MULTI-IMAGE BLIND DEBLURRING USING A SMOOTHED NUV PRIOR AND ITERATIVELY REWEIGHTED COORDINATE DESCENT

Boxiao Ma, Jelena Trisovic, and Hans-Andrea Loeliger

ETH Zurich, Dept. of Information Technology & Electrical Engineering
{ma, loeliger}@isi.ee.ethz.ch, tjelena@student.ethz.ch

ABSTRACT

A new method for blind image deblurring is proposed that relies on a smoothed-NUV (normal with unknown variance) prior for images, which promotes piecewise smooth images with crisp edges. The proposed method can use multiple blurred versions of the same image.

The variational representation of the prior allows the joint estimation of the image and the blurring kernel(s) to be decomposed into descent steps in reweighted least-squares problems and nonlinear scalar updates of the individual variances of the prior. Specifically, we propose an iteratively reweighted coordinate descent algorithm that has no parameters. Simulation results demonstrate that the proposed approach compares favorably to state-of-the-art methods.

Index Terms— Blind image deblurring, smoothed NUV, sparsity, iteratively reweighted coordinate descent.

1. INTRODUCTION

Image deblurring has long been an important task in computer vision, and the ubiquity of hand-held photo devices (in particular, smart phones) has increased the demand for robust deblurring. Such devices are in general capable of shooting multiple images of the same scene within a short time period but with inevitable corruptions (camera shake, out of focus, etc.), which result in blurry photographs.

The deblurring problem is usually formulated in a statistical setting as follows. A color image is represented by $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_L)^\mathsf{T}$ with L color pixels $\mathbf{X}_\ell \in \mathbb{R}^3$, and we are given a set of observations (i.e., blurred images) $\mathbf{Y}_m = \mathbf{y}_m,\ m \in \{1, \ldots, M\}$. The latent image $\mathbf{X}$ and these observations are related by

$$\mathbf{Y}_m = \mathbf{K}_m * \mathbf{X} + \mathbf{Z}_m \quad m \in \{1, \ldots, M\}, \tag{1}$$

where $\mathbf{K}_m$ is the blur kernel, the symbol "$*$" denotes convolution and $\mathbf{Z}_m$ is white Gaussian noise, i.e., $\mathbf{Z}_m \sim \mathcal{N}(0, \sigma^2_{Z_m}\mathbf{I}_L)$. The problem is to produce estimates $\hat{\mathbf{x}}$ and $\hat{\mathbf{k}}_1, \ldots, \hat{\mathbf{k}}_M$ of the latent image and the blur kernels, respectively.

A great variety of algorithms for tackling the (multi-) image blind deblurring problem exists. There are methods that use priors on natural images, blur kernels, or both [1–7], methods that model the general motion blur procedure [8, 9], and methods proposed in recent years using convolutional neural networks [10–12].

Xu et al. [2] proposed a new regularizer to approximate the L_0 cost in order to recover the latent image and blur kernel. Krishnan et al. [4] used an L_1/L_2 regularization scheme which adapts L_1 regularization by reweighting the iterations using the L_2 norm of image gradients. Shan et al. [7] incorporated spatial parameters and a ringing suppression step to enforce natural image statistics. In [3], Zhang et al. proposed a normal prior which couples the latent image, blur kernels and noise level together. The prior in [3] is similar to the prior in this paper, but it lacks the smoothing capability of our prior and it makes additional assumptions on the blur kernels. Related pertinent work also includes [13] and [14].

Smoothed-NUV priors for images were introduced in [15, 16] and further elaborated in [17]. Using the plain SNUV prior of [15–17] amounts to penalizing small pixel differences quadratically and large pixel differences logarithmically; the former favors smooth areas in the image while the latter favors crisp edges between such areas. In this prior work, the linear operator relating the image and the observations was assumed to be known. In the present paper, we explore the use of the plain SNUV prior for blind image deblurring, where the operator (i.e., the blurring kernels) needs to be learned.

In the mentioned earlier work [15, 16], the image estimation was carried out by approximative versions of expectation maximization. However, iteratively reweighted coordinate descent (IRCD) as introduced in [17] has turned out to be more robust even for the tasks considered in [15, 16]. In this paper, we therefore use IRCD, which we here extend to estimating the blurring kernels as well.

The paper is structured as follows. In Section 2, we review the smoothed-NUV prior for images and the resulting cost function. In Section 3, we describe the IRCD algorithm for minimizing the cost function and we address a few technicalities. Some simulation results are presented in Section 4. Section 5 concludes the paper.

2. THE ESTIMATION PROBLEM

In this paper, we assume the pixels $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_L)$ to be arranged in a 2-dimensional rectangular grid. The set of neighbors $\Delta \subset \{1, \ldots, L\}^2$ is defined by $(k, \ell) \in \Delta$ if and only if $k < \ell$ and the pixels $\mathbf{X}_k$ and $\mathbf{X}_\ell$ are immediate neighbors in some row or column, i.e., the Manhattan distance between these pixels equals one.

The measurement model in (1) defines a Gaussian likelihood $p(\mathbf{y}_1, \ldots, \mathbf{y}_M | \mathbf{x}, \mathbf{k}_1, \ldots, \mathbf{k}_M)$. Our joint estimate of the image and the kernel(s) is of the form

$$\underset{\mathbf{x};\, \mathbf{k}_1, \ldots, \mathbf{k}_M}{\operatorname{argmin}} \left(\sum_{m=1}^{M} \frac{\|\mathbf{y}_m - \mathbf{k}_m * \mathbf{x}\|^2}{2\sigma_{Z_m}^2} + \tilde{\kappa}(\mathbf{x}) \right), \qquad (2)$$

with a regularization term $\tilde{\kappa}(\mathbf{x})$ as described below in Section 2.1. In statistical terms, the function $\rho(\mathbf{x}) \triangleq e^{-\tilde{\kappa}(\mathbf{x})}$ may be viewed as a (possibly improper) prior on $\mathbf{X}$, in which case (2) is a joint maximum-a-posteriori (MAP) estimate of the image and the kernel(s).

The minimization in (2) is not a convex optimization problem. In consequence, there may be multiple local minima, one of which will be chosen by the algorithm (to be described) and its initialization. Empirically, this does not seem to be a problem.

For the sake of clarity, we restrict ourselves for the moment to a single observation $\mathbf{y}$ of a grayscale image $\mathbf{X} = (X_1, \ldots, X_L)^\mathsf{T}$ (with $X_\ell \in \mathbb{R}$) and the corresponding blur kernel $\mathbf{k} = (k_1, \ldots, k_J)$. The extension to color images and to multiple observations is straightforward and described in Section 3.4.

2.1. The Plain SNUV Prior [17]

In this paper, the function $\tilde{\kappa}(\mathbf{x})$ in (2) has the form

$$\tilde{\kappa}(\mathbf{x}) = \sum_{(k,\ell) \in \Delta} \kappa(x_k - x_\ell) \qquad (3)$$

with

$$\kappa(u_{k,\ell}) \triangleq \min_{\sigma_{k,\ell} \geq 0} \left(\frac{u_{k,\ell}^2}{2(\sigma_0^2 + \sigma_{k,\ell}^2)} + \frac{1}{2} \log(\sigma_0^2 + \sigma_{k,\ell}^2) \right) \qquad (4)$$

and $u_{k,\ell} \triangleq x_k - x_\ell$. As shown in [17], the corresponding prior $\rho(u_{k,\ell}) \triangleq e^{-\kappa(u_{k,\ell})}$ is the sum of two zero-mean Gaussians, one with fixed variance σ_0^2 and the other with unknown variance $\sigma_{k,\ell}^2$, as illustrated in Fig. 1. Following [17], this prior is called *plain smoothed* NUV, where "smoothed" refers to $\sigma_0^2 > 0$ and "plain" refers to the absence of an extra prior on $\sigma_{k,\ell}^2$.

The variational representation (4) is the key to the algorithm that will be discussed in Section 3.

The minimizing $\sigma_{k,\ell}$ in (4) is easily determined to be

$$\hat{\sigma}_{k,\ell}^2 = \begin{cases} u_{k,\ell}^2 - \sigma_0^2, & \text{if } u_{k,\ell}^2 > \sigma_0^2 \\ 0, & \text{otherwise,} \end{cases} \qquad (5)$$

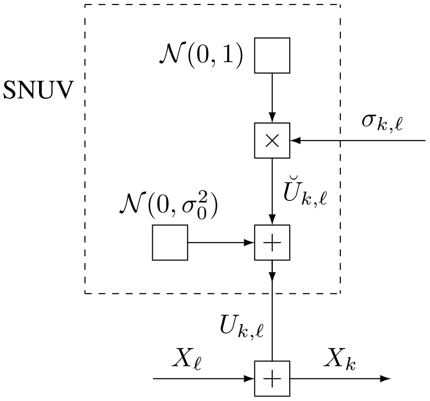


Fig. 1. Factor graph of the plain SNUV prior. $\mathcal{N}(m, \sigma^2)$ denotes a normal distribution with mean m and variance σ^2. The variable $U_{k,\ell} = X_k - X_\ell$ is the difference between two neighboring pixels X_k and X_ℓ.

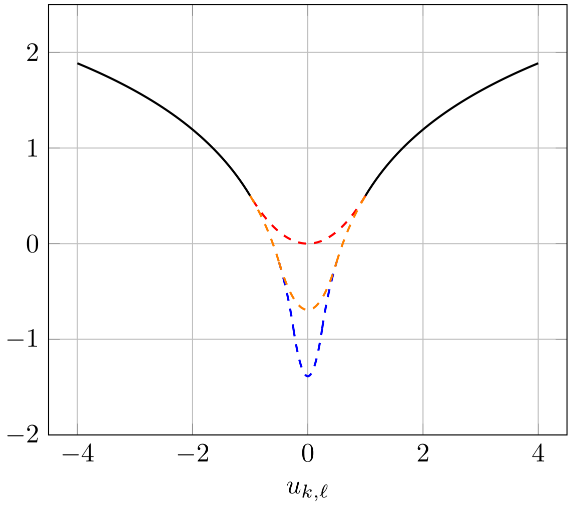


Fig. 2. Plain SNUV cost function (6). Dashed red: $\sigma_0^2 = 1$. Dashed orange: $\sigma_0^2 = 1/4$. Dashed blue: $\sigma_0^2 = 1/16$.

resulting in

$$\kappa(u_{k,\ell}) = \begin{cases} \log|u_{k,\ell}| + \frac{1}{2}, & \text{if } u_{k,\ell}^2 > \sigma_0^2 \\ \dfrac{u_{k,\ell}^2}{2\sigma_0^2} + \log|\sigma_0|, & \text{otherwise.} \end{cases} \qquad (6)$$

For $\sigma_0^2 \neq 0$, the function (6) is continuous and everywhere differentiable (even at the points $u_{k,\ell} = \pm\sigma_0$), cf. Fig. 2.

The convex (quadratic) part of (6) smoothes small pixel differences while the concave (logarithmic) part encourages sharp edges.

2.2. The Cost Function

Using (3) and (4), the estimation problem (2) can now be written as

$$(\hat{\mathbf{x}}, \hat{\mathbf{k}}) = \underset{\mathbf{x},\, \mathbf{k}}{\operatorname{argmin}} \left(\frac{\|\mathbf{y} - \mathbf{k} * \mathbf{x}\|^2}{2\sigma_Z^2} + \sum_{(k,\ell) \in \Delta} \kappa(x_k - x_\ell) \right) \qquad (7)$$

$$= \operatorname*{argmin}_{\mathbf{x},\,\mathbf{k}} \min_{\boldsymbol{\sigma}} f(\mathbf{x},\mathbf{k},\boldsymbol{\sigma}), \tag{8}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^{|\Delta|}$ is the vector of all $\sigma_{k,\ell}$ with $(k,\ell) \in \Delta$ (in some arbitrary order) and

$$f(\mathbf{x},\mathbf{k},\boldsymbol{\sigma}) \triangleq \frac{\|\mathbf{y}-\mathbf{k}*\mathbf{x}\|^2}{2\sigma_Z^2} + \sum_{(k,\ell)\in\Delta} \frac{(x_k - x_\ell)^2}{2(\sigma_0^2+\sigma_{k,\ell}^2)} + \frac{1}{2}\sum_{(k,\ell)\in\Delta} \log(\sigma_0^2+\sigma_{k,\ell}^2). \tag{9}$$

3. MINIMIZING THE COST FUNCTION

We now extend the IRCD algorithm of [17] to cope with the minimization of (9).

3.1. Iteratively Reweighted Coordinate Descent (IRCD)

The minimization (i.e., finding a local minimum) of (9) can be carried out by iterating the following three steps:

1. A descent step in $\mathbf{x}$ with fixed $\mathbf{k} = \mathbf{k}^{(\nu-1)}$, and $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(\nu-1)}$.
2. A descent step in $\mathbf{k}$ with fixed $\mathbf{x} = \mathbf{x}^{(\nu)}$, and $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{(\nu-1)}$.
3. Minimization over $\boldsymbol{\sigma}$ with fixed $\mathbf{x} = \mathbf{x}^{(\nu)}$, $\mathbf{k} = \mathbf{k}^{(\nu)}$.

Beginning with initial values $\mathbf{x}^{(0)}$, $\mathbf{k}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$, such an algorithm computes a sequence $\mathbf{x}^{(1)}$, $\mathbf{k}^{(1)}$, $\boldsymbol{\sigma}^{(1)}$, $\mathbf{x}^{(2)}$, $\mathbf{k}^{(2)}$, $\boldsymbol{\sigma}^{(2)}$, ... that normally converges to a local minimum of (9). (We have no proof of convergence, but we have never observed nonconvergence in practice.)

It is obvious from (9) that Step 3 splits into scalar minimizations which have the closed-form solution (5) with $u_{k,\ell} = x_k^{(\nu)} - x_\ell^{(\nu)}$.

Both Steps 1 and 2 are descent steps in a quadratic cost function, cf. [17]. In this paper (as in [17]), we implement each of these steps by a round of coordinate descent as follows. For Step 1, for $\ell = 1 \ldots L$,

$$x_\ell^{(\nu)} = \operatorname*{argmin}_{x_\ell} f\Big(\tilde{\mathbf{x}}_\ell^{(\nu)}, \mathbf{k}^{(\nu-1)}, \boldsymbol{\sigma}^{(\nu-1)}\Big), \tag{10}$$

with $\tilde{\mathbf{x}}_\ell^{(\nu)} = \big(x_1^{(\nu)}, \ldots, x_{\ell-1}^{(\nu)}, x_\ell, x_{\ell+1}^{(\nu-1)}, \ldots, x_L^{(\nu-1)}\big)$. Likewise for Step 2, for $j = 1 \ldots J$,

$$k_j^{(\nu)} = \operatorname*{argmin}_{k_j} f\Big(\mathbf{x}^{(\nu)}, \tilde{\mathbf{k}_j}^{(\nu)} \boldsymbol{\sigma}^{(\nu-1)}\Big), \tag{11}$$

with $\tilde{\mathbf{k}}_j^{(\nu)} = \big(k_1^{(\nu)}, \ldots, k_{j-1}^{(\nu)}, k_j, k_{j+1}^{(\nu-1)}, \ldots, k_J^{(\nu-1)}\big)$. Each of these scalar minimizations is a simple least-squares problem with a closed-form solution.

One round of this IRCD algorithm requires about the same number of computations as computing the gradient in a steepest-descent algorithm. We find IRCD to work very well in practice (but many iterations may be needed).

3.2. Rescaling the Estimates

Another preferable property of this algorithm is scale invariance, which many other models do not possess. It means that if $\hat{\mathbf{x}}$ and $\hat{\mathbf{k}}$ are the estimates obtained with the initialization $\mathbf{x}^{(0)}$, $\mathbf{k}^{(0)}$, $\boldsymbol{\sigma}^{(0)}, \sigma_0$, then $\alpha\hat{\mathbf{x}}$ and $\alpha^{-1}\hat{\mathbf{k}}$ are the optimal solutions with the scaled initialization $\alpha\mathbf{x}^{(0)}$, $\alpha^{-1}\mathbf{k}^{(0)}$, $\alpha\boldsymbol{\sigma}^{(0)}, \alpha\sigma_0$.

To see why this is important, we notice that the scaling of the solution $\alpha\hat{\mathbf{x}}$ and $\alpha^{-1}\hat{\mathbf{k}}$ affects only the second term in (8). Thus the algorithm always produces some scaled solution which exaggerates the blur kernel $\mathbf{k}$ and contracts the image $\mathbf{x}$.

To address this issue, we incorporate the constraint $\sum_{j=1}^{M} k_j = 1$ into the minimization problem (8) as follows. After each IRCD iteration, the corresponding quantities are scaled as $\mathbf{x}^{(\nu)} = \alpha^{(\nu)}\mathbf{x}^{(\nu)}$, $\mathbf{k}^{(\nu)} = (\alpha^{(\nu)})^{-1}\mathbf{k}^{(\nu)}$, $\boldsymbol{\sigma}^{(\nu)} = \alpha^{(\nu)}\boldsymbol{\sigma}^{(\nu)}$, $\sigma_0 = \alpha^{(\nu)}\sigma_0$, with $\alpha^{(\nu)} = \sum_{j=1}^{M} k_j^{(\nu)}$.

3.3. Finding σ_0

As we can see from (9), the main parameters of this algorithm are σ_0 and σ_Z. We now propose an empirical method for choosing σ_0.

The parameter σ_0^2 relates to the variation in the smooth regions of the image. Therefore, σ_0 should only depend on the image. However, since the clean image is absent, we approximate the clean image with the measurement $\mathbf{y}$. With this idea, we proposed the following way to choose σ_0.

1. For $i = 1, \ldots, L$, we calculate the variance of pixel values in a $N \times N$ patch $P_i(N)$ centered around y_i

$$\sigma_i^2 = \frac{1}{N^2} \sum_{y_n \in P_i(N)} (y_n - \mu_i)^2, \tag{12}$$

where $\mu_i = \frac{1}{N^2}\sum_{y_n\in P_i(N)} y_n$ is the mean of all pixels in the patch $P_i(N)$.

2. The estimate of σ_0^2 is formed as

$$\hat{\sigma}_0^2 = \frac{1}{L}\sum_{j=i}^{L} \sigma_i^2. \tag{13}$$

The common choice for the patch size is $N \approx \sqrt{L}/100$. This method is purely empirical, but based on our experiences, it works well in practice.

3.4. Extension to Multiple Color Measurements

For color measurements, the three color channels are treated independently except that they share the same parameters σ_0 and σ_Z.

Extending the estimation algorithm to $M > 1$ measurements (i.e., multiple blurred images) is straightforward.

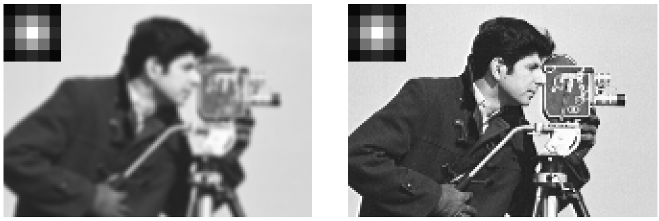


Fig. 3. Deblurring results of a grayscale image of a cameraman. Left: blurred image; right: deblurred image.

Step 1 in Section 3.1 remains essentially unchanged (since all the blur kernels $\mathbf{k}_1 = \mathbf{k}_1^{(\nu-1)}, \ldots, \mathbf{k}_M = \mathbf{k}_M^{(\nu-1)}$ are fixed). Step 2 turns into M separate descent steps in $\mathbf{k}_1, \ldots, \mathbf{k}_M$, with all other variables fixed. Step 3 remains unchanged. The additional scaling step described in Section 3.2 needs to be adapted as well: in this case, instead of scaling the image $\mathbf{x}$ and σ_0^2, we scale the measurements $\mathbf{y}_m$ and $\sigma_{Z_m}^2$ with the corresponding kernel sum $\alpha_m^{(\nu)}$.

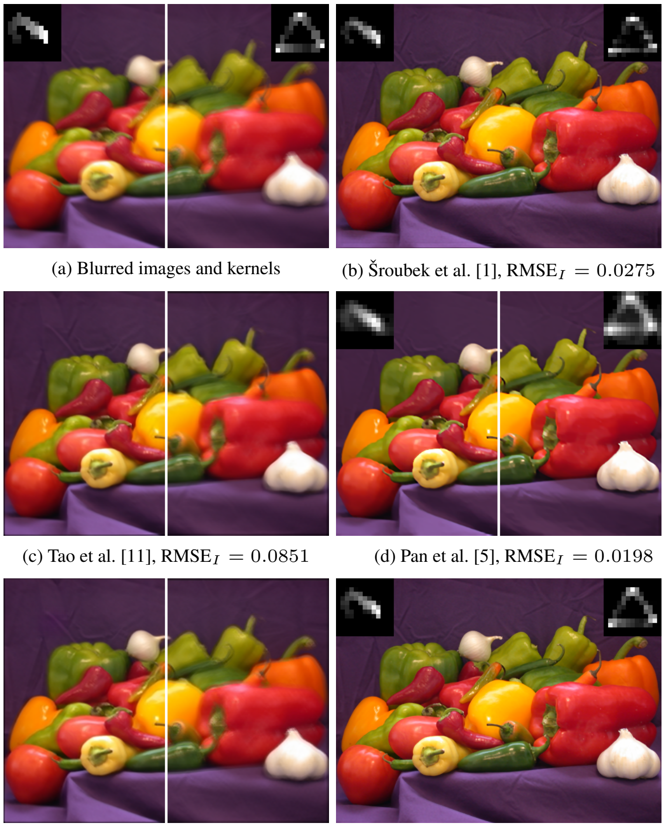
4. RESULTS

In this section we present some experimental results using two test images: a 256×256 grayscale image of a cameraman and a 384×512 color image of peppers.

Fig. 3 shows the deblurring results of the camera image using only a single measurement (partly shown in the left part of Fig. 3). The image on the left side is blurred with a 5×5 Gaussian blur kernel (with the original kernel shown at the upper left corner). The right side image is the deblurred image (with the estimated kernel shown at the upper left corner). The root mean square error (RMSE) of the deblurred image and the estimated kernel are $\text{RMSE}_I = 0.0043$ and $\text{RMSE}_K = 0.0005$. The algorithm was run with the following parameters $\sigma_0^2 = 2.1 \times 10^{-3}$, $\sigma_Z^2 = 1 \times 10^{-7}$ for 8000 iterations, and the parameter σ_0^2 was determined using the method in Section 3.3. The algorithm is able to reconstruct the sharp image and the blur kernel well.

While Fig. 3 is rather a simple example, Fig. 4 shows the deblurring results of the peppers image using two synthetic measurements corrupted by two 13×13 blur kernels. The two blurry measurements with corresponding blur kernels are partly shown in Fig. 4 (a). Fig. 4 (b)–(e) are the deblurred results using the methods of Šroubek et al. [1], Tao et al. [11], Pan et al. [5] and Kupyn et al. [12] which are obtained using their own software. Since the approach of [5, 11, 12] cannot deal with multiple measurements, we run their software for each measurement separately and show both results (partly), and the RMSE is obtained by averaging two deblurred images. Our result was obtained with the following parameters: $\sigma_0^2 = 9.1 \times 10^{-4}$, $\sigma_Z^2 = 4 \times 10^{-5}$ using 12000 iterations.

The result produced by our method is both visually and quantitatively better than the other methods in Fig. 4. The method of Šroubrek et al. [1] gives a good deblurring result and a good kernel estimate, but it produces some line artifacts at the bottom of the image. The software provided by Tao et al. [11] and Kupyn et al. [12] do not work well with the image blurred with the second kernel. While Pan et al.'s method [5] produces a good deblurred image, the estimated kernel is not quite satisfactory.



(a) Blurred images and kernels

(b) Šroubek et al. [1], $\text{RMSE}_I = 0.0275$

(c) Tao et al. [11], $\text{RMSE}_I = 0.0851$

(d) Pan et al. [5], $\text{RMSE}_I = 0.0198$

(e) Kupyn et al. [12], $\text{RMSE}_I = 0.0847$

(f) Our Results, $\text{RMSE}_I = 0.0085$

Fig. 4. Deblurring results of a color image of peppers using different methods.

5. CONCLUSION

We proposed a new approach to blind image deblurring that relies on the smoothed-NUV prior from [15, 16]. The variational representation of the prior from [17] allows the joint estimation of the image and the blurring kernel(s) to be decomposed into descent steps in reweighted least-squares problems and nonlinear scalar updates of the individual variances of the prior. Accordingly, we proposed an iteratively reweighted coordinate descent algorithm, which is parameter free and works well in practice. The proposed approach is shown by simulations to compare favorably with four state-of-the-art methods.

6. REFERENCES

[1] F. Šroubek and P. Milanfar, "Robust multichannel blind deconvolution via fast alternating minimization," *IEEE Trans. on Image Processing*, vol. 21, no. 4, pp. 1687–1700, 2011.

[2] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 sparse representation for natural image deblurring," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 1107–1114.

[3] H. Zhang, D. P. Wipf, and Y. Zhang, "Multi-image blind deblurring using a coupled adaptive sparse prior," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 1051–1058.

[4] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011, pp. 233–240.

[5] J. Pan, D. Sun, H. Pfister, and M. Yang, "Blind image deblurring using dark channel prior," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 1628–1636.

[6] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.

[7] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. on Graphics*, vol. 27, no. 3, pp. 73, 2008.

[8] Z. Hu, L. Xu, and M. Yang, "Joint depth estimation and camera shake removal from single blurry image," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 2893–2900.

[9] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *European Conf. on Computer Vision*. Springer, 2010, pp. 171–184.

[10] S. Nah, T. Kim, and K. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 3883–3891.

[11] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 8174–8182.

[12] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE Int. Conf. on Computer Vision*, 2019, pp. 8878–8887.

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

[14] S. Cho and S. Lee, "Convergence analysis of MAP based blur kernel estimation," in *Proc. IEEE Int. Conf. on Computer Vision*, 2017, pp. 4808–4816.

[15] N. Zalmai, C. Luneau, C. Stritt, and H.-A. Loeliger, "Tomographic reconstruction using a new voxel-domain prior and Gaussian message passing," in *Proc. 24th European Signal Processing Conf. (EUSIPCO)*, 2016, pp. 2295–2299.

[16] B. Ma, N. Zalmai, R. Torfason, C. Stritt, and H.-A. Loeliger, "Color image segmentation using iterative edge cutting, NUV-EM, and Gaussian message passing," in *Proc. IEEE Global Conf. on Signal and Information Processing (GlobalSIP)*, 2017, pp. 161–165.

[17] H.-A. Loeliger, B. Ma, H. Malmberg, and F. Wadehn, "Factor graphs with NUV priors and iteratively reweighted descent for sparse least squares and more," *Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, pp. 1–5, 2018.