A BLUR-SURE-LET ALGORITHM TO BLIND PSF ESTIMATION FOR DECONVOLUTION

Feng Xue, Jiaqi Liu, Chengguo Liu, Gang Meng

National Key Laboratory of Science and Technology on Test Physics and Numerical Mathematics
Beijing, 100076, China

ABSTRACT

In this paper, we consider blind deconvolution that consists of PSF (point spread function) estimation and non-blind deconvolution. It has been proved that blur-SURE — a modified version of SURE (Stein's unbiased risk estimate) — is a valid criterion for parametric PSF estimation.

The key contribution of this work is to propose a fast algorithm for the blur-SURE minimization. Incorporating a linear combination of multiple smoother matrices with different but fixed regularization parameters, the optimal regularized processing is obtained by a closed-form solution. This linear parametrization of the processing greatly accelerates the minimization.

The extensive experiments show the significant improvement of computational time. The low computational complexity enables the blur-SURE criterion to be readily applied to more complicated parametric forms of PSF.

***Index Terms*—** PSF estimation, blur-SURE minimization, linear parametrization

1. INTRODUCTION

In this article, we are interested in blind deconvolution, where the image formation is often modeled as [1]:

$$\mathbf{y} = \mathbf{H}_0\mathbf{x} + \mathbf{b} \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^N$ and $\mathbf{x} \in \mathbb{R}^N$ denote the observed and original clean images, respectively, N is pixel number of the image, $\mathbf{H}_0 \in \mathbb{R}^{N\times N}$ is the underlying true convolution matrix constructed by point spread function (PSF) $\mathbf{h}_0$. $\mathbf{b} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is an additive Gaussian white noise with variance σ^2. Blind deconvolution attempts to estimate both $\mathbf{x}$ and $\mathbf{H}_0$ from the measured data $\mathbf{y}$ only [1].

PSF estimation is essential to the deconvolution performance. Typically, PSF is simultaneously estimated with the original image by regularization or Bayesian approaches, which enforce a certain constraints on PSF and original image as regularization terms, and formulate blind deconvolution as minimization of an objective functional [2].

In some particular applications, the PSF is of known specific parametric form and is completely characterized by a small number of parameters [3]. Thus, the parametric blind deconvolution is to estimate the PSF parameters from the observed image [1]. Most parametric approaches are only applied for specific types of PSF, e.g. [4] estimated the length and direction of linear motion based on Radon transform.

In [1], we proposed a blur-SURE — a modified version of SURE (Stein's unbiased risk estimate) — as a novel criterion for parametric PSF estimation. It has been shown that the blur-SURE minimization yields the highly accurate estimates of PSF parameters. This blur-SURE framework is not limited to any particular form of PSF. The blur-SURE minimization, however, is computationally expensive, especially when the PSF form involves more than two unknown parameters. Proceeding the work of [1], we now propose a fast algorithm to implement the minimization, based on a linear parametrization of processing, called *blur-SURE-LET* approach. Thus, the computational complexity of PSF estimation is greatly reduced.

2. BRIEF REVIEW AND RE-INTERPRETATION OF THE BLUR-SURE FRAMEWORK

2.1. Blur-SURE as a criterion for PSF estimation

Denote the linear function (or processing) by matrix $\mathbf{U}$. [1] proved that if the matrix U is given as:

$$\mathbf{U} = \mathbf{HSH}^{\mathrm{T}}(\mathbf{HSH}^{\mathrm{T}} + \mathbf{C})^{-1} \tag{2}$$

the minimization of blur-MSE:

$$\text{blur-MSE} = \frac{1}{N}\mathbb{E}\left\{\|\mathbf{Uy} - \mathbf{H}_0\mathbf{x}\|^2\right\} \tag{3}$$

over $\mathbf{H}$ yields exact estimate of PSF, where $\mathbf{S} = \mathbb{E}\left\{\mathbf{xx}^{\mathrm{T}}\right\}$, $\mathbf{C} = \mathbb{E}\left\{\mathbf{bb}^{\mathrm{T}}\right\} = \sigma^2\mathbf{I}$. Due to the inaccessible $\mathbf{H}_0\mathbf{x}$ and $\mathbf{S}$, [1] proposed two practical approximations. The first one is to use blur-SURE — an unbiased estimate of blur-MSE — to replace blur-MSE:

$$\text{blur-SURE} = \frac{1}{N}\left\{\|\mathbf{Uy} - \mathbf{y}\|^2 + 2\sigma^2\mathrm{Tr}(\mathbf{U})\right\} - \sigma^2 \tag{4}$$

depending on the observed data $\mathbf{y}$ only. The proof of the unbiasedness can be completed using *Stein's lemma* [5].

2.2. Smoother matrix

Another approximation is to use the following matrix:

$$\mathbf{U}_\lambda = \mathbf{H}\left(\mathbf{H}^\mathrm{T}\mathbf{H} + \lambda\mathbf{R}\right)^{-1}\mathbf{H}^\mathrm{T} \quad \text{i.e.} \quad U_\lambda(\omega) = \frac{|H(\omega)|^2}{|H(\omega)|^2 + \lambda\omega^2} \tag{5}$$

to approximate (2), where the eigenvalues of the circulant matrix $\mathbf{R}$ is ω^2, λ is a regularization parameter. In model selection problem, $\mathbf{U}_\lambda$ is called *smoother matrix* or *smoother filtering* [6]. If the PSF is of known function form, depending on a small number of parameters $\mathbf{s} = [s_1, s_2, ..., s_P]^\mathrm{T}$ (parameter vector), the PSF estimation is finally formulated as the following joint minimization problem:

$$\min_{\mathbf{s},\lambda} \underbrace{\frac{1}{N}\left\{\left\|\mathbf{U}_{\mathbf{s},\lambda}\mathbf{y} - \mathbf{y}\right\|^2 + 2\sigma^2\mathrm{Tr}(\mathbf{U}_{\mathbf{s},\lambda})\right\} - \sigma^2}_{\text{blur-SURE}(\mathbf{s},\lambda)} \tag{6}$$

where the smoother matrix $\mathbf{U}_{\mathbf{s},\lambda}$ is given by (5), with $\mathbf{H}_\mathbf{s}$ depending on the parameter $\mathbf{s}$.

3. A FAST ALGORITHM OF BLUR-SURE MINIMIZATION

3.1. Outline of the proposed algorithm

Notice that it is difficult to minimize the blur-SURE (6), since its convexity is not guaranteed, though exhaustive search has shown that the global minimum of blur-SURE is the desired estimate of PSF parameter $\mathbf{s}$. However, the greedy algorithm is computationally expensive. In our previous work of [1], we performed an alternating minimization algorithm between $\mathbf{s}$ and λ, until the convergence is reached. This iterative algorithm, however, has very slow convergence rate. Now, we propose a fast algorithm to substantially reduce the computational complexity.

First, we set a number of tentative values for parameter $\mathbf{s}$, and find optimal value of λ (denoted by $\lambda_{\text{opt}}(\mathbf{s})$) for each fixed $\mathbf{s}$ by minimizing the blur-SURE. Then, incorporating the tentative optimal pairs $(\mathbf{s}, \lambda_{\text{opt}}(\mathbf{s}))$ into the blur-SURE, we find the optimal parameter $\mathbf{s}$ by the blur-SURE minimization. This procedure is equivalent to the global search in spirit, by separating the joint minimization into two independent optimizations, and seeking the solution along the optimal path $(\mathbf{s}, \lambda_{\text{opt}}(\mathbf{s}))$. The algorithm is outlined in Algorithm 1.

Consider the first step for finding corresponding optimal λ_{opt} for each tentative value $\mathbf{s}$. The most straightforward way is to perform exhaustive search, to find the exact solution to the blur-SURE minimization, denoted by λ_{opt}. The line search is of high computational cost. Now, we propose a fast algorithm for the first step, by providing a closed-form solution to the value of λ for each tentative $\mathbf{s}_j$, such that the produced value of λ is as close as possible to the exact λ_{opt}. The details are described as follows.

Algorithm 1: Separating Minimization Algorithm

Input: blur-SURE$(\mathbf{s}, \lambda)$: objective function (6)
Output: optimal $\mathbf{s}_{\text{opt}}$
begin
  1. set J tentative values of $\mathbf{s}$: $\mathbf{s}_1, ..., \mathbf{s}_J$;
  for $j = 1 \to J$ **do**
    for each fixed $\mathbf{s}_j$, find the corresponding optimal $\lambda_{\text{opt}}(\mathbf{s}_j) = \arg\min_\lambda \text{blur-SURE}(\mathbf{s}_j, \lambda)$;
  end
  2. for each pair $(\mathbf{s}_j, \lambda_{\text{opt}}(\mathbf{s}_j))$, find the optimal $\mathbf{s}_{\text{opt}} = \text{argmin}_{\mathbf{s}_j} \text{blur-SURE}(\mathbf{s}_j, \lambda_{\text{opt}}(\mathbf{s}_j))$.
end

3.2. Linear parametrization of smoother matrix

The LET (linear expansion of thresholds) strategy was firstly proposed in [7], where the authors linearly parametrized the thresholding function for image denoising. We now adopt the linear parametrization to the smoother matrix $\mathbf{U}$ (5) as:

$$\mathbf{U}_{\mathbf{s},\lambda} = \sum_{k=1}^{K} a_k \underbrace{\mathbf{H}_\mathbf{s}\left(\mathbf{H}_\mathbf{s}^\mathrm{T}\mathbf{H}_\mathbf{s} + \lambda_k\mathbf{R}\right)^{-1}\mathbf{H}_\mathbf{s}^\mathrm{T}}_{\mathbf{U}_k} \tag{7}$$

where λ_k are different but fixed values. Substituting (7) into (6), the blur-SURE becomes quadratic function of the linear coefficients a_k, for any fixed parameter $\mathbf{s}$. Thus, the minimization finally boils down to solving a simple linear system of equations:

$$\sum_{k'=1}^{K} \underbrace{\left(\mathbf{y}^\mathrm{T}\mathbf{U}_k^\mathrm{T}\mathbf{U}_{k'}\mathbf{y}\right)}_{\mathbf{M}_{k,k'}} a_{k'} = \underbrace{\mathbf{y}^\mathrm{T}\mathbf{U}_k\mathbf{y} - \sigma^2\mathrm{Tr}(\mathbf{U}_k)}_{c_k} \tag{8}$$

for $k = 1, 2, ..., K$, i.e. $\mathbf{Ma} = \mathbf{c}$ in matrix notation, which is exact and fast. Thus, the weights a_k obtained by the blur-SURE minimization automatically constitute the best linear combination of the given regularized candidates $\mathbf{U}_k$ in terms of blur-MSE.

Closer inspection of the relation among a_k, λ_k and exact solution λ_{opt} (by line search) reveals two indications of the linear weights a_k: (1) the weight a_k for any k reflects the proportion and significance of its corresponding candidate $\mathbf{U}_k$ in the linearly combined estimate $\mathbf{U}_{\mathbf{s},\lambda}$; (2) the value of a_k indicates the closeness of the associated λ_k to optimal λ_{opt}: if a_k is close to 1 for some k, it implies that its related λ_k is very close to λ_{opt}; conversely, if λ_k is very close to λ_{opt} for some k, the weight a_k by the blur-SURE minimization will be approximately unity. Overall, the blur-SURE minimization naturally favors the best candidate with λ_k closest to λ_{opt}, and always assigns a largest weight a_k to it.

Taking $K = 2$ regularized candidates in (7) for example, Table 1 shows the relation among λ_k, a_k and λ_{opt}. Assuming the exact λ_{opt} is known in advance by line search, we set $\lambda_1 = \lambda_{\text{opt}}$ and $\lambda_2 = \lambda_{\text{opt}}$, respectively, and obtain a_k by (8). It can be seen that if $\lambda_k = \lambda_{\text{opt}}$ for some k, the blur-SURE minimization

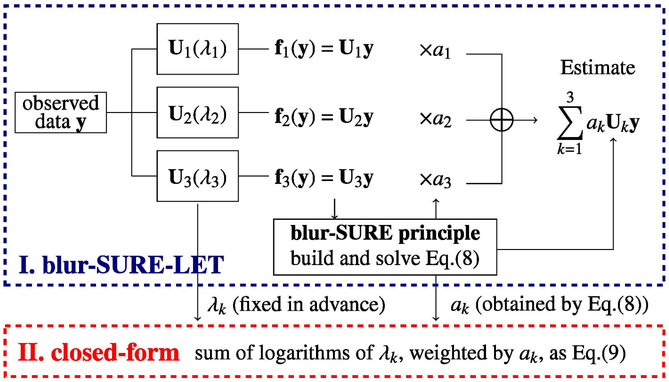

Fig. 1. The flowchart of the proposed blur-SURE-LET procedure.

will lead to $a_k \approx 1$, whereas other candidates make very little contribution to the finally combined estimate $\mathbf{U}_{\mathbf{s},\lambda}$.

Table 1. The relation among λ_k, a_k and λ_{opt}

parameter setting	λ_1	a_1	λ_2	a_2	λ_{opt}
Case I	λ_{opt}	1.00	$0.1\lambda_{\text{opt}}$	0.00	4.55×10^{-4}
Case II	$10\lambda_{\text{opt}}$	0.01	λ_{opt}	0.99	3.47×10^{-3}

Based on the observations mentioned above, we empirically found that the following summation of logarithms of λ_k with optimal weights a_k could predict the value of λ, which is very close to optimal λ_{opt}:

$$\log_{10}\lambda = \sum_{k=1}^{K} a_k \log_{10}\lambda_k \Longleftrightarrow \lambda = \prod_{k=1}^{K}\lambda_k^{a_k} \qquad (9)$$

Furthermore, if $\sum_{k=1}^{K} a_k = 1$ (it holds for most cases, e.g. see Table 1), Eq.(9) implies that the selected value is a *generalized* geometric mean of λ_k, weighted by a_k. We use the term *generalized*, since a_k obtained by the blur-SURE minimization may be negative.

3.3. Short summary

In Section 3.2, we have proposed a fast algorithm to implement the first step of Algorithm 1, which can be summarized as follows:

1. set K values for $\lambda_1, ..., \lambda_K$;
2. solve (8) to obtain a_k;
3. compute λ by (9).

Figure 1 shows the flowchart for the idea of Section 3.2.

Now, we incorporate Section 3.2 into Algorithm 1, and present the fast algorithm for parametric PSF estimation, which is described in Algorithm 2. This fast algorithm combines the linear parametrization (LET) with the blur-SURE minimization, hence, is called *blur-SURE-LET* approach.

4. EXPERIMENTAL RESULTS

4.1. Parameter setting

The test image is *Cameraman*, blurred by the following two typical kernels:

- **Gaussian kernel:** $\mathbf{h}_s(r;s) = C\cdot\exp[-r^2/(2s^2)]$

Algorithm 2: Fast Minimization Algorithm

Input: blur-SURE($\mathbf{s},\lambda$): objective function (6)
Output: optimal $\mathbf{s}_{\text{opt}}$
begin
  1. set J tentative values of $\mathbf{s}$: $\mathbf{s}_1, ..., \mathbf{s}_J$, and K values for $\lambda_1, ..., \lambda_K$;
  for $j = 1 \to J$ **do**
    (*1*) compute a_k by (8);
    (*2*) compute $\lambda_{\text{opt}}(\mathbf{s}_j)$ by (9).
  end
  2. for each pair $(\mathbf{s}_j, \lambda_{\text{opt}}(\mathbf{s}_j))$, find the optimal $\mathbf{s}_{\text{opt}} = \text{argmin}_{\mathbf{s}_j}\,\text{blur-SURE}(\mathbf{s}_j, \lambda_{\text{opt}}(\mathbf{s}_j))$.
end

- *jinc* **function**[1]**:** $\mathbf{h}_s(r;s) = C\cdot\left[\frac{J_1(r/s)}{r/s}\right]^2$

where $r = \sqrt{i^2+j^2}$. $J(\cdot)$ is the first-order Bessel function of first kind. The level of Gaussian noise is evaluated by BSNR[2].

The PSF estimation amounts to estimate the parameter s of Gaussian and *jinc* functions. We use $\lambda_1 = 1\times10^{-4}\sigma^2$, $\lambda_2 = 1\times10^{-3}\sigma^2$, $\lambda_3 = 1\times10^{-2}\sigma^2$ (i.e. $K = 3$) throughout ALL the experiments.

4.2. The accuracy of the selected λ

In this section, we will focus on the first step of Algorithm 1 and 2. The exact solution to optimal λ_{opt} is obtained by line search, which serves as a benchmark: the closer value to λ_{opt} implies the better selection. Hence, we evaluate the accuracy of the blur-SURE-LET algorithm by relative error:

$$\text{error} = \frac{\lambda - \lambda_{\text{opt}}}{\lambda_{\text{opt}}} \qquad (10)$$

where λ is selected by the proposed blur-SURE-LET algorithm. Figure 2 shows the relative error under various tentative values of $s \in [1.0, 3.0]$, where the horizontal zero black line denotes λ_{opt} by line search. We can see that the relative error of blur-SURE-LET are always within 30%.

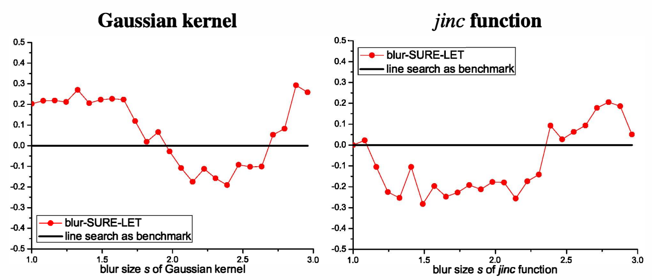

Fig. 2. The regularization parameters λ selected by blur-SURE-LET procedure (BSNR=40dB).

Table 2 shows the computational time for one blur-SURE minimization for a fixed s. For precise computation, it is nec-

[1]The name *jinc* is due to the structural similarity to *sinc* function [8], it is often used to describe optical diffraction-limited condition [8].

[2]The blurred SNR is defined as: $\text{BSNR} = 10\log_{10}\frac{\text{var}(\mathbf{H}_0\mathbf{x})}{N\sigma^2}$ (in dB).

essary to perform the minimization for at least $J = 50$ values of s in a certain interval. Compared to line search, the blur-SURE-LET algorithm results in considerable reduction of the computational time, which will be shown in next section.

Table 2. Computational time (in seconds)

Methods	line search	blur-SURE-LET
Time	0.15	**0.005**

4.3. The performance of PSF estimation

In this part, we use the fast blur-SURE-LET algorithm to estimate the parameter s of Gaussian and *jinc* functions. Figure 3 shows the estimation results, where the black horizontal line denotes the true value $s_0 = 2.0$. We can see that the proposed blur-SURE-LET approach always yields highly accurate estimation of PSF parameter (estimated $s \in [1.91, 2.04]$), even under severe noise corruption (e.g. BSNR=10dB).

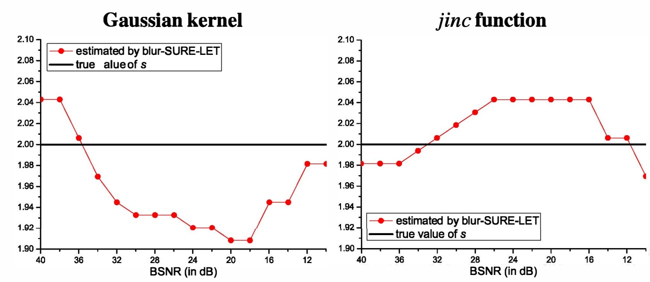

Fig. 3. The estimated parameter s by blur-SURE-LET procedure under various BSNR (from 40dB to 10dB).

Table 3 reports the computational time. We can see that the proposed blur-SURE-LET algorithm is considerably faster than line search.

Table 3. Computational time (in seconds)

Methods	line search	blur-SURE-LET
Time	8.75	**0.60**

4.4. The performance of blind deconvolution

In our last set of experiments, we apply our developed SURE-LET algorithm to perform deconvolution [9], with the estimated PSF by the blur-SURE-LET algorithm[3]. Figure 4 shows a visual example. Compared to using exact PSF, there is only 0.04dB of PSNR loss with the estimated PSF.

We also perform experiment with a real image of *Text*, shown in Figure 5. We approximate the out-of-focus blur as Gaussian function. Figure 5 shows the restored image using Gaussian kernel with $s = 2.62$, estimated by the proposed algorithm.

5. CONCLUSIONS

The blur-SURE has been verified as a valid criterion for parametric PSF estimation [1]. In this paper, we proposed a novel algorithm for the blur-SURE minimization. The experiments have demonstrated that the developed algorithm is considerably faster than line search or alternating algorithm. As emphasized in [1], the blur-SURE framework is not limited to any specific PSF parametric form. The proposed fast algorithm provides a huge potential for the application of this framework to more complicated PSF forms, e.g. fluorescence microscopy [3].

[3]To keep clarity, our blind deconvolution consists of two steps: the blur-SURE-LET algorithm presented in this paper is for *PSF estimation* (the first step), whereas the SURE-LET algorithm proposed in [9] is devoted to *non-blind deconvolution* (the second step using the PSF estimated in the first step).

Fig. 4. A visual comparison of *Cameraman*.

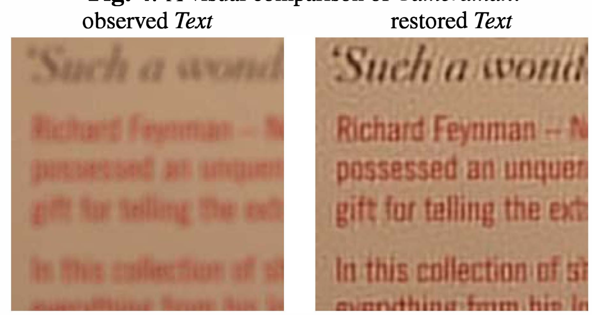

Fig. 5. The restoration of real image: *Text*.

6. REFERENCES

[1] F. Xue and T. Blu, "SURE-based blind Gaussian deconvolution," in *IEEE Statistical Signal Processing Workshop*, 2012, pp. 452–455.

[2] H. Liao and M. Ng, "Blind deconvolution using generalized cross validation approach to regularization parameter estimation," *IEEE Transactions on Image Processing*, , no. 99, pp. 1–1, 2011.

[3] Tal Kenig, Zvi Kam, and Arie Feuer, "Blind image deconvolution using machine learning for three-dimensional microscopy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2191–2204, 2010.

[4] J.P. Oliveira, M.A.T. Figueiredo, and J.M. Bioucas-Dias, "Parametric blur estimation for blind restoration of natural images: Linear motion and out-of-focus," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 466–477, 2014.

[5] C.M. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, pp. 1135–1151, 1981.

[6] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani, *The Elements of Statistical Learning*, vol. 2, Springer, 2009.

[7] F. Luisier, T. Blu, and M. Unser, "A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 593–606, 2007.

[8] Q. Cao, "Generalized jinc functions and their application to focusing and diffraction of circular apertures," *J. Opt. Soc. Am. A*, vol. 20, no. 4, pp. 661–667, 2003.

[9] F. Xue, F. Luisier, and T. Blu, "Multi-wiener SURE-LET deconvolution," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1954–1968, 2013.