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]:

\[ y = H_0 x + b \]  

where \( y \in \mathbb{R}^N \) and \( x \in \mathbb{R}^N \) denote the observed and original clean images, respectively, \( N \) is pixel number of the image, \( H_0 \in \mathbb{R}^{N \times N} \) is the underlying true convolution matrix constructed by point spread function (PSF) \( h_0 \). \( b \sim N(0, \sigma^2 I) \) is an additive Gaussian white noise with variance \( \sigma^2 \). Blind deconvolution attempts to estimate both \( x \) and \( H_0 \) from the measured data \( y \) 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 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 \( U \). [1] proves that if the matrix \( U \) is given as:

\[ U = HSH^T (HSH^T + C)^{-1} \]  

the minimization of blur-MSE:

\[ \text{blur-MSE} = \frac{1}{N} E \left\{ \| Uy - H_0 x \|^2 \right\} \]  

over \( H \) yields exact estimate of PSF, where \( S = E\{xx^T\} \), \( C = E\{bb^T\} = \sigma^2 I \). Due to the inaccessible \( H_0 x \) and \( S \), [1] proposed two practical approximations. The first one is to use blur-SURE — an unbiased estimate of blur-MSE — to replace blur-MSE:

\[ \text{blur-SURE} = \frac{1}{N} \left\{ \| Uy - y \|^2 + 2\sigma^2 \text{Tr}(U) \right\} - \sigma^2 \]  

depending on the observed data \( y \) 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:

\[ U_\lambda = H(H^T H + \lambda R)^{-1} H^T \]

i.e. \[ U_\lambda(\omega) = \frac{|H(\omega)|^2}{|H(\omega)|^2 + \lambda \omega^2} \]  

(5)
to approximate (2), where the eigenvalues of the circulant matrix R is \( \omega^2 \), \( \lambda \) is a regularization parameter. In model selection problem, \( U_\lambda \) is called smoother matrix or smoother filtering [6]. If the PSF is of known function form, depending on a small number of parameters \( s = [s_1, s_2, \ldots, s_p]^T \) (parameter vector), the PSF estimation is finally formulated as the following joint minimization problem:

\[ \min_{s, \lambda} \frac{1}{N} \left\{ \|U_{s, \lambda} y - y\|^2 + 2\sigma^2 \text{Tr}(U_{s, \lambda}) \right\} - \sigma^2 \]  

(6)

where the smoother matrix \( U_{s, \lambda} \) is given by (5), with \( H_s \) depending on the parameter \( s \).

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 \( s \). However, the greedy algorithm is computationally expensive. In our previous work of [1], we performed an alternating minimization algorithm between \( s \) and \( \lambda \), 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 \( s \), and find optimal value of \( \lambda \) (denoted by \( \lambda_{opt}(s) \)) for each fixed \( s \) by minimizing the blur-SURE. Then, incorporating the tentative optimal pairs \((s, \lambda_{opt}(s))\) into the blur-SURE, we find the optimal parameter \( s \) 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 \((s, \lambda_{opt}(s))\). The algorithm is outlined in Algorithm 1.

Consider the first step for finding corresponding optimal \( \lambda_{opt} \) for each tentative value \( s \). The most straightforward way is to perform exhaustive search, to find the exact solution to the blur-SURE minimization, denoted by \( \lambda_{opt} \). 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 \( \lambda \) for each tentative \( s \), such that the produced value of \( \lambda \) is as close as possible to the exact \( \lambda_{opt} \). The details are described as follows.

Algorithm 1: Separating Minimization Algorithm

<table>
<thead>
<tr>
<th>Input:</th>
<th>blur-SURE(s, \lambda): objective function (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
<td>optimal ( s_{opt} )</td>
</tr>
<tr>
<td>begin</td>
<td>1. set ( J ) tentative values of ( s: s_1, \ldots, s_J );</td>
</tr>
<tr>
<td></td>
<td>for ( j = 1 \rightarrow J ) do</td>
</tr>
<tr>
<td></td>
<td>for each fixed ( s_j ), find the corresponding optimal ( \lambda_{opt}(s_j) = \arg\min_{\lambda} \text{blur-SURE}(s_j, \lambda) );</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
<tr>
<td></td>
<td>2. for each pair ((s_j, \lambda_{opt}(s_j))), find the optimal ( s_{opt} = \arg\min_{s_j} \text{blur-SURE}(s_j, \lambda_{opt}(s_j)) ).</td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

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 \( U \) (5) as:

\[ U_{s, \lambda} = \sum_{k=1}^{K} a_k H_s (H_s^T H_s + \lambda R)^{-1} H_s^T U_k \]  

(7)

where \( \lambda_k \) 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 \( s \). Thus, the minimization finally boils down to solving a simple linear system of equations:

\[ \sum_{k'=1}^{K} \left( y^T U_k^T U_{k'} y \right) a_{k'} = y^T U_k y - \sigma^2 \text{Tr}(U_k) \]  

(8)

for \( k = 1, 2, \ldots, K \), i.e. \( M_{k,k'} a_k = c_k \) 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 \( U_k \) in terms of blur-MSE.

Closer inspection of the relation among \( a_k, \lambda_k \) and exact solution \( \lambda_{opt} \) (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 \( U_k \) in the linearly combined estimate \( U_{s, \lambda} \); (2) the value of \( a_k \) indicates the closeness of the associated \( \lambda_k \) to optimal \( \lambda_{opt} \): if \( a_k \) is close to 1 for some \( k \), it implies that its related \( \lambda_k \) is very close to \( \lambda_{opt} \); conversely, if \( \lambda_k \) is very close to \( \lambda_{opt} \) 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 \( \lambda_k \) closest to \( \lambda_{opt} \), 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 \( \lambda_k, a_k \) and \( \lambda_{opt} \). Assuming the exact \( \lambda_{opt} \) is known in advance by line search, we set \( \lambda_1 = \lambda_{opt} \) and \( \lambda_2 = \lambda_{opt} \), respectively, and obtain \( a_k \) by (8). It can be seen that if \( \lambda_k = \lambda_{opt} \) for some \( k \), the blur-SURE minimization...
will lead to $a_k \approx 1$, whereas other candidates make very little contribution to the finally combined estimate $U_s$. 

### Table 1. The relation among $\lambda_k$, $a_k$ and $\lambda_{opt}$

<table>
<thead>
<tr>
<th>parameter setting</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$\lambda_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case I</td>
<td>1.00</td>
<td>0.1</td>
<td>0.00</td>
<td>4.55×10^{-4}</td>
</tr>
<tr>
<td>Case II</td>
<td>100</td>
<td>0.01</td>
<td>0.99</td>
<td>3.47×10^{-3}</td>
</tr>
</tbody>
</table>

Based on the observations mentioned above, we empirically found that the following summation of logarithms of $\lambda_k$ with optimal weights $a_k$ could predict the value of $\lambda$, which is very close to optimal $\lambda_{opt}$:

$$\log_{10} \lambda = \sum_{k=1}^{K} a_k \log_{10} \lambda_k \iff \lambda = \prod_{k=1}^{K} \lambda_k^{a_k}$$

(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 $\lambda_k$, 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 $\lambda$ 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**: $h_g(r,s) = C \cdot \exp[-r^2/(2s^2)]$
- **jinc function**: $h_j(r,s) = C \cdot \left[\frac{J_1(s|s)}{r|s}\right]^2$

where $r = \sqrt{s^2 + j^2}$. $J_1(s)$ is the first-order Bessel function of first kind. The level of Gaussian noise is evaluated by BSNR2.

The PSF estimation amounts to estimate the parameter $s$ of Gaussian and jinc functions. We use $\lambda_1 = 1 \times 10^{-4} \sigma^2$, $\lambda_2 = 1 \times 10^{-3} \sigma^2$, $\lambda_3 = 1 \times 10^{-2} \sigma^2$ (i.e. $K = 3$) throughout ALL the experiments.

#### 4.2. The accuracy of the selected $\lambda$

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

$$\text{error} = \frac{\lambda - \lambda_{opt}}{\lambda_{opt}}$$

where $\lambda$ 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 $\lambda_{opt}$ by line search. We can see that the relative error of blur-SURE-LET are always within 30%.

### Algorithm 2: Fast Minimization Algorithm

**Input:** blur-SURE($s, \lambda$): objective function (6)

**Output:** optimal $s_{opt}$

```
begin
  1. set $J$ tentative values of $s$: $s_1, ..., s_J$, and $K$ values for $\lambda_1, ..., \lambda_K$;
  for $j = 1 \rightarrow J$ do
    1. compute $a_k$ by (8);
    2. compute $\lambda_{opt}(s_j)$ by (9).
  end
  2. for each pair $(s_j, \lambda_{opt}(s_j))$, find the optimal $s_{opt} = \arg\min_{s_j} \text{blur-SURE}(s_j, \lambda_{opt}(s_j))$.
end
```

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

2The blurred SNR is defined as: $\text{BSNR} = 10 \log_{10} \frac{\text{var}(x)}{\text{var}(\text{blur}(x))}$ (in dB).
SURE-LET algorithm presented in this paper is 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].

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).

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. 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

$$J = \sum_{i=1}^{N} (y_i - \hat{x}_i)^2 + \lambda \sum_{i=1}^{N} \log(\hat{x}_i)$$

where $y_i$ is the observed image, $\hat{x}_i$ is the estimate and $\lambda$ is the regularization parameter.

The performance of blind deconvolution is only 0.04dB of PSNR loss with the estimated PSF. 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).

Table 2. Computational time (in seconds)

<table>
<thead>
<tr>
<th>Methods</th>
<th>line search</th>
<th>blur-SURE-LET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>0.15</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Table 3. Computational time (in seconds)

<table>
<thead>
<tr>
<th>Methods</th>
<th>line search</th>
<th>blur-SURE-LET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>8.75</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Fig. 4. A visual comparison of Cameraman.

<table>
<thead>
<tr>
<th>observed PSNR=20.74dB</th>
<th>by exact PSF PSNR=22.85dB</th>
<th>by estimated PSF PSNR=22.81dB</th>
</tr>
</thead>
</table>

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

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).

REFERENCES