Photon-Limited Deblurring using Algorithm Unrolling

Anonymous Author(s)
Affiliation
Address
email

Abstract

Image deblurring in a photon-limited condition is ubiquitous in a variety of low-light applications such as photography, microscopy and astronomy. However, presence of photon shot noise due to low-illumination and/or short exposure time makes the deblurring task substantially more challenging . This paper presents an algorithm unrolling approach for the photon-limited deblurring problem that unrolls a Plug-and-Play algorithm using a fixed-iteration network. By modifying the typical two-variable splitting to a three-variable splitting, our unrolled network is differentiable and can be trained end-to-end. We demonstrate the usage of our algorithm on real photon-limited image data.

1 Introduction

Non-blind image deblurring is a restoration problem where the aim is to obtain a clean image from an image corrupted by spatially invariant blur due to motion, camera shake or defocus. Traditionally, the problem is formulated as follows: $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where the $\mathbf{x}$ is the clean image to be recovered from the corrupted image $\mathbf{y}$, $\mathbf{H}$ represents the blur operation in matrix form, and $\mathbf{n}$ is the additive i.i.d Gaussian noise. Non-blind deblurring methods assume that the blur kernel $\mathbf{H}$ is known.

An overwhelming majority of current solutions [8, 10, 5, 13, 3, 4] are able to deblur images under the presence of i.i.d Gaussian noise. However, in low-illumination settings, the images captured by the sensor are corrupted with Poisson shot noise and often these solutions fail to adequately recover the clean image. We refer to this situation as the *photon limited* setting i.e. when the number of photon arriving at the image sensor during the exposure time is small compared to that of a well-illuminated or *photon-abundant scene*. In this paper, we address the problem of non-blind deblurring in photon-limited scenes.

1.1 Problem Formulation

Assume the clean image $\mathbf{x}$ to be normalized from $[0, 1]$ and monochrome which is blurred by a motion kernel $\mathbf{H}$. The signal-dependent shot noise is represented using the term *photon level* α and hence the sensor output is given by

$$\mathbf{y} = \text{Poisson}(\alpha \cdot \mathbf{H}\mathbf{x}), \tag{1}$$

where $\text{Poisson}(\mathbf{u})$ represents an instance of Poisson random vector with mean equal to $\mathbf{u}$. Therefore, the likelihood of the blurred and noisy image $\mathbf{y}$ given the clean signal $\mathbf{x}$ is as follows:

$$p(\mathbf{y}|\mathbf{x};\alpha) = \prod_{j=1}^{N} \frac{(\alpha\mathbf{H}\mathbf{x})_j^{\mathbf{y}_j} e^{-(\alpha\mathbf{H}\mathbf{x})_j}}{\mathbf{y}_j!}, \quad \text{where}(\cdot)_j \text{ represents the } j\text{th entry of the vector} \tag{2}$$

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

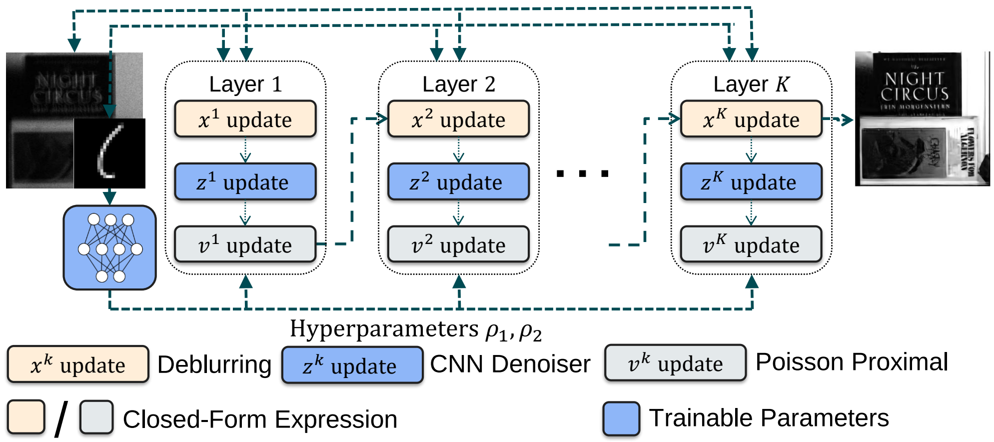


Figure 1: **Proposed unrolled Plug-and-Play for deblurring.** For conventional PnP, one of the update requires a convex optimization solver, making it infeasible for end-to-end training. Through the alternate formulation of the problem, each sub-module in an iteration is in closed form and more importantly, differentiable.

1.2 Contributions and scope

In this paper, we formulate photon-limited image deblurring problem as a Poisson inverse problem. Classical methods for the poisson inverse problem [9, 14, 12] are available but they don't tap into the power of convolutional neural networks. We approach Poisson deblurring by unrolling the Plug-and-Play algorithm [16, 2] using a fixed-iteration unrolled network. Compared to prior work such as [15] which requires an inner iteration solver, our three-operator splitting strategy makes all the sub-problems differentiable. This allows us to train the unfolded network end-to-end, as illustrated in Figure 1.

2 Method

In this paper, we present a unrolled iterative method as a solution for the Poisson deblurring problem as formulated in Section 1.1. First, the cost function corresponding to the MAP estimate of the clean image $\mathbf{x}$ given a Poisson log-likelihood and a prior $p(x)$ is formulated

$$\mathbf{x}^* = \underset{\mathbf{x}}{\text{argmin}} \left[\alpha \mathbf{1}^T \mathbf{H}\mathbf{x} - \mathbf{y}^T \log(\alpha \mathbf{H}\mathbf{x}) - \log p(\mathbf{x})\right], \text{ where } \mathbf{1} \text{ represents the all-ones vector} \quad (3)$$

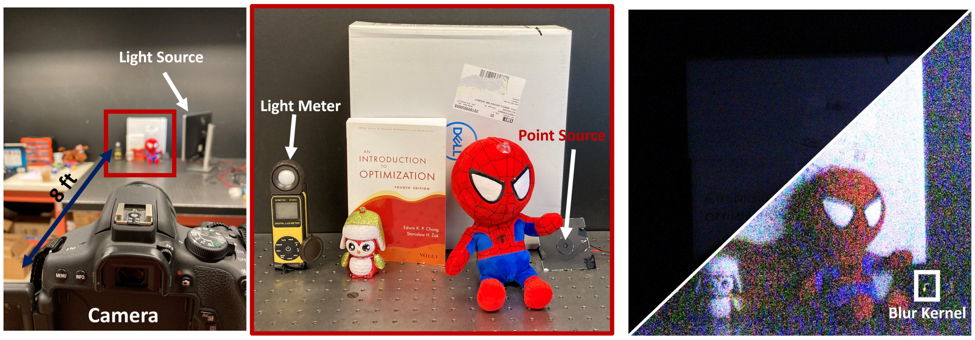
The cost function shown above can solved using the Plug-and-Play framework where we first convert the unconstrained optimization problem to a constrained one by performing variable splitting $\mathbf{x} = \mathbf{z}$ i.e.

$$\{\mathbf{x}^*, \mathbf{z}^*\} = \underset{\mathbf{x},\mathbf{z}}{\text{argmin}} \left[-\mathbf{y}^T \log(\alpha \mathbf{H}\mathbf{x}) + \alpha \mathbf{1}^T \mathbf{H}\mathbf{x} + \log p(\mathbf{z})\right] \quad \text{subject to} \;\; \mathbf{x} = \mathbf{z} \quad (4)$$

The constrained optimization problem above is solved by using the ADMM method. *For fixed iteration unrolling, Plug-and-Play framework in the conventional form is not feasible.* To unroll the ADMM algorithm in [16, 2], all the iterative updates need to be differentiable. This allows for end-to-end training of all the parameters of the fixed iteration network via backpropagation. For the Poisson inverse problem, the data-subproblem is another iterative method [15] and hence differentiating through it is fundamentally inefficient.

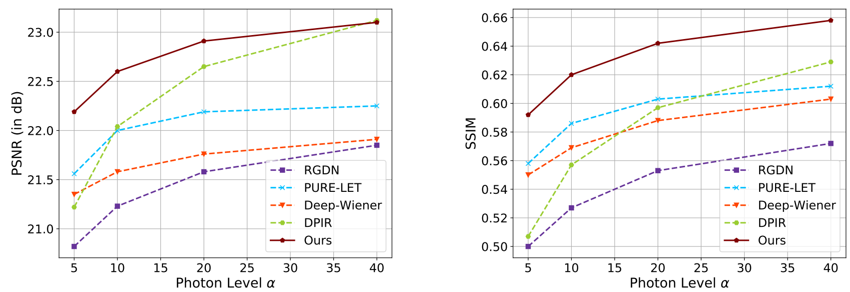
Since the current framework doesn't allow for iterative unrolling, we use an alternate formulation [6, 7] of the PnP-framework. Through this reformulation of Plug-and-Play, we are able to derive a series of iterative updates where each step can be implemented as a single-step and differentiable computation. Specifically, in addition to splitting the variable as $\mathbf{x} = \mathbf{z}$, we introduce a third variable $\mathbf{v}$ corresponding to blurred image $\mathbf{H}\mathbf{x}$ and hence the constraint $\mathbf{H}\mathbf{x} = \mathbf{v}$.

$$\{\mathbf{x}^*, \mathbf{z}^*, \mathbf{v}^*\} = \underset{\mathbf{x},\mathbf{z},\mathbf{v}}{\text{argmin}} \left[-\mathbf{y}^T \log(\alpha \mathbf{v}) + \alpha \mathbf{1}^T \mathbf{v} + \log p(\mathbf{z})\right] \quad \text{subject to} \;\; \mathbf{x} = \mathbf{z}, \mathbf{H}\mathbf{x} = \mathbf{v} \quad (5)$$



(a) Experimental setup, well illuminated scene (b) Real capture

Figure 2: **Experimental Setup** For evaluation of the proposed method on real images, we collect noisy and blurred images using a DSLR as shown in the setup shown above.



(a) (b)

Figure 3: **Quantitative evaluation.** Comparison of PSNR and SSIM of the different methods on Levin et. al. dataset [11].

After forming the corresponding the augmented Lagrangian [1], we arrive at the following iterative updates:

$$\mathbf{x}^{k+1} = (\mathbf{I} + (\rho_2/\rho_1)\mathbf{H}^T\mathbf{H})^{-1}(\widetilde{\mathbf{x}}_0^k + (\rho_2/\rho_1)\mathbf{H}^T\widetilde{\mathbf{x}}_1^k), \tag{6}$$

$$\mathbf{z}^{k+1} = D_\sigma(\widetilde{\mathbf{z}}^k), \tag{7}$$

$$\mathbf{v}^{k+1} = \frac{(\rho_2\widetilde{\mathbf{v}}^k - \alpha) + \sqrt{(\rho_2\widetilde{\mathbf{v}}^k - \alpha)^2 + 4\rho_2\mathbf{y}}}{2\rho_2}, \tag{8}$$

$$\mathbf{u}_1^{k+1} = \mathbf{u}_1^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1}, \tag{9}$$

$$\mathbf{u}_2^{k+1} = \mathbf{u}_2^k + \mathbf{H}\mathbf{x}^{k+1} - \mathbf{v}^{k+1}, \tag{10}$$

where $\widetilde{\mathbf{x}}_0^k \stackrel{\text{def}}{=} \mathbf{z}^{k+1} - \mathbf{u}_1^k$, $\widetilde{x}_1^k \stackrel{\text{def}}{=} \mathbf{v}^{k+1} - \mathbf{u}_2^k$, $\mathbf{v}^k \stackrel{\text{def}}{=} \mathbf{H}\mathbf{x}^k + \mathbf{u}_2^k$, $\widetilde{\mathbf{z}}^k \stackrel{\text{def}}{=} \mathbf{x}^k + \mathbf{u}_1^k$. With an end-to-end trainable iterative process, we can now describe the unfolded iterative network. The Plug-and-Play updates described above are now unfolded for $K = 8$ iterations and the entire differentiable pipeline is trained in a supervised manner such that the final output i.e. $\mathbf{x}^{K-1}$ matches the clean image $\mathbf{x}$ using multi-scale ℓ_1 loss.

To initialize $\mathbf{x}^0$, we use the Wiener filtering step as follows (not to be confused with [3]). The parameters used in updates (6), (8) - ρ_1^k, ρ_2^k for $k = 1, 2, 3, .., K$ are changed for each iteration and determined in one-shot by the blurring kernel $\mathbf{h}$ and photon level α using a fully convolutional layer followed by a fully connected layer. For the denoiser in (7), we use the architecture provided in [18] which introduces skip connections in a U-Net architecture known as ResUNet.

3 Experiments

For quantitative evaluation of our method, we test our method along with other contemporary deblurring approaches on the Levin et. al [11] dataset which provides 32 images generated by blurring

4 different clean images by 8 different motion kernels and the blurred images are synthetically corrupted with shot noise at photon levels $\alpha = 5, 10, 20, 40$. We compare our method with the following deblurring methods - **RGDN** [8], **PURE-LET** [12], **Deep-Wiener** [3], and **DPIR** [17].

To demonstrate that the proposed scheme is able to deblur images real images in low-light, we capture blurred images taken in low light (estimated photon level $\alpha \approx 20$). A kernel is also captured by placing a point source in the the scene. The reconstruction results are shown and compared to other deblurring methods in Figure 4 and Figure 5.

Figure 4: **Proposed method on real data.** (Top) Noisy and blurred image captured using Canon EOS Rebel T6i camera. (Bottom) Reconstruction using our method. For a qualitative comparison of other deblurring approaches on these images, refer to Figure 5.

Raw Image | RGDN [8] | PURE-LET [12] | DPIR [17] | Wiener-Net [3] | Ours

Figure 5: **Qualitative Comparison on real data.** We look at zoomed in regions of the reconstructed images from Figure 4 using competing methods. From visual inspection one can see that our method is able to recover finer details compared to other methods.

4 Conclusion

In this paper, we present an unrolled algorithm for the photon-limited deblurring problem. An alternate three-operator splitting strategy was used to the Plug-and-Play framework to obtain a series of iterative steps which could be trained end-to-end. We demonstrated that the proposed method was able to recover deblurred images from both real and synthetic data.

References

[1] S. Boyd, N. Parikh, and E. Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[2] S. H. Chan, X. Wang, and O. A. Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2016.

[3] J. Dong, S. Roth, and B. Schiele. Deep wiener deconvolution: Wiener meets deep learning for image deblurring. In *34th Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2020.

[4] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu. Denoising prior driven deep neural network for image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2305–2318, 2018.

[5] T. Eboli, J. Sun, and J. Ponce. End-to-end interpretable learning of non-blind image deblurring. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 314–331. Springer, 2020.

[6] M. A. Figueiredo and J. M. Bioucas-Dias. Deconvolution of poissonian images using variable splitting and augmented lagrangian optimization. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 733–736. IEEE, 2009.

[7] M. A. Figueiredo and J. M. Bioucas-Dias. Restoration of poissonian images using alternating direction optimization. *IEEE transactions on Image Processing*, 19(12):3133–3145, 2010.

[8] D. Gong, Z. Zhang, Q. Shi, A. van den Hengel, C. Shen, and Y. Zhang. Learning deep gradient descent optimization for image deconvolution. *IEEE Transactions on Neural Networks and Learning Systems*, 31 (12):5468–5482, 2020. doi: 10.1109/TNNLS.2020.2968289.

[9] Z. T. Harmany, R. F. Marcia, and R. M. Willett. This is spiral-tap: Sparse poisson intensity reconstruction algorithms—theory and practice. *IEEE Transactions on Image Processing*, 21(3):1084–1096, 2011.

[10] J. Kruse, C. Rother, and U. Schmidt. Learning to push the limits of efficient fft-based image deconvolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4586–4594, 2017.

[11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971. IEEE, 2009.

[12] J. Li, F. Luisier, and T. Blu. Pure-let image deconvolution. *IEEE Transactions on Image Processing*, 27(1): 92–105, 2017.

[13] Y. Nan, Y. Quan, and H. Ji. Variational-em-based deep learning for noise-blind image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3635, 2020.

[14] R. D. Nowak and E. D. Kolaczyk. A statistical multiscale framework for poisson inverse problems. *IEEE Transactions on Information Theory*, 46(5):1811–1825, 2000.

[15] A. Rond, R. Giryes, and M. Elad. Poisson inverse problems by the plug-and-play scheme. *Journal of Visual Communication and Image Representation*, 41:96–108, 2016.

[16] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[17] K. Zhang, W. Zuo, S. Gu, and L. 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.

[18] K. Zhang, L. V. Gool, and R. 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.