ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.6

Reconstruct an Image from a Blurred image by using Shrinkage-Thresholding Operators

Kattavinod kumar[1], Muvva srikanth[2]
M.Tech Student[1], Assistant Professor[2]
Department of ECE
GVR & S College of Engineering & Technology, Andhra Pradesh, India

Abstract:
Significant edge selection and time-varying regularization are two critical techniques to guarantee the success of maximum a posterior (MAP)-based unsighted deconvolution. However, the existing approaches usually rely on carefully designed regularizers and handcrafted parameter alteration to obtain acceptable inference of the shape kernel. Many regularizers exhibit the configuration-preserving smooth capability, but fail to enhance salient edges. In this paper, under the MAP framework, we propose the iteration-wise `p-norm regularizers together with data-driven strategy to address these issues. First, we extend the generalized shrinkage-thresholding (GST) operator for `p- norm minimization with negative p value, which can grind salient edges while suppressing trivial details. Then, the iteration wise GST parameters are specified to allow dynamical salient edge choice and time-varying regularization. Finally, instead of handcrafted correction, a principled discriminative learning approach is proposed to learn the iteration-wise GST operators from the training dataset. Furthermore, the multi-scale scheme is developed to improve the competence of the algorithm. Experimental results show that, negative p value is more effective in estimating the coarse shape of smudge kernel at the early stage, and the learned GST operators can be well generalized to other dataset and real world blurry images. Compared with the state of- the-art methods, our method achieves better deblurring results in terms of both quantitative metrics and optical quality, and it is much faster than the state-of-the-art patch-based blind deconvolution method.

Index Terms: unsighted deconvolution, kernel judgment, image deblurring, hyper-Laplacian, discriminative learning.

I. INTRODUCTION

Blind image deconvolution aims to recover the latent sharp image x and blur kernel k from the blurry observation
y = k x + n; (1)
where denotes 2D convolution and n is additive Gaussian white noise. Blind image deconvolution generally involves two stages, i.e., blur kernel estimation and non-blind deconvolution, where the former is crucial to the success of the algorithm. There are two classes of popular blur kernel estimation strategies: variational Bayes (VB)-based and maximum a posterior (MAP)-based ones. Levin et al. showed that naive MAP prefers trivial delta kernel solution while the VB-based approaches are more robust in estimating the blur kernel. This observation has motivated several VB based methods for image deconvolution. However, the approximation of integration is required, making VB-based methods computationally inefficient. Recently, with the introduction of salient edge selection and time-varying regularization, interest in improved MAP has been revived for efficient blind deconvolution with many representative methods. MAP jointly estimates the pair (k; x) by maximizing a posterior probability,

$$\max_{\mathbf{x},\mathbf{k}} \Pr(\mathrm{x},\mathrm{k}|\mathrm{y}) \propto \max_{\mathbf{x},\mathbf{k}} \Pr(\mathrm{y}|\mathrm{x},\mathrm{k}) \Pr(\mathrm{x}) \Pr(\mathrm{k}), \qquad (2)$$

where Pr(x; kjy) denotes a posterior on (x; k), Pr(x) and Pr(k) are the priors of the latent sharp image and the blur kernel, and Pr(yjx; k) denotes the likelihood of the observation y. The MAP model can be equivalently rewritten as

$$\min_{\mathbf{k},\mathbf{x}} \frac{\lambda}{2\sigma_n^2}\|\mathrm{k}\otimes\mathrm{x}-\mathrm{y}\|^2+\phi(\mathrm{x})+\mu\varphi(\mathrm{k}), \qquad (3)$$

Our work is motivated by the two key techniques, i.e., salient edge selection and time-varying regularization, which have been widely adopted in MAP-based blind deconvolution. However, we re-analyze these techniques by raising three questions:
1) Salient edge selection is widely used to explicitly or implicitly recover salient edges to facilitate kernel estimation In shock and bilateral filters are employed in each iteration to enhance strong edges
while suppressing harmful small-scale textures. Actually, bilateral filter is a smoothing operator and shock filter is a sharpening operator, while the regularizers like `0-norm result in a structure-preserving smoothing operator. Thus, our first question is: is it possible to extend the existing regularizers, e.g., `p-norm, to achieve both smoothing and sharpening capability?
2) Time-varying regularization is also widely adopted in blind deconvolution. To better estimate the blur kernel k, the salient edges should be dynamically recovered to guide the algorithm gradually converge to the desired solution. For example, in parameters of the shock and bilateral filters are tuned to select the strongest edges at first, and subsequently the estimated kernel is refined by the gradually added details. In the regularization parameter _ is set small in the first a few iterations to preserve strong edges, and then gradually increases along with iteration to produce accurate blur kernel. So, our second question is: is there a family of priors (each iteration has its own parameters) for blind deconvolution?
3) Most existing approaches involve carefully designed regularizers and handcrafted parameter tuning to guide the algorithms to converge to the desired solution. It is interesting to ask the question: can we learn the iteration-wise regularization parameters using the data-driven strategy?

Algorithm 1 Multi-scale image deconvolution

Input: Blurry image y, scale number S
Output: Blur kernel k and latent image x
1: Initializing $\mathrm{d}^{(S-1)}$ and $\mathrm{k}^{(S-1)}$
2: **for** $s = S-1$ to 0 **do**
3: Downsampling y to $\mathrm{y}^{(s)}$
4: Inputing $\mathrm{y}^{(s)}$, $\mathrm{d}^{(s)}$, $\mathrm{k}^{(s)}$ and $\{\theta^{(s,1)}, ..., \theta^{(s,t)}, ..., \theta^{(s,T)}\}$ to Algorithm 2 that returns $\mathrm{d}^{(s)}$ and $\mathrm{k}^{(s)}$
5: **if** $s > 0$ **then**
6: Upsamping $\mathrm{d}^{(s)}$ and $\mathrm{k}^{(s)}$ to $\mathrm{d}^{(s-1)}$ and $\mathrm{k}^{(s-1)}$
7: **end if**
8: **end for**
9: $\mathrm{k} = \mathrm{k}^{(0)}$
10: Given k, recovering x by non-blind deconvolution

Input: Blurry image y, $\mathrm{d}^{(0)}$, $\mathrm{k}^{(0)}$ and $\{\theta^{(1)}, ..., \theta^{(t)}, ..., \theta^{(T)}\}$
Output: Blur kernel k and latent gradient d
1: **for** $t = 1$ to T **do**
2: *// Lines 3-4 solve* **d***-step* Eq. (18)
3: $\mathrm{w}^{(t)} = \mathrm{GST}\left(\mathrm{d}^{(t-1)}, p^{(t)}, 1/\beta^{(t)}\right)$
4: $\mathrm{d}^{(t)} = \Omega^{-1}\left(\eta + \beta^{(t)}\mathrm{w}^{(t)}\right)$
5: *// Lines 6-8 solve* **k***-step* Eq. (27)
6: $\mathrm{h}^{(t)} = \arg\min_{\mathrm{h}} \frac{\delta_1^{(t)}}{2}\|\mathrm{k}^{(t-1)} - \mathrm{h}\|^2 + B(\mathrm{h})$
7: $\mathrm{g}^{(t)} = \mathrm{GST}\left(\mathrm{k}^{(t-1)}, 0.5, \mu^{(t)}/\delta_1^{(t)}\right)$
8: $\mathrm{k}^{(t)} = \Phi^{-1}\zeta$
9: Updating $\beta^{(t+1)}$, $\delta_1^{(t+1)}$, $\delta_2^{(t+1)}$, $\delta_3^{(t+1)}$
10: **end for**
11: $\mathrm{k} = \mathrm{k}^{(t)}$ and $\mathrm{d} = \mathrm{d}^{(t)}$

Algorithm 3 Learning GST operators over scales

Input: Training set $\mathcal{D}$, scale number S
Output: $\left\{\{\theta^{(S-1,t)}\}_{t=1}^{T}, ..., \{\theta^{(s,t)}\}_{t=1}^{T}, ..., \{\theta^{(0,t)}\}_{t=1}^{T}\right\}$
1: Denoting θ' as α, μ, β, δ_1, δ_2, δ_3 and initializing $\theta'^{(S,T)}$
2: **for** $s = S-1$ to 0 **do**
3: Downsampling training set $\mathcal{D}$ to $\mathcal{D}^{(s)}$
4: Initializing $\theta'^{(s,1)}$ as $\theta'^{(s+1,T)}$
5: Inputing $\mathcal{D}^{(s)}$ and $\theta'^{(s,1)}$ to Algorithm 4 to learn the optimal parameters of scale s, i.e. $\{\theta^{(s,t)}\}_{t=1}^{T}$
6: **end for**

Algorithm 4 Learning GST operators on scale s

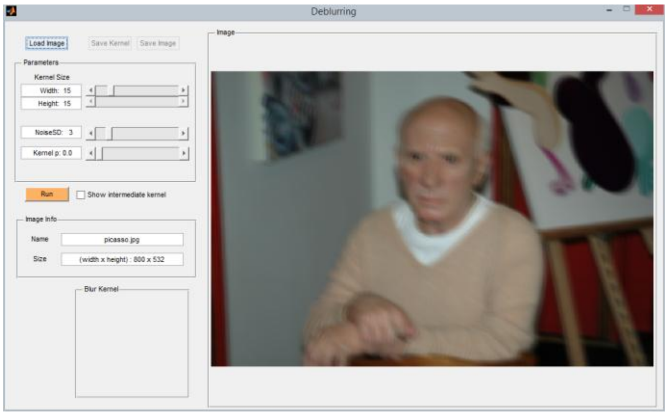
Input: Training set $\mathcal{D}$ and $\alpha^{(1)}$, $\mu^{(1)}$, $\beta^{(1)}$, $\delta_1^{(1)}$, $\delta_2^{(1)}$, $\delta_3^{(1)}$
Output: $\left\{\theta^{(1)}, ..., \theta^{(t)}, ..., \theta^{(T)}\right\}$
1: **for** $t = 1$ to T **do**
2: grad = 0
3: **for** $i = 1$ to N **do**
4: Updating $\mathrm{d}_i^{(t)}$ Eq. (18) and $\mathrm{k}_i^{(t)}$ Eq. (27)
5: $\mathrm{grad} = \mathrm{grad} + \partial L_i^{(t)}(\theta)/\partial\theta$
6: **end for**
7: Using gradient based L-BFGS method to search optimal $\theta^{(t)}$
8: Updating $\alpha^{(t+1)}$, $\mu^{(t+1)}$ and penalty parameters $\beta^{(t+1)}$, $\delta_1^{(t+1)}$, $\delta_2^{(t+1)}$, $\delta_3^{(t+1)}$
9: **end for**

II. IMPLEMENTATION

In our implementation, some extra constraints are taken to improve the robustness and stability of the learned iteration wise GST operators. In [8, 15, 19, 20], the regularization parameter _ begins with some small value and gradually increases along with the iteration numbers. As to the p value, p = 0 is first adopted to estimate the blur kernel, and then p = 0:5 is adopted for the final restoration. As a summary, both _ and p values should be non-decreasing along with the iteration numbers. Thus, in our greedy learning procedure, for each scale s, the non-decreasing constraints on both _ and p are imposed, i.e., _(s;t+1) _ _(s;t) and p(s;t+1) _ p(s;t), and over the scales, it is reasonable to consider the constraints _(s□1;1) _ _(s;T) and p(s□1;1) _ p(s;T). As to the search range, _ is constrained in [0:5; 5], and p is constrained. Moreover, we set the scale number S = 5, the number of inner iterations in each scale T = 20, and the dowmsampling rate as 2. Thus, there are 200 GST parameters to be learned in the proposed iteration-wise MAP framework. Other parameters, including regularization weight _ and penalty parameters _; _1; _2; _3, should also be non-decreasing along with the scales and iterations. Specifically, the regularization weight _ on blur kernel k is initialized as 1 _ 10□6,

III. EXPERIMENTAL RESULTS

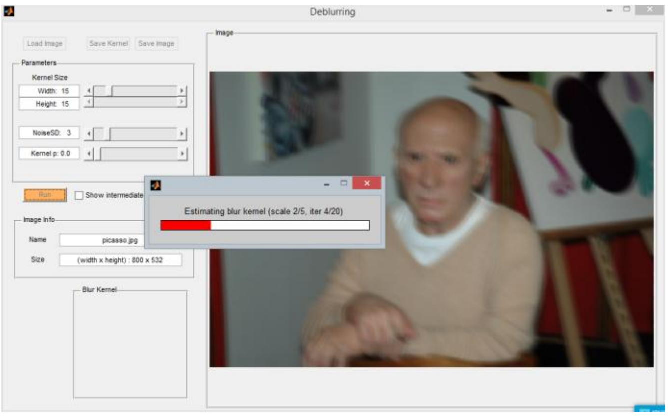
In this subsection we evaluate the performance of the proposed method on real blurry photographs, and compare it with the the top two competing methods based on Tables and III, i.e., Fig. 13 shows
Run “Deblurring.m” to open GUI, and click “Load Image”.



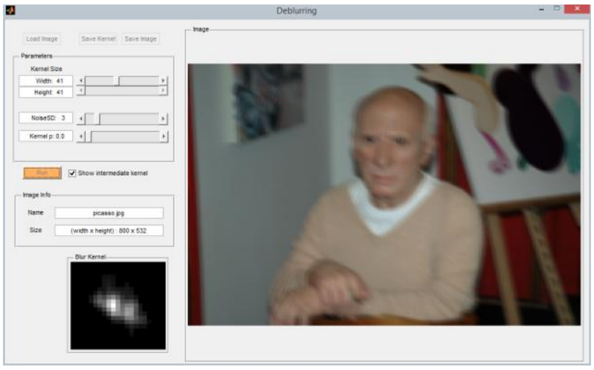
For the parameters, the default setting can work well for most images. If necessary, 3 parameters can be tuned.

1) **Kernel Size**: width and height keep the same.
2) **Noise SD**: std. of Gaussian noise, which ranges in [1, 25]. For real blurry images, the value in 2-10 can provide satisfactory results.
3) **Kernel p**: sparsity of blur kernel, p value ranges in [0, 1], and the value in [0, 0.5] works well.

Click “Run” to start kernel estimation.



If “Show intermediate kernel”, the current estimated kernel will be shown.



The final blur kernel and clear image can be saved by "save kernel" and "save image".

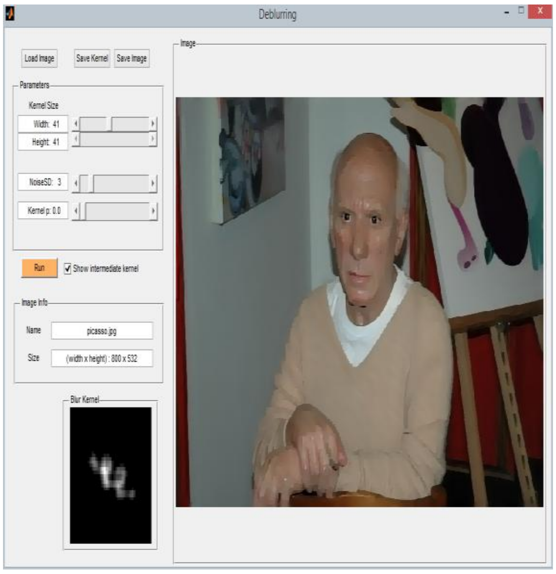


Table.1.Recommended settings for demo images

	Kernel size	Noise std.	Kernel p value
Blurred11	Default	Default	Default
Blurred13	Default	Default	Default
Blurred14	Default	Default	Default
Blurred15	Default	Default	Default
Boat	Default	Default	Default
Fishes	Default	5	Default
Flower	Default	Default	Default
Fountain	Default	Default	Default
Mukta	Default	Default	Default
Nv	Default	Default	Default
Picasso	Default	Default	Default
Postcard	91	5	Default
Roma_large	91	5	0.2
summerhouse	Default	Default	Default

the deblurring results on three real blurry images. For the image roma, our method and Sun et al.'s method can achieve satisfactory deblurring results, while the result has visible color distortions in the red close-up. The second image is taken in low light condition. The results suffer severe distortions and noise in the red and green close-ups, while the result by our method is clearer and visually plausible. Moreover, the third image vehicle is severely blurred, and our method achieves much better result. For example, the license number can be roughly read as"N15 5826" from the deblurring result by our method, but it is difficult to be recognized from the results by both and Sun et al. For the green and yellow close-ups, although all the results are not good, the result by our method is visually more pleasant, while the distortions like ringing effects are much more severe.

IV. CONCLUSIONS

In this paper, by generalizing the GST operator to the case with $p < 0$, we proposed an iteration-wise MAP framework for blind deconvolution. Then a discriminative learning method was developed to learn iteration-wise GST operators from a blurry image set. The learned GST operators begin with $p < 0$ to avoid trivial delta kernel solution, and gradually increase with iterations for accurate blur kernel estimation. The proposed method can be directly applied to other dataset and real world blurry images. Experimental results showed that the proposed method performs better than the competing methods in terms of both quantitative metrics and visual effect, and is much faster than the state-of-the-art patch-based method. The proposed iteration-wise learning method was designed on the image gradients, and thus has limitations to model patch-level structures. In our future work, we will investigate the appropriate framework to learn iteration-wise priors for image patches or filter responses. Moreover, to improve kernel estimation performance, joint learning over iterations can also be used to fine-tune the greedily learned parameters.

V. REFERENCES

[1]. A. Levin, Y.Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in CVPR, 2009.

[2] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational bayesian blind deconvolution using a total variation prior," IEEE Transactions on Image Processing, vol. 18, no. 1, pp. 12–26, 2009.

[3] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," ACM Transactions on Graphics (TOG), vol. 25, no. 3, pp. 787– 794, 2006.

[4] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos, "Bayesian blind deconvolution with general sparse image priors," in ECCV, 2012.

[5] D. Wipf and H. Zhang, "Analysis of bayesian blind deconvolution," in Energy Minimization Methods in Computer Vision and Pattern Recognition, 2013.

[6]. D. Perrone, R. Diethelm, and P. Favaro, "Blind deconvolution via lower-bounded logarithmic image priors," in Energy Minimization Methods in Computer Vision and Pattern Recognition, 2015.

[7]. T. F. Chan and C. Wong, "Total variation blind deconvolution," IEEE Transactions on Image Processing, vol. 7, no. 3, pp. 370– 375, 1998.

[8]. D. Perrone and P. Favaro, "Total variation blind deconvolution - the devil is in the details," in CVPR, 2014.

[9]. R. M. Rameshan, S. Chaudhuri, and R. Velmurugan, "Joint map estimation for blind deconvolution: when does it work?" in Indian Conference on Computer Vision, Graphics and Image Processing, 2012.

[10]. Z. Hu, S. Cho, J. Wang, and M.-H. Yang, “Deblurring low-light images with light streaks,” in CVPR, 2014.

[11].S. Cho and S. Lee, “Fast motion deblurring,” ACM Transactions on Graphics (TOG), vol. 28, no. 5, p. 145, 2009.

[12]. L. Xu and J. Jia, “Two-phase kernel estimation for robust motion deblurring,” in ECCV, 2010.

[13]. N. Joshi, R. Szeliski, and D. Kriegman, “Psf estimation using sharp edge prediction,” in CVPR, 2008.

[14]. D. Krishnan, T. Tay, and R. Fergus, “Blind deconvolution using a normalized sparsity measure,” in CVPR, 2011.

[15]. L. Xu, S. Zheng, and J. Jia, “Unnatural l0 sparse representation for natural image deblurring,” in CVPR, 2013.

[16]. J. Jia and L. Xu, “Structure extraction from texture via relative total variation,” ACM Transactions on Graphics (TOG), vol. 31, no. 6, p. 139, 2012.

[17]. W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, “A generalized iterated shrinkage algorithm for non-convex sparse coding,” in ICCV, 2013.

[18]. Q. Shan, J. Jia, and A. Agarwala, “High-quality motion deblurring from a single image,” ACM Transactions on Graphics (TOG), vol. 27, no. 3, p. 73, 2008.

[17]. A. N. Rajagopalan and R. Chellappa, Motion deblurring algorithms and systems. Cambridge University Press, 2014.

[20]. 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), 2013