

Robust Kernel Estimation in Blind Deconvolution

Zhiming Wang^{1,a}, Xing Li^{1,b}

¹School of Computer & Communication Engineering, University of Science and Technology Beijing, 100083, Beijing, China

^awangzhiming@ies.ustb.edu.cn, ^bli_xing_student@sina.com

Keywords: Blind Deconvolution; Kernel Estimation; Normalized Sparsity Measure

Abstract. Due to the loss of information about image and the interference of noise, blind deconvolution is an ill-posed problem. In this paper, we study this problem based on the algorithm of Krishnan et al.[1], which uses a normalized sparsity measure to solve the problem. By assuming the random high frequency property of the difference between true kernel and intermediate estimated kernel, we add a Gaussian smoothing filtering during sharp image update step. The filtering process can improve robustness of the algorithm. Experimental results show that our algorithm estimates more precise kernel and run fast than Krishnan's original algorithm.

Introduction

Image blurring is caused by many factors, such as the effect of atmosphere, technology problems of camera, motion, defocus, and so on.

Under supposing of linear position-invariant, a blurred image B can be modeled by the convolution of a blur kernel k with a sharp image S , along with the addition of the noise N :

$$B = k \otimes S + N. \quad (1)$$

B is known and both k and S are unknown. Our goal is to estimate the blur kernel k and sharp image S . Since B can be combined by an infinite number of pairs (k, S) , we should use additional conditions to regular the problem so that we can gain the expected solution.

Recent research in natural image statistics have shown that images obey heavy-tailed distributions in their gradients[2,3]. We can use this prior knowledge to solve the deconvolution problem in gradient domain.

A common method is the MAP estimation, which estimates a pair (k, S) by maximizing the posterior probability:

$$(k, S) = \operatorname{argmax} p(k, S|B). \quad (2)$$

We usually use the negative logarithm of (2), that is:

$$(k, S) = \operatorname{argmin} -\log p(k, S|B). \quad (3)$$

In next section, we give the processing of MAP estimation a simple presentation.

A simple presentation of MAP estimation

Based on Bayesian probability theory, we have:

$$\text{posterior} \propto \text{likelihood} \times \text{prior}. \quad (4)$$

$$p(k, S|B) \propto p(B|k, S) \times p(S) \times p(k). \quad (5)$$

where we use the factor: $p(k, S) = p(S) \times p(k)$.

so, Eq.3 can be transformed into:

$$(k, S) = \operatorname{argmin} -\log p(B|k, S) - \log p(S) - \log p(k). \quad (6)$$

We assume that the gradient x of S obeys the distribution:

$$p(x_i) = \beta_1 e^{-\beta_2 |x_i|^\alpha}. \quad (7)$$

where β_1 and β_2 are positive. For simple, we assume every pixel gradient shares the same β_1 and β_2 , and obeys the independent distribution(i.i.d), So

$$p(S) = p(x) = \prod_i \beta_1 e^{-\beta_2 |x_i|^\alpha} = \beta_1 e^{-\beta_2 \|x\|_\alpha^\alpha}. \quad (8)$$

where we assume the relation between S and x is one to one.

Also, we assume an i.i.d Gaussian noise N with variance η^2 , so we have:

$$p(B|k, S) = \frac{1}{(\sqrt{2\pi}\eta)^n} e^{-\frac{\|B - k \otimes S\|_2^2}{2\eta^2}}. \quad (9)$$

Finally, we have the expression about Eq.2:

$$(k, S) = \operatorname{argmin} \lambda_1 \|B - k \otimes S\|_2^2 + \lambda_2 \|x\|_\alpha^\alpha + \lambda_3 G(k). \quad (10)$$

$G(k)$ can be $\|k\|_1^1$. λ_1, λ_2 and λ_3 are positive.

Finally, we have the general form:

$$(k, S) = \operatorname{argmin} \lambda_1 \|B - k \otimes S\|_2^2 + \lambda_2 J(S) + \lambda_3 G(k). \quad (11)$$

Here, $J(S) = \|x\|_\alpha^\alpha = \|\nabla S\|_\alpha^\alpha$.

For more details, we refer the reader to papers of [4] and [6].

The MAP pair (k, S) should minimize the Eq.11, but as pointed out by Levin et al.[5] and Fergus et al.[3], blurred image usually has a lower cost than sharp image. MAP objective function attempts to minimize all gradients (even large ones), so that we usually obtain a blurred image rather than a sharp image. Therefore, Fergus et al.[3] approximates the full posterior distribution and adopts a variational Bayesian approach based on the algorithm of [7]. Based on the algorithm of [3], Levin et al.[6] develops a MAP_k estimation that can be optimized easily. But they all are more complex. Krishnan et al.[1] use a normalized sparsity measure that favors sharp image more than blurred image. Although it can not be explained clearly in probability theory, it can be understood in the MAP view yet. Our improvement is based on this algorithm.

Kernel estimation

From Eq.1, B is known and both k and S are unknown. Our goal is to estimate the blur kernel k and sharp image S . As the way of Fergus et al.[3], we deal with the problem in gradient domain. That is:

$$f_h \otimes B = k \otimes (f_h \otimes S) + n_h, f_v \otimes B = k \otimes (f_v \otimes S) + n_v. \quad (12)$$

$$\{f_h, f_v\} = \{[1, -1], [1, -1]^T\}.$$

We denote $y = \{f_h \otimes B; f_v \otimes B\}$, $x = \{f_h \otimes S; f_v \otimes S\}$, and $n = \{n_h; n_v\}$, so we have:

$$y = k \otimes x + n \quad (13)$$

so Eq.11 is transformed into:

$$(k, x) = \operatorname{argmin} \lambda_1 \|y - k \otimes x\|_2^2 + \lambda_2 J(x) + \lambda_3 G(k). \quad (14)$$

Following Krishnan et al.[1], we use the form:

$$(k, x) = \operatorname{argmin} \frac{\lambda'}{2} \|y - k \otimes x\|_2^2 + \frac{\|x\|_1}{\|x\|_2} + \psi' \|k\|_1^1. \quad (15)$$

$\frac{\|x\|_1}{\|x\|_2}$ is a normalized sparsity measure that favors sharp image more than blurred image, and k is subjected to that $\sum_i k_i = 1, k_i \geq 0$. For more details, please read the [1]. We use the iterative method to estimate k , which alternates between x and k updates. Also we use multi-scale strategy to improve performance of the algorithm, particularly for a large kernel[3].

x-Update

We update x by the following expression, when the k is fixed:

$$x = \arg\min \frac{1}{2} \|y - k \otimes x\|_2^2 + \frac{\|x\|_1}{\lambda' \|x\|_2}. \quad (16)$$

This expression can not be optimized easily, because $\frac{\|x\|_1}{\|x\|_2}$ is non-convex. However, it can be transformed into a convex l_1 -regularized problem by fixing the l_2 norm denominator from previous iteration[1]. So we solve the problem:

$$x = \arg\min \frac{1}{2} \|y - k \otimes x\|_2^2 + \frac{1}{\lambda' \|x'\|_2} \|x\|_1. \quad (17)$$

x' is fixed in the inner iteration. We use ISTA(iterative shrinkage-thresholding algorithm)[8] in the inner iteration.

Analysis of ISTA and smooth filtering

In the inner iteration of x -update, we have the following procedure:

0:Require: Observed image y , Threshold t , Maximum iterations N , Initial iterate x^0

1: for $j = 0$ to $N - 1$ do

2: $v = x^j - tK^T(Kx^j - y)$

3: $x^{j+1} = S_{t\lambda}(v)$

4: end for

5: return: image x^N

From Eq.17, λ is $\frac{1}{\lambda' \|x'\|_2}$ not $\lambda' \|x'\|_2$, and K is the convolution matrix of kernel k . We make $\Delta = t\lambda$, so $t = \frac{\Delta}{\lambda}$. At procedure 0, we require threshold Δ instead of t . From Eq.17, we have $t = \Delta \lambda' \|x'\|_2$. S is the soft shrinkage operation, that is:

$$S_{\Delta}(v)_i = \max(|v_i| - \Delta, 0) \text{sign}(v_i). \quad (18)$$

At procedure 2, we use K , which is not the convolution matrix of true kernel. We assume that $K + ek$ is equal to the convolution matrix of true kernel. So, if we use the true kernel, procedure 2 will become:

$$v = x^j - t(K + ek)^T((K + ek)x^j - y). \quad (19)$$

Finally, we have:

$$v = x^j - tK^T(Kx^j - y) + \text{err}. \quad (20)$$

$\text{err} = -K^T ekx^j - ek^T((K + ek)x^j - y)$. We assume that err has the high frequency feature, so we add a Gaussian smoothing filtering f to suppress it. So we should add a procedure after procedure 2. That is:

$$v = f \otimes v. \quad (21)$$

By Gaussian smoothing filtering, the robustness of the algorithm is improved for different initialization at kernel k .

k -Update

We update k by the following expression, when the x is fixed:

$$k = \arg\min \|y - k \otimes x\|_2^2 + \psi \|k\|_1. \quad (22)$$

We use IRLS(Iterative Reweighted Least Squares)[9] to deal with the problem in the outer iteration. So Eq.22 is transformed into:

$$k = \arg\min \|y - k \otimes x\|_2^2 + \psi \sum_i \text{weight}_i k_i^2. \quad (23)$$

$\text{weight}_i = |k_i|^{-1}$. When k_i is zero, we threshold it.

To solve the Eq.23, we make the derivative of Eq.23 being zero:

$$2X^T(Xk - y) + 2\psi Wk = 0. \quad (24)$$

W is the matrix combined by weight_i . Finally, we have:

$$(X^T X + \psi W)k = X^T y. \quad (25)$$

These equations can be solved by many iterative algorithms, and we use CG(conjugate gradient)[10] to solve it in inner. In the process of solving the problem, we can use $x \otimes m = Xm$, and $x^r \otimes m = X^T m$, of which x^r is that x rotates 180° around the center.

Multi-scale strategy

We use multi-scale strategy to improve performance of the algorithm, particularly for a large kernel. At first, kernel size is 3×3 , increases at each scale and becomes the same size as the original size at the final scale. At the same time, x and y are reduced at the same scale.

In the section experiment, we will consider the different initialization at kernel k .

Image restoration

When the kernel k has been estimated, we can use non-blind deconvolution methods to restore sharp image S from blurred image B . we solve the following expression:

$$S = \argmin \lambda_1 \|B - k \otimes S\|_2^2 + \|\nabla_h S\|_\alpha^\alpha + \|\nabla_v S\|_\alpha^\alpha. \quad (26)$$

We use the method from [2]. This method transforms Eq.26 into a quadratic form using half-quadratic method[11].

Experimental results

In this section, we present results' comparing for different initialization of kernel k . We use two criterions to evaluate the performance. The grayscale images that we use come from: <http://www.wisdom.weizmann.ac.il/~vision/BlindDeblur.html>.

Two criterions

First, SNR(Signal Noise Ratio) is defined as:

$$\text{SNR} = 10 \log_{10} \frac{\|S\|_2^2}{\|S - \hat{S}\|_2^2}. \quad (27)$$

Where S is the known ground truth image, and \hat{S} is the image that we estimate using the estimated kernel. The value of SNR is the bigger the better.

Second, Error Ratio[5,12] is defined as:

$$\text{Error Ratio} = \frac{\|S - \hat{S}_k\|_2^2}{\|S - \hat{S}_k\|_2^2}. \quad (28)$$

Where S is the known ground truth image, \hat{S}_k is the image that we estimate using the known ground truth kernel k , and $\hat{S}_{\hat{k}}$ is the image that we estimate using the estimated kernel \hat{k} . The value of Error Ratio is the smaller the better.

Results

We consider the two different initialization of kernel k . The first is $[1/2 \ 1/2 \ 0; 0 \ 0 \ 0; 0 \ 0 \ 0]$, and the second is $[1/9 \ 1/9 \ 1/9; 1/9 \ 1/9 \ 1/9; 1/9 \ 1/9 \ 1/9]$.

As shown in Fig.1, we use the image '20' and the kernel '4' and '8' from dataset.



Fig.1. Left: the ground truth image; right-top-left: the 27×27 ground truth kernel; right-top-right: the 23×23 ground truth kernel; right-bottom-left: the blurred image of left kernel; right-bottom-right: the blurred image of right kernel.

Fig.2 shows the results about the 27×27 kernel.

Table 1 shows the values of SNR, Error Ratio, and elapsed time about the 27×27 kernel.

Fig.3 shows the results about the 23×23 kernel.

Table 2 shows the values of SNR, Error Ratio, and elapsed time about the 23×23 kernel.

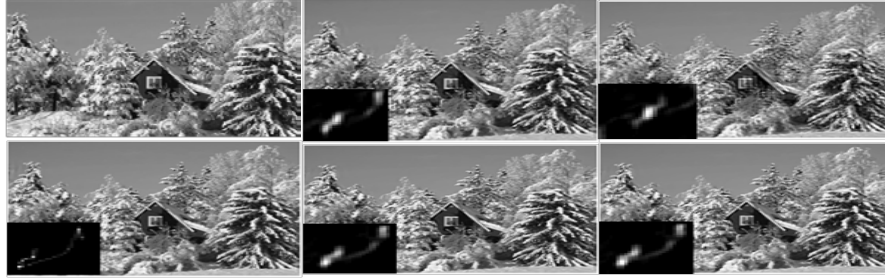


Fig.2. The results of 27×27 kernel. Left-top: the ground truth image; left-bottom: deblurred image using the ground truth kernel; middle-top: deblurred image using kernel of Krishnan et al. with the first initialization; middle-bottom: deblurred image using our kernel with the first initialization; right-top: deblurred image using kernel of Krishnan et al. with the second initialization; right-bottom: deblurred image using our kernel with the second initialization. Corresponding kernels are shown as insets at the bottom left of each image.

Table 1. Performance comparison on 27×27 kernel.

Algorithm	SNR	Error Ratio	Elapsed Time(s)
Krishnan's, first initialization	10.29	7.15	218.09
Ours, first initialization	11.53	5.37	214.17
Krishnan's, second initialization	8.33	11.21	222.23
Ours, second initialization	11.53	5.37	218.46

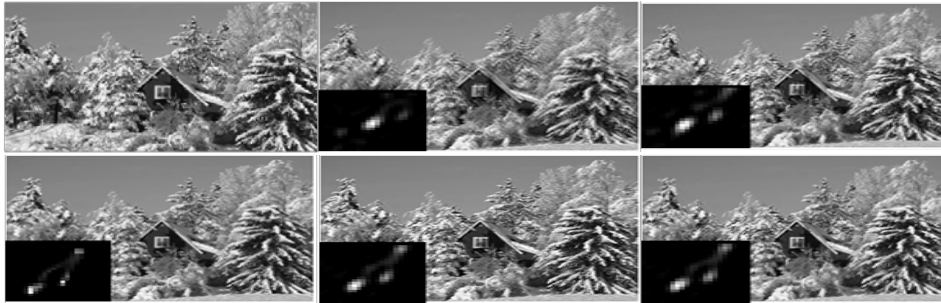


Fig.3. The results of 23×23 kernel.

Table 2. Performance comparison on 23×23 kernel.

Algorithm	SNR	Error Ratio	Elapsed Time(s)
Krishnan's, first initialization	9.87	9.63	245.55
Ours, first initialization	12.39	5.40	227.19
Krishnan's, second initialization	10.47	8.40	236.07
Ours, second initialization	12.38	5.41	228.12

Remarks

From Fig.2 and Table 1, we can see that Krishnan et al. favors the first initialization of kernel. From Fig.3 and Table 2, we can see that Krishnan et al. favors the second initialization of kernel. However, our algorithm gives almost the same results and gives better performance than Krishnan's algorithm. It means that we improved its robustness. By comparing elapsed times, one can see that ours runs a little fast than Krishnan's algorithm.

Fig.4 shows the results of a real-data image provided by Fergus et al.[3]. From that one can see that our algorithm gives more robust result than Krishnan's original algorithm with equal parameters.



Fig.4. The results of a real-data image. First: the blurred image; second: deblurred image using kernel of Krishnan et al. with the first initialization; third: deblurred image using our kernel with the first initialization; fourth: deblurred image using kernel of Krishnan et al. with the second initialization; fifth: deblurred image using our kernel with the second initialization.

Conclusions

In general, Krishnan's algorithm gives a good performance in blind image restoration, but estimation error is inevitable during kernel estimation. By assuming the high frequency property of this error, we improve robustness of Krishnan's algorithm by adding a Gaussian smoothing filtering in sharp image estimation iteration. Experimental results show it is an effective way to guarantee robustness.

References

- [1]D. Krishnan, T. Tay and R. Fergus. Blind deconvolution using a normalized sparsity measure[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
- [2]D. Krishnan and R. Fergus. Fast image deconvolution using hyper-Laplacian priors[C]. Advances in Neural Information Processing Systems. 2009.
- [3]R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph[J]. ACM Transactions on Graphics (TOG), 2006, 25(3): 787-794.
- [4]C. Bishop. Pattern recognition and machine learning[M]. springer, 2006.
- [5]A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms[C]. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2009.
- [6]A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
- [7]J. Miskin and D. J. C. Mackay. Ensemble learning for blind image separation and deconvolution[M]. Advances in independent component analysis. Springer London, 2000: 123-141.
- [8]A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J]. SIAM journal on imaging sciences, 2009, 2(1): 183-202.
- [9]C. S. Burrus. Iterative Reweighted Least Squares[J]. <http://cnx.org/content/m45285/1.12/>.
- [10]J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain[J], Carnegie Mellon University, Pittsburgh, PA, 1994.
- [11]D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization[J]. IEEE Transactions on Image Processing, 1995, 4(7): 932-946.
- [12]T. Michaeli, M. Irani. Blind deblurring using internal patch recurrence[M]. European Conference on Computer Vision. Springer International Publishing, 2014: 783-798.