RECURSIVE EVALUATION OF SURE FOR TOTAL VARIATION DENOISING

Feng Xue[a], Thierry Blu[b], Jiaqi Liu[a] and Xia Ai[a]

[a]National Key Laboratory of Science and Technology on Test Physics and Numerical Mathematics, Beijing, 100076, China
[b]Dept. of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong

ABSTRACT

Recently, total variation (TV)-based regularization has become a standard technique for signal denoising. The reconstruction quality is generally sensitive to the value of regularization parameter. In this work, based on Chambolle's algorithm, we develop two data-driven optimization schemes based on minimization of Stein's unbiased risk estimate (SURE)—statistically equivalent to mean squared error (MSE). First, we propose a recursive evaluation of SURE to monitor the estimation error during Chambolle's iteration; the optimal value is then identified by the minimum SURE. Second, for fast optimization, we perform alternating update between regularization parameter and solution within Chambolle's iteration. We exemplify the proposed methods with both 1-D and 2-D signal denoising. Numerical experiments show that the proposed methods lead to highly accurate estimate of regularization parameter and nearly optimal denoising performance.

Index Terms— Signal denoising, total variation (TV), Stein's unbiased risk estimate (SURE), Chambolle's algorithm

1. INTRODUCTION

Consider the standard signal denoising problem: find a good estimate of $\mathbf{x}_0 \in \mathbb{R}^N$ from the following observation model [1–4]:

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

where $\mathbf{y} \in \mathbb{R}^N$ is the observed noisy data, $\epsilon \in \mathbb{R}^N$ is an additive Gaussian white noise with known variance $\sigma^2 > 0$.

Since the seminal work of ROF [5], total variation(TV)-based regularization has become a standard technique for signal denoising, which is particularly effective for recovering those signals with piecewise constant region while preserving edges [5, 6]. It formulates the denoising problem as the following TV-minimization [2, 3, 5, 7, 8]:

$$\min_{\mathbf{x}} \underbrace{\frac{1}{2}\left\|\mathbf{y}-\mathbf{x}\right\|_2^2 + \lambda \cdot \mathrm{TV}(\mathbf{x})}_{\mathcal{L}(\mathbf{x})} \tag{2}$$

In this paper, we choose Chambolle's algorithm to solve the TV minimization [9], since it is quite fast and probably one of the most popular algorithms in the recent decade. In the original algorithm of [9], the TV follows the standard non-smooth definition. However, we consider here a smooth approximation of isotropic TV, defined as:

$$\mathrm{TV}(\mathbf{x}) = \begin{cases} \sum_n \sqrt{\left|(\mathbf{D}\mathbf{x})_n\right|^2 + \beta} & \text{(1-D case)} \\ \sum_n \sqrt{\left|(\mathbf{D}_1\mathbf{x})_n\right|^2 + \left|(\mathbf{D}_2\mathbf{x})_n\right|^2 + \beta} & \text{(2-D case)} \end{cases} \tag{3}$$

for some small $\beta > 0$. Here $\mathbf{D}_1$ and $\mathbf{D}_2$ denote the horizontal and vertical first-order differences, respectively. Such continuously differentiable definition of TV would facilitate the computation of SURE (see Eq.(10) in Section 2.2.1).

For a pleasant denoising quality, it is essential to select a proper value of the regularization parameter λ, to keep a good balance between data fidelity and TV enforcement [7,9]. Discrepancy principle has been used to tune this parameter in Chambolle's algorithm [9].

We denote the solution to (2) by $\widehat{\mathbf{x}}_\lambda$, to emphasize the strong dependency of the estimate upon λ. In this paper, we quantify the denoising performance by the mean squared error (MSE) [1, 10]:

$$\mathrm{MSE} = \frac{1}{N}\mathbb{E}\left\{\left\|\widehat{\mathbf{x}}_\lambda - \mathbf{x}_0\right\|_2^2\right\} \tag{4}$$

and attempt to select a value of λ, such that the corresponding solution $\widehat{\mathbf{x}}_\lambda$ achieves minimum MSE. Notice that MSE (4) is inaccessible due to the unknown $\mathbf{x}_0$. In practice, Stein's unbiased risk estimate (SURE) has been proposed as a statistical substitute for MSE [1, 11]:

$$\mathrm{SURE} = \frac{1}{N}\left\|\widehat{\mathbf{x}}_\lambda - \mathbf{y}\right\|_2^2 + \frac{2\sigma^2}{N}\mathrm{Tr}\Big(\mathbf{J}_{\mathbf{y}}(\widehat{\mathbf{x}}_\lambda)\Big) - \sigma^2 \tag{5}$$

since it depends on the observed data $\mathbf{y}$ only. Here, $\mathbf{J}_{\mathbf{y}}(\widehat{\mathbf{x}}_\lambda) \in \mathbb{R}^{N\times N}$ is a Jacobian matrix defined as:

$$\Big[\mathbf{J}_{\mathbf{y}}(\widehat{\mathbf{x}}_\lambda)\Big]_{n,m} = \frac{\partial(\widehat{\mathbf{x}}_\lambda)_n}{\partial y_m}$$

Recently, SURE has become a popular criterion for optimization, in the context of non-linear denoising and deconvolution [1, 10], and ℓ_1-based sparse reconstruction [12–14]. However, to our best knowledge, there are very few literature on the application of SURE to TV-minimization.

This paper is to optimize the regularization parameter λ for TV denoising, based on minimization of SURE (5). Our main contributions are twofold. First, we develop a recursive evaluation of SURE for Chambolle's algorithm, which finally provides a reliable estimate of the MSE for the non-linear reconstruction. The optimal λ can then be identified by exhaustive search for minimum SURE. Furthermore, for fast optimization, we perform alternating update between regularization parameter and solution within the Chambolle's iteration, which yields very accurate estimate of optimal λ and nearly optimal denoising performance.

2. RECURSIVE EVALUATION OF SURE FOR CHAMBOLLE'S ALGORITHM

2.1. Basic scheme of Chambolle's algorithm

The original Chambolle's algorithm is for image denoising of 2-D case only [9]. Now, we will present this algorithm in matrix language for both 1-D and 2-D cases, which helps to keep a succinct style for the development of SURE later.

For 2-D case, introducing an auxiliary vector $\mathbf{u}$ in TV domain, Chambolle's iteration can be expressed as:

$$\mathbf{u}^{(i+1)} = \overline{\mathbf{V}}^{(i)}\Big(\overbrace{\mathbf{u}^{(i)} - \frac{\tau}{\lambda}\mathbf{D}\underbrace{(\mathbf{y}+\lambda\mathbf{D}^{\mathrm{T}}\mathbf{u}^{(i)})}_{\mathbf{x}^{(i)}}}^{\mathbf{w}^{(i)}}\Big) \tag{6}$$

where τ is a step-size, $\mathbf{x}^{(i)}$ can be obtained by the update of $\mathbf{u}^{(i)}$. $\mathbf{D}$ and diagonal matrix $\overline{\mathbf{V}}^{(i)}$ are

$$\mathbf{D} = \begin{bmatrix}\mathbf{D}_1\\ \mathbf{D}_2\end{bmatrix} \in \mathbb{R}^{2N\times N}; \quad \overline{\mathbf{V}}^{(i)} = \begin{bmatrix}\mathbf{V}^{(i)} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}^{(i)}\end{bmatrix} \in \mathbb{R}^{2N\times 2N}$$

with diagonal $\mathbf{V}^{(i)} \in \mathbb{R}^{N\times N}$ given by:

$$\mathbf{V}^{(i)}_{n,n} = \left(1+\frac{\tau}{\lambda}\sqrt{\left((\mathbf{D}_1\mathbf{x}^{(i)})_n\right)^2 + \left((\mathbf{D}_2\mathbf{x}^{(i)})_n\right)^2 + \beta}\right)^{-1}$$

For 1-D case, Chambolle's iteration is also expressed by (6), with the diagonal matrix $\mathbf{V}^{(i)} \in \mathbb{R}^{N\times N}$ given by:

$$\mathbf{V}^{(i)}_{n,n} = \left(1+\frac{\tau}{\lambda}\sqrt{\left((\mathbf{D}\mathbf{x}^{(i)})_n\right)^2 + \beta}\right)^{-1}$$

2.2. Recursive evaluation of SURE

From (5), the SURE for the i-th iterate is:

$$\mathrm{SURE} = \frac{1}{N}\left\|\mathbf{x}^{(i)} - \mathbf{y}\right\|_2^2 + \frac{2\sigma^2}{N}\mathrm{Tr}\Big(\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)})\Big) - \sigma^2 \tag{7}$$

The computation of SURE requires to compute $\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)})$, which can be evaluated in a recursive manner, as shown later. We first develop SURE for 2-D case, and then 1-D case will be readily obtained.

2.2.1. *SURE for 2-D case*

We rewrite (6) as:

$$\underbrace{\begin{bmatrix}\mathbf{u}_1^{(i+1)}\\ \mathbf{u}_2^{(i+1)}\end{bmatrix}}_{\mathbf{u}^{(i+1)}} = \underbrace{\begin{bmatrix}\mathbf{V}^{(i)} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}^{(i)}\end{bmatrix}}_{\overline{\mathbf{V}}^{(i)}}\underbrace{\begin{bmatrix}\mathbf{w}_1^{(i)}\\ \mathbf{w}_2^{(i)}\end{bmatrix}}_{\mathbf{w}^{(i)}}$$

Let us consider the first part—$\mathbf{u}_1^{(i+1)}$:

$$\mathbf{u}_1^{(i+1)} = \mathbf{V}^{(i)}\Big(\underbrace{\mathbf{u}_1^{(i)} - \frac{\tau}{\lambda}\mathbf{D}_1\mathbf{x}^{(i)}}_{\mathbf{w}_1^{(i)}}\Big)$$

The Jacobian matrix of $\mathbf{u}_1^{(i+1)}$ is:

$$\begin{aligned}\left[\mathbf{J}_{\mathbf{y}}(\mathbf{u}_1^{(i+1)})\right]_{m,n} &= \frac{\partial(\mathbf{u}_1^{(i+1)})_m}{\partial y_n} = \frac{\partial}{\partial y_n}\Big(\mathbf{V}^{(i)}_{m,m}(\mathbf{w}_1^{(i)})_m\Big)\\ &= (\mathbf{w}_1^{(i)})_m\frac{\partial \mathbf{V}^{(i)}_{m,m}}{\partial y_n} + \mathbf{V}^{(i)}_{m,m}\frac{\partial(\mathbf{w}_1^{(i)})_m}{\partial y_n}\end{aligned} \tag{8}$$

Let $\mathbf{a} = \mathbf{D}_1\mathbf{x}^{(i)}$ and $\mathbf{b} = \mathbf{D}_2\mathbf{x}^{(i)}$ (ignoring the superscript (i) of $\mathbf{a}$ and $\mathbf{b}$ for brevity). The continuous differentiability of TV definition (3) makes (8) easy to compute:

$$\begin{aligned}\frac{\partial \mathbf{V}^{(i)}_{m,m}}{\partial y_n} &= \frac{\partial \mathbf{V}^{(i)}_{m,m}}{\partial a_m}\cdot\frac{\partial a_m}{\partial y_n} + \frac{\partial \mathbf{V}^{(i)}_{m,m}}{\partial b_m}\cdot\frac{\partial b_m}{\partial y_n}\\ &= -\frac{\tau}{\lambda}\cdot\underbrace{\frac{a_m(\mathbf{V}^{(i)}_{m,m})^2}{\sqrt{a_m^2+b_m^2+\beta}}}_{(\mathbf{C}_1^{(i)})_{m,m}}\cdot\frac{\partial a_m}{\partial y_n} - \frac{\tau}{\lambda}\cdot\underbrace{\frac{b_m(\mathbf{V}^{(i)}_{m,m})^2}{\sqrt{a_m^2+b_m^2+\beta}}}_{(\mathbf{C}_2^{(i)})_{m,m}}\cdot\frac{\partial b_m}{\partial y_n}\\ &= -\frac{\tau}{\lambda}\Big[\mathbf{C}_1^{(i)}\mathbf{D}_1\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)})\Big]_{m,n} - \frac{\tau}{\lambda}\Big[\mathbf{C}_2^{(i)}\mathbf{D}_2\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)})\Big]_{m,n}\end{aligned} \tag{9}$$

Substituting (9) into (8), we obtain:

$$\mathbf{J}_{\mathbf{y}}(\mathbf{u}_1^{(i+1)}) = -\frac{\tau}{\lambda}\mathbf{W}_1^{(i)}\Big(\mathbf{C}_1^{(i)}\mathbf{D}_1 + \mathbf{C}_2^{(i)}\mathbf{D}_2\Big)\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)}) + \mathbf{V}^{(i)}\mathbf{J}_{\mathbf{y}}(\mathbf{w}_1^{(i)})$$

where $\mathbf{J}_{\mathbf{y}}(\mathbf{w}_1^{(i)}) = \mathbf{J}_{\mathbf{y}}(\mathbf{u}_1^{(i)}) - \frac{\tau}{\lambda}\mathbf{D}_1\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)})$ by the basic property of Jacobian matrix [14]. Similarly, $\mathbf{J}_{\mathbf{y}}(\mathbf{u}_2^{(i+1)})$ is given by:

$$\mathbf{J}_{\mathbf{y}}(\mathbf{u}_2^{(i+1)}) = -\frac{\tau}{\lambda}\mathbf{W}_2^{(i)}\Big(\mathbf{C}_1^{(i)}\mathbf{D}_1 + \mathbf{C}_2^{(i)}\mathbf{D}_2\Big)\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)}) + \mathbf{V}^{(i)}\mathbf{J}_{\mathbf{y}}(\mathbf{w}_2^{(i)})$$

Finally, the recursion of $\mathbf{J}_{\mathbf{y}}(\mathbf{u}^{(i+1)})$ is summarized as:

$$\begin{aligned}\mathbf{J}_{\mathbf{y}}(\mathbf{u}^{(i+1)}) = \begin{bmatrix}\mathbf{J}_{\mathbf{y}}(\mathbf{u}_1^{(i+1)})\\ \mathbf{J}_{\mathbf{y}}(\mathbf{u}_2^{(i+1)})\end{bmatrix} &= \begin{bmatrix}\mathbf{V}^{(i)} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}^{(i)}\end{bmatrix}\mathbf{J}_{\mathbf{y}}(\mathbf{u}^{(i)})\\ &-\frac{\tau}{\lambda}\begin{bmatrix}\mathbf{W}_1^{(i)}\mathbf{C}_1^{(i)} + \mathbf{V}^{(i)} & \mathbf{W}_1^{(i)}\mathbf{C}_2^{(i)}\\ \mathbf{W}_2^{(i)}\mathbf{C}_1^{(i)} & \mathbf{W}_2^{(i)}\mathbf{C}_2^{(i)} + \mathbf{V}^{(i)}\end{bmatrix}\mathbf{D}\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)})\end{aligned} \tag{10}$$

Note that $\mathbf{x}^{(i)} = \mathbf{y} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{u}^{(i)}$, we have:

$$\mathbf{J}_{\mathbf{y}}(\mathbf{x}^{(i)}) = \mathbf{I} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{J}_{\mathbf{y}}(\mathbf{u}^{(i)})$$

2.2.2. *SURE for 1-D case*

Based on the above discussions of 2-D case, the recursions of Jacobian matrices for 1-D case are given by:

$$\begin{cases} \mathbf{J_y}(\mathbf{u}^{(i+1)}) = \mathbf{V}^{(i)}\mathbf{J_y}(\mathbf{u}^{(i)}) - \frac{\tau}{\lambda}\Big(\mathbf{W}^{(i)}\mathbf{C}^{(i)} + \mathbf{V}^{(i)}\Big)\mathbf{D}\mathbf{J_y}(\mathbf{x}^{(i)}) \\ \mathbf{J_y}(\mathbf{x}^{(i)}) \quad = \mathbf{I} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{J_y}(\mathbf{u}^{(i)}) \end{cases} \tag{11}$$

where the diagonal matrix $\mathbf{C}^{(i)}$ is $[\mathbf{C}^{(i)}]_{n,n} = \frac{(\mathbf{D}\mathbf{x}^{(i)})_n(\mathbf{V}^{(i)}_{n,n})^2}{\sqrt{((\mathbf{D}\mathbf{x}^{(i)})_n)^2+\beta}}$.

2.3. Summary of Chambolle's algorithm with SURE evaluation

Finally, we summarize the proposed algorithm as **Algorithm 1**, which enables us to solve (2) with a prescribed value of λ, and simultaneously evaluate the SURE during the Chambolle's iterations.

Algorithm 1: SURE evaluation for Chambolle's algorithm of 1-D and 2-D cases

Input: $\mathbf{y}$, λ, β, τ, initial $\mathbf{u}^{(0)}$ and $\mathbf{x}^{(0)}$
Output: reconstructed $\widehat{\mathbf{x}}_\lambda$ and SURE($\widehat{\mathbf{x}}_\lambda$)
for $i = 1, 2, \ldots$ *(Chambolle's iteration)* **do**
  1 compute $\mathbf{x}^{(i)}$ by (6);
  2 update $\mathbf{J_y}(\mathbf{x}^{(i)})$ by (10) or (11);
  3 compute SURE of i-th iterate by (7);
end

To find the optimal value of λ, an intuitive idea is to repeatedly implement **Algorithm 1** for various tentative values of λ, then, the minimum SURE indicates the optimal λ (see Fig.2-(2) for example). This *global search* has been frequently used in [12–14].

2.4. Monte-Carlo for practical computation

For 2-D case, due to the limited computational resources (e.g. RAM), it is impractical to store the huge matrix $\mathbf{J_y}(\mathbf{x}^{(i)})$ and compute the trace. Monte-Carlo simulation provides an alternative way to compute the trace by the following fact [1]:

$$\mathrm{Tr}\Big(\mathbf{J_y}(\mathbf{x}^{(i)})\Big) = \mathbb{E}\Big\{\mathbf{n}_0^{\mathrm{T}}\mathbf{J_y}(\mathbf{x}^{(i)})\mathbf{n}_0\Big\} \tag{12}$$

with $\mathbf{n}_0 \sim \mathcal{N}(0, \mathbf{I}_N)$. We rewrite (10) as:

$$\begin{cases} \mathbf{J_y}(\mathbf{u}_1^{(i+1)}) = \mathbf{V}^{(i)}\mathbf{J_y}(\mathbf{u}_1^{(i)}) - \frac{\tau}{\lambda}\mathbf{P}_1^{(i)}\mathbf{J_y}(\mathbf{x}^{(i)}) \\ \mathbf{J_y}(\mathbf{u}_2^{(i+1)}) = \mathbf{V}^{(i)}\mathbf{J_y}(\mathbf{u}_2^{(i)}) - \frac{\tau}{\lambda}\mathbf{P}_2^{(i)}\mathbf{J_y}(\mathbf{x}^{(i)}) \end{cases}$$

where $\mathbf{P}_1^{(i)} = (\mathbf{W}_1^{(i)}\mathbf{C}_1^{(i)} + \mathbf{V}^{(i)})\mathbf{D}_1 + \mathbf{W}_1^{(i)}\mathbf{C}_2^{(i)}\mathbf{D}_2$ and $\mathbf{P}_2^{(i)} = \mathbf{W}_2^{(i)}\mathbf{C}_1^{(i)}\mathbf{D}_1 + (\mathbf{W}_2^{(i)}\mathbf{C}_2^{(i)} + \mathbf{V}^{(i)})\mathbf{D}_2$. With the input $\mathbf{n}_0$, the noise evolution with the Chambolle's iteration is:

$$\begin{cases} \overbrace{\mathbf{J_y}(\mathbf{u}_1^{(i+1)})\mathbf{n}_0}^{\mathbf{n}_{\mathbf{u}_1}^{(i+1)}} = \mathbf{V}^{(i)}\overbrace{\mathbf{J_y}(\mathbf{u}_1^{(i)})\mathbf{n}_0}^{\mathbf{n}_{\mathbf{u}_1}^{(i)}} - \frac{\tau}{\lambda}\mathbf{P}_1^{(i)}\overbrace{\mathbf{J_y}(\mathbf{x}^{(i)})\mathbf{n}_0}^{\mathbf{n}_{\mathbf{x}}^{(i)}} \\ \underbrace{\mathbf{J_y}(\mathbf{u}_2^{(i+1)})\mathbf{n}_0}_{\mathbf{n}_{\mathbf{u}_2}^{(i+1)}} = \mathbf{V}^{(i)}\underbrace{\mathbf{J_y}(\mathbf{u}_2^{(i)})\mathbf{n}_0}_{\mathbf{n}_{\mathbf{u}_2}^{(i)}} - \frac{\tau}{\lambda}\mathbf{P}_2^{(i)}\underbrace{\mathbf{J_y}(\mathbf{x}^{(i)})\mathbf{n}_0}_{\mathbf{n}_{\mathbf{x}}^{(i)}} \end{cases} \tag{13}$$

and

$$\underbrace{\mathbf{J_y}(\mathbf{x}^{(i)})\mathbf{n}_0}_{\mathbf{n}_{\mathbf{x}}^{(i)}} = \mathbf{n}_0 + \lambda\mathbf{D}_1^{\mathrm{T}}\underbrace{\mathbf{J_y}(\mathbf{u}_1^{(i)})\mathbf{n}_0}_{\mathbf{n}_{\mathbf{u}_1}^{(i)}} + \lambda\mathbf{D}_2^{\mathrm{T}}\underbrace{\mathbf{J_y}(\mathbf{u}_2^{(i)})\mathbf{n}_0}_{\mathbf{n}_{\mathbf{u}_2}^{(i)}} \tag{14}$$

Thus, instead of using (10), we can successfully compute the trace of $\mathbf{J_y}(\mathbf{x}^{(i)})$ by (13) and (14), without the storage of huge matrices. The MC evaluation is summarized as follows.

Algorithm 2: MC for SURE evaluation (for 2-D case)

for $i = 1, 2, \ldots$ *(Chambolle's iteration)* **do**
  1 compute $\mathbf{x}^{(i)}$ by (6);
  2 compute $\mathbf{n}_{\mathbf{u}_1}^{(i)}$, $\mathbf{n}_{\mathbf{u}_2}^{(i)}$ and $\mathbf{n}_{\mathbf{x}}^{(i)}$ by (13) and (14);
  3 compute the trace of $\mathbf{J_y}(\mathbf{x}^{(i)})$ by (12);
  4 compute SURE of i-th iterate by (7);
end

3. SURE-BASED ALTERNATING OPTIMIZATION WITHIN CHAMBOLLE'S ALGORITHM

Note that the global search for the optimal λ requires repeated implementations of the Chambolle's algorithm, which is rather computationally expensive. It is possible to reduce the computational complexity, if the optimization of λ can be completed within ONE execution of the Chambolle's iteration. A possible solution is to alternatively optimize the parameter $\lambda^{(i)}$ (by the SURE minimization) and update the solution $\mathbf{x}^{(i)}$ (using $\lambda^{(i)}$) within the i-th iterate—so-called *alternating optimization*, summarized in Fig.1.

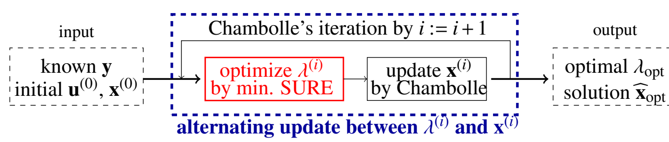

Fig. 1. Alternating optimization between parameter $\lambda^{(i)}$ and solution $\mathbf{x}^{(i)}$ within i-th step of Chambolle's iteration.

Let us consider the highlighted step of Fig.1—optimization of $\lambda^{(i)}$ by minimizing SURE, which is written as:

$$\lambda^{(i)} = \arg\min_{\lambda} \underbrace{\frac{1}{N}\|\mathbf{x}_\lambda^{(i)} - \mathbf{y}\|_2^2 + \frac{2\sigma^2}{N}\mathrm{Tr}\Big(\mathbf{J_y}(\mathbf{x}_\lambda^{(i)})\Big) - \sigma^2}_{\text{SURE of } \mathbf{x}_\lambda^{(i)}} \tag{15}$$

Here, $\mathbf{x}_\lambda^{(i)}$ is obtained by (6) with the previous iterate $\mathbf{x}^{(i-1)}$ and tentative λ. Since the SURE is not a simple function of λ, a straightforward method is to perform exhaustive search for the optimal value of $\lambda^{(i)}$ during each iterate.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

4.1. Parameter setting and initialization

Recalling that to facilitate the computation of SURE, we defined a differentiable TV with a smoothing parameter β.

Extensive experiments found that $\beta = 10^{-12}$ is sufficient to keep both differentiability and accuracy of TV. We always use this value throughout this paper.

Typically, $\mathbf{u}^{(0)}$ is initialized as $\mathbf{u}^{(0)} = \mathbf{0} \in \mathbb{R}^{2N}$, and thus, the resultant $\mathbf{x}^{(0)} = \mathbf{y}$ by (6).

4.2. 1-D signal denoising

We first consider a 1-D signal denoising, where $\mathbf{x}_0 \in \mathbb{R}^{256}$ denotes a 1-D signal. Then, we add the noise ϵ with noise variance $\sigma^2 = 100$. First, we apply **Algorithm 1** to solve (2) with fixed $\lambda = 1$. Fig.2-(1) shows the Chambolle's convergence and the evolution of SURE during the iterations. Fig.2-(2) shows the global search for optimal λ. We can see that the SURE is always a reliable substitute for MSE.

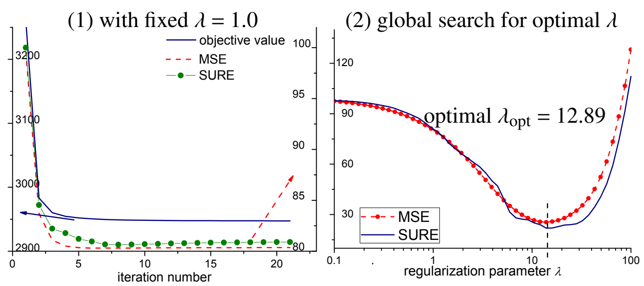

Fig. 2. The convergence of Chambolle's algorithm and the global optimization (1-D case).

We implement the procedure of Fig.1 to perform alternating optimization, and show the results in Fig.3.

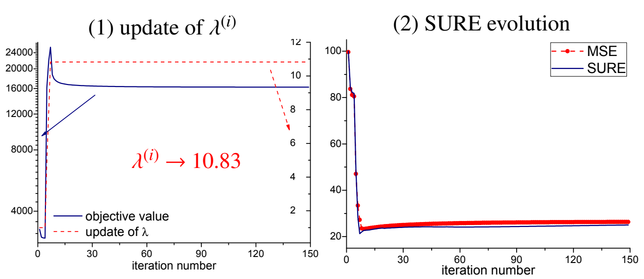

Fig. 3. Alternating optimization (1-D case).

4.3. Image denoising

We now consider a 2-D image *Cameraman*, the noise variance is $\sigma^2 = 100$. First, we apply **Algorithm 2** to solve (2) with fixed $\lambda = 1$, where the SURE is evaluated by MC. Fig.4 shows the Chambolle's convergence and global optimization of λ. Fig.5 shows the results of alternating optimization. Fig.6 shows the optimally denoised images.

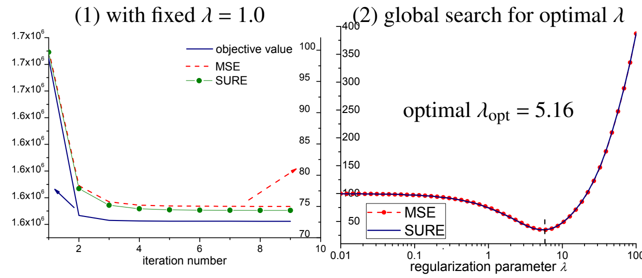

Fig. 4. The convergence and the global optimization (2-D case).

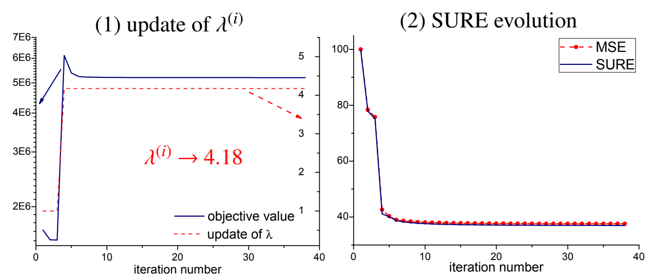

Fig. 5. Alternating optimization (2-D case).

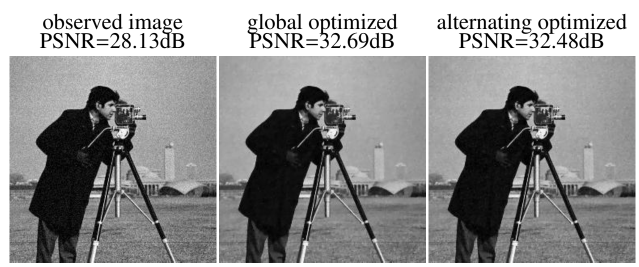

Fig. 6. A visual example of *Cameraman*.

Table 1 reports the results and computational time (in seconds) of global and alternating ('ALT' in the table) optimizations. We also compare both methods with discrepancy principle used in [9] ('DP [9]' in the table). We can see that the alternating strategy yields nearly optimal denoising performance, with considerably faster computational speed.

Table 1. Comparisons between global and alternating optimization

case	1-D signal			2-D image *Cameraman*		
methods	est. λ	MSE	time	est. λ	MSE	time
DP [9]	16.11	30.13	15	8.13	45.55	121
global	12.89	25.89	75	5.16	35.83	767
ALT	10.83	26.34	**10**	4.18	36.58	**70**

5. CONCLUSIONS

In this paper, we presented two SURE-based automatic methods of tuning regularization parameter for total variation denoising, based on Chambolle's algorithm [9]. Future work will deal with extension of this technique to handle other regularizers [12–14] and image deconvolution [10, 13].

6. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under grant number 61401013 and the Hong Kong Research Grant Council (CUHK410012).

7. REFERENCES

[1] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2778–2786, 2007.

[2] I. W. Selesnick, A. Parekh, and I. Bayram, "Convex 1-D total variation denoising with non-convex regularization," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 141–144, 2015.

[3] I. W. Selesnick and P.Y. Chen, "Total variation denoising with overlapping group sparsity," in *IEEE Int. Conf. Acoust., Speech, Signal Proc. (ICASSP)*, 2013, pp. 5696–5700.

[4] I. W. Selesnick, H. L. Graber, D. S. Pfeil, and R. L. Barbour, "Simultaneous low-pass filtering and total variation denoising," *IEEE Trans. Signal Processing*, vol. 62, no. 5, pp. 1109–1124, 2014.

[5] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[6] J. Oliveira, M. Figueiredo, and J. Bioucas-Dias, "Adaptative total variation image deblurring: a majorization-minimization approach," *Signal Processing*, vol. 89, no. 9, pp. 1683–1693, 2009.

[7] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W.T. Yin, "An iterative regularization method for total variation-based image restoration," *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 2, pp. 460–489, 2005.

[8] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM J. Sci. Comput.*, vol. 17, no. 1, pp. 227–238, 1996.

[9] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89–97, 2004.

[10] 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.

[11] Charles M Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, pp. 1135–1151, 1981.

[12] R. Giryes, M. Elad, and Y.C. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 407–422, 2010.

[13] C. Vonesch, S. Ramani, and M. Unser, "Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint," in *IEEE Int. Conf. on Image Proc.*, 2008, pp. 665–668.

[14] F. Xue, Anatoly G. Yagola, J.Q. Liu, and G. Meng, "Recursive SURE for iterative reweighted least square algorithms," *Inverse Problems in Science and Engineering*, vol. 24, no. 4, pp. 625–646, 2016.