AN ITERATIVE DENOISING AND BACKWARDS PROJECTIONS METHOD AND ITS ADVANTAGES FOR BLIND DEBLURRING

Tom Tirer and Raja Giryes

School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel

ABSTRACT

In blind deblurring, the goal is to recover a latent sharp image from its blurred version when the blur kernel is unknown. In this case, natural image priors often lead to intractable algorithms or failures if used with maximum a posteriori (MAP) estimation. Therefore, the ruling approach is to start with estimating only the kernel, and then use it to recover the latent image via non-blind deblurring. While many blind deblurring works focus on the kernel estimation, we consider the second phase, where we build on the recently proposed Iterative Denoising and Backward Projections (IDBP) strategy. The proposed method uses an automatic parameters tuning mechanism, which can tune the parameters differently for each kernel and image, contrary to other deblurring algorithms that are restricted to a uniform tuning in the blind-deblurring setting. We demonstrate the advantages of our method over widely used deblurring algorithms.

Index Terms— Plug-and-Play, blind deblurring, image denoising, parameter tuning

1. INTRODUCTION

We consider the uniform blind deblurring task, where the goal is to reconstruct an image from its blurred version, obtained by an unknown spatially invariant blur kernel. Many works have studied this ill-posed problem in the recent years.

It has been observed that naive maximum a posteriori (MAP) estimation, which jointly estimates the latent sharp image and the kernel, tends to fail for many popular natural image priors [1, 2], as it prefers a trivial no-blur solution, i.e. the kernel is estimated as a delta function and the latent image as the given blurred image. In order to tackle this behavior, some works estimate the kernel alone from a marginalized distribution (integrated over the latent image) [1, 3], and then use the estimated kernel to recover the latent image via a non-blind deblurring method. Other works avoid the no-blur solution by ad-hoc strategies that strengthen dominant edges in the estimated latent image. Such strategies include changing the amount of regularization between iterations [4, 5], applying shock filter [6] or computing an edge map, which is used to refine the edges [7]. Another recent approach avoids ad-hoc operations by exploiting an unnatural image prior [8]. All of these methods produce an unnaturally looking image with high-contrast structures, aimed only at facilitating the kernel estimation. Therefore, they also require a final non-blind deblurring step.

Interestingly, the work in [9] makes use of a sophisticated natural image prior, which inherently avoids the MAP no-blur solution. However, its implementation is simplified to reduce the computational complexity. Presumably, this relaxation leads to an unsatisfactory image recovery, that once again, makes the use of a final non-blind deblurring step inevitable.

As described above, almost all relevant works basically differ in their kernel estimation strategy, while having a similar final non-blind deblurring step. In this work, we consider this last step. We demonstrate the advantages of using a deblurring method that incorporates an automatic parameter tuning mechanism. Such a method can outperform state-of-the-art algorithms, e.g. EPLL [10] and IDD-BM3D [11], which are restricted to a uniform tuning in the blind-deblurring setting. We build on the recently proposed Iterative Denoising and Backward Projections (IDBP) strategy [12], where we employ an automatic parameter tuning mechanism that can set parameters differently for each kernel and image.

While in this paper we focus on image deblurring, IDBP can solve general linear inverse problems using denoising algorithms, similar to the Plug-and-Play priors method [13, 14], but without the burdensome manual parameter tuning of the latter. The fact that the prior term being used by IDBP is determined by the chosen denoiser, allows us to compare IDBP with popular deblurring algorithms that use the same prior. For example, IDBP with a plugged-in BM3D denoiser [15] uses the same image prior as IDD-BM3D, and IDBP with a plugged-in EPLL denoiser [10] uses the same image prior as EPLL deblurring.

Lastly, we note that there are other works that consider automatic parameter selection in inverse problems. However, in these works the prior term is restricted to certain types of penalty functions, e.g. Tikhonov regularization [16, 17, 18] or smoothed versions of the ℓ_p ($1 < p < 2$) norm [19, 20]. As far as we know, the literature does not offer similar tuning mechanism for sophisticated image priors like EPLL and

The authors are thankful to Tomer Michaeli for fruitful discussion. This work is partially supported by ERC-StG SPADE.

BM3D. In contrast, the IDBP tuning considerations do not depend on the prior, which is specified by the chosen denoiser.

2. IDBP METHOD

For completeness, we briefly describe the image restoration problem addressed by IDBP, and the algorithm in its general form[1]. Later, we specialize it to the deblurring problem, and provide an automatic parameter tuning mechanism, which is suitable to the case of *inexact* blur kernel.

Many image restoration tasks can be formulated by

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ represents the unknown original image, $\boldsymbol{y} \in \mathbb{R}^m$ represents the observations, $\boldsymbol{H}$ is a *known* $m \times n$ degradation matrix and $\boldsymbol{e} \in \mathbb{R}^m$ is a vector of i.i.d. Gaussian random variables $e_i \sim \mathcal{N}(0, \sigma_e^2)$. This model can represent different image restoration problems; for example: denoising when $\boldsymbol{H}$ is the $n \times n$ identity matrix $\boldsymbol{I}_n$, inpainting when $\boldsymbol{H}$ is a selection of m rows of $\boldsymbol{I}_n$, and deblurring when $\boldsymbol{H}$ is a blurring operator. Note that $\boldsymbol{H}$ is ill-conditioned in the case of image deblurring, thus, in practice it can be approximated by a rank-deficient matrix, or alternatively by a full rank $m \times n$ matrix ($m < n$). For a unified formulation, it is assumed that $m < n$.

Similar to almost any approach for recovering $\boldsymbol{x}$, we define the following typical cost function

$$f(\tilde{\boldsymbol{x}}) = \frac{1}{2\sigma_e^2}\|\boldsymbol{y} - \boldsymbol{H}\tilde{\boldsymbol{x}}\|_2^2 + s(\tilde{\boldsymbol{x}}), \tag{2}$$

where $\|\cdot\|_2$ stands for the Euclidean norm, and $s(\boldsymbol{x})$ is a prior image model. Next, we reformulate (2) as

$$\begin{aligned} f(\tilde{\boldsymbol{x}}) &= \frac{1}{2\sigma_e^2}\|\boldsymbol{H}(\boldsymbol{H}^\dagger\boldsymbol{y} - \tilde{\boldsymbol{x}})\|_2^2 + s(\tilde{\boldsymbol{x}}) \\ &= \frac{1}{2\sigma_e^2}\|\boldsymbol{H}^\dagger\boldsymbol{y} - \tilde{\boldsymbol{x}}\|_{\boldsymbol{H}^T\boldsymbol{H}}^2 + s(\tilde{\boldsymbol{x}}), \end{aligned} \tag{3}$$

where

$$\boldsymbol{H}^\dagger \triangleq \boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1}, \tag{4}$$

$$\|\boldsymbol{u}\|_{\boldsymbol{H}^T\boldsymbol{H}}^2 \triangleq \boldsymbol{u}^T\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{u}. \tag{5}$$

Note that $\boldsymbol{H}^\dagger$ is the pseudoinverse of the full row rank matrix $\boldsymbol{H}$, and $\|\boldsymbol{u}\|_{\boldsymbol{H}^T\boldsymbol{H}}$ is not a real norm, since $\boldsymbol{H}^T\boldsymbol{H}$ is not a positive definite matrix in our case.

The optimization problem $\min_{\tilde{\boldsymbol{x}}} f(\tilde{\boldsymbol{x}})$ can be equivalently written with a degenerate constraint

$$\min_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}} \ \frac{1}{2\sigma_e^2}\|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{x}}\|_{\boldsymbol{H}^T\boldsymbol{H}}^2 + s(\tilde{\boldsymbol{x}}) \ \text{ s.t. } \ \tilde{\boldsymbol{y}} = \boldsymbol{H}^\dagger\boldsymbol{y}. \tag{6}$$

Now, the basic idea is to loose the variable $\tilde{\boldsymbol{y}}$ in a restricted manner, which can facilitate the estimation of $\boldsymbol{x}$. First, we give some degrees of freedom to $\tilde{\boldsymbol{y}}$ by using the constraint $\boldsymbol{H}\tilde{\boldsymbol{y}} = \boldsymbol{y}$ instead of $\tilde{\boldsymbol{y}} = \boldsymbol{H}^\dagger\boldsymbol{y}$. Next, we prevent large components of $\tilde{\boldsymbol{y}}$ in the null space of $\boldsymbol{H}$ (that may strongly disagree with the prior $s(\tilde{\boldsymbol{x}})$), by replacing the multiplication by $\frac{1}{\sigma_e^2}\boldsymbol{H}^T\boldsymbol{H}$ in the fidelity term with multiplication by $\frac{1}{(\sigma_e+\delta)^2}\boldsymbol{I}_n$, where δ is a design parameter. This leads to the following optimization problem

$$\min_{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}} \ \frac{1}{2(\sigma_e+\delta)^2}\|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{x}}\|_2^2 + s(\tilde{\boldsymbol{x}}) \ \text{ s.t. } \ \boldsymbol{H}\tilde{\boldsymbol{y}} = \boldsymbol{y}. \tag{7}$$

Note that δ introduces a tradeoff. On the one hand, too large value of δ may over-reduce the effect of the fidelity term. On the other hand, too small value of $(\sigma_e + \delta)^2$ may over-penalize $\tilde{\boldsymbol{x}}$ unless it is very close to the affine subspace $\{\boldsymbol{H}\mathbb{R}^n = \boldsymbol{y}\}$. This limits the effective feasible set of $\tilde{\boldsymbol{x}}$ in problem (7), such that it may not include potential solutions of the original problem (6). Therefore, we suggest setting the value of δ as

$$\delta = \underset{\tilde{\delta}}{\text{argmin}} \ (\sigma_e + \tilde{\delta})^2 \tag{8}$$

$$\text{s.t. } \frac{1}{\sigma_e^2}\|\boldsymbol{H}^\dagger\boldsymbol{y} - \tilde{\boldsymbol{x}}\|_{\boldsymbol{H}^T\boldsymbol{H}}^2 \geq \frac{1}{(\sigma_e+\tilde{\delta})^2}\|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{x}}\|_2^2 \ \ \forall\, \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}} \in \mathcal{S}$$

where $\mathcal{S}$ denotes the feasible set of problem (7). Note that the feasibility of $\tilde{\boldsymbol{x}}$ is dictated by $s(\tilde{\boldsymbol{x}})$ and the feasibility of $\tilde{\boldsymbol{y}}$ is dictated by the constraint in (7).

The δ that minimizes (8) implies that $\frac{1}{\sigma_e^2}\|\boldsymbol{H}^\dagger\boldsymbol{y} - \tilde{\boldsymbol{x}}\|_{\boldsymbol{H}^T\boldsymbol{H}}^2 \approx \frac{1}{(\sigma_e+\delta)^2}\|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{x}}\|_2^2$ for feasible $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$. Using this property with the fact that $\tilde{\boldsymbol{y}} = \boldsymbol{H}^\dagger\boldsymbol{y}$ is one of the solutions of the underdetermined system $\boldsymbol{H}\tilde{\boldsymbol{y}} = \boldsymbol{y}$, prevents increasing the penalty on potential solutions of the original optimization problem (6). Therefore, roughly speaking, we do not lose solutions when we solve (7) instead of (6) with δ from (8).

The IDBP algorithm uses alternating minimization to solve (7). Iteratively, $\tilde{\boldsymbol{x}}_k$ is estimated by solving

$$\tilde{\boldsymbol{x}}_k = \underset{\tilde{\boldsymbol{x}}}{\text{argmin}} \ \frac{1}{2(\sigma_e+\delta)^2}\|\tilde{\boldsymbol{y}}_{k-1} - \tilde{\boldsymbol{x}}\|_2^2 + s(\tilde{\boldsymbol{x}}), \tag{9}$$

and $\tilde{\boldsymbol{y}}_k$ is estimated by projecting $\tilde{\boldsymbol{x}}_k$ onto $\{\boldsymbol{H}\mathbb{R}^n = \boldsymbol{y}\}$

$$\begin{aligned} \tilde{\boldsymbol{y}}_k &= \underset{\tilde{\boldsymbol{y}}}{\text{argmin}} \ \|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{x}}_k\|_2^2 \ \text{ s.t. } \ \boldsymbol{H}\tilde{\boldsymbol{y}} = \boldsymbol{y} \\ &= \boldsymbol{H}^\dagger\boldsymbol{y} + (\boldsymbol{I}_n - \boldsymbol{H}^\dagger\boldsymbol{H})\tilde{\boldsymbol{x}}_k. \end{aligned} \tag{10}$$

The variable $\tilde{\boldsymbol{y}}_k$ is expected to be closer to the true signal $\boldsymbol{x}$ than the raw observations $\boldsymbol{y}$. Thus, IDBP alternates between estimating the signal and using this estimation in order to obtain improved measurements that still comply with $\boldsymbol{y}$. Note that (9) describes obtaining $\tilde{\boldsymbol{x}}_k$ using a white Gaussian denoiser with noise variance of $\sigma^2 = (\sigma_e + \delta)^2$, applied on the image $\tilde{\boldsymbol{y}}_{k-1}$. Thus, it can be written compactly as $\tilde{\boldsymbol{x}}_k = \mathcal{D}(\tilde{\boldsymbol{y}}_{k-1}; \sigma)$, where $\mathcal{D}(\cdot;\sigma)$ is a denoising operator. Therefore, similar to the P&P approach [13], IDBP does not require explicitly specifying the prior function $s(\boldsymbol{x})$. Instead, $s(\boldsymbol{x})$ is implicitly defined by the choice of $\mathcal{D}(\cdot;\sigma)$.

[1] Additional explanations and a theoretic analysis can be found in [12].

2.1. Setting the value of the parameter δ

In practice, it is not clear how to obtain such δ that solves (8). Therefore, in order to relax the condition in (8), which should be satisfied by all $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ in $\mathcal{S}$, we can focus only on the sequences $\{\tilde{\boldsymbol{x}}_k\}$ and $\{\tilde{\boldsymbol{y}}_k\}$ generated by the proposed alternating minimization process. Then, we can use the following proposition, which is proved in [12].

Proposition 1. *Set $\delta = \tilde{\delta}$. If there exist an iteration k of IDBP that violates the following condition*

$$\frac{1}{\sigma_e^2}\|\boldsymbol{y} - \boldsymbol{H}\tilde{\boldsymbol{x}}_k\|_2^2 \geq \frac{1}{(\sigma_e + \tilde{\delta})^2}\|\boldsymbol{H}^\dagger(\boldsymbol{y} - \boldsymbol{H}\tilde{\boldsymbol{x}}_k)\|_2^2, \quad (11)$$

then $\delta = \tilde{\delta}$ also violates the condition in (8).

Note that (11) can be easily evaluated for each iteration. Thus, violation of (8) can be spotted by violation of (11), and used for stopping the process, increasing δ and running the algorithm again. The opposite direction does not hold. Even when (11) is satisfied for all iterations, it does not guarantee satisfying (8). However, the relaxed condition (11) provides an easy way to set δ with an approximation to the solution of (8), which gives good empirical results [12].

2.2. The deblurring case

In the (non-blind) deblurring problem, for a circular shift-invariant blur operator whose kernel is $\boldsymbol{h}$, IDBP can be efficiently implemented using Fast Fourier Transform (FFT). Recall that in this case $\boldsymbol{H}$ is an ill-conditioned $n \times n$ matrix. Therefore, $\boldsymbol{H}^\dagger$ is replaced with a regularized inversion of $\boldsymbol{H}$, using standard Tikhonov regularization, which is given in the Fourier domain by

$$\tilde{\boldsymbol{g}} \triangleq \frac{\mathcal{F}^*\{\boldsymbol{h}\}}{|\mathcal{F}\{\boldsymbol{h}\}|^2 + \epsilon \cdot \sigma_e^2}, \quad (12)$$

where ϵ is a parameter that controls the amount of regularization in the approximation of $\boldsymbol{H}^\dagger$, and $\mathcal{F}\{\cdot\}$ denotes the FFT operator. Then, (10) can be computed by

$$\tilde{\boldsymbol{y}}_k = \mathcal{F}^{-1}\Big\{\tilde{\boldsymbol{g}}\Big(\mathcal{F}\{\boldsymbol{y}\} - \mathcal{F}\{\boldsymbol{h}\}\mathcal{F}\{\tilde{\boldsymbol{x}}_k\}\Big)\Big\} + \tilde{\boldsymbol{x}}_k. \quad (13)$$

For an *exact* blur kernel, it was observed (see Fig. 3 in [12]) that pairs of (δ, ϵ) that give the best results indeed satisfy condition (11) for all iterations, while pairs of (δ, ϵ) that lead to bad results often violate this condition at early iteration. Note that if the division of the LHS of (11) by its RHS is less than 1, even for a single iteration, it means that the original condition in (8) is violated by the associated (δ, ϵ). However, since the opposite direction does not hold, a small confidence margin (e.g. 3) on the ratio should be kept.

The above observations led to the auto-tuned IDBP (Algorithm 3 in [12]), where δ is fixed and ϵ is automatically tuned. Starting with a small value of ϵ, the ratio LHS/RHS of (11) is evaluated at the end of each IDBP iteration (except the first one, which strongly depends on the initial $\tilde{\boldsymbol{y}}_0$). If the ratio is smaller than a threshold τ, then ϵ is slightly increased and IDBP is restarted. Auto-tuned IDBP, with plugged-in BM3D denoiser, has shown competitiveness with IDD-BM3D, whose parameters are *manually optimized per different kernel*.

In this work, we examine the use of the auto-tuned IDBP as a final step in the blind deblurring task, where $\boldsymbol{h}$ is unknown, and only its estimate is available. First, we observe that a stable convergence requires a larger confidence margin, presumably due to the uncertainty in the kernel. While this increase obviously depends on the quality of the kernel estimation, in this work we use a fixed $\tau = 9$ for the examined kernel estimation methods [5, 9]. Second, we observe that while for a known $\boldsymbol{h}$ the LHS/RHS ratio is similar for almost all iterations, for an estimated kernel the ratio significantly grows between iterations. This can be interpreted as increasing the regularization too much, and results in loss of fine details. Thus, we slightly modify the auto-tuning scheme: if the LHS/RHS ratio is above $\tau + 1$ then we decrease δ by a factor Π_δ close to 1. The modified scheme is presented in Algorithm 1. We note that we have not encountered any restart due to decreasing δ using the default Π_δ. The discussed behavior and the efficiency of the modification are demonstrated in Fig. 1.

Algorithm 1: Auto-tuned IDBP for deblurring with inexact kernel

Input: $\boldsymbol{h}, \boldsymbol{y}, \sigma_e$, denoising operator $\mathcal{D}(\cdot;\sigma)$, stopping criterion. $\boldsymbol{y} \approx \boldsymbol{x} * \boldsymbol{h} + \boldsymbol{e}$, such that $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma_e^2 \boldsymbol{I}_n)$ and $\boldsymbol{x}$ is an unknown signal whose prior model is specified by $\mathcal{D}(\cdot;\sigma)$.

Output: $\hat{\boldsymbol{x}}$ an estimate for $\boldsymbol{x}$.

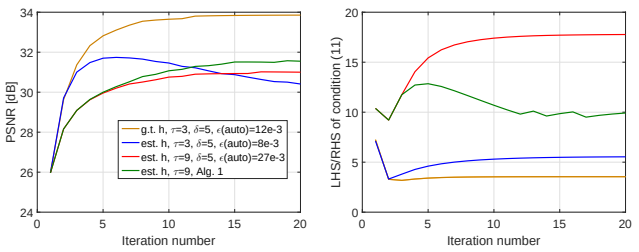
Params.: $\tilde{\boldsymbol{y}}_0 =$ some initialization, $k = 0$, $\delta =$ moderate value, $\epsilon =$ small initial value, $\Delta\epsilon =$ small increment, $\Pi_\delta =$ decreasing factor close to 1, $\tau =$ confidence margin.

Default init.: $\tilde{\boldsymbol{y}}_0 = \boldsymbol{y}$, $\delta = 5$, $\epsilon =$10e-3, $\Delta\epsilon$=1e-3, $\Pi_\delta = 0.92$, $\tau = 9$.

while *stopping criterion not met* **do**
- $k = k + 1$;
- $\tilde{\boldsymbol{x}}_k = \mathcal{D}(\tilde{\boldsymbol{y}}_{k-1}; \sigma_e + \delta)$;
- Compute $\tilde{\boldsymbol{y}}_k$ using (13) (note that $\tilde{\boldsymbol{g}}$ depends on ϵ);
- **if** $k > 1$ *and LHS/RHS of (11)* $< \tau$ **then**
  - $\epsilon = \epsilon + \Delta\epsilon$;
  - Restart process: $k = 0$;
- **else if** *LHS/RHS of (11)* $> \tau + 1$ **then**
  - $\delta = \delta \cdot \Pi_\delta$;

end

$\hat{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_k$;



(a) (b)

Fig. 1: Auto-tuned IDBP (with EPLL denoiser) for *image 1* and *kernel 3* from [2] for confidence margin $\tau = \{3, 9\}$ and $\boldsymbol{h} = \{$ground truth, estimated by [9]$\}$: (a) PSNR vs. iteration number; (b) LHS / RHS of (11) vs. iteration number.

Table 1: Average PSNR [dB] for different kernel estimation and non-blind deblurring methods for the dataset of [2].

	EPLL	IDD-BM.	IDBP-EPLL	IDBP-BM3D
Kernel [5]	27.47	26.98	**27.50**	27.43
Kernel [9]	28.53	27.16	**28.94**	28.87

3. EXPERIMENTS

We compare the performance of Algorithm 1 (with the default setting) to the state-of-the-art deblurring methods EPLL [10] and IDD-BM3D [11]. For Algorithm 1, we use BM3D and EPLL denoisers, and denote the resulting methods by IDBP-BM3D and IDBP-EPLL, respectively. The stopping criterion for both IDBP versions is 15 iterations. Two methods [5, 9] are used to obtain 31×31 kernel estimations. In all cases, we use the Matlab code supplied by the authors. Contrary to EPLL's implementation that uses a uniform parameter setting, IDD-BM3D has different settings for each scenario in [11]. Since in blind deblurring the kernel is unknown, we use only the tuning of Scenario 5[2] in [11], which seems to give the best results in general. Similar to [7, 9], all evaluations are done using the center portion, hence we discard 15 pixels from each border of the ground truth image. Also, since the blur kernel can only be recovered up to a global translation, we align the deblurred image with the ground-truth image using a fast sub-pixel rigid translation [21] before computing the error.

The average PSNR results for the natural blurred images dataset of [2] are shown in Table 1. We note that all methods use noise level $\sigma_e = 2/255$.[3] Note the small performance gap between the two IDBP versions. Also, we remark that IDBP-BM3D is faster due to the fast implementation of the BM3D denoiser.

Figures 2 and 3 visually compare the results of EPLL and IDBP-BM3D for two colored natural blurry images (due to space limitation, only fragments of the images are presented). We use the kernel estimation of [5], with zeroing of elements bellow 0.15 of the maximal value. This slight modification improves the kernel estimation, and hence the results for all methods. While EPLL is performed on each RGB channel separately, IDBP-BM3D is implemented efficiently using a fast colored version of the BM3D denoiser [22] (we note that condition (11) is evaluated for the grayscale version). In Fig. 2 it can be seen that EPLL has more artifacts than IDBP-BM3D (e.g. at the chin). In Fig. 3, IDBP-BM3D result looks sharper (e.g. notice the paving-stones).

[2]This scenario assumes a Gaussian blur with spatial standard deviation of 1.6 and noise level of $\sigma_e = 2/255$.

[3]This is approximately the standard deviation of the residual images, obtained by the ground truth images and kernels.

Fig. 2: From left to right: blurry image and estimated kernel, EPLL recovery, and IDBP-BM3D recovery.

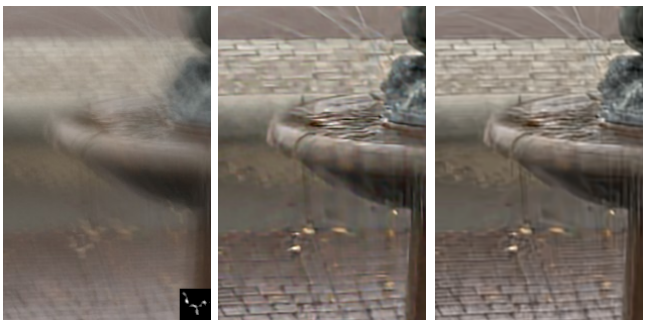


Fig. 3: From left to right: blurry image and estimated kernel, EPLL recovery, and IDBP-BM3D recovery.

4. CONCLUSION

We considered the widely used approach for blind deblurring, which requires a final non-blind deblurring step. We showed that implementing this last step using the IDBP method, equipped with a suitable automatic parameter tuning mechanism, can improve the performance compared to other deblurring algorithms, which are restricted to a uniform tuning in the blind-deblurring setting. Especially, IDBP-BM3D seems to be a good and fast BM3D-based alternative to EPLL, which is currently used more often in the considered setting [7, 9]. Future work may combine the proposed approach with iterative kernel estimation, in which the estimated kernel is improved using the deblurred image, or a new kernel is estimated for the deblurred image, as done in [23].

5. REFERENCES

[1] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *ACM transactions on graphics (TOG)*, vol. 25, pp. 787–794, 2006.

[2] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1964–1971, 2009.

[3] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2657–2664, 2011.

[4] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *ACM transactions on graphics (TOG)*, vol. 27, p. 73, 2008.

[5] J. Kotera, F. Šroubek, and P. Milanfar, "Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors," in *International Conference on Computer Analysis of Images and Patterns*, pp. 59–66, Springer, 2013.

[6] S. Cho and S. Lee, "Fast motion deblurring," in *ACM Transactions on Graphics (TOG)*, vol. 28, p. 145, ACM, 2009.

[7] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *IEEE International Conference on Computational Photography (ICCP)*, pp. 1–8, 2013.

[8] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1107–1114, 2013.

[9] T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *European Conference on Computer Vision (ECCV)*, pp. 783–798, Springer, 2014.

[10] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 479–486, 2011.

[11] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, 2012.

[12] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *arXiv preprint arXiv:1710.06647*, 2017.

[13] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 945–948, 2013.

[14] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 408–423, 2016.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[16] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[17] P. C. Hansen and D. P. OLeary, "The use of the L-curve in the regularization of discrete ill-posed problems," *SIAM Journal on Scientific Computing*, vol. 14, no. 6, pp. 1487–1503, 1993.

[18] E. Haber and D. Oldenburg, "A GCV based method for nonlinear ill-posed problems," *Computational Geosciences*, vol. 4, no. 1, pp. 41–63, 2000.

[19] Y. C. Eldar, "Generalized SURE for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2009.

[20] R. Giryes, M. Elad, and Y. C. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 407–422, 2011.

[21] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," *Optics letters*, vol. 33, no. 2, pp. 156–158, 2008.

[22] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *IEEE Int. Conf. Imag. Proc (ICIP)*, pp. I–313, 2007.

[23] N. Zon, R. Hanocka, and N. Kiryati, "Fast and easy blind deblurring using an inverse filter and probe," in *Computer Analysis of Images and Patterns*, (Cham), pp. 269–281, Springer International Publishing, 2017.