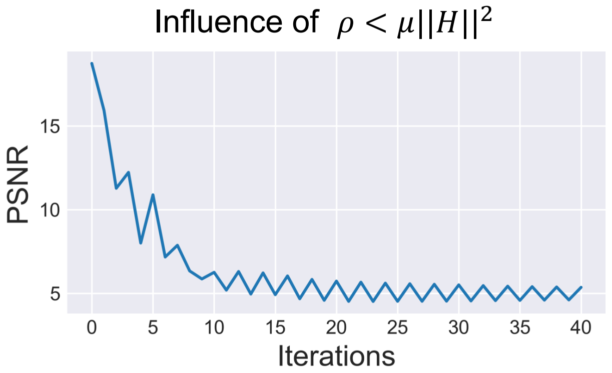
then the sequence $\{\mathcal{L}_\mu(x_k, z_k, w_k)\}$ is convergent and the primal residues $\|x_{k+1} - x_k\|$, $\|z_{k+1} - z_k\|$ and the dual residue $\|w_{k+1} - w_k\|$ converge to 0 as k approaches infinity.
If in addition Φ is differentiable then $\lim_{k\to\infty} \nabla E(x_k) = 0$.

[Qinghua Liu, Xinyue Shen, Yuantao Gu. Linearized ADMM for Non-convex Non-smooth Optimization with Convergence Analysis, 2019.]

Parameters influence







Convergence speed

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**.

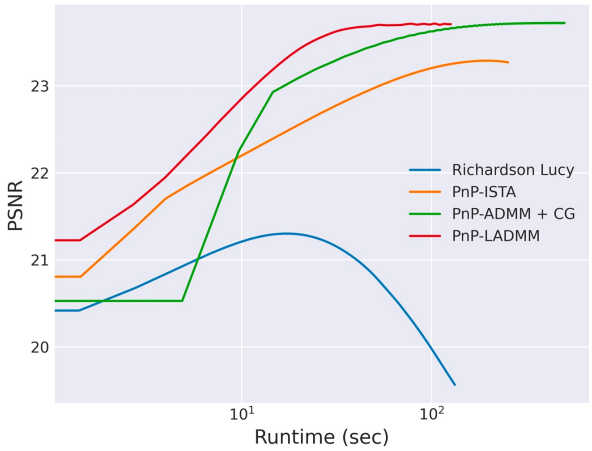


Fig. 2: Convergence speed of the different methods, we use 40 images with spatially-varying blur and Gaussian noise with $\sigma = 10/255$.

Deep **unfolding** networks

$$x_{k+1} = \mathcal{P}_\beta(x_k - \gamma H^T(Hx_k - z_k + u_k))$$

$$z_{k+1} = \frac{y + \alpha(Hx_{k+1} + u_k)}{1 + \alpha}$$

$$u_{k+1} = u_k + Hx_{k+1} - z_{k+1}$$

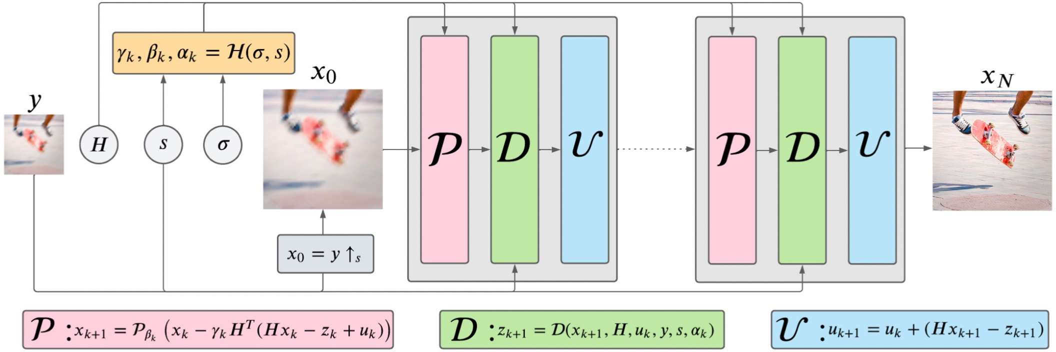


Figure 3: Model architecture, the low-resolution image is upsampled and alternately fed to the prior module $\mathcal{P}$, the data module $\mathcal{D}$ and the update module $\mathcal{U}$ during N iterations

[Kai Zhang, Luc Van Gool, Radu Timofte, Deep Unfolding Network for Image Super-Resolution, CVPR 2020.]

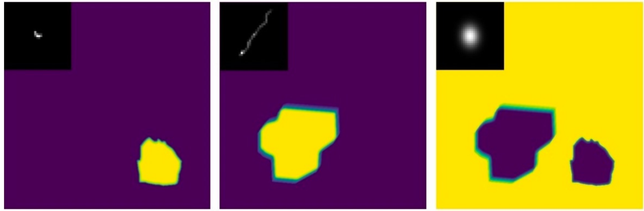
Dataset

- We use COCO segmentation dataset and O'Leary blur model:

$$y = (\sum_{i=1}^{P} U_i K_i x) \downarrow_s + \epsilon$$

- The U_i's are segmentation masks (not necessary binary) and K_i's (convolution matrix block circulant matrix with circulant blocks)

- Limitations: COCO HR images are not always of a good quality



(a) Object masks U_i and kernels K_i

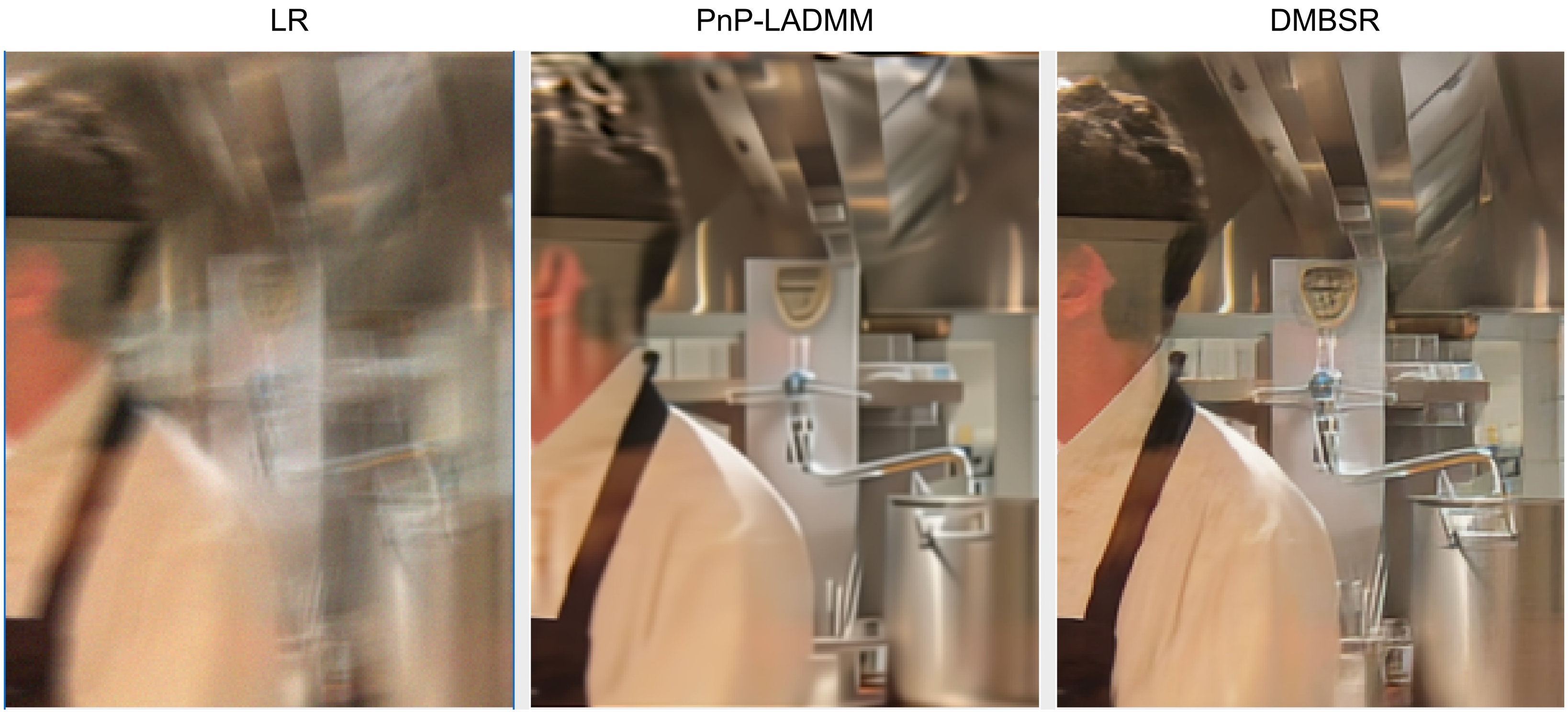
(b) Generated pairs

Figure 4: Example of data generated by our pipeline

Results

Results

PnP vs Unfolding



Blind model

LR

[Carbajal 2021] + Ours

Joint training

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

Conclusion

- Plug & Play algorithm for inverse problems with untractable data fitting term proximal operators based on linearized ADMM.

- Deep unfolding of our algorithm.

- Experiments on spatially-varying blur with the O’Leary blur model

Futur work

- Blind model (Guillermo Carbajal and al)

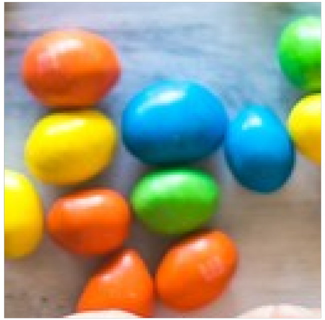
- Camera Shake using homography model:

$$y = K(x) = \frac{1}{n}\sum_{i=1}^{n} H_i(x)$$

where H_i is the homographic deformation induced by the camera rotation between $t = 0$ and $t = t_i$.

- Application to demosaicking

Sharp



Blurry

PnP

Unfolding

Thank You !