2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506502

IMPROVING NEURAL BLIND DECONVOLUTION

Jan Kotera, Filip Šroubek, Václav Šmídl

Czech Academy of Sciences, Institute of Information Theory and Automation, Prague, Czech Republic

ABSTRACT

The field of blind image deblurring was for a long time dominated by Maximum-A-Posteriori methods seeking the optimal pair of sharp image–blur of a suitable functional. Recently, learning-based methods, especially those based on deep convolutional neural networks, are proving effective and are receiving increasing attention by the research community. In 2020, Ren et al. proposed a deblurring method called SelfDeblur which combines the model-driven approach of traditional MAP methods and the generative power of neural nets. The method is capable of producing very high-quality results, yet it inherits some problems of MAP methods, especially possible convergence to a wrong local optimum. In this paper we propose several easy-to-implement modifications of SelfDeblur, namely suitable initialization, multiscale processing, and regularization, that improve the average performance of the original method and decrease the probability of failure.

Index Terms— blind deblurring, SelfDeblur, deep image prior, multiscale

1. INTRODUCTION

Blind image deblurring is a long-standing image processing problem for which hundreds of solutions have been proposed. The task consists of recovering the sharp image corresponding to the input image corrupted by an unknown blur. Let us denote by y the observed blurred image and by x the corresponding sought sharp image. If the blur is spatially invariant within the input, it can be modeled by a convolution with an unknown kernel k and the degradation model has the form

$$y = k * x + n, \tag{1}$$

where n denotes observation noise. The objective is to find x (and usually k) given y.

Vast majority of proposed solutions are formulated as the MAP estimation of the image–blur pair (x, k), which typically results in an iterative procedure alternating over the image- and blur-estimation [1–8]. Recently the field of image deblurring has been flooded with methods based on deep-learning, where models trained on large datasets of natural images predict either the blur or the sharp image directly from the blurred input [9–13]. Both approaches have their advantages and disadvantages. MAP methods are rather slow, sensitive to parameter tuning, and can end up in a local optimum which does not correspond to the desired or visually appealing result. On the other hand, the efficacy of learning-based methods depends strongly on the quality of the training dataset and procedure; they have difficulty generalizing to unseen types of images or large and complex blurs.

In an attempt to mitigate some of these shortcomings, Ren et al. [14] recently proposed a method called *SelfDeblur* that combines the MAP approach with the approximation power of neural nets. The method is based on a concept called *deep image prior*, proposed earlier in [15], in which both the estimated image and blur are represented by a generative net and this implicitly serves as a corresponding prior (a short overview of this approach follows in Sec. 2.2). These nets are not pre-trained on any datasets, they are used merely for their structure which serves as a regularizer of the optimization process. SelfDeblur then follows essentially the MAP approach and, by "training" the aforementioned nets solely on the blurred input, seeks the optimum of the posterior corresponding to the deblurring problem. The resulting method is capable of achieving impressive deblurring results and reportedly beats many established SOTA methods on several benchmark datasets.

Unfortunately, the method does not entirely escape some of the typical problems inherent to MAP approach. As an iterative method applied to a non-convex problem, it often fails to converge to the correct optimum. On top of that, the neural nets are randomly initialized and as a result the method is non-deterministic – multiple runs on the same input produce different results with a wide range in quality. In this paper we address some of these problems and propose several easy-to-implement modifications or usage protocols, inspired by similar ideas used in traditional MAP methods, which do not change to core concept of SelfDeblur or its net architecture yet substantially improve the results in terms of quality and consistency.

2. OVERVIEW OF THE PROBLEM

2.1. Maximum-a-posteriori image deblurring

In the MAP deblurring approach, the solution has the form

$$(x^*, k^*) = \underset{x,k}{\operatorname{argmax}}\, P(x, k \mid y) = \underset{x,k}{\operatorname{argmax}}\, P(y \mid x, k) P(x) P(k), \tag{2}$$

This work was supported by the Czech Science Foundation grant GA20-27939S and by the Praemium Academiae of the Czech Academy of Sciences.

where $P(y|x,k)$ is the likelihood (data-driven) term and $P(x)$, $P(k)$ are priors. This corresponds to the minimization problem

$$\min_{x,k} L(x,k;y) + R(x) + R(k), \tag{3}$$

where L is the data-term (corresponding to the likelihood) and R are regularizers (priors). Solving (3) then most commonly results in the following alternating procedure:

1. Initialize x, k.
2. Update x by solving (3) w.r.t. x.
3. Update k by solving (3) w.r.t. k.

The optimized objective is highly non-convex, containing many spurious local minima, which presents serious practical problems. The outlined procedure often gets trapped in a local minimum with incorrect kernel k and, by nature of the blur operation, deblurring with such kernel produces a visually unpleasant image. A particularly strong minimum is the so-called *no-blur* solution, in which $x^* = y$ and $k^* = \delta$ (delta function), as this optimizes the data-term. Many solutions have been proposed to mitigate this problem and certain recurring and time-tested trends can be identified, such as:

- Suitably initialize x or k [16].
- Process y in a multiscale fashion, i.e. downsample y to a scale corresponding to small size of k, estimate x, k at this scale level and upsample these estimates to the next higher level to be used as initialization. Repeat until the original scale is reached [2].
- Use regularizers designed to increase the probability of the correct solution and possibly adjust those during the optimization procedure [1,4,6,17,18].
- Similarly, use data-term designed to facilitate kernel estimation rather than one implied by noise distribution; possibly even use a different data-term for x-estimation and k-estimation [3].
- Perform artificial processing of the current estimate of either k or x during optimization before estimating the other variable. Examples include edge enhancement (shock-filtering, unsharp-masking, over-regularization) of x or suppression of small values in k [3, 18].

This list is by no means exhaustive but provides the general idea of the means typically employed to make MAP deblurring work.

2.2. SelfDeblur and Deep Image Prior

For an inverse problem in imaging formulated as

$$x^* = \operatorname*{argmin}_x L(x;\, y) + R(x), \tag{4}$$

the idea of *deep image prior* (DIP) [15] is to reformulate (4) as

$$x^* = \mathcal{G}(\theta^*), \text{ where } \theta^* = \operatorname*{argmin}_\theta L(\mathcal{G}(\theta),\, y). \tag{5}$$

In other words, the image x is represented using a function $\mathcal{G}$ (typically a generative neural net) parametrized by θ and this representation implicitly serves as a regularizer. The net is not pre-trained in any way, it is in fact initialized randomly and its parameters θ optimized as a way of solving (5). After its publication, DIP was quickly adopted and utilized in many image processing tasks.

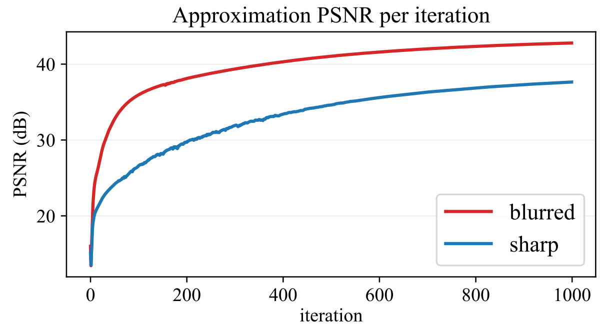

Fig. 1. Error of the image approximation problem (8) averaged over several sharp and blurred images. Sharp images are less easily approximated by the net and therefore are not inherently favored over the incorrect blurred solutions.

SelfDeblur [14] is an application of DIP in MAP image deblurring. The problem is then reformulated as

$$\min_{\theta_x,\theta_k} \|\mathcal{G}_k(\theta_k) * \mathcal{G}_x(\theta_x) - y\|_2^2, \tag{6}$$

$$\text{s.t. } 0 \le \mathcal{G}_x(\theta_x) \le 1 \text{ and } 0 \le \mathcal{G}_k(\theta_k),\ \|\mathcal{G}_k(\theta_k)\|_1 = 1. \tag{7}$$

The functions $\mathcal{G}_x$ and $\mathcal{G}_k$ are neural nets (deep multiscale encoder-decoder and shallow fully-connected net, respectively) and their architecture is such that the constraints (7) are satisfied automatically. The optimization then consists of a gradient-descent of (6) in the space of net parameters θ_x and θ_k, starting from their random initialization.

Despite being essentially a direct MAP approach without any artificial additives, the method works impressively well, which is demonstrated in the original paper [14]. Part of its appeal is that it does not contain any explicit blur inversion step which normally causes visual artifacts in the image if the blur estimation is incorrect. However, it is susceptible to the previously discussed problems of MAP methods and has its own shortcomings.

The core idea of DIP, as presented by its authors, is that the generative nets $\mathcal{G}_x$ and $\mathcal{G}_k$, by means of their structure, inherently favor generation of the correct x and k – supposedly by being able to approximate e.g. a sharp image more easily than a blurred one. In case of deblurring this premise is not fulfilled, as can be readily demonstrated. Fig. 1 shows the approximation error of the simple problem $\min_{\theta_x} \|\mathcal{G}_x(\theta_x) - y\|_2^2$ for both sharp and blurred input image y. It is apparent that blurred images are in fact better and more easily approximated by the encoder-decoder net, so it cannot be the approximation capacity of the net that plays the role of a prior favoring sharp images. It is probably the interplay between the net structure (i.e. the fact that the image is optimized in terms of θ_x rather than x directly) and the optimizer (Adam) what causes that the no-blur solution is often avoided, though there is still no guarantee that the non-convex optimization

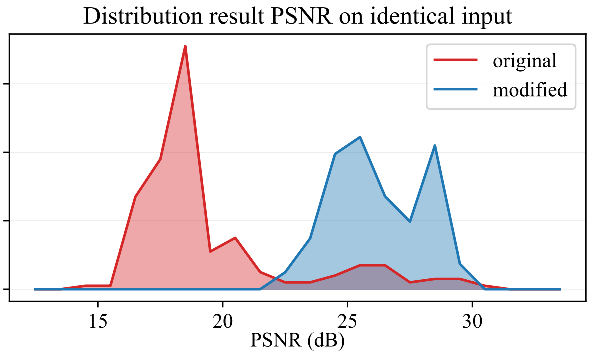

Fig. 2. Multiple runs on an identical input; distribution of the result PSNR of the original SelfDeblur (red) and the proposed modification (blue) on a random image from the test dataset.

will end up in the correct minimum. Similar phenomenon was observed and analyzed e.g. in [19]. Another problem of SelfDeblur is that due to the random initialization of the nets it is strongly non-deterministic – multiple runs on the same input produce very different results (in terms of e.g. PSNR; see demonstration in Fig. 2) and it is therefore difficult to fine-tune the method's hyperparameters.

In this paper we propose several additions to SelfDeblur that to a certain degree alleviate these problems and make the method more consistent, without substantially increasing the computational burden and altering the core concept or proposed architecture of the original method.

3. PROPOSED SOLUTION

Inspired by the techniques outlined in Sec. 2.1 we augmented SelfDeblur by the following techniques:

Multiscale processing We initially estimate x and k for a $2\times$ downsampled input y, upsample these estimates by a factor of $\sqrt{2}$ and use them as initialization (see the next paragraph) of the deblurring at the next higher scale, until we reach the original input resolution (i.e. we use three scales in total). The last scale is processed for 5000 iterations (per original SelfDeblur implementation) while the previous scales run for only 500 iterations, which is sufficient for initialization of the next scale. The increase of processing time is then approx. 15 % (7.5 % if only two scales are used).

Initialization In the original SelfDeblur, the nets $\mathcal{G}_x$ and $\mathcal{G}_k$ are initialized randomly. We propose to initialize them such that $\mathcal{G}_x(\theta_x) = x_0$ and $\mathcal{G}_k(\theta_k) = k_0$ for some x_0, k_0, so that the deblurring loop has a meaningful starting point. The first (smallest) scale is initialized with k_0 as Gaussian blur of σ equal to the kernel half-width, while $\mathcal{G}_x$ is left random. Each next scale is initialized with k_0 and x_0 corresponding to upsampled estimates from the previous scale. To initialize the parameters of the net, the following problem must be solved

$$\min_{\theta_x} \|\mathcal{G}_x(\theta_x) - x_0\|_2^2 \quad (8)$$

(and correspondingly for $\mathcal{G}_k$). This optimization converges quickly to an acceptable tolerance (we use RMSE=$10/255$ for $\mathcal{G}_x$ and correspondingly scaled value for $\mathcal{G}_k$).

Regularization The loss of the original SelfDeblur consists only of the data-term measuring the degradation model error $k * x - y$. This error is measured either by its ℓ^2 norm as in (6) or by SSIM [20] as $\mathrm{SSIM}(x * k, y)$. This is not mentioned in the original paper and on the other hand the paper mentions regularizing the image by total variation (TV) – it can be seen in the code, however, that TV is removed and the data-term is switched from ℓ^2 to SSIM after 1000 iterations.

We learned that using TV for image regularization strongly pushes the optimization to the no-blur solution, but it can improve the image quality when the optimization is already close to the correct solution – we therefore use TV after 2000 iterations in the last scale. In addition, we use ℓ^2 regularization for the kernel values in the initial part of the optimization, as this increases the loss of the no-blur solution (the kernel values sum to 1, therefore ℓ^2 is decreased when the kernel contains many small values rather than few large). To summarize, we replace the problem (6) by

$$\min_{\theta_x,\theta_k} L\left(\mathcal{G}_k(\theta_k) * \mathcal{G}_x(\theta_x);\, y\right) + \alpha_x \|\nabla \mathcal{G}_x(\theta_x)\|_1 + \alpha_k \|\mathcal{G}_x(\theta_x)\|_2^2, \quad (9)$$

where L is ℓ^2 if $\texttt{iter} < 2000$, SSIM otherwise, $\alpha_x = .01$ if $\texttt{iter} > 2000$, 0 otherwise, and $\alpha_k = .1$ if $\texttt{iter} < 2000$, 0 otherwise.

The deblurring optimization procedure is identical to that of SelfDeblur. The proposed modifications are simple in their implementation and the original code then requires only a few changes. In the next section we demonstrate their practical effect.

4. EXPERIMENTAL EVALUATION

We tested the proposed modification and compared its performance to that of the original SelfDeblur method on a dataset of images with synthetic blur. Eight images from the Kodak dataset [21] were each blurred with eight different blur kernels, resulting in 64 test images in total. Because the method is non-deterministic, each of these inputs was processed ten times. To better demonstrate the behavior of the tested methods, instead of reporting only the mean result we present the results in the form of a distribution of the restoration performance on the whole dataset, as this better illustrates the spread and modes of the method performance. Also, we report the performance in terms of ISNR (improvement in PSNR w.r.t. the input), otherwise results for inputs of different difficulty would be compared absolutely, which is incorrect. The corresponding distribution is in Fig. 3. It is apparent that the modified method in general performs better (the whole distribution is shifted to the right), the mode of complete fails (subzero or near-zero results) is lower while the mode of great successes (over $+5$dB improvement) is significantly more prominent. The numerical results in terms of mean ISNR and mean standard deviation (across identical inputs) is in Tab. 1. Due to the proposed modifications the

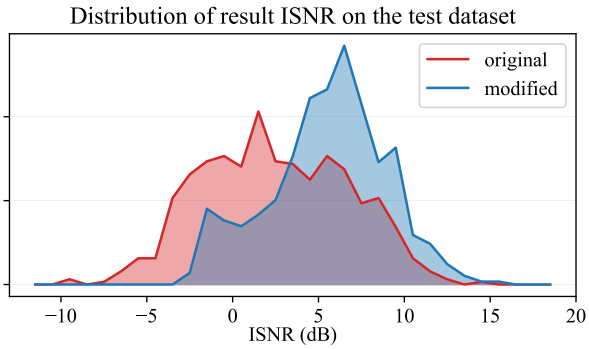

Fig. 3. Distribution of result ISNR (improvement in PSNR w.r.t. the input) on the Kodak test dataset; the original SelfDeblur (red) and the proposed modification (blue).

deblurring method achieved significantly better performance in average with even slight decrease in the randomness of results (naturally, the randomness can be attenuated by suitable regularization but that in turn compromises the top achievable performance).

The proposed modification was also tested on the well-known dataset of real-blurred images by Levin et al. [22], on which the original SelfDeblur method was developed and presented. The original method already performs very well on this dataset, yet the modification still slightly improves the mean performance, as can be seen in Tab. 1. Examples results are in Fig. 4, which contains the best (left) and the worst (right) results (in terms of ISNR) of the proposed method on the Levin dataset.

	Kodak dataset		*Levin dataset*	
Method	ISNR	Std. dev.	ISNR	Std. dev.
Original SD	2.48 dB	2.52	7.62 dB	1.36
Proposed	**5.49 dB**	2.12	**8.23 dB**	1.21

Table 1. Mean performance of the original method and the proposed modification on the synthetic Kodak dataset and real Levin dataset. The proposed modifications substantially improve the restoration performance with even slight decrease in randomness on the synthetic set; on the real set the improvement is less significant but still there.

4.1. Other attempted modifications

To make our contribution more complete, we will also list and briefly describe other modifications, mostly based on techniques that were previously successfully used in traditional MAP deblurring, which proved ineffective or detrimental when combined with deep image prior in the optimization framework utilized in SelfDeblur.

Image over-regularization It has been relatively common in MAP to apply strong image regularization during the initial iterations and decrease it slowly to its natural value throughout the optimization. We observed that with SelfDeblur this results in very quick convergence to the no-blur solution.

Overuse of multiscale In MAP it is common to start with very small kernels (3×3 or 5×5) and use many multiscale levels to reach the original resolution. We observed that for too small kernel the method again tends to converge to the no-blur solution from which it does not escape. We therefore advocate using larger initial kernels (the smallest we used is 7×7) and then fewer multiscale levels are required.

ISNR 13.1 dB   ISNR 3.6 dB

Fig. 4. The best (left) and the worst (right) result of the proposed method on the Levin dataset of images with real blur; the top row contains input images, the bottom row are results.

Image pre-filtering It was proposed by Xu et al. [3] and then adopted by many authors to artificially enhance edges in the current image estimate before the kernel-estimation step in the optimization loop. We tried applying shock-filtering to the image and while the technique slightly decreases the probability of no-blur or similarly "underestimated" solution, in average on the whole test set it did not help and produced slightly inferior results.

Combined SSIM and ℓ^2 loss Using SSIM as the data-term in the loss function instead of ℓ^2 can significantly improve the results (and it is surprising that such finding was not reported by the authors of SelfDeblur), but in certain cases the optimization starts to deteriorate after switching from ℓ^2 to SSIM and the deblurring fails. We experimented with combining SSIM and ℓ^2 but were not successful in getting the "best of both worlds." Depending on hyperparameters, the optimization behaves either like ℓ^2 or SSIM alone.

5. CONCLUSION

We proposed and evaluated several modifications of the SelfDeblur method by [14]. These modifications are simple to implement and increase the processing time only minimally while improving the average performance and consistency of the original method. They do not alter the core principle of the original method nor do they depend on or modify the architecture of the nets, which makes them universal and not particular to the SelfDeblur implementation.

6. REFERENCES

[1] Qi Shan, Jiaya Jia, and Aseem Agarwala, "High-quality motion deblurring from a single image," in *ACM SIGGRAPH 2008 Papers*, New York, NY, USA, 2008, SIGGRAPH '08, Association for Computing Machinery.

[2] Sunghyun Cho and Seungyong Lee, "Fast motion deblurring," *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 1–8, Dec. 2009.

[3] Li Xu and Jiaya Jia, "Two-phase kernel estimation for robust motion deblurring," in *Computer Vision – ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios, Eds., Berlin, Heidelberg, 2010, pp. 157–170, Springer Berlin Heidelberg.

[4] Li Xu, Shicheng Zheng, and Jiaya Jia, "Unnatural L0 sparse representation for natural image deblurring," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 1107–1114.

[5] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays, "Edge-based blur kernel estimation using patch priors," in *IEEE International Conference on Computational Photography (ICCP)*, April 2013, pp. 1–8.

[6] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang, "Blind image deblurring using dark channel prior," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1628–1636.

[7] Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao, "Image deblurring via extreme channels prior," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6978–6986.

[8] Liang Chen, Faming Fang, Tingting Wang, and Guixu Zhang, "Blind image deblurring with local maximum gradient prior," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 1742–1750.

[9] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 257–265.

[10] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia, "Scale-recurrent network for deep image deblurring," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 8174–8182.

[11] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. H. Lau, and M. Yang, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2521–2529.

[12] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019, pp. 8877–8886.

[13] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 3843–3851.

[14] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3338–3347.

[15] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[16] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 787–794, July 2006.

[17] Dilip Krishnan, Terence Tay, and Rob Fergus, "Blind deconvolution using a normalized sparsity measure," in *CVPR 2011*, June 2011, pp. 233–240.

[18] Subeesh Vasu, Venkatesh Reddy Maligireddy, and A. N. Rajagopalan, "Non-blind deblurring: Handling kernel uncertainty with cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] Kenji Kawaguchi, "Deep learning without poor local minima," in *Advances in Neural Information Processing Systems*, 2016.

[20] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[21] "Kodak image set," http://r0k.us/graphics/kodak/, Accessed: 2020-01-13.

[22] 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*, June 2009, pp. 1964–1971.