Iterative Weighted Risk Estimation for Nonlinear Image Restoration with Analysis Priors

Sathish Ramani, Jeffrey Rosen, Zhihao Liu and Jeffrey A. Fessler

EECS Department, University of Michigan, Ann Arbor, MI, USA.

ABSTRACT

Image acquisition systems invariably introduce blur, which necessitates the use of deblurring algorithms for image restoration. Restoration techniques involving regularization require appropriate selection of the regularization parameter that controls the quality of the restored result. We focus on the problem of automatic adjustment of this parameter for nonlinear image restoration using *analysis*-type regularizers such as total variation (TV). For this purpose, we use two variants of Stein's unbiased risk estimate (SURE), Predicted-SURE and Projected-SURE, that are applicable for parameter selection in inverse problems involving Gaussian noise. These estimates require the Jacobian matrix of the restoration algorithm evaluated with respect to the data. We derive analytical expressions to recursively update the desired Jacobian matrix for a fast variant of the iterative reweighted least-squares restoration algorithm that can accommodate a variety of regularization criteria. Our method can also be used to compute a nonlinear version of the generalized cross-validation (NGCV) measure for parameter tuning. We demonstrate using simulations that Predicted-SURE, Projected-SURE, and NGCV-based adjustment of the regularization parameter yields near-MSE-optimal results for image restoration using TV, an *analysis*-type ℓ_1-regularization, and a smooth convex edge-preserving regularizer.

Keywords: Image restoration, regularization parameter, generalized cross-validation, Stein's unbiased risk estimate, Jacobian matrix.

1. INTRODUCTION

Restoration is an important image-processing step in many applications where the goal is to recover an underlying image of interest $\mathbf{x}$ from blurred and noisy measurements $\mathbf{y}$. One way to solve this problem is to minimize a cost function Ψ composed of a data-fidelity term (that measures the consistency of the restored result to the measurements) and a regularization term (that reduces the effect of noise by "smoothing" the solution). Nonquadratic regularization criteria such as edge-preserving ones, e.g., total variation (TV) or those that are based on sparsity (e.g., ℓ_1-regularization) can provide better quality[1] compared to quadratic Tikhonov-like regularizers. However proper application of regularization demands careful selection of the so-called regularization parameter, β, that balances the trade-off between noise amplification and image-smoothing. This task is nontrivial and is often performed manually based on visual inspection, so a means of automation is desirable.

Generalized cross-validation (GCV) is a popular quantitative method[2] for automatic selection of β, especially for linear restoration algorithms. The primary advantage of GCV is that it does

Further author information: Sathish Ramani, Email: sramani@umich.edu; Jeffrey Rosen, Email: jayar@umich.edu; Zhihao Liu, Email: zhihao@umich.edu and Jeffrey A. Fessler, Email: fessler@umich.edu, Telephone: +1 734 763 1434. Send correspondence to Jeffrey A. Fessler.

Computational Imaging X, edited by Charles A. Bouman, Ilya Pollak,
Patrick J. Wolfe, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 8296, 82960N
© 2012 SPIE-IS&T · CCC code: 0277-786X/12/$18 · doi: 10.1117/12.915643

not require knowledge of the noise variance σ^2. Additionally, GCV is known to yield β that asymptotically minimizes the *true* mean squared-error (MSE, also known as *risk*) for linear algorithms. Extensions of GCV to nonlinear algorithms (NGCV) are available,[3,4] but they are more involved analytically[4] and more expensive to compute compared to the linear version. An alternative to GCV is to use methods that directly attempt to minimize the MSE (or its weighted variants) via the principles of *risk estimation*.[5–11] Such techniques are appealing since MSE-type measures are commonly used to quantify reconstruction quality.

When noise is modeled as Gaussian, Stein's principle[5] can be used to derive an estimate (*Stein's unbiased risk estimate*—SURE) of MSE for denoising applications. Unlike GCV, SURE requires the knowledge of σ^2, but is optimal even in the nonasymptotic regime and has been used for optimally adjusting parameters of a variety of denoising algorithms.[9–11] For ill-posed inverse problems however, it may not be possible to directly estimate MSE since $\mathbf{y}$ may only contain partial information[6] about $\mathbf{x}$. In such cases, weighted variants such as Predicted-MSE and Projected-MSE (that evaluate the error only on those components of $\mathbf{x}$ that are accessible[6,12] through $\mathbf{y}$) may be used. Applying Stein's principle to Predicted-MSE and Projected-MSE yields Predicted-SURE,[12] and Projected-SURE,[8] respectively, which also require σ^2.

In this paper, we propose to employ and assess NGCV, Predicted-SURE and Projected-SURE for selecting the regularization parameter β for nonlinear image restoration. Computation of these quantities requires[6–8] the Jacobian matrix of the restoration algorithm evaluated with respect to $\mathbf{y}$. We evaluate the desired Jacobian matrix for a fast variant of the iterative reweighted least-squares (IRLS) restoration algorithm[13,14] that can accommodate a variety of regularization criteria. Our work can be interpreted as an extension to those of Vonesch *et al.*[7] and Giryes *et al.*[8] that apply to nonlinear restoration specifically using iterative shrinkage-thresholding (IST)-type algorithms with *synthesis* priors. In *synthesis* formulations, the cost Ψ is minimized as a function of the coefficients of a suitable basis (e.g., wavelets) and the solution is obtained as a linear combination of the resulting optimized coefficients. Such formulations preclude popular edge-preserving regularizers like TV that belong to the category of *analysis* priors where Ψ is directly specified in terms of the desired solution. Moreover, Selesnick *et al.*[15] recently illustrated that *analysis* priors can provide better restoration quality compared to *synthesis* ones. Our method can be readily applied to a variety of *analysis*-type regularizers including TV and can also be easily extended to handle *synthesis* formulations. We demonstrate using various simulations that β selected to minimize NGCV, Predicted-SURE and Projected-SURE yield near-MSE-optimal results for image restoration using TV, an *analysis*-type ℓ_1-regularization and a smooth edge-preserving regularizer with the Fair potential.[16]

The paper is structured as follows. We describe image restoration and parameter selection problems mathematically in Section 2. We present details related to GCV, NGCV and the use of Stein's principle for estimating Predicted-MSE and Projected-MSE in Section 3. We describe our restoration algorithm and provide an analytical development of the evaluation of the desired Jacobian matrix in Section 4. We present numerical results in Section 5 and conclude the paper with Section 6.

2. PROBLEM FORMULATION

We adopt the following linear model for the restoration problem,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^N$ denotes blurred and noisy data, $\mathbf{x} \in \mathbb{R}^N$ is the unknown noise-free image, $\mathbf{b} \in \mathbb{R}^N$ is an *i.i.d.* zero-mean Gaussian random vector with variance σ^2, i.e., $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_N)$, and $\mathbf{I}_N$ is the identity matrix of size N. We assume that $\mathbf{A} \in \mathbb{R}^{N \times N}$, which represents the blur of the imaging system, is circulant. Throughout the paper, $(\cdot)^\top$ denotes the transpose of a vector or a matrix and the m-th element of any vector $\mathbf{y}$ is denoted by y_m and the mn-th element of any matrix $\mathbf{A}$ is written as $[\mathbf{A}]_{mn}$.

Recovering $\mathbf{x}$ from $\mathbf{y}$ is an ill-posed problem, especially when $\mathbf{A}$ has a nontrivial null space $\mathsf{N}\{\mathbf{A}\} \neq \emptyset$. Since regularization is an effective means of tackling the ill-posedness of (1), we formulate image restoration as a regularized reconstruction problem where we minimize a cost function Ψ composed of a quadratic data-fidelity term (that measures how closely the restored result agrees with the model in (1)) and a regularization term $\mathcal{R}$ (that enforces "smoothness" or regularity in the reconstruction thereby reducing the effect of noise):

$$\hat{\mathbf{x}} = \mathbf{f}_\beta(\mathbf{y}) \triangleq \underset{\mathbf{u}}{\operatorname{argmin}} \left\{ \Psi(\mathbf{u}) \triangleq \|\mathbf{y} - \mathbf{A}\mathbf{u}\|_2^2 + \beta \mathcal{R}(\mathbf{u}) \right\}. \tag{2}$$

Convex nonquadratic regularizers that preserve edges, e.g., total variation (TV), or those that promote sparsity, e.g., based on the ℓ_1-norm, are popular in image reconstruction problems as they can provide good quality reconstructions.[8, 13] Therefore, we consider

$$\mathcal{R}_\phi(\mathbf{u}) = \sum_{l=0}^{PM-1} \phi(|[\mathbf{R}\mathbf{u}]_l|),$$

where $\phi(\cdot)$ is a convex nonquadratic potential function, $\mathbf{R} = [\mathbf{R}_1{}^\top \cdots \mathbf{R}_P{}^\top]^\top$, and $\{\mathbf{R}_i\}_{i=1}^P$ are $M \times N$ (typically, $M \geq N$) regularization operators (e.g., finite differences along different directions). The general form $\mathcal{R}_\phi$ includes *analysis* ℓ_1-regularization for $\phi(x) = x$:

$$\mathcal{R}_{\ell_1}(\mathbf{u}) = \sum_{l=0}^{PM-1} |[\mathbf{R}\mathbf{u}]_l|, \tag{3}$$

and many convex smooth edge-preserving regularizers: as an instance, we consider

$$\mathcal{R}_{\mathrm{FP}}(\mathbf{u}) = \sum_{l=0}^{PM-1} \phi_{\mathrm{FP}}(|[\mathbf{R}\mathbf{u}]_l|), \tag{4}$$

where $\phi_{\mathrm{FP}}(x) \triangleq \delta^2[x/\delta - \log(1 + x/\delta)]$ is the Fair potential (FP)[16] with $\delta > 0$. The methods proposed in this work also apply to the strongly edge-preserving TV regularization:

$$\mathcal{R}_{\mathrm{TV}}(\mathbf{u}) = \sum_{l=0}^{M-1} \sqrt{\sum_{p=1}^{P} |[\mathbf{R}_p\mathbf{u}]_l|^2}. \tag{5}$$

For notational ease, we introduced the nonlinear operator $\mathbf{f}_\beta : \mathbb{R}^N \to \mathbb{R}^N$, representative of the minimization process in (2), that acts on the data $\mathbf{y}$ to yield the restored result $\hat{\mathbf{x}}$. The regularization parameter β in (2) (and $\mathbf{f}_\beta$) balances the effect of the data-fidelity and regularization terms on $\hat{\mathbf{x}}$: small β-values lead to noisy reconstructions while large values result in over-smoothing and loss of details.[8, 10] A proper choice of β is therefore crucial for obtaining good quality restoration.

3. SELECTION OF REGULARIZATION PARAMETER β

3.1 Generalized Cross Validation (GCV)

GCV[2] is an attractive method for selecting β, especially in the context of linear algorithms. For a generic linear reconstruction of the form, $\mathbf{f}_\beta(\mathbf{y}) = \mathbf{F}_\beta \mathbf{y}$ ($\mathbf{F}_\beta$ is matrix representing some type of inverse filtering), GCV selects β by minimizing

$$\mathrm{GCV}(\beta) = \frac{N^{-1}||\mathbf{A}\mathbf{f}_\beta(y) - \mathbf{y}||^2}{(1 - N^{-1}\mathrm{tr}\{\mathbf{A}\mathbf{F}_\beta\})^2}. \tag{6}$$

Calculation of the trace, $\mathrm{tr}\{\mathbf{A}\mathbf{F}_\beta\}$ in the denominator of (6), can be performed either analytically in some special cases, e.g., when $\mathbf{F}_\beta$ is circulant, or stochastically using Monte-Carlo methods[17] for a general $\mathbf{F}_\beta$. GCV(β) is simple to implement and is know to yield β that asymptotically provides an optimal reconstruction for linear algorithms.[2]

For nonlinear algorithms (denoted by $\mathbf{f}_\beta$), Deshpande *et al.*[3] proposed the following NGCV measure[3,4] (GCV for nonlinear algorithms) based on the principles of cross-validation:

$$\mathrm{NGCV}(\beta) = \frac{N^{-1}\|\mathbf{y} - \mathbf{A}\mathbf{f}_\beta(\mathbf{y})\|_2^2}{(1 - N^{-1}\mathrm{tr}\{\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\})^2}, \tag{7}$$

where $\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})$ is the Jacobian matrix consisting of partial derivatives of the components $\{f_{\beta,n}(\mathbf{y})\}_{n=1}^N$ of $\mathbf{f}_\beta(\mathbf{y})$ with respect to the components $\{y_n\}_{n=1}^N$ of $\mathbf{y}$: the kl-th element of $\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})$ is given by

$$[\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})]_{kl} = \left.\frac{\partial f_{\beta,k}(\mathbf{z})}{\partial z_l}\right|_{\mathbf{z}=\mathbf{y}}. \tag{8}$$

NGCV(β) is a generalization of GCV(β) for nonlinear algorithms.[3,4] It is more involved and computation intensive compared to GCV(β) (6) as it requires the evaluation of $\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})$.

3.2 Stein's Principle for Estimating MSE-type Measures

In image reconstruction problems, mean squared error (MSE),

$$\mathrm{MSE}(\beta) \triangleq N^{-1}\|\mathbf{x} - \mathbf{f}_\beta(\mathbf{y})\|_2^2, \tag{9}$$

is commonly used to determine quality of a reconstructed image and is an attractive alternative to (N)GCV for tuning β. However, MSE(β) cannot be directly used in practice due to its dependence on the unknown $\mathbf{x}$. For denoising applications, i.e., $\mathbf{A} = \mathbf{I}_N$ in (1), one can use Stein's principle* to estimate MSE(β) when noise is modeled as Gaussian. This process leads to the so-called Stein's Unbiased Risk Estimate (SURE)[5,9] given by $\mathrm{SURE}(\beta) = N^{-1}\|\mathbf{y} - \mathbf{f}_\beta(\mathbf{y})\|_2^2 - \sigma^2 + 2\sigma^2 N^{-1}\mathrm{tr}\{\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$. SURE($\beta$) is unbiased, i.e., $\mathcal{E}_\mathbf{b}\{\mathrm{MSE}(\beta)\} = \mathcal{E}_\mathbf{b}\{\mathrm{SURE}(\beta)\}$ (where $\mathcal{E}_\mathbf{b}\{\cdot\}$ denotes the expectation operation with respect to $\mathbf{b}$), but requires the knowledge of the noise variance σ^2 unlike (N)GCV. Evaluation of $\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})$ in SURE(β) can be performed analytically for some special nonlinear denoising algorithms (e.g., wavelets-based denoising[9] and nonlocal means[11]) or numerically using the Monte-Carlo method[10] for an arbitrary linear/nonlinear, iterative/noniterative algorithm $\mathbf{f}_\beta$.

*Application of Stein's principle requires the hypotheses that $\mathbf{f}_\beta$ is (weakly) differentiable[6,9] and decays sufficiently rapidly[6,9] such that $\lim_{b_n \to \infty} p(\mathbf{b}) f_{\beta,n}(\mathbf{y}) = 0 \;\forall\; n$, for the Gaussian probability density function $p(\mathbf{b}) = (2\pi\sigma^2)^{-N/2}\exp(-\|\mathbf{b}\|_2^2/2\sigma^2)$.

However, for inverse problems (1) involving $\mathbf{A}$ with a nontrivial null-space $\mathsf{N}(\mathbf{A})$, it is not possible to estimate MSE since information concerning $\mathbf{x}$ that lies in $\mathsf{N}(\mathbf{A})$ cannot be recovered from $\mathbf{y}$. In such cases, Projected-MSE may be used as an alternative[6,8] to MSE. Projected-MSE computes squared-error based on those components of $\mathbf{x}$ that lie in the orthogonal complement[6,8] of $\mathsf{N}(\mathbf{A})$ (equivalent to $\mathsf{R}(\mathbf{A}^\top)$, the range space of $\mathbf{A}^\top$) that are in turn accessible from $\mathbf{y}$ and thus allows for its estimation:[6,8]

$$\text{Projected-MSE}(\beta) = N^{-1}\|\mathbf{P}(\mathbf{x} - \mathbf{f}_\beta(\mathbf{y}))\|_2^2, \tag{10}$$

where $\mathbf{P} = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^\dagger\mathbf{A}$ is the matrix corresponding to a projection onto $\mathsf{R}(\mathbf{A}^\top)$ and $(\cdot)^\dagger$ denotes pseudo-inverse. Another squared-error measure that is amenable to estimation is Predicted-MSE (PMSE)[12] that computes the error in the measurement domain:

$$\text{Predicted-MSE}(\beta) = N^{-1}\|\mathbf{A}(\mathbf{x} - \mathbf{f}_\beta(\mathbf{y}))\|_2^2. \tag{11}$$

Both Projected-MSE and Predicted-MSE are special cases of the following weighted error measure (WMSE) that we consider for generality of exposition:

$$\text{WMSE}(\beta) = N^{-1}\|\mathbf{A}(\mathbf{x} - \mathbf{f}_\beta(\mathbf{y}))\|_{\mathbf{W}}^2, \tag{12}$$

where $\mathbf{W}$ is a positive definite ($\mathbf{W} \succ \mathbf{0}$),[§] symmetric ($\mathbf{W}^\top = \mathbf{W}$) weighting matrix. It is easy to see that $\mathbf{W} = (\mathbf{A}\mathbf{A}^\top)^\dagger$ for Projected-MSE in (10) and $\mathbf{W} = \mathbf{I}_N$ for Predicted-MSE in (11). Similar to SURE(β), one can arrive at an estimate for WMSE(β) using Stein's principle[6] as stated in the following theorem.

Theorem 3.1. *Let $\mathbf{f}_\beta$ in (2) be (weakly) differentiable and satisfy $\mathcal{E}_\mathbf{b}\{|[\mathbf{W}\mathbf{A}\,\mathbf{f}_\beta(\mathbf{y})]_n|\} < \infty\ \forall\ n$. Then, the random variable*

$$\text{WSURE}(\beta) = N^{-1}\|\mathbf{y} - \mathbf{A}\mathbf{f}_\beta(\mathbf{y})\|_{\mathbf{W}}^2 - \sigma^2 N^{-1}\text{tr}\{\mathbf{W}\} + 2\sigma^2 N^{-1}\text{tr}\{\mathbf{W}\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\} \tag{13}$$

is an unbiased estimator of WMSE(β) *(12), i.e.,* $\mathcal{E}_\mathbf{b}\{\text{WMSE}(\beta)\} = \mathcal{E}_\mathbf{b}\{\text{WSURE}(\beta)\}$. ■

A proof of this result can be obtained by a straightforward extension of previous results, e.g., of that of Eldar.[6] As with SURE(β), the key step in computing WSURE(β) is the calculation of $\text{tr}\{\mathbf{W}\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$ that in turn requires $\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})$. We propose to evaluate $\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})$ analytically for an iterative reweighted least squares (IRLS)-type algorithm[13,14] that can be applied to (2) for several regularizers including TV (5), ℓ_1-regularization (3) and smooth edge-preserving criteria such as (4).

4. EVALUATING THE JACOBIAN MATRIX $\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})$

4.1 Standard IRLS Algorithm

The iterative reweighted least squares (IRLS) algorithm performs the minimization in (2) by repetitively solving iteration-dependent linear systems[13] of the form

$$\mathbf{H}_{(i)}\mathbf{u}_{(i+1)} = \mathbf{A}^\top\mathbf{y}. \tag{14}$$

[§] A matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ is said to be positive definite, i.e., $\mathbf{W} \succ \mathbf{0}$, if $\mathbf{z}^\top\mathbf{W}\mathbf{z} > 0\ \forall\ \mathbf{z} \in \mathbb{R}^N$.

At iteration i, $\mathbf{H}_{(i)} = \mathbf{A}^\top \mathbf{A} + \mathbf{R}^\top \boldsymbol{\Omega}_{(i)}^{-1} \mathbf{R}$ and $\boldsymbol{\Omega}_{(i)} = \operatorname{diag}\{\boldsymbol{\omega}_{(i)}\}$ is a diagonal weighting matrix, where the n-th component of $\boldsymbol{\omega}_{(i)}$ is given by $\omega_{(i)n} = \left.\frac{t}{\beta\phi'(t)}\right|_{t=|[\mathbf{R}\mathbf{u}_{(i)}]_n|}$, $n = 0 \ldots PM-1$, and ϕ' is the derivative of ϕ. For the ℓ_1-regularization in (3), we have that

$$\omega_{(i)n} = \beta^{-1}\, |[\mathbf{R}\mathbf{u}_{(i)}]_n|, \tag{15}$$

while for the Fair potential (4),

$$\omega_{(i)n} = (\beta\,\delta)^{-1}(\delta + |[\mathbf{R}\mathbf{u}_{(i)}]_n|). \tag{16}$$

In case of TV, $\boldsymbol{\omega}_{(i)} = \mathbf{1}_P \otimes \breve{\boldsymbol{\omega}}_{(i)}$, where $\otimes$ denotes Kronecker product, $\mathbf{1}_P$ is a $P \times 1$ vector of 1s and the m-th element of $\breve{\boldsymbol{\omega}}_{(i)} \in \mathbb{R}^M$ is given by

$$\breve{\omega}_{(i)m} = \beta^{-1} \left(\sum_{p=1}^{P} |[\mathbf{R}_p \mathbf{u}_{(i)}]_m|^2 \right)^{1/2}. \tag{17}$$

Standard iterative solvers, e.g., preconditioned conjugate gradient (PCG), for (14) may converge slowly for nonsmooth regularizers such as ℓ_1-regularization (3) and TV (5) since $\mathbf{R}\mathbf{u}_{(i)}$ tends to become sparse with increasing i and correspondingly $\boldsymbol{\Omega}_{(i)}^{-1}$ becomes poorly conditioned for (15) and (17). To enhance numerical stability of IRLS (14), a small positive constant is often added to (15) and (17)—this is usually referred to as *corner rounding*[13] and is often administered to IRLS (14) when nonsmooth regularizers (such as ℓ_1-regularization and TV) are used.

4.2 The IRLS-MIL Algorithm

One can circumvent the use of corner rounding for IRLS by using a matrix splitting scheme.[13, 18] To see this, we rewrite (14) as

$$\mathbf{B}_{(i)} \mathbf{u}_{(i+1,j+1)} = \mathbf{A}^\top \mathbf{y} + (\mathbf{C}_{(i)} - \mathbf{A}^\top \mathbf{A}) \mathbf{u}_{(i+1,j)}, \tag{18}$$

where $\mathbf{B}_{(i)} = \mathbf{C}_{(i)} + \mathbf{R}^\top \boldsymbol{\Omega}_{(i)}^{-1} \mathbf{R}$ and $\mathbf{C}_{(i)}$ is an invertible matrix such that $\mathbf{C}_{(i)} - \mathbf{A}^\top \mathbf{A} \succ \mathbf{0}$. Solving for $\mathbf{u}_{(i+1,j+1)}$ in (18), we obtain the following iterative scheme

$$\mathbf{u}_{(i+1,j+1)} = \mathbf{B}_{(i)}^{-1} (\mathbf{A}^\top \mathbf{y} + (\mathbf{C}_{(i)} - \mathbf{A}^T \mathbf{A}) \mathbf{u}_{(i+1,j)}), \tag{19}$$

that is guaranteed[18] to converge on the j-iteration to a solution of (14). By applying matrix inversion lemma (MIL)[19] to $\mathbf{B}_{(i)}^{-1}$, we get $\mathbf{B}_{(i)}^{-1} = \mathbf{C}_{(i)}^{-1} - \mathbf{C}_{(i)}^{-1} \mathbf{R}^\top \mathbf{G}_i^{-1} \mathbf{R} \mathbf{C}_{(i)}^{-1}$, where $\mathbf{G}_{(i)} = \boldsymbol{\Omega}_{(i)} + \mathbf{R} \mathbf{C}_{(i)}^{-1} \mathbf{R}^\top$. Thus, (19) with MIL leads to

$$\mathbf{u}_{(i+1,j+1)} = \mathbf{z}_{(i+1,j)} - \mathbf{C}_{(i)}^{-1} \mathbf{R}^\top \mathbf{v}_{(i+1,j)}, \tag{20}$$

$$\text{solve } \{\mathbf{G}_{(i)} \mathbf{v}_{(i+1,j)} = \mathbf{R} \mathbf{z}_{(i+1,j)}\} \text{ for } \mathbf{v}_{(i+1,j)}, \tag{21}$$

where $\mathbf{z}_{(i+1,j)} \triangleq \mathbf{C}_{(i)}^{-1} \mathbf{A}^\top \mathbf{y} + (\mathbf{I}_N - \mathbf{C}_{(i)}^{-1} \mathbf{A}^\top \mathbf{A}) \mathbf{u}_{(i+1,j)}$. We use the following iterative solver[18] for (21) in the inner loop of the algorithm:

$$\mathbf{v}_{(i+1,j,k+1)} = \boldsymbol{\Gamma}_{(i)}^{-1} (\mathbf{R} \mathbf{z}_{(i+1,j)} + \mathbf{Q} \mathbf{v}_{(i+1,j,k)}), \tag{22}$$

where $\mathbf{\Gamma}_{(i)} \triangleq \mathbf{\Omega}_{(i)} + \lambda \mathbf{I}_{PM}$ is a diagonal matrix, $\mathbf{Q} \triangleq \lambda\, \mathbf{I}_{PM} - \mathbf{R}\mathbf{C}_{(i)}^{-1}\mathbf{R}^{\top}$, and λ is the maximum eigenvalue of $\mathbf{R}\mathbf{C}_{(i)}^{-1}\mathbf{R}^{\top}$. The update-rule (22) guarantees convergence[15, 18] of $\mathbf{v}_{(i+1,j,k)}$ to a solution of (21) and is implemented in favor of a CG-based solver since (22) is linear in both $\mathbf{z}_{(\cdot)}$ and $\mathbf{v}_{(\cdot)}$ and also decouples the shift-variant component $\mathbf{\Omega}_{(i)}$ from the rest of the terms in $\mathbf{G}_{(i)}$: these features allow us to more easily evaluate the desired Jacobian matrix in (7) and (13) at each iteration as elucidated in the sequel. An important observation is that IRLS-MIL (20)-(22) depends on $\mathbf{\Gamma}_{(i)}$ (that in turn relies on $\mathbf{\Omega}_{(i)}$ rather than $\mathbf{\Omega}_{(i)}^{-1}$) that is well-defined and thus does not require corner rounding for handling (15) and (17).

4.3 Implementation of IRLS-MIL

The IRLS-MIL algorithm (20)-(21) generally exhibits faster convergence[13] over standard IRLS (14). The convergence speed of IRLS-MIL (20)-(22) depends primarily on the "proximity" of $\mathbf{C}_{(i)}$ to $\mathbf{A}^{\top}\mathbf{A}$ while ensuring $\mathbf{C}_{(i)} - \mathbf{A}^{\top}\mathbf{A} \succ \mathbf{0}$. For image restoration with circulant $\mathbf{A}$ in (1), $\mathbf{A}^{\top}\mathbf{A} \in \mathbb{R}^{N \times N}$ is also circulant. So we used $\mathbf{C}_{(i)} = \mathbf{C}_{\nu} \triangleq \mathbf{A}^{\top}\mathbf{A} + \nu \mathbf{I}_N \ \forall\ i$ and implemented $\mathbf{C}_{\nu}^{-1}$ using FFTs. The parameter $\nu > 0$ was chosen to achieve a prescribed condition number of $\mathbf{C}_{\nu}$, $\kappa(\mathbf{C}_{\nu})$, that can be easily computed as a function of ν. In general, setting $\kappa(\mathbf{C}_{\nu})$ to a large value can lead to numerical instabilities in $\mathbf{C}_{\nu}^{-1}$ and IRLS-MIL, while a small $\kappa(\mathbf{C}_{\nu})$ reduces convergence speed of IRLS-MIL.[13] In our experiments, we found that ν leading to $\kappa(\mathbf{C}_{\nu}) \in [20, 100]$ yielded good acceleration over standard IRLS for a fixed number of outer (i.e., index by i) iterations, so we set ν such that $\kappa(\mathbf{C}_{\nu}) = 100$.

4.4 Jacobian Matrix Derivation for IRLS-MIL

We propose to evaluate $\mathbf{J}_{\mathbf{f}_{\beta}}(\mathbf{y})$, or equivalently, $\mathbf{J}_{\mathbf{u}_{(\cdot)}}(\mathbf{y})$ (since $\mathbf{u}_{(\cdot)}$ is the reconstructed output at any stage of the IRLS-MIL algorithm in (20)-(22)) analytically as follows. Using linearity of (8), we have from (20) that

$$\mathbf{J}_{\mathbf{u}_{(i+1,j+1)}}(\mathbf{y}) = \mathbf{J}_{\mathbf{z}_{(i+1,j)}}(\mathbf{y}) - \mathbf{C}_{(i)}^{-1}\mathbf{R}^{\top}\mathbf{J}_{\mathbf{v}_{(i+1,j,K)}}(\mathbf{y}), \tag{23}$$

where $\mathbf{J}_{\mathbf{z}_{(i+1,j)}}(\mathbf{y}) = \mathbf{C}_{(i)}^{-1}\mathbf{A}^{\top} + (\mathbf{I}_N - \mathbf{C}_{(i)}^{-1}\mathbf{A}^{\top}\mathbf{A})\mathbf{J}_{\mathbf{u}_{(i+1,j)}}(\mathbf{y})$. The term $\mathbf{J}_{\mathbf{v}_{(i+1,j,K)}}(\mathbf{y})$ in (23) corresponds to $\mathbf{v}_{(i+1,j,K)}$ that we obtain after performing K iterations of (22) and that we substitute in place of $\mathbf{v}_{(i+1,j)}$ in (20). We use product rule for Jacobian matrices[20] to obtain $\mathbf{J}_{\mathbf{v}_{(i+1,j,K)}}(\mathbf{y})$ from (22) as follows:

$$\mathbf{J}_{\mathbf{v}_{(i+1,j,k+1)}}(\mathbf{y}) = \mathbf{\Gamma}_{(i)}^{-1}(\mathbf{R}\mathbf{J}_{\mathbf{z}_{(i+1,j)}}(\mathbf{y}) + \mathbf{Q}\mathbf{J}_{\mathbf{v}_{(i+1,j,k)}}(\mathbf{y})) - \mathbf{\Gamma}_{(i)}^{-2}\mathbf{D}_{\mathbf{v}_{(i+1,j,k)}}\mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{y}), \tag{24}$$

where $\mathbf{D}_{\mathbf{v}_{(i+1,j,k)}} \triangleq \mathrm{diag}\{\mathbf{R}\mathbf{z}_{(i+1,j)} + \mathbf{Q}\mathbf{v}_{(i+1,j,k)}\}$. Since $\boldsymbol{\omega}_{(i)}$ is a function of $\mathbf{R}\mathbf{u}_{(i)}$, we use chain rule[20] on $\mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{y})$ to get that

$$\mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{y}) = \mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{u}_{(i)})\ \mathbf{J}_{\mathbf{u}_{(i)}}(\mathbf{y}). \tag{25}$$

We analytically evaluate $\mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{u}_{(i)})$ for $\boldsymbol{\omega}_{(i)}$ in (15)-(17), respectively, and obtain

$$\begin{aligned}
(\ell_1\text{-regularization})\ \mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{u}_{(i)}) &= \beta^{-1}\mathrm{diag}\{\boldsymbol{\tau}_{(i)}\}\,\mathbf{R}, && (26)\\
(\text{Fair potential})\ \ \mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{u}_{(i)}) &= (\beta\,\delta)^{-1}\mathrm{diag}\{\boldsymbol{\tau}_{(i)}\}\,\mathbf{R}, && (27)\\
(\text{TV})\ \mathbf{J}_{\boldsymbol{\omega}_{(i)}}(\mathbf{u}_{(i)}) &= \mathbf{1}_P \otimes \left(\beta^{-2}\sum_{p=1}^{P}\mathrm{diag}\{\boldsymbol{\varrho}_{(i)p}\}\mathbf{R}_p\right), && (28)
\end{aligned}$$

```
1. Initialization: $\mathbf{u}_{(0,0)} \triangleq \mathbf{A}^\top\mathbf{y}$, $\mathbf{J}_{\mathbf{u}_{(0,0)}}(\mathbf{y})\mathfrak{n} \triangleq \mathbf{A}^\top\mathfrak{n}$, $i = 0$
2. Repeat Steps 3-17 until Stop Criterion is met
3. If $i = 0$
4. $\mathbf{u}_{(i+1,0)} = \mathbf{u}_{(i,0)}$, $\mathbf{v}_{(i+1,0,0)} = \mathbf{R}\mathbf{u}_{(i,0)}$, $\mathbf{J}_{\mathbf{u}_{(i+1,0)}}(\mathbf{y})\mathfrak{n} \triangleq \mathbf{J}_{\mathbf{u}_{(i,0)}}(\mathbf{y})\mathfrak{n}$, $\mathbf{J}_{\mathbf{v}_{(i+1,0,0)}}(\mathbf{y})\mathfrak{n} \triangleq \mathbf{R}\mathbf{J}_{\mathbf{u}_{(i,0)}}(\mathbf{y})\mathfrak{n}$
5. Else
6. $\mathbf{u}_{(i+1,0)} = \mathbf{u}_{(i,J)}$, $\mathbf{v}_{(i+1,0,0)} = \mathbf{v}_{(i,J-1,K)}$, $\mathbf{J}_{\mathbf{u}_{(i+1,0)}}(\mathbf{y})\mathfrak{n} \triangleq \mathbf{J}_{\mathbf{u}_{(i,J)}}(\mathbf{y})\mathfrak{n}$, $\mathbf{J}_{\mathbf{v}_{(i+1,0,0)}}(\mathbf{y})\mathfrak{n} \triangleq \mathbf{J}_{\mathbf{v}_{(i,J-1,K)}}(\mathbf{y})\mathfrak{n}$
7. Compute $\mathbf{\Gamma}_{(i)}$; set $j = 0$
8.      Run $J$ iterations of Steps 9-14
9.      Compute $\mathbf{z}_{(i+1,j)}$ and $\mathbf{J}_{\mathbf{z}_{(i+1,j)}}(\mathbf{y})\mathfrak{n}$
10.       If $j > 0$ set $\mathbf{v}_{(i+1,j,0)} = \mathbf{v}_{(i+1,j-1,K)}$ and $\mathbf{J}_{\mathbf{v}_{(i+1,j,0)}}(\mathbf{y})\mathfrak{n} \triangleq \mathbf{J}_{\mathbf{v}_{(i+1,j-1,K)}}(\mathbf{y})\mathfrak{n}$
12.       Run $K$ iterations of (22) and (24) to get $\mathbf{v}_{(i+1,j,K)}$ and $\mathbf{J}_{\mathbf{v}_{(i+1,j,K)}}(\mathbf{y})\mathfrak{n}$
13.       Compute $\mathbf{u}_{(i+1,j+1)}$ (20) and $\mathbf{J}_{\mathbf{u}_{(i+1,j+1)}}(\mathbf{y})\mathfrak{n}$ (23)
14.       Set $j = j+1$ and return to Step 9
16. Compute NGCV($\beta$) and / or WSURE($\beta$) at iteration $i$ using (29)-(30), (7) and (13), respectively
17. Set $i = i + 1$ and return to Step 3
```

Figure 1: Iterative computation of WSURE(β) and NGCV(β) for image restoration using IRLS-MIL algorithm (with J iterations of (20)-(21) and K iterations of (22)). We use a pregenerated binary random vector $\mathfrak{n} = \mathfrak{n}_{\pm 1}$ for Monte-Carlo computation (29)-(30) of the required traces in (7) and (13), respectively. Vectors of the form $\mathbf{J}_{\cdot}(\cdot)\mathfrak{n}$ are stored and manipulated in place of actual matrices $\mathbf{J}_{\cdot}(\cdot)$.

where the n-th component of $\boldsymbol{\tau}_{(i)} \in \mathbb{R}^{PM}$ is $\tau_{(i)n} = \text{sign}([\mathbf{R}\mathbf{u}_{(i)}]_n)$, and the m-th component of $\boldsymbol{\varrho}_{(i)p} \in \mathbb{R}^M$, $p = 1 \ldots P$, is $\varrho_{(i)pm} = [\mathbf{R}_p\mathbf{u}^{(i)}]_m \, \breve{\omega}^{-1}_{(i)m}$. In (25)-(28), $\mathbf{u}_{(i)}$ is to be interpreted as $\mathbf{u}_{(i)} = \mathbf{u}_{(i+1,0)}$, the initialization for the j-iterations (19) at $(i+1)$-th outer iteration.

To summarize our method, we run (20), (22) for restoration, and additionally execute the updates in (23)-(24) using (25)-(28) to iteratively evaluate $\mathbf{J}_{\mathbf{u}_{(\cdot)}}(\mathbf{y})$ at any stage of IRLS-MIL.

4.5 Monte-Carlo Trace Estimation

The Jacobian matrices $\mathbf{J}_{\cdot}(\cdot)$ have enormous sizes for typical restoration settings and cannot be stored and manipulated directly to compute the desired traces, $\text{tr}\{\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$ in (7) and $\text{tr}\{\mathbf{W}\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$ in (13), respectively. So we use a Monte-Carlo method[17] to estimate $\text{tr}\{\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$ and $\text{tr}\{\mathbf{W}\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$ that is based on the following straightforward identity: for any deterministic matrix $\mathbf{T} \in \mathbb{R}^{N\times N}$,

$$\mathcal{E}_{\mathfrak{n}}\{\mathfrak{n}^\top \, \mathbf{T} \, \mathfrak{n}\} = \text{tr}\{\mathbf{T}\},$$

where $\mathfrak{n} \in \mathbb{R}^M$ is an i.i.d. zero-mean random vector with unit variance. To use this type of stochastic estimation for $\text{tr}\{\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$ and $\text{tr}\{\mathbf{W}\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\}$, we adopt the procedure proposed by Vonesch *et al.*[7] where we take products with $\mathfrak{n}$ in (23)-(25) and store and update vectors of the form $\mathbf{J}_{\mathbf{u}_{(\cdot)}}(\cdot)\mathfrak{n}$ and $\mathbf{J}_{\mathbf{v}_{(\cdot)}}(\cdot)\mathfrak{n}$ in IRLS-MIL. At any point during the course of IRLS-MIL, the desired traces in (7) and (13) are stochastically approximated as

$$\text{tr}\{\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\} \;\approx\; \hat{t}_{\mathbf{A}\mathbf{J}_\mathbf{u}} \;\triangleq\; \mathfrak{n}^\top \mathbf{A}\,\mathbf{J}_{\mathbf{u}_{(\cdot)}}(\mathbf{y})\mathfrak{n}, \tag{29}$$

$$\text{tr}\{\mathbf{W}\mathbf{A}\mathbf{J}_{\mathbf{f}_\beta}(\mathbf{y})\} \;\approx\; \hat{t}_{\mathbf{W}\mathbf{A}\mathbf{J}_\mathbf{u}} \;\triangleq\; \mathfrak{n}^\top \mathbf{W}\,\mathbf{A}\,\mathbf{J}_{\mathbf{u}_{(\cdot)}}(\mathbf{y})\mathfrak{n}, \tag{30}$$

respectively. To improve accuracy of (29)-(30), $\mathfrak{n}$ can be designed to decrease the variance of $\hat{t}_{\mathbf{A}\mathbf{J}_\mathbf{u}}$ and $\hat{t}_{\mathbf{W}\mathbf{A}\mathbf{J}_\mathbf{u}}$: it has been shown[17] that the variance of a Monte-Carlo trace estimate (such as $\hat{t}_{\mathbf{A}\mathbf{J}_\mathbf{u}}$ or $\hat{t}_{\mathbf{W}\mathbf{A}\mathbf{J}_\mathbf{u}}$) is lower for a binary random vector $\mathfrak{n}_{\pm 1}$ whose elements are either $+1$ or -1 with

Table 1: Setup for image restoration (IR) experiments

Experiment	Test image (256×256)	Blur	Regularization
IR-A	Cameraman	Uniform 9×9	$\mathcal{R}_{\mathrm{TV}}(5)$
IR-B	Peppers	$(1 + x_1 + x_2)^{-1}$, $-7 \leq x_1, x_2 \leq 7$	$\mathcal{R}_{\ell_1}$ (3)
IR-C	House	Gaussian with standard deviation 2	$\mathcal{R}_{\mathrm{FP}}$ (4)

Table 2: ISNR (in dB) of Restored Images for Experiments IR-A, IR-B, and IR-C and varying BSNR

Experiment	BSNR	σ^2	MSE (oracle)	Projected-SURE	Predicted-SURE	NGCV	GCV
IR-A	20	3.08×10^1	3.85	3.73	3.84	3.84	2.45
	30	3.08	5.85	5.84	5.85	5.85	2.40
	40	3.08×10^{-1}	8.50	8.50	8.49	8.49	2.41
	50	3.08×10^{-2}	11.02	10.97	11.00	11.01	2.38
IR-B	20	1.99×10^1	4.44	4.28	4.34	4.34	-0.26
	30	1.99	8.44	8.43	8.44	8.44	-0.51
	40	1.99×10^{-1}	12.41	12.41	12.41	12.28	-0.55
	50	1.99×10^{-2}	15.54	15.53	15.44	15.53	-0.55
IR-C	20	1.76×10^1	3.98	3.57	3.62	3.61	-0.59
	30	1.76	4.71	4.70	4.55	4.58	-1.01
	40	1.76×10^{-1}	6.19	6.14	6.12	6.12	-1.06
	50	1.76×10^{-2}	6.79	6.64	6.63	6.63	-1.07

probability 0.5 than for a Gaussian random vector $\boldsymbol{\mathfrak{n}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_M)$. So in our experiments, we used one realization of $\boldsymbol{\mathfrak{n}}_{\pm 1}$ in (29)-(30). Fig. 1 presents an outline for implementing IRLS-MIL with recursions for $\mathbf{J}_{.}(\cdot)\boldsymbol{\mathfrak{n}}_{\pm 1}$ to compute and monitor NGCV(β) and WSURE(β) as IRLS-MIL evolves.

5. EXPERIMENTAL RESULTS

We performed simulations with standard test images and blur kernels[8] outlined in Table 1. In all experiments, data $\mathbf{y}$ was simulated by synthetically adding Gaussian noise whose variance was chosen to meet a prescribed blurred signal to noise ratio, BSNR $\triangleq 10 \log_{10}(\mathrm{Var}(\mathbf{Ax})/\sigma^2)$. We ran 100 iterations of IRLS-MIL (with $J = 1$ and $K = 1$ inner-iterations, respectively, see Figure 1) with the following regularizers: ℓ_1-wavelets (3) with two levels of undecimated Haar excluding the approximation level for $\mathbf{R}$, a smooth convex regularizer (4) with Fair Potential ($\delta = 10^{-4}$) and finite differences for $\mathbf{R}$ and TV (5). We measured quality of restored results using improvement in signal to noise ratio, $\mathrm{ISNR}(\beta) \triangleq 10 \log_{10}(\|\mathbf{y} - \mathbf{x}\|_2^2 / \|\mathbf{x} - \mathbf{f}_\beta(\mathbf{y})\|_2^2)$.

In all experiments, we selected β so as to minimize Projected-SURE(β), Predicted-SURE(β) and NGCV(β) using golden-mean search method. We assumed that σ was available (in practice, σ can be reliably estimated) for computing Projected-SURE(β), Predicted-SURE(β). We implemented $\mathbf{W} = (\mathbf{AA}^{\mathrm{H}})^\dagger$ in Projected-SURE using FFTs, where we set the eigenvalues of $\mathbf{AA}^{\mathrm{H}}$ below a threshold of 10^{-3} to zero for numerical stability of $(\mathbf{AA}^{\mathrm{H}})^\dagger$. We also include results for β selected by minimizing GCV(β) that applies only to linear algorithms, but has been suggested for use with nonlinear algorithms as well (with $\mathbf{F}_\beta$ replaced by $\mathbf{f}_\beta(\mathbf{y})$ in (6)) by Giryes et al.[8] We compare our results with those obtained by minimizing the true "unknown" (oracle) MSE(β).

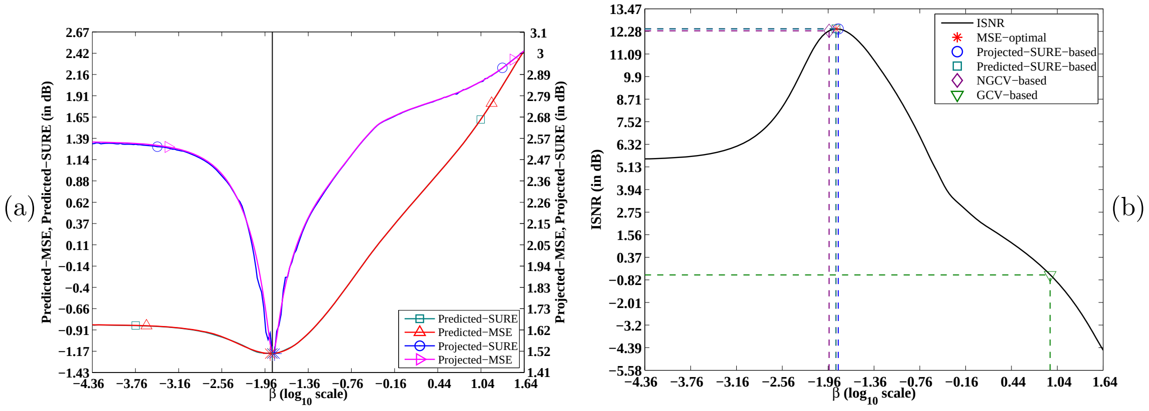

Figure 2: Experiment corresponding to third row in IR-B in Table 2: (a) Plot of Projected-MSE, Projected-SURE(β), Predicted-MSE and Predicted-SURE(β), respectively, as functions of β. The estimates closely capture the trends of the respective MSE-curves and their minima (indicated by *) are close to that of the (oracle) MSE indicated by the solid vertical line. (b) Plot of ISNR(β) as a function of β along with indication of ISNR values corresponding to MSE-, Predicted-SURE-, Projected-SURE-, NGCV-, and GCV-selections. ISNR values obtained using Projected-SURE, Predicted-SURE, and NGCV are agreeably close to the oracle (MSE) while GCV selection is noticeably worse.

In Table 2, we present ISNR of restoration results obtained by optimizing MSE(β), GCV(β), NGCV(β), Predicted-SURE(β), and Projected-SURE(β) for various BSNRs in each experiment. NGCV, Predicted-SURE, and Projected-SURE lead to ISNRs reasonably close to oracle-ISNR corresponding to minimum-MSE in all experiments. In contrast, GCV (that is appropriate for linear algorithms) yielded significantly lower ISNR values; this is likely because of the strong nonlinearity of our algorithm for the considered nonquadratic regularizers.

We plot Projected-MSE(β), Predicted-MSE(β), Projected-SURE(β) and Predicted-SURE(β) in Figure 2a and ISNR(β) in Figure 2b, respectively, as functions of β for an instance of Experiment IR-B. Predicted-SURE is almost an exact replica of Predicted-MSE, while Projected-SURE follows the trend of Projected-MSE very closely (Figure 2a), deviating only slightly for some β-values away from the minimum. NGCV-, Predicted-SURE-, and Projected-SURE-based selections lead to ISNRs close to that of the MSE-based selection (Figure 2b), while the GCV-based one produces a much lower ISNR value.

Figure 3 shows a visual comparison of images restored with βs that minimized MSE(β), NGCV(β), Projected-SURE(β), Predicted-SURE(β), and GCV(β) for the same instance of Experiment IR-B. Figures 3d, 3e, and 3f corresponding to Projected-SURE, Predicted-SURE, and NGCV, respectively, are visually similar to the minimum-MSE result Figure 3(c). GCV-based restoration Figure 3(g) is noticeably over-smoothed for reasons explained earlier. We obtained similar results in various other experiments[14] (results not shown due to space limitations) indicating the consistency of our approach.

6. CONCLUSION

Selection of the regularization parameter β is an important step in regularized reconstruction methods. When data is corrupted by Gaussian noise, NGCV(β) (the nonlinear version of GCV)[3,4] and

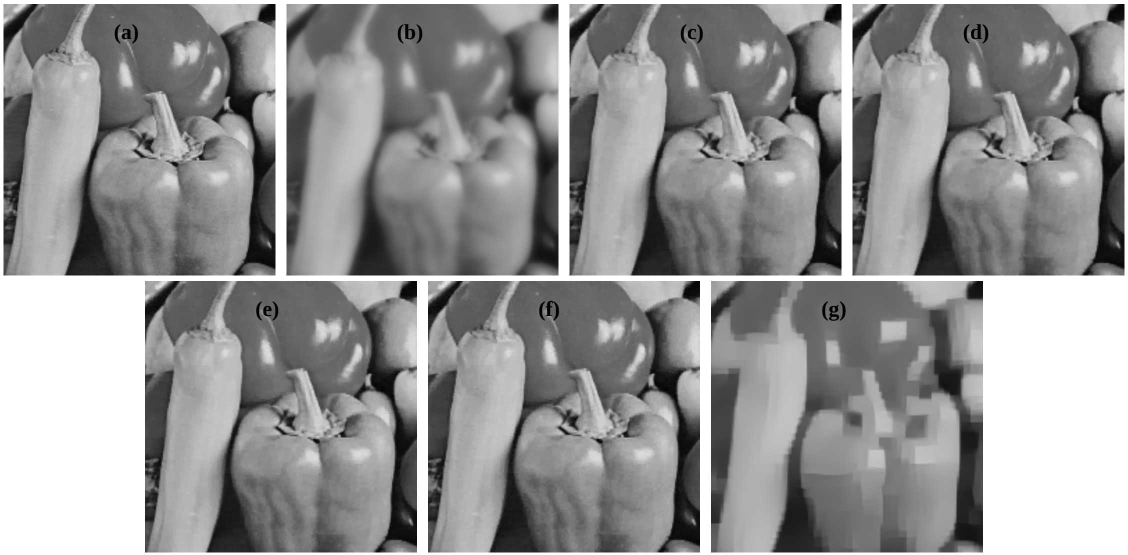

Figure 3: Experiment corresponding to third row in IR-B in Table 2: Zoomed images of (a) Noise-free Peppers; (b) Blurred and noisy data; and TV-restored images with regularization parameter β selected to minimize (c) (oracle) MSE(β) (12.41 dB); (d) Projected-SURE(β) (12.41 dB); (e) Predicted-SURE(β) (12.41 dB); (f) NGCV(β) (12.28 dB); (g) GCV(β) (-0.55 dB). Projected-SURE-, Predicted-SURE- and NGCV-based results, (d)-(f), respectively, visually resemble the oracle MSE-based result (c) very closely, while the GCV-based result is considerably over-smoothed.

weighted MSE-estimates,[6] i.e., Predicted-SURE(β), Projected-SURE(β) can be used to tune β, but their computation necessitates the evaluation[6] of the trace of a linear transform of the Jacobian matrix $\mathbf{J}_{\mathbf{f}_\beta}$. In this paper, we introduced a method to recursively evaluate $\mathbf{J}_{\mathbf{f}_\beta}$ for the IRLS-MIL algorithm that can handle a variety of regularizers including TV, ℓ_1-regularization and smooth edge-preserving criteria. We estimated the desired trace using a Monte-Carlo scheme.[17] We demonstrated through simulations that β selected by minimizing NGCV, Predicted-SURE and Projected-SURE provide near-MSE-optimal results for restorations using TV, ℓ_1-wavelets and smooth edge-preserving regularization with the Fair potential. Our results suggest that NGCV, Predicted-SURE and Projected-SURE can be useful for regularization parameter selection in non-linear restoration problems involving Gaussian noise. The proposed method can also be extended to handle synthesis formulations, various other regularizers and restoration algorithms.

ACKNOWLEDGMENTS

This work was supported in part by NIH grants R01 HL 098686 and P01 CA 87634.

REFERENCES

[1] Karl, W. C., “Regularization in image restoration and reconstruction,” in [*Handbook of Image & Video Processing*], Bovik, A., ed., 183–202, ELSEVIER, 2nd ed. (2005).

[2] Craven, P. and Wahba, G., “Smoothing noisy data with spline functions,” *Numer. Math.* **31**, 377–403 (1979).

[3] Deshpande, L. N. and Girard, D. A., "Fast computation of cross-validated robust splines and other non-linear smoothing splines," in [*Curves and Surfaces*], Laurent, L. M. and Schumaker, eds., 143–8, Academic, Boston (1991).

[4] Girard, D. A., "The fast Monte-Carlo Cross-Validation and C_L procedures: Comments, new results and application to image recovery problems - Rejoinder," *Computation. Stat.* **10**, 251–258 (1995).

[5] Stein, C., "Estimation of the mean of a multivariate normal distribution," *Ann. Stat.* **9**, 1135–51 (Nov. 1981).

[6] Eldar, Y. C., "Generalized SURE for exponential families: applications to regularization," *IEEE Trans. Sig. Proc.* **57**, 471–81 (Feb. 2009).

[7] Vonesch, C., Ramani, S., and Unser, M., "Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint," *Proc. IEEE Intl. Conf. Img. Proc. (ICIP'08)* , 665–8 (2008).

[8] Giryes, R., Elad, M., and Eldar, Y. C., "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis* **30**, 407–22 (May 2011).

[9] Blu, T. and Luisier, F., "The SURE-LET approach to image denoising," *IEEE Trans. Im. Proc.* **16**, 2778–86 (Nov. 2007).

[10] Ramani, S., Blu, T., and Unser, M., "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. Im. Proc.* **17**, 1540–54 (Sept. 2008).

[11] Ville, D. V. D. and Kocher, M., "Nonlocal means with dimensionality reduction and SURE-based parameter selection," *IEEE Trans. Im. Proc.* **20**(9), 2683–90 (2011).

[12] Thompson, A. M., Brown, J. C., Kay, J. W., and Titterington, D. M., "A study of methods of choosing the smoothing parameter in image restoration by regularization," *IEEE Trans. Patt. Anal. Mach. Intell.* **13**(4), 326–339 (1991).

[13] Ramani, S. and Fessler, J. A., "An accelerated iterative reweighted least squares algorithm for compressed sensing MRI," in [*Proc. IEEE Intl. Symp. Biomed. Imag.*], 257–60 (2010).

[14] Ramani, S., Rosen, J., Liu, Z., and Fessler, J. A., "Regularization parameter selection for nonlinear iterative image restoration and mri reconstruction using GCV and SURE-based methods," *IEEE Trans. Im. Proc.* (submitted).

[15] Selesnick, I. W. and Figueiredo, M. A. T., "Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors," in [*SPIE-7446*], 74460D (2009). Wavelets XIII.

[16] Fair, R. C., "On the robust estimation of econometric models," *Ann. Econ. Social Measurement* **2**, 667–77 (Oct. 1974).

[17] Hutchinson, M. F., "A stochastic estimator for the trace of the influence matrix for Laplacian smoothing splines," *Comm. in Statistics - Simulation and Computation* **19**(2), 433–50 (1990).

[18] Hackbusch, W., [*Iterative Solution of Large Sparse Systems of Equations*], Springer-Verlag, New York (1994).

[19] Golub, G. H. and Loan, C. F. V., [*Matrix Computations*], The Johns Hopkins University Press, Baltimore, MD, USA, third ed. (1996, p. 50).

[20] Hjorungnes, A. and Gesbert, D., "Complex-valued matrix differentiation: techniques and key results," *IEEE Trans. Sig. Proc.* **55**(6), 2740–6 (2007).