Appl. Comput. Harmon. Anal. 30 (2011) 407–422

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Letter to the Editor

The projected GSURE for automatic parameter tuning in iterative shrinkage methods

R. Giryes[a,*], M. Elad[a], Y.C. Eldar[b]

[a] The Department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel
[b] The Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa 32000, Israel

ARTICLE INFO

Article history:
15 November 2009
Revised 10 October 2010
Accepted 23 November 2010
Available online 25 November 2010
Communicated by Martin Vetterli

Keywords:
Iterated shrinkage
Stein unbiased risk estimator
Separable surrogate function
Inverse problems

ABSTRACT

Linear inverse problems are very common in signal and image processing. Many algorithms that aim at solving such problems include unknown parameters that need tuning. In this work we focus on optimally selecting such parameters in iterative shrinkage methods for image deblurring and image zooming. Our work uses the projected Generalized Stein Unbiased Risk Estimator (GSURE) for determining the threshold value λ and the iterations number K in these algorithms. The proposed parameter selection is shown to handle any degradation operator, including ill-posed and even rectangular ones. This is achieved by using GSURE on the projected expected error. We further propose an efficient greedy parameter setting scheme, that tunes the parameter while iterating without impairing the resulting deblurring performance. Finally, we provide extensive comparisons to conventional methods for parameter selection, showing the superiority of the use of the projected GSURE.

1. Introduction

In many applications in signal and image processing there is a need for solving a linear inverse problem [1,2]. Here we consider the scenario in which an original image $\mathbf{x}$ is degraded by a linear operator $\mathbf{H}$ followed by an additive white Gaussian noise $\mathbf{w}$. Our observation can be written as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{1}$$

where the goal is to reconstruct the original image $\mathbf{x}$ from $\mathbf{y}$. In our setting, there is no Bayesian prior on $\mathbf{x}$; it is therefore treated as a deterministic unknown.

There are many techniques in the literature for this task that focus on different objectives [1–11]. Most of these methods include parameters that require tuning. Some of the algorithms are iterative, and thus the number of iterations needs to be set as well. Tuning of such parameters is generally not a trivial task. Very often, the objective in these problems is recovery of the signal with a minimal Mean-Squared Error (MSE) between the original image and the estimated one [12]. In this case, ideally, the parameters should be chosen such that this MSE is minimized. However, since the original image $\mathbf{x}$ is deterministic, the MSE will depend in general on the unknown $\mathbf{x}$. Consequently, we cannot know what choice of parameters minimizes the true MSE. In practice, therefore, a common approach to parameter tuning is still manual, by looking at the reconstructed result and judging its quality subjectively.

[*] Corresponding author.
E-mail addresses: raja@cs.technion.ac.il (R. Giryes), elad@cs.technion.ac.il (M. Elad), yonina@ee.technion.ac.il (Y.C. Eldar).

1063-5203/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.acha.2010.11.005

The literature offers several automatic ways for choosing the parameters of the reconstruction algorithm [1]. One such method is the Unbiased Predictive Risk Estimator (UPRE) [13], which is valid only for the case that the reconstruction operator is linear. Other popular tools are the Generalized Cross Validation (GCV) technique with its many variants [14], and the L-Curve method [15], both available for general reconstruction algorithms. The number of iterations can be chosen using the discrepancy principle [16], such that the norm of the error between the degraded image and reconstructed image is close to the noise variance.

An alternative technique for parameter tuning is the Stein Unbiased Risk Estimator (SURE) [17,18]. SURE provides an unbiased estimate of the MSE for a candidate solver, including non-linear ones. Minimizing this estimated MSE, leads to the automatic tuning desired. The original SURE by Stein is limited to the case of denoising of white additive Gaussian noise. One of the best known algorithms that uses SURE is Donoho's SureShrink denoising algorithm [19]. There are also other contributions along these lines, as in [20–22].

Recently, a generalization of SURE (GSURE) [23] has been developed for more general models that can be described in the form of exponential family distribution. This provides a new opportunity for choosing the parameters automatically in a much more diverse set of inverse problems. In the case where $\mathbf{H}$ is rank-deficient, or even rectangular (with more columns than rows), the inverse problem becomes ill-posed. In such a scenario, GSURE can at best estimate the MSE in the range of $\mathbf{H}^T$. We refer hereafter to a tuning of the algorithm's parameters with this estimation as the projected GSURE.

The reconstruction algorithms used in our work belong to the *iterated shrinkage* family of methods [7,8,24,25], In these algorithms, the image is known to have a sparse representation over a dictionary. Reconstruction is obtained by an iterative process composed of applying the dictionary and the degradation operator with their conjugates, in addition to a point-wise shrinkage operation. In these algorithms the threshold value λ in the shrinkage parameter needs to be tuned, along with the number of iterations K to be performed.

The work reported in [26] develops a special case of GSURE for the case where the blur operator $\mathbf{H}$ is assumed to be full-rank. This work also addresses automatic tuning of the parameters λ and K in an iterated shrinkage algorithm, used for the deblurring problem. For a pre-chosen number of iterations, the GSURE is used to select a fixed value of λ that optimizes the overall estimated MSE. For a given λ, the GSURE is again used to determine the number of iterations. We refer to this method of setting λ and K as the *global method*.

Similar to the work in [26], this paper proposes to tune λ and K for iterated shrinkage algorithms, which are geared towards solving inverse problems in image processing. Our work extends the applicability of GSURE to ill-posed inverse problems, using the projected GSURE. We show that turning from the plain GSURE to its projected version, the performance of the parameter tuning improves substantially.

In addition, we propose an alternative to the global method mentioned above, where the value of λ is chosen by minimizing the estimated MSE at each iteration, thereby obtaining a different value of λ per iteration. We refer to this technique as the *greedy method*. The main benefit of such a strategy is the natural way with which the number of iterations is set simultaneously – the iterations can be stopped when the estimated MSE improvement is under a certain threshold. In order to further improve the performance with regard to the overall estimated MSE in this greedy approach, λ in each iteration can be chosen with a look-ahead on the estimated MSE in the next few iterations.

This paper is organized as follows: In Section 2 we present the deblurring and image scaling (zooming) problems, and the iterative shrinkage algorithms used for solving them. Section 3 describes several conventional techniques for parameter setting (GCV, L-curve, and the discrepancy method). Section 4 presents the SURE with its generalization, as developed in [23]. In Section 5 we use GSURE for the task of parameter selection in the iterative shrinkage algorithms using the two approaches – global and greedy. The results of these methods are presented in Section 6. We conclude in Section 7 with a summary of this work, and a discussion on future research directions.

2. Iterative shrinkage methods

To define a linear inverse problems, we need to specify the model of the original image, the noise characteristics, and the degradation operator. We consider two types of operators $\mathbf{H}$. The first is a blur operator, and the second is a scale-down operator that defines a scale-up problem. In the blur case, $\mathbf{H}$ is a square matrix that blurs each pixel by replacing it with a weighted average of its neighbors. In the scale-down scenario, $\mathbf{H}$ is composed of two operators applied sequentially, the first blurs the image and the second down-samples it. In both settings we assume that the operators are known, and the additive noise $\mathbf{w}$ is white Gaussian with known variance σ^2. Our goal in both settings is to reconstruct the original image $\mathbf{x}$ without assuming any Bayesian prior.

In order to evaluate the quality of the reconstruction, we measure the MSE between the reconstructed image and the original one,

$$\mathrm{MSE}(\hat{\mathbf{x}}) = E\left[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\right]. \tag{2}$$

This criterion is very commonly used in image processing despite its known shortcomings [27]. Two related measures that are also common are the Peak Signal to Noise Ratio (PSNR), and the Improvement in Signal to Noise Ratio (ISNR),[1] which are both inversely proportional to the MSE. Thus, all of the criteria favor low MSE.

Since the goodness of our fit is measured by the MSE, a natural approach would be to choose an estimate $\hat{\mathbf{x}}$ that directly minimizes the MSE. Unfortunately, however, the MSE depends in general on $\mathbf{x}$ and therefore cannot be minimized directly. This difficulty is explained in more detail in [12]. To avoid direct MSE minimization, in the context of imaging it is often assumed that $\mathbf{x}$ has a sparse representation $\boldsymbol{\alpha}$ under a given dictionary $\mathbf{D}$ [28]. This knowledge of $\mathbf{x}$ is then used together with a standard least-squares criterion to find an estimate $\hat{\mathbf{x}}$. More specifically, a popular approach to recover $\mathbf{x}$ is to first seek the value of $\boldsymbol{\alpha}$ that minimizes an objective of the form:

$$f(\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{y} - \mathbf{HD}\boldsymbol{\alpha}\|_2^2 + \lambda\rho(\boldsymbol{\alpha}). \tag{3}$$

The first term requires that the reconstructed image, after degradation, should be close to the measured image $\mathbf{y}$. The second term forces sparsity using the penalty function ρ. The minimizer of this function is the representation of the desired image, and its multiplication by $\mathbf{D}$ gives the reconstructed image. Although intuitive, this strategy does not in general minimize the MSE.

The minimization of $f(\boldsymbol{\alpha})$ in (3) is typically performed using an iterative procedure. An appealing choice is the iterative shrinkage methods [7,8]. These algorithms are iterative, applying the matrices $\mathbf{H}$ and $\mathbf{D}$ and their adjoints, and an additional element-wise thresholding operation in each iteration. Consequently, these methods are well-suited for high-dimensional problems with sparse unknowns. This family of techniques includes the Separable Surrogate Functionals (SSF) method [7], the Parallel Coordinate Descent (PCD) method [29,24], and some others. In this work we focus on the SSF and PCD techniques. The SSF algorithm proposes the iteration formula

$$\boldsymbol{\alpha}_{k+1} = \mathcal{S}_{\rho,\lambda/c}\left(\frac{1}{c}(\mathbf{HD})^T(\mathbf{y} - \mathbf{HD}\boldsymbol{\alpha}_k) + \boldsymbol{\alpha}_k\right), \tag{4}$$

which is proven to converge [7] to a local minimum of (3). In the case where ρ is convex, this is also the global minimum. The term $\mathcal{S}_{\rho,\lambda/c}$ is a shrinkage operator dependant on ρ with threshold λ/c. The constant c depends on the combined operator $\mathbf{HD}$. The resulting estimator is

$$\hat{\mathbf{x}} = h_{\lambda,K}(\mathbf{y}) = \mathbf{D}\boldsymbol{\alpha}_K, \tag{5}$$

where K is the number of SSF iterations performed. When our task is minimizing the objective in (3), K is chosen to be the point that the decreasing rate is under a certain threshold. When aiming at minimization of the MSE, both λ and K need to be tuned carefully: in many cases, from a certain iteration, the objective value continues to descend, while the MSE value increases.

In [29,24] an alternative iterated shrinkage approach is presented, which leads to faster convergence. In this technique, termed PCD, the iteration formula is given by

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k - \mu(\mathbf{v}_k - \boldsymbol{\alpha}_k), \tag{6}$$

where

$$\mathbf{v}_k = \mathcal{S}_{\rho,\mathbf{W}\lambda}\big(\mathbf{W}(\mathbf{HD})^T(\mathbf{y} - \mathbf{HD}\boldsymbol{\alpha}_k) + \boldsymbol{\alpha}_k\big). \tag{7}$$

The diagonal matrix $\mathbf{W}$ contains the norms of the columns of $\mathbf{HD}$, which are computed off line. Once $\mathbf{v}_k$ is computed, a line search is performed along the trajectory $\mathbf{v}_k - \boldsymbol{\alpha}_k$, using the parameter μ. This method is also proven to converge [24]. Our estimator in this case is computed by (5) like in the SSF.

In order to use the SSF and the PCD methods for the inverse problems mentioned, the penalty function ρ and the dictionary $\mathbf{D}$ need to be chosen as well as the λ and the number of iterations. Following Elad et al. [24], we use the function

$$\rho_s(\alpha) = |\alpha| - s\log\left(1 + \frac{|\alpha|}{s}\right) \quad s \in (0, \infty). \tag{8}$$

This is a smoothed version of the ℓ_p-norm in the range $1 < p < 2$. The parameter s defines which ℓ_p-norm we are approximating in this range. For small values of s (e.g., $s = 0.001$) this function serves as a smoothed version of the ℓ_1-norm, and we use it as the penalty function throughout this work. This choice of ρ dictates the shape of the shrinkage operator $\mathcal{S}$ [24].

There are various possibilities for choosing $\mathbf{D}$. In our simulations, and following the work reported in [9,25,8], we use the undecimated Haar wavelet with 3 layers of resolution. The motivation for this choice is that when looking for sparse representations, redundant dictionaries are favorable.

[1] These are defined as $\mathrm{PSNR} = 10\log_{10}(255^2/\mathrm{MSE})$ and $\mathrm{ISNR} = 10\log_{10}(\mathrm{MSE}_{ref}/\mathrm{MSE})$, where MSE_{ref} is a reference-MSE to compare against.

The parameter λ which determines the weight of the penalty function also needs setting, as do the number of iterations. Ideally, we would like to choose the parameters that minimize the MSE between the original image $\mathbf{x}$ and its estimate. However, as we have seen already, the dependence of the MSE on $\mathbf{x}$ precludes its direct minimization. We now turn to present several methods that aim to bypass this difficulty.

3. Conventional parameter tuning methods

The main approaches the literature offers for automatic parameter setting are the GCV [14], the L-curve [15] and the discrepancy principle [16]. For linear deconvolution methods, the parameter selection can also be based on the Unbiased Predictive Risk Estimator (UPRE) [13]; however, the algorithms we treat here are non-linear.

In this section we accompany the description of these methods with an image deblurring experiment along the following lines: Working on the image `cameraman`, we use the blur kernel

$$\text{ker 1:}\quad h(x_1, x_2) = \frac{1}{1 + x_1^2 + x_2^2}, \quad x_1, x_2 = -7, \dots, 7 \tag{9}$$

and an additive white Gaussian noise with variance $\sigma^2 = 2$. This experiment and its chosen parameters are adopted from Figueiredo and Nowak [9].

3.1. The Generalized Cross Validation (GCV) method

For a reconstruction algorithm $h_{\lambda,K}(\cdot)$ and a degraded image $\mathbf{y}$, we define the residual of the algorithm to be

$$\mathbf{r}_{\mathbf{h}_{\lambda,\mathbf{K}}(\mathbf{y})} = \mathbf{H}h_{\lambda,K}(\mathbf{y}) - \mathbf{y}. \tag{10}$$

In the case where $h_{\lambda,K}(\cdot)$ is linear, the GCV method selects the parameters to be those that minimize the GCV functional given by

$$\mathrm{GCV}(\lambda, K) = \frac{\frac{1}{n_\mathbf{y}} \|\mathbf{r}_{h_{\lambda,K}(\mathbf{y})}\|_2^2}{[\frac{1}{n_\mathbf{y}} \operatorname{trace}(\mathbf{I} - \mathbf{H}h_{\lambda,K}(\cdot))]^2}, \tag{11}$$

where $\mathbf{I}$ is the identity matrix, and $n_\mathbf{y}$ is the length of $\mathbf{y}$. In high dimensions, the trace in the denominator of (11) is costly to compute directly. Stochastic estimation of this expression can be used instead [30]. Specifically, noting that $E[\mathbf{n}^T h_{\lambda,K}(\mathbf{n})] = \operatorname{trace}(h_{\lambda,K}(\cdot))$ where $\mathbf{n}$ is a random vector $\mathbf{n} \sim N(0, \mathbf{I})$, we can approximate the trace by one such instance of the random vector. More than one instance of the random variable $\mathbf{n}$ can be used for better estimation of the trace by averaging, but in this work we use just one such realization of $\mathbf{n}$. Replacing the trace in (11) yields

$$\mathrm{GCV}(\lambda, K) = \frac{\frac{1}{n_\mathbf{y}} \|\mathbf{r}_{h_{\lambda,K}(\mathbf{y})}\|_2^2}{[1 - \frac{1}{n_\mathbf{y}} \mathbf{n}^T \mathbf{H}h_{\lambda,K}(\mathbf{n})]^2}. \tag{12}$$

Some extensions of the GCV to non-linear cases were treated in [31–33,1], but as far as we know the literature does not offer extensions of the GCV to non-linear deconvolution algorithms like the one we address in this work. We suggest using (12) as the GCV objective to be minimized for the parameters selection, despite the non-linearity of $h_{\lambda,K}(\cdot)$.

Fig. 1 presents an experiment using GCV. For fixed number of iterations, $K = 43$, the GCV as a function of λ is presented in this figure. The value of λ is chosen to be the minimizer of the GCV function. This minimal point can be found by a line search, but such a procedure may get trapped in a local minimum of the GCV curve, since the obtained curve is not necessarily unimodal. Here and in later figures, the value chosen based on the true MSE is marked by an asterisk. In this case, as can be seen, the GCV performs well.

3.2. The L-curve method

For setting a parameter with the L-curve method, a log of the squared norm of the result of the algorithm versus the squared norm of its residual is plotted over a range of parameter values. The result is typically a curve with an L shape. The L-curve criterion is to choose the parameter value that corresponds to the “corner” of the curve – the point with maximum curvature [15]. For a parameter of interest λ, the curve is defined by the x-axis $X(\lambda) = \log \|r_{h_\lambda(\mathbf{y})}\|_2^2$ and the y-axis $Y(\lambda) = \log \|h_\lambda(\mathbf{y})\|_2^2$. The curvature is then given by

$$\kappa(\lambda) = \frac{X''(\lambda)Y'(\lambda) - X'(\lambda)Y''(\lambda)}{(X'(\lambda)^2 + Y'(\lambda)^2)^{3/2}}. \tag{13}$$

The value of λ is selected to maximize (13). Since the L-curve can only be sampled at specific discrete points, the calculation of the second order derivatives is very sensitive and thus should be avoided. Following Hansen and O’Leary [15],

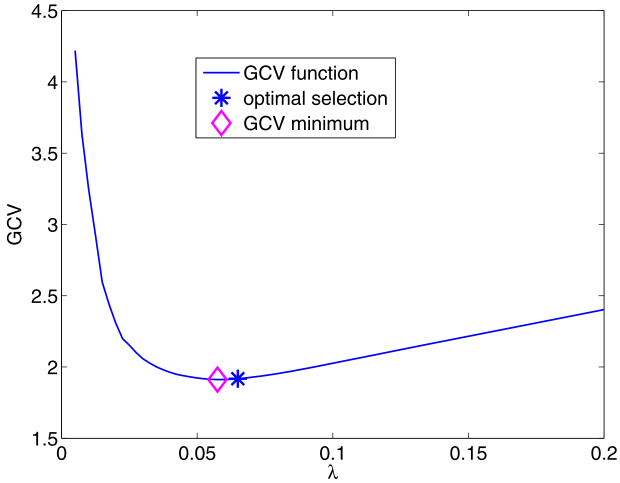
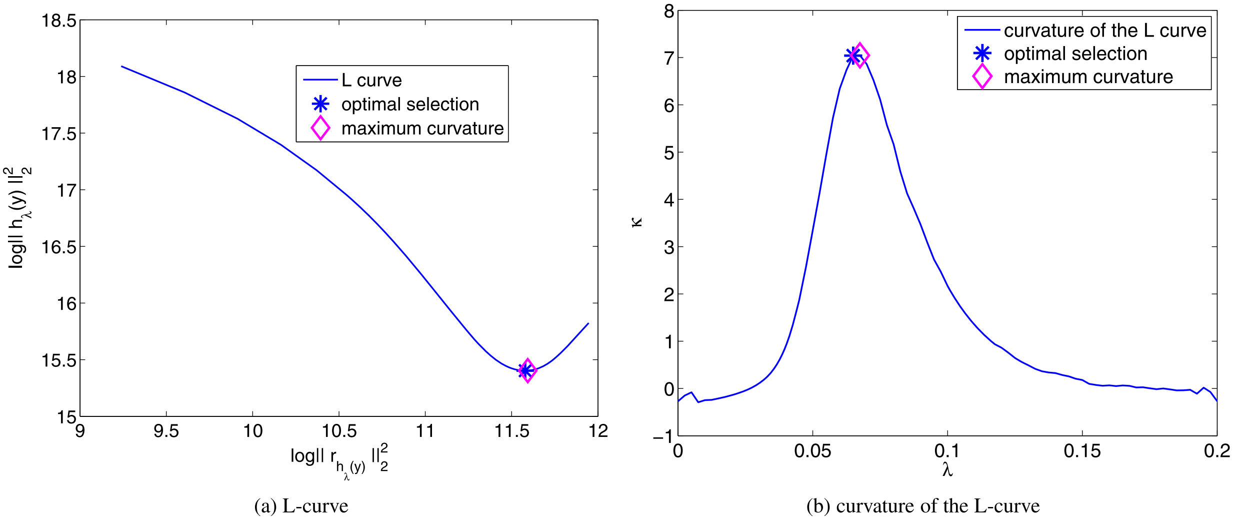


Fig. 1. The GCV curve as a function of λ for the PCD algorithm, with ker 1 as blur and with noise power of $\sigma^2 = 2$ where $K = 43$.



(a) L-curve

(b) curvature of the L-curve

Fig. 2. The L-curve results for setting λ for the PCD algorithm with ker 1, $\sigma^2 = 2$, and fixing $K = 43$. The L-curve is on the top and its curvature is on bottom.

we apply a local smoothing on the L-curve points and use the new smoothed points as control points for a cubic spline. We then calculate the derivatives of the spline at those points and get the curvature on each of them. The smoothing is done by fitting five points, where the smoothed point is the center of them, to a straight line in the least squares sense.

As in the case of the GCV, according to our knowledge, the L-curve has not been applied to non-linear deconvolution methods. Limitations of this approach for linear deconvolution are presented in [1,34]. In addition, the curvature is very sensitive and thus we can very easily be diverted from the appropriate value. In addition to this shortcoming, each calculation of a curvature of a point needs the values of the L-curve of its neighbors. Therefore, it is hard to tune the parameters in an efficient way.

Fig. 2 presents the experiment results using the L-curve method. This figure shows both the L-curve (top) and its curvature (bottom) for a fixed $K = 43$. The location of the chosen parameter λ is marked by a diamond. As can be seen, the λ value chosen as the curvature maximizer, in this case, is very close to the ideal value.

3.3. The discrepancy principle

A very intuitive parameter tuning idea is to use the discrepancy principle. Based on the noise properties, the original image approximately satisfies $\|\mathbf{Hx} - \mathbf{y}\|_2^2 \approx n_{\mathbf{y}}\sigma^2$. The discrepancy principle assumes that the same should hold for the

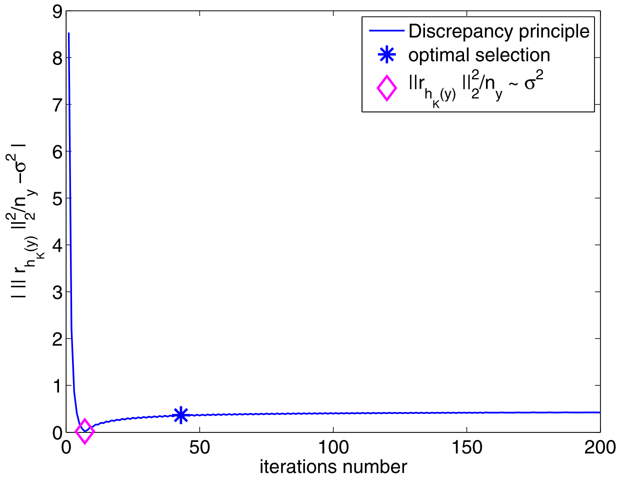


Fig. 3. The discrepancy principle for selecting the number of iterations for the PCD algorithm with ker 1, $\sigma^2 = 2$, and fixing $\lambda = 0.065$.

reconstructed image and thus selects the parameters based on the assumption that

$$\left\|\mathbf{H}h_{\lambda,K}(\mathbf{y}) - \mathbf{y}\right\|_2^2 \approx n_{\mathbf{y}}\sigma^2. \tag{14}$$

The iteration number, K, is chosen to be the first iteration in which $\|\mathbf{H}h_K(\mathbf{y}) - \mathbf{y}\|_2^2 < n_{\mathbf{y}}\sigma^2$.

Experimenting with the discrepancy principle, Fig. 3 presents the value of $|\|\mathbf{H}h_K(\mathbf{y}) - \mathbf{y}\|_2^2 - n_{\mathbf{y}}\sigma^2|$ as a function of K, where $\lambda = 0.065$ is fixed. The discrepancy principle selects the parameter value that achieves the minimal value on the graphs. As can be seen, although this method is very appealing, in many instances like this one, it tends to select a value of K far from the optimal one.

In this section we have described three parameter tuning methods: The GCV, the L-curve and the discrepancy principle. The discrepancy principle, which is commonly used as a stopping criterion, tends to fail in many cases, as can be seen in this section. As for the GCV and the L-curve methods, as far as we know, they were not applied before to non-linear deconvolution algorithms of the kind we use here. While their deployment is reasonable, their performance tends to be overly sensitive, possibly leading to poor results, as will be shown in Section 6. The fact that these three methods may give good parameter tuning in part of the cases may be explained by the fact that these methods do not aim at minimizing the MSE but use different considerations for the tuning. This leads us naturally to consider ways to select the parameters directly based on an estimated value of the MSE.

4. GSURE and projected GSURE

In this section we discuss an unbiased estimator for the MSE obtained by any reconstruction algorithm. This MSE estimate will later be used for parameter selection. The Stein Unbiased Risk Estimator (SURE) was proposed in [17,18] as an estimate of the MSE in the Gaussian denoising problem, namely, when $\mathbf{H} = \mathbf{I}$ and the noise is i.i.d. and Gaussian. This idea was then extended in [23] to handle more general inverse problems of the form discussed here, and a wider class of noise distributions. We refer to the later MSE estimate as the generalized SURE (GSURE).

The essential concept behind the GSURE principle is to seek an unbiased estimate for the MSE that is only a function of the estimate $h(\mathbf{y})$ and $\mathbf{y}$ (up to a constant which may depend on $\mathbf{x}$ but is independent of $h(\mathbf{y})$). Denote such an MSE estimate by $\eta(\hat{\mathbf{x}}, \mathbf{y})$. Then,

$$E\left[\eta(\hat{\mathbf{x}}, \mathbf{y})\right] = E\left[\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2\right] = E\left[\left\|h(\mathbf{y}) - \mathbf{x}\right\|_2^2\right]. \tag{15}$$

If we have access to such a function $\eta(\hat{\mathbf{x}}, \mathbf{y})$ that estimates the MSE, while being also dependent on a set of parameters Θ (that control the reconstruction performance), then we may choose the values of Θ so as to minimize this function, and in that way aim at minimizing the true MSE as a consequence.

For the case where $\mathbf{H}$ has full column rank, the GSURE is given by Eldar [23]

$$\eta\left(h(\mathbf{u}), \mathbf{y}\right) = c_1 + \left\|h(\mathbf{u})\right\|_2^2 - 2\mathbf{x}_{ML}^T h(\mathbf{u}) + 2\nabla_{\mathbf{u}} \cdot h(\mathbf{u}). \tag{16}$$

Here, $\mathbf{u} = (1/\sigma^2)\mathbf{H}^T\mathbf{y}$ is the sufficient statistic for the model (1), ∇ is the divergence operator

$$\nabla_{\mathbf{z}} h(\mathbf{u}) = \sum_{i=1}^{n} \frac{dh(\mathbf{u})[i]}{d\mathbf{z}[i]} = \operatorname{trace}\left(\frac{dh(\mathbf{u})}{d\mathbf{z}}\right), \tag{17}$$

and c_1 is a constant independent of $h(\mathbf{u})$. Given that our goal is to minimize the MSE by setting $h(\mathbf{u})$, the constant c_1 is irrelevant for the minimization process. The term $\mathbf{x}_{ML}$ stands for the Maximum Likelihood (ML) estimator—the minimizer of

$$\min_{\mathbf{z}} \|\mathbf{y}-\mathbf{H}\mathbf{z}\|. \tag{18}$$

If $\mathbf{H}^T\mathbf{H}$ is invertible, then this estimate is given by

$$\mathbf{x}_{ML} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{y}. \tag{19}$$

The MSE estimate presented in (16) is valid only if $\mathbf{H}^T\mathbf{H}$ is invertible. When this condition is not satisfied, $\mathbf{H}$ has a non-trivial null space. Therefore, we have no information about the components of $\mathbf{x}$ that lie in the null space $\mathcal{N}(\mathbf{H})$. Thus we can estimate only the projected MSE on the range of $\mathbf{H}^T$ (denoted by $\mathcal{R}(\mathbf{H}^T)$), which is equal to the orthogonal complement of the null space $\mathcal{N}(\mathbf{H})^{\perp}$. Therefore, instead of attempting to minimize an unbiased estimate of the MSE, we consider an unbiased estimate of the projected MSE: $E[\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)}\|h(\mathbf{u})-\mathbf{x}\|_2^2]$, where

$$\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)} = \mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T\right)^{\dagger}\mathbf{H}, \tag{20}$$

is the orthogonal projection onto $\mathcal{R}(\mathbf{H}^T) = \mathcal{N}(\mathbf{H})^{\perp}$. Here † denotes the Moore–Pernose pseudo inverse.

As shown in [23], an unbiased estimator for the projected MSE is given by the same formula as (16), but applied on the projected estimate $h(\mathbf{u})$:

$$\eta_{\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)}}\big(h(u), y\big) = c_2 + \left\|\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)}h(\mathbf{u})\right\|_2^2 - 2\mathbf{x}_{ML}^T h(\mathbf{u}) + 2\nabla_{\mathbf{u}}\cdot\big(\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)}h(\mathbf{u})\big). \tag{21}$$

Again, c_2 is a constant independent of $h(\mathbf{y})$. Here the ML estimate is chosen as the minimum-norm minimizer of (18), given by

$$\mathbf{x}_{ML} = \mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T\right)^{\dagger}\mathbf{y}. \tag{22}$$

5. Parameter tuning with GSURE

The generalization of SURE provides the possibility of using this estimator for the case of general $\mathbf{H}$, and specifically, for the deblurring and image scale-up problems. The tuning will be demonstrated for the SSF and PCD algorithms. As seen in Section 2, these methods have mainly two unknown parameters, the iterations number K and the thresholding value λ. Vonesch, Ramani and Unser, in [26], have already addressed this problem for the case where the operator $\mathbf{H}$ has full column rank. In their method, referred to as the *global method*, one parameter is being set at a time, given the other. We extend this for general $\mathbf{H}$ using the projected GSURE, showing the advantage of its usage over the regular GSURE, and over the conventional parameter setting methods that were presented in Section 3.

Though the global method for parameter setting provides us with a good estimation of the parameters as demonstrated in Section 6, it sets only one parameter given the other. We present an additional approach that sets both together. The parameter λ takes on a different value in each iteration of the algorithm in a greedy fashion. This way, we choose K together with λ, setting it to be the point where the estimated MSE stops decreasing. This method has the same computational cost for choosing two parameters as the global method has for selecting only one, without degrading the MSE performance. We also suggest a *greedy look-ahead method* that aims at better reconstruction results at the expense of higher complexity. We compare between these algorithms and the global method demonstrating the advantage of using these approaches for parameter tuning when both λ and K are unknown.

5.1. The global method using GSURE

In order to use GSURE with the iterated shrinkage algorithms, described in Section 2, we need to calculate the ML estimator, the derivative of the reconstruction algorithm with respect to $\mathbf{u}$, and the projection operator in the rank-deficient case. In our experiments, the ML estimator and the projection matrix are easily calculated by exploiting the fact that the blur operators considered are block circulant. As such, they are diagonalized by the 2D discrete Fourier transform (DFT). This helps both for the deblurring and the scale-up problems we handle.

To calculate the derivatives with respect to $\mathbf{u}$, we first reformulate the iterative equation as a function of $\mathbf{u}$ instead of $\mathbf{y}$. Rewriting (4) leads to

$$\boldsymbol{\alpha}_{k+1} = \mathcal{S}_{\rho,\lambda/c}\left(\frac{1}{c}(\mathbf{D})^T\left(\sigma^2\mathbf{u} - \mathbf{H}^T\mathbf{H}\mathbf{D}\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\right), \tag{23}$$

where $\hat{\mathbf{x}} = h_{\lambda,K}(\mathbf{u}) = \mathbf{D}\boldsymbol{\alpha}_K$ as in (5). The derivatives of $\hat{\mathbf{x}}$ are calculated recursively by first noticing that

$$\frac{d\boldsymbol{\alpha}_{k+1}}{d\mathbf{u}} = \mathcal{S}'_{\rho,\lambda/c}\left(\frac{1}{c}\mathbf{D}^T\left(\sigma^2\mathbf{u} - \mathbf{H}^T\mathbf{H}\mathbf{D}\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\right)\left[\frac{\sigma^2}{c}\mathbf{D}^T - \frac{1}{c}(\mathbf{H}\mathbf{D})^T\mathbf{H}\mathbf{D}\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}} + \frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\right], \tag{24}$$

where $\mathcal{S}'_{\rho,\lambda/c}(\cdot)$ is an element wise derivative of the shrinkage function, organized as a diagonal matrix. From here, the divergence of the estimator can be directly obtained by multiplying the above by $\mathbf{D}$ from the left, gathering the diagonal of the matrix and summing it up:

$$\nabla_{\mathbf{u}} h_{\lambda,K}(\mathbf{u}) = \operatorname{trace}\left(\frac{h_{\lambda,K}(\mathbf{u})}{d\mathbf{u}}\right) = \operatorname{trace}\left(\mathbf{D}\frac{d\boldsymbol{\alpha}_K}{d\mathbf{u}}\right). \tag{25}$$

It is impracticable to calculate the Jacobian of (24) in high dimensions. Instead we use the trace estimator that was used for the GCV in Section 3.1. Following Vonesch et al. [26] we choose a random vector $\mathbf{n} \sim N(0, \mathbf{I})$. Using this estimator we iterate over $\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n}$, which can be calculated instead of $\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}$, leading to the following iterative formula

$$\frac{d\boldsymbol{\alpha}_{k+1}}{d\mathbf{u}}\mathbf{n} = S'_{\rho,\lambda/c}\left(\frac{1}{c}\mathbf{D}^T\left(\mathbf{u} - \mathbf{H}^T\mathbf{H}\mathbf{D}\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\right)\left[\frac{\sigma^2}{c}\mathbf{D}^T\mathbf{n} - \frac{1}{c}(\mathbf{H}\mathbf{D})^T\mathbf{H}\mathbf{D}\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n} + \frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n}\right]. \tag{26}$$

For the estimation of the MSE for the PCD algorithm we need to calculate the trace as was done for the SSF. We rewrite the PCD iteration in terms of $\mathbf{u}$

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \mu(\mathbf{v}_k - \boldsymbol{\alpha}_k) = (1-\mu)\boldsymbol{\alpha}_k + \mu\left(\mathcal{S}_{\rho,\mathbf{W}\lambda}\left(\mathbf{W}\mathbf{D}^T\left(\sigma^2\mathbf{u} - \mathbf{H}^T\mathbf{H}\mathbf{D}\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\right)\right). \tag{27}$$

The derivative with respect to $\mathbf{u}$ is given by

$$\frac{d\boldsymbol{\alpha}_{k+1}}{du} = (1-\mu)\frac{d\boldsymbol{\alpha}_k}{du} + \mu\mathcal{S}'_{\rho,\mathbf{W}\lambda}\left(\mathbf{W}\mathbf{D}^T\left(\sigma^2\mathbf{u} - \mathbf{H}^T\mathbf{H}\mathbf{D}\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\right)\left[\mathbf{W}\mathbf{D}^T\left(\sigma^2\mathbf{I} - \mathbf{H}^T\mathbf{H}\mathbf{D}\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\right) + \frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\right]. \tag{28}$$

Using randomization we get the following iterative formula for estimating the trace value,

$$\frac{d\boldsymbol{\alpha}_{k+1}}{du}\mathbf{n} = (1-\mu)\frac{d\boldsymbol{\alpha}_k}{du}\mathbf{n} + \mu\mathcal{S}'_{\rho,\mathbf{W}\lambda}\left(\mathbf{W}\mathbf{D}^T\left(\sigma^2\mathbf{u} - \mathbf{H}^T\mathbf{H}\mathbf{D}\boldsymbol{\alpha}_k\right) + \boldsymbol{\alpha}_k\right)\left[\mathbf{W}\mathbf{D}^T\left(\sigma^2\mathbf{n} - \mathbf{H}^T\mathbf{H}\mathbf{D}\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n}\right) + \frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n}\right]. \tag{29}$$

Now that we hold a complete expression for the GSURE MSE estimate, K and λ can be chosen by minimizing it. This strategy was developed in [26] and is referred to as the global method. In the full-rank case, the projection matrix is the identity and the projected GSURE coincides with the regular GSURE. Thus we can use the projected GSURE equation in (21) for both cases. For a fixed and pre-chosen number of iterations K, the λ which minimizes the GSURE expression is chosen. Repeating this process for various values of K, one can minimize the overall global MSE with respect to both K and λ. As an analytical minimization of the GSURE is hard to achieve, we use a golden section search [35]. We initialize these algorithms by

$$\boldsymbol{\alpha}_0 = (\mathbf{H}\mathbf{D})^T\mathbf{y}, \tag{30}$$

and

$$\frac{d\boldsymbol{\alpha}_0}{d\mathbf{u}} = \sigma^2\mathbf{D}^T. \tag{31}$$

Denoting by T the GSURE calculation time (per-iteration), n_{gs} the golden-section number of iterations, and n_{is} the number of iterations of the iterated shrinkage algorithm, the time complexity of the global method for setting λ, when K is fixed, is $O(n_{is}n_{gs}T)$. For setting K another golden section can be performed over the number of iterations.

5.2. The greedy method using GSURE

In the greedy method, instead of choosing one global λ for all the iterations, we take a different route. Turning to a local alternative, the value of λ can be determined as the minimizer of the estimated MSE of the current iteration. In this strategy, K is set together with λ as the point where the estimated MSE (almost) stops decreasing. This allows us to avoid setting each of the parameters separately, as done in the global method. The algorithm proposed is the following:

- Initialize $\boldsymbol{\alpha}_0$ and calculate its derivative w.r.t. $\mathbf{u}$.
- Repeat:
  1. Set $\lambda_k^* = \arg\min_\lambda(\eta_{\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)}}(h_\lambda(\mathbf{u}, \boldsymbol{\alpha}_{k-1})))$.
  2. Perform the iterations in (27) and (29) (PCD case) or (23) and (26) (SSF case) for the calculation of $\boldsymbol{\alpha}_k$ and $\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n}$ using λ_k^*.
  3. Compute $\mathrm{MSE}_k^* = \eta_{\mathbf{P}_{\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)}}}(\mathbf{D}\boldsymbol{\alpha}_k, \mathbf{n}^T\mathbf{D}\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n}, \mathbf{x}_{ML})$ using (21), (5), (25), (20) and (19) or (22).
  4. If $\mathrm{MSE}_{k-1}^* - \mathrm{MSE}_k^* \leqslant \delta$, stop.
- $\hat{\mathbf{x}} = \mathbf{D}\boldsymbol{\alpha}_k$.

Table 1
A list of the experiments presented.

Problem	Blur	σ^2	Decimation	Algorithm	Cond-#
1. Deblurring	Ker 1: $(1+x_1^2+x_2^2)^{-1}$, $-7 \leqslant x_1, x_2 \leqslant 7$	2 or 8	None	PCD	77
2. Deblurring	Ker 2: 1/81, $-4 \leqslant x_1, x_2 \leqslant 4$	0.308	None	PCD	2.5e + 5
3. Deblurring	Ker 3: [1, 4, 6, 4, 1], separable	49	None	PCD	∞
4. Scale-up	Ker 4: [1, 2, 1]/4, separable	49	2 : 1, in each axis	SSF	∞
5. Scale-up	Ker 5: [1, 1]/2, separable	16	2 : 1, in each axis	SSF	∞

The complexity of the greedy algorithm is also $O(n_{is}n_{gs}T)$. When one of the parameters is fixed in the global method and we need to set only the other parameter we get the same complexity in the two approaches. In the case where both parameters are to be set, the greedy technique has an advantage since it chooses the number of iterations along with the parameter λ. In contrast, in the global method the number of iterations is either pre-chosen and thus suboptimal, or searched using yet another golden-search for optimizing K increasing the overall complexity.

To further decrease the MSE, we can modify the greedy algorithm by introducing a look-ahead option. One of the problems of the greedy strategy is that it minimizes the MSE of the current iteration but can harm the overall MSE. Thus, instead of choosing λ that minimizes the estimated MSE of the current iteration, we select it to minimize the estimated MSE of r iterations ahead, assuming that these r iterations are performed using the greedy approach described above. This change provides a look-ahead of r iterations, formulated as the following algorithm:

- Initialize $\boldsymbol{\alpha}_0$ and calculate its derivative w.r.t. $\mathbf{u}$.
- Repeat:
  1. Set λ_k^* by minimizing the estimated MSE r iterations ahead using the above-described greedy algorithm.
  2. Perform a single iteration as in (27) and (29) (PCD case) or (23) and (26) (SSF case) for the calculation of $\boldsymbol{\alpha}_k$ and $\frac{d\boldsymbol{\alpha}_k}{d\mathbf{u}}\mathbf{n}$ using λ_k^*.
  3. Compute $\mathrm{MSE}_k^* = \eta_{\mathbf{P}_{\mathbf{P}_{\mathcal{R}(\mathbf{H}^T)}}}(\mathbf{D}\boldsymbol{\alpha}_K, \mathbf{n}^T\mathbf{D}\frac{d\boldsymbol{\alpha}_K}{d\mathbf{u}}\mathbf{n}, \mathbf{x}_{ML})$ using (21), (5), (25), (20) and (19) or (22).
  4. If $\mathrm{MSE}_{k-1}^* - \mathrm{MSE}_k^* \leqslant \delta$, stop.
- $\hat{\mathbf{x}} = \mathbf{D}\boldsymbol{\alpha}_k$.

In step 1, for each test λ in the golden-section, r iterations of the greedy method are performed, as described for the greedy algorithm. Finally, λ of the current iteration is chosen such that the estimated MSE of the last rth greedy iteration is minimized. The time complexity of the r look-ahead greedy algorithm is $n_{is}(n_{gs})^2rT$, which is $n_{gs}r$ times slower than the other two methods. However, this is partially compensated for in some cases by getting smaller n_{is} due to faster convergence. Furthermore, the resulting MSE is often lower, as we illustrate in Section 6.

6. Results

The greedy methods were previously presented, with preliminary results on low dimension signals, in [36]. In this work we extended their use for images. In this section we present a set of experiments that demonstrate parameter tuning for the deblurring and the scale-up problems, using the PCD and the SSF algorithms. The representative experiments chosen correspond to different scenarios, which enable a demonstration of the various algorithms and their behavior. The kernels chosen for the debluring tests are standard kernels that are usually used to check debluring results [9,5]. In reality the blur that occurs in images may vary from a narrow blur to a wide one, and with weak to strong additive noise. This is exactly the variety of cases we are checking. For the image scale-up tests we use standard anti-aliasing filters before applying the down-sampling operator on the image. We should note that our work included various more tests, which are not brought here because of space limitation, and which lead to the same final conclusions. Table 1 lists the five tests performed. The tests we present divide into two groups: (i) preliminary ones that demonstrate the superiority of the projected GSURE over all the other alternatives, and then (ii) advanced tests that compare the global and the greedy methods.

6.1. Preliminary tests

We start by demonstrating the core ability of the GSURE and the classical techniques to choose the required parameters. In Fig. 4 we present the GSURE ISNR estimation[2] for Problem-1 applied on `Cameraman` with $\sigma^2 = 2$ as a function of the iterations. Similarly, Fig. 5 presents the GSURE ISNR as a function of λ for the same problem with $\sigma^2 = 8$. In both cases, the other parameter is kept fixed, choosing it to be the optimal one based on the true MSE. It can be observed in both graphs that the GSURE gives a very good estimation for the MSE and thus also a high-quality estimate of both λ and the number of iterations K. It can also be seen that it selects the parameters to be much closer to the optimal selection,

[2] The reference MSE is obtained by considering $\mathbf{y}$ as the estimate.

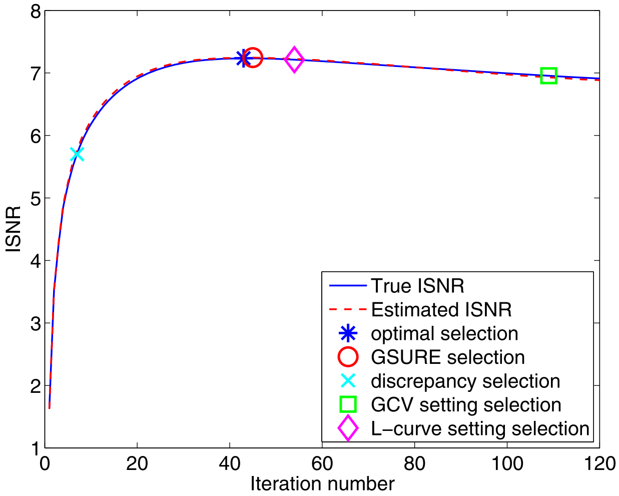


Fig. 4. The GSURE-estimated and the true ISNR as a function of the iterations number for Problem-1 with $\sigma^2 = 2$ and fixing $\lambda = 0.065$. The selection of the iterations number based on the various discussed methods is marked.

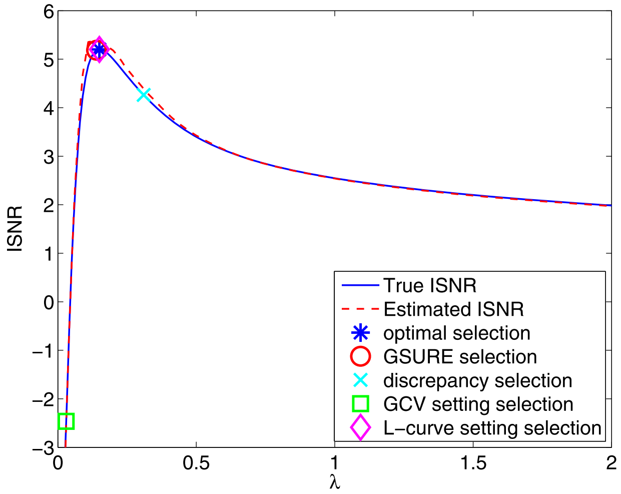


Fig. 5. The GSURE-estimated and the true ISNR as a function of λ for Problem-1 with $\sigma^2 = 8$ and fixing $K = 27$. The selection of λ based on the various discussed methods is marked.

Table 2
ISNR deblurring results for Problem-1 on `Boat` with $\sigma^2 = 2$, fixing $\lambda = 0.0725$ and setting K using various methods.

Parameter tuning method	ISNR
Optimal selection	6.461 dB
GSURE	6.458 dB
L-curve	6.43 dB
GCV	5.21 dB
Discrepancy principle	5.47 dB

compared to all the conventional methods. Among these conventional methods, the L-curve gives the best approximation for the parameters. However, it is hard to compute in an efficient way. The other methods approximate λ relatively well but fail in choosing K.

Repeating the experiment reported in Fig. 4 on the image `Boat` and fixing $\lambda = 0.0725$, Table 2 summarizes the ISNR reconstruction results for setting K. It can be seen again that the GSURE gives the best reconstruction results, being approximately the same as those achieved by the optimal selection.

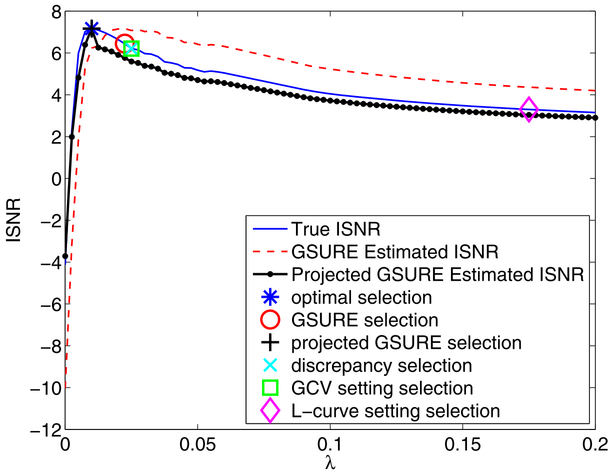


Fig. 6. The true and estimated ISNR as a function of λ, using the GSURE and its projected version. Applied on Problem-2, the iterations number is set to $K = 150$, and λ is estimated using the various proposed methods.

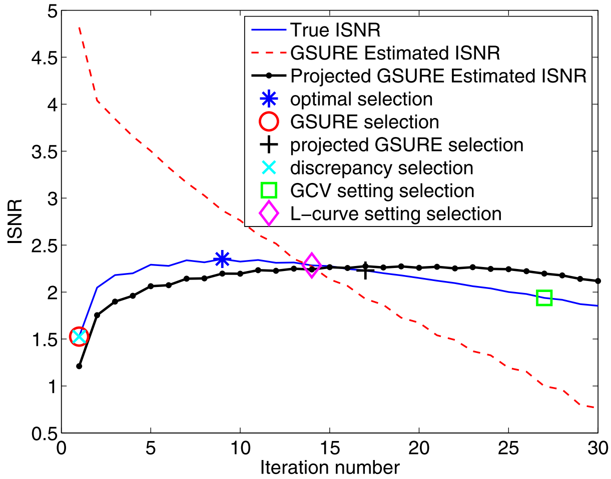


Fig. 7. The true and estimated ISNR as a function of K, using the GSURE and its projected version. Applied on Problem-3, λ is set to $\lambda = 0.67$, and K is estimated using the various proposed methods.

In Figs. 6 and 7 a comparison is made between the ISNR estimation using the regular and the projected GSURE. The problem considered in the first graph is Problem-2 and in the second is Problem-3, where both are applied on `Cameraman`. The selected parameters of the various methods are marked. The first graph is a function of λ and the second is a function of the iterations K. It can be seen that for both operators the estimation of λ and the iterations number using the projected GSURE is reasonable and always better than using the regular GSURE. Also, we see that in the case where we have a blur with real null space we get a totally unreliable estimation of the ISNR with the regular GSURE. We also see that L-curve fails this time in estimating λ.

Table 3 compares the various parameter setting methods for a deblurring experiment based on Problem-2 for reconstructing the image `Peppers` when λ is being tuned. The visual effect of the various parameter selections can be observed in Fig. 8. We see the same behavior as before, indicating that the projected GSURE is the best choice, leading to the best reconstruction results. The results again show that the regular GSURE (and the conventional methods) tend to fail, while the projected version performs very well.

In Figs. 9 and 10 a comparison is made between the estimation of the PSNR using the regular and the projected GSURE for Problem 4 (scale-up). As before, a comparison is done between the parameter values selected by the various methods. In addition, we show the reconstruction quality achieved by the bicubic, the Lanczos, and the bilinear interpolation methods,[3] as a reference performance to compare against.

[3] These scale-up techniques do not require parameter tuning.

Table 3
The deblurring results for Problem-2 applied on Peppers with $\sigma^2 = 2$, fixing $K = 92$, and setting λ using the different parameter tuning techniques.

Parameter tuning method	ISNR
Optimal selection	8.86 dB
Projected GSURE	8.86 dB
Discrepancy principle	8.28 dB
GSURE	8.17 dB
GCV	8.1 dB
L-curve	7.51 dB

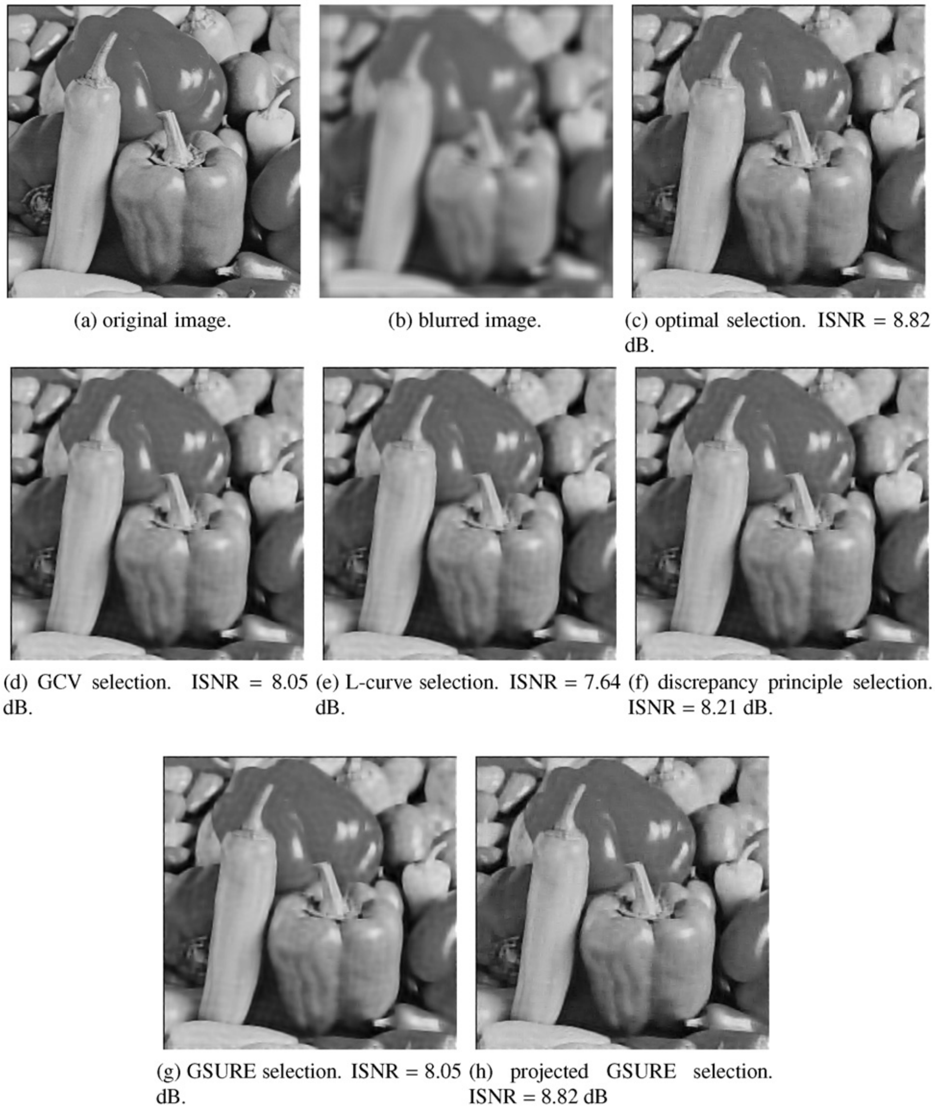

(a) original image.
(b) blurred image.
(c) optimal selection. ISNR = 8.82 dB.
(d) GCV selection. ISNR = 8.05 dB.
(e) L-curve selection. ISNR = 7.64 dB.
(f) discrepancy principle selection. ISNR = 8.21 dB.
(g) GSURE selection. ISNR = 8.05 dB.
(h) projected GSURE selection. ISNR = 8.82 dB

Fig. 8. The reconstruction result for Peppers for Problem-2, when $K = 92$ and λ is being tuned based on the different parameter tuning techniques. Form top left to bottom right: original image, blurred image, tuning using true ISNR, GCV, L-curve, discrepancy principle, GSURE and projected GSURE.

The first graph is a function of the iterations K and the second is a function of λ. It can be seen that the reconstruction algorithm with correctly tuned parameters achieves better reconstruction results than the conventional scale-up techniques, demonstrating the importance of the parameter tuning. Here as well, it can be observed that the projected GSURE provides the best estimation for the selection of λ and K.

As in the case of deblurring, the scale-up results show that the projected GSURE gives the best reconstruction results. Because of its superiority, we shall use it hereafter in the various experiments that follow. Since in the case of a full rank

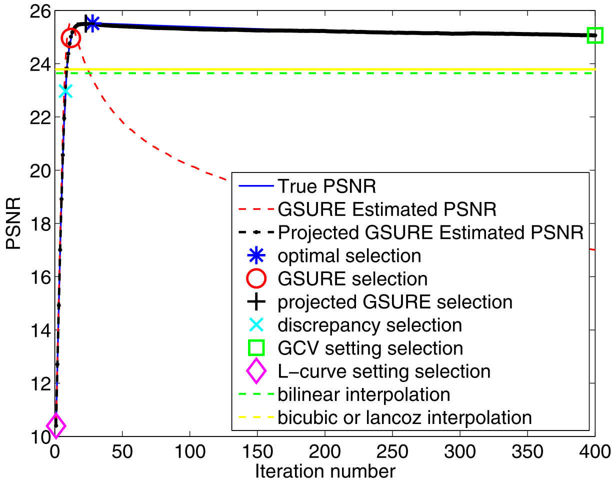


Fig. 9. The true and estimated PSNR as a function of K, using the GSURE and its projected version. Applied on Problem-4, λ is set to be $\lambda = 0.5$, and K is estimated using the various proposed methods.

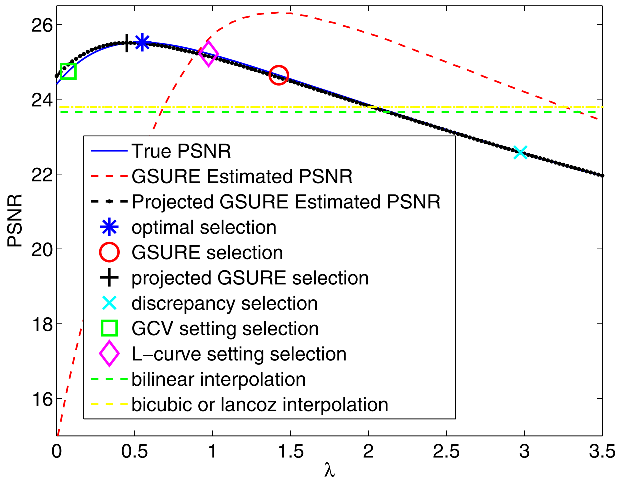


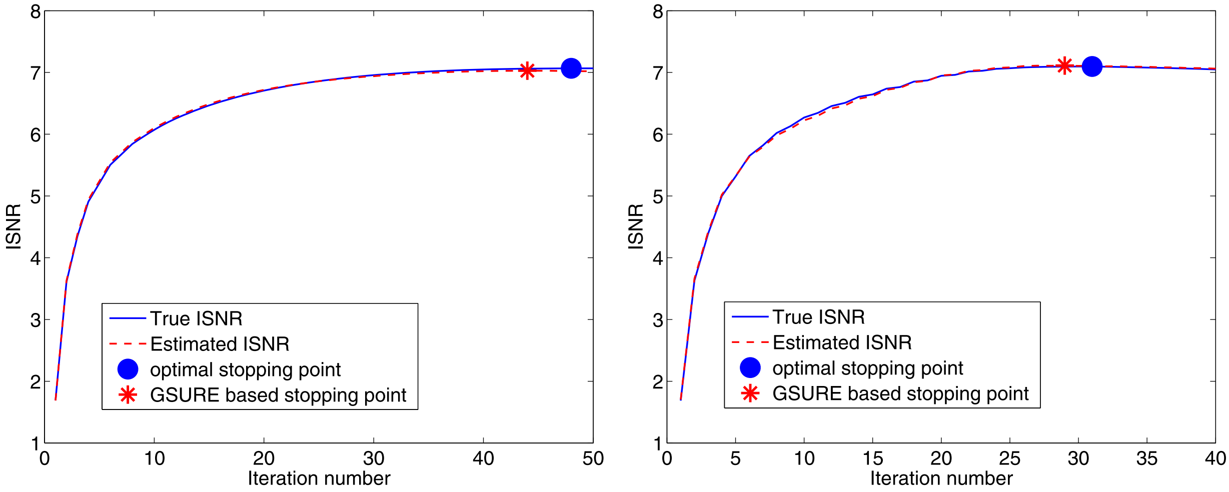
Fig. 10. The true and estimated PSNR as a function of λ, using the GSURE and its projected version. Applied on Problem-4, K is set to be $K = 27$, and λ is estimated using the various proposed methods.

operator, the GSURE and projected GSURE coincide, in every place that GSURE is referenced, the projected GSURE would be the one being intended.

6.2. Advanced tests

We now turn to look at the greedy strategies and compare them to the global method, all aiming to jointly estimate K and λ. In Fig. 11 we present the results obtained for Problem-1 applied on `Cameraman`. The graph shows the true ISNR after each iteration of the greedy and the greedy look-ahead methods (with $r = 1$), together with their ISNR estimation. As can be seen, the estimation has the same behavior as the true ISNR. In particular, in both graphs the maxima are very close. Choosing the number of iterations based on GSURE yields an ISNR very close to the one based on the true ISNR. This justifies our claim about using the greedy algorithm as an automatic stopping rule, by detecting an increase in the ISNR. The maxima in both graphs (marked in a circle as before) are very close, both in the greedy, as well as in the greedy look-ahead case. Fig. 12 presents the actual result obtained for a similar experiment applied on the image `House` with Problem-2.

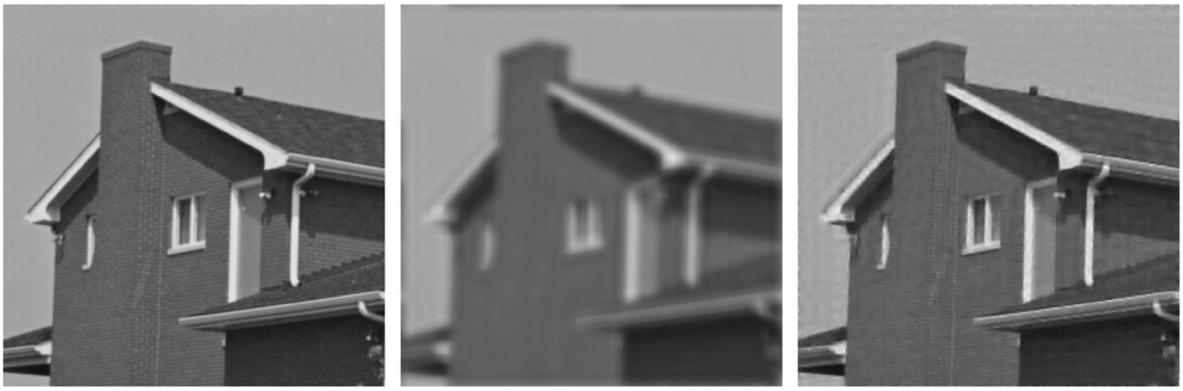
A comparison of the MSE of the global, the greedy, and the look-ahead techniques for the deblurring problem in images is presented in Fig. 13 for Problems-1, -2, and -3 on the image `Cameraman`. In all cases, no real advantage is seen in terms of ISNR by each of the techniques. Since the greedy algorithm sets the iterations number together with λ, this means that the greedy methods are more efficient in the overall performance, at a possible slight costs in terms of the final ISNR.



(a) greedy.

(b) greedy look-ahead with $r = 1$.

Fig. 11. The GSURE-estimated and the true ISNR as a function of the iteration number for the greedy algorithm (top) and for the greedy look-ahead algorithm with $r = 1$ (bottom) for Problem-1.



(a) original image.

(b) blurred image.

(c) greedy look-ahead method with $r = 1$. ISNR = 8.65dB.

Fig. 12. The reconstruction result for `House` for Problem-2, where λ is set based on the greedy look-ahead method with $r = 1$, and K is set as a stopping rule.

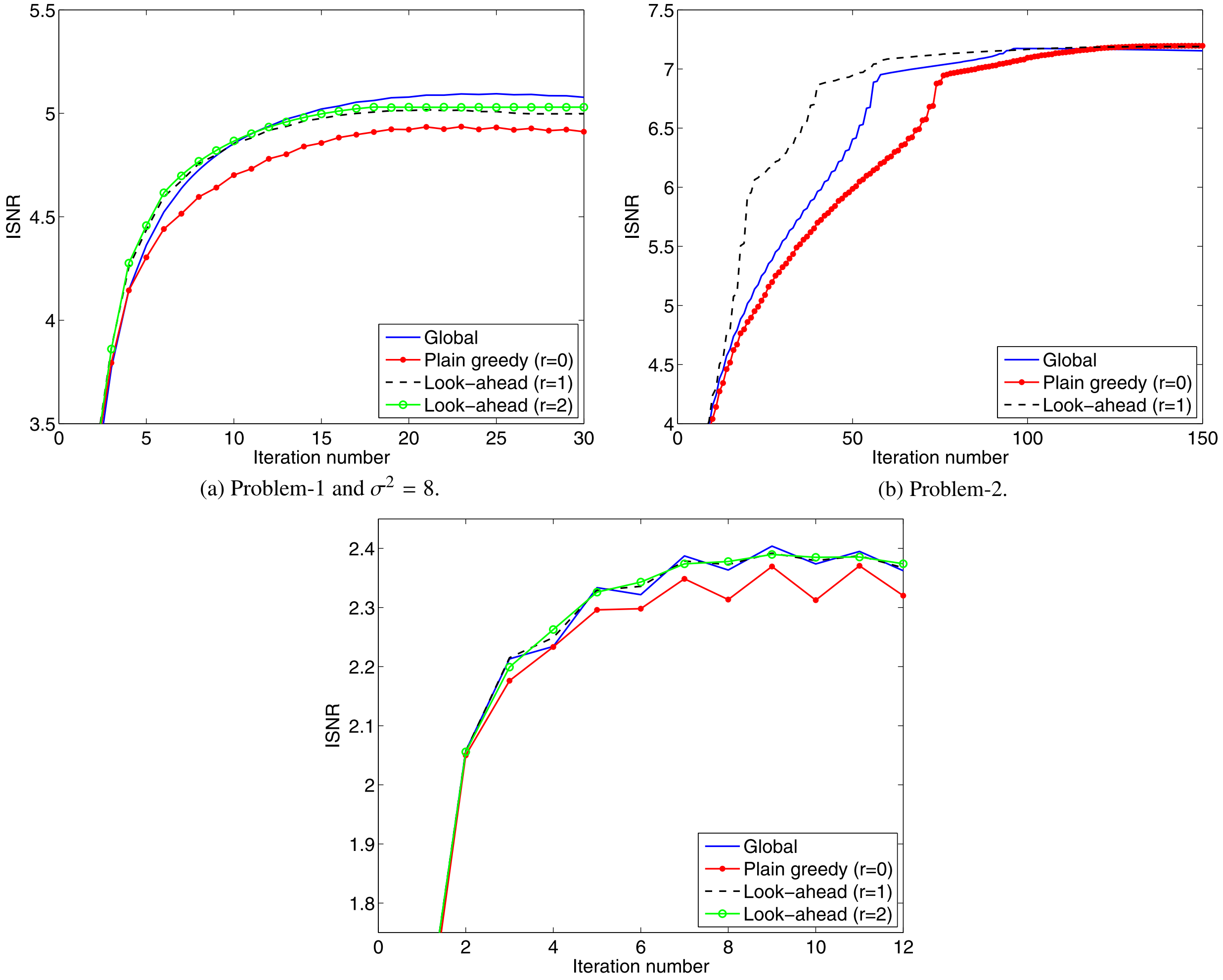
Turning to the scale-up Problems-4 and -5, the same comparison is done in Fig. 14. Here again we get similar results in terms of PSNR for the three methods. This implies that the two approaches (global and greedy) are equivalent in terms of the output quality, and the differences are mostly in their complexities, with a clear advantage to the greedy techniques.

To summarize, for the various deblurring and scale-up tests, there is no real advantage to one of the methods over the other, in terms of the quality of the reconstruction results. Thus, when we need to set both λ and K, the greedy methods are to be favored.

7. Conclusion and discussion

In this work we have considered automatic tuning of parameters for inverse problems. Our focus has been iterated shrinkage algorithms for deblurring and image scale-up, targeting an automatic way for choosing λ in each iteration in these methods, and choosing the number of iterations K. We extended the global tuning method developed in [26] to general ill-posed problems, by exploiting a projected version of the GSURE. We also applied the GCV, the L-curve, and the discrepancy-principle methods for this task of parameter tuning, and compared their selection with those of the (projected) GSURE. The GSURE was shown to give better approximation for the true values of the tuned parameters, leading to better results in the reconstruction.

Two greedy methods for parameter tuning – a simple and a look-ahead version – were presented. These two methods are shown to perform as well as the global method, with the advantage of setting an automatic stopping rule together with the λ parameter, whereas the previous methods set only one of the parameters, given the other.

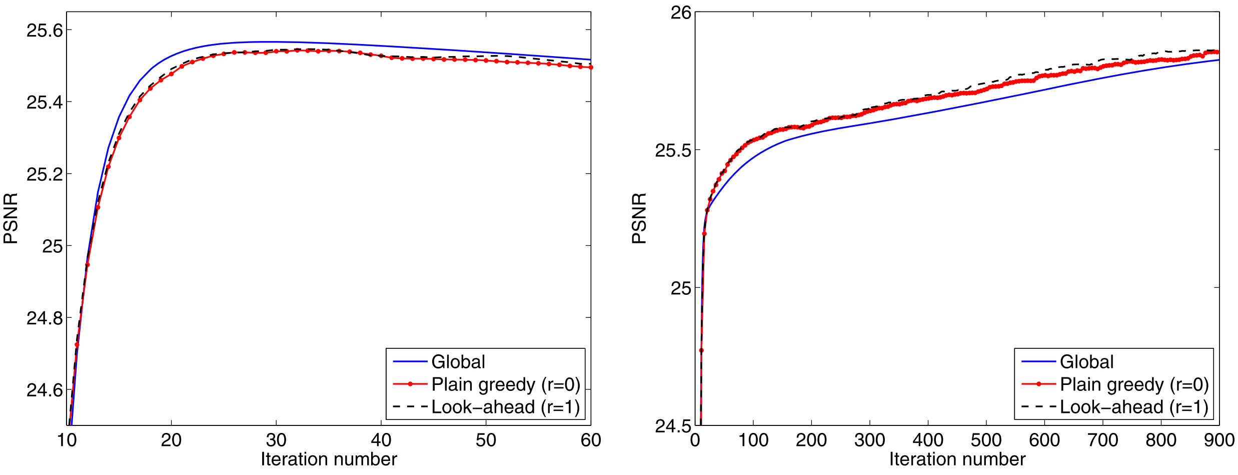


(a) Problem-1 and $\sigma^2 = 8$.

(b) Problem-2.

(c) Problem-3.

Fig. 13. The true ISNR as a function of the iterations number for the global, greedy, and look-ahead-greedy methods for Problems 1, 2 and 3 with different kernels and noise power.



(a) Problem-4

(b) Problem-5

Fig. 14. The true PSNR as a function of the iterations number for the global, greedy, and greedy look-ahead methods for Problems 4 and 5.

We are aware of the fact that there are better algorithms today for image deblurring and scale-up. However, the scope of this paper is to demonstrate the applicability of the GSURE for parameter tuning for such methods and not to compete with current state of the art reconstruction results. Using the concepts presented in this paper, automatic parameter tuning can be performed also for other techniques.

Acknowledgments

The authors are grateful to Mario A.T. Figueiredo for sharing his image restoration software with them.

This research was partly supported by the European Community's FP7-FET program, SMALL project, under grant agreement no. 225913, by the ISF grant number 599/08 and by the Israel Science Foundation under Grant no. 1081/07 and by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract no. 216715).

References

[1] C. Vogel, Computational Methods for Inverse Problems, Society for Industrial and Applied Mathematics, 2002.
[2] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, Institute of Physics Publishing, 1998.
[3] H. Takeda, S. Farsiu, P. Milanfar, Deblurring using regularized locally adaptive kernel regression, IEEE Trans. Image Process. 17 (4) (2008) 550–563.
[4] A. Foi, K. Dabov, V. Katkovnik, K. Egiazarian, Shape-adaptive DCT for denoising and image reconstruction, in: Proc. SPIE El. Imaging 2006, in: Image Process.: Algorithms and Systems V, vol. 6064A-18, San Jose, CA, 2006.
[5] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.
[6] M. Fadili, J. Starck, Sparse representation-based image deconvolution by iterative thresholding, in: Astronomical Data Analysis ADA'06, Marseille, France, 2006.
[7] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math. 57 (11) (2004) 1413–1457.
[8] M. Elad, B. Matalon, J. Shtok, M. Zibulevsky, A wide-angle view at iterated shrinkage algorithms, in: Wavelets XII, vol. 6701, SPIE, 2007.
[9] M. Figueiredo, R. Nowak, A bound optimization approach to wavelet-based image deconvolution, in: Proceedings of the 2005 IEEE International Conference on Image Processing (ICIP'05), vol. 2, 2005, pp. II–782–5.
[10] Z. Ben-Haim, Y.C. Eldar, Blind minimax estimation, IEEE Trans. Inform. Theory 53 (2007) 3145–3157.
[11] S. Kay, Y.C. Eldar, Rethinking biased estimation, IEEE Signal Process. Mag. 25 (3) (2008) 133–136.
[12] Y.C. Eldar, Rethinking biased estimation: Improving maximum likelihood and the cramér–rao bound, Found. Trends Signal Process. 1 (4) (2008) 305–449.
[13] C. Mallows, Some comments on C_P, Technometrics 15 (4) (1973) 661–675.
[14] G. Golub, M. Heath, G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics 21 (2) (1979) 215–223.
[15] P. Hansen, D. O'Leary, The use of the L-curve in the regularization of discrete ill-posed problems, SIAM J. Sci. Comput. 14 (6) (1993) 1487–1503.
[16] V. Morozov, On the solution of functional equations by the method of regularization, Soviet Math. Dokl. 7 (1966) 414–417.
[17] C.M. Stein, Estimation of the mean of a multivariate distribution, in: Proc. Prague Symp. Asymptotic Statist., 1973, pp. 345–381.
[18] C. Stein, Estimation of the mean of a multivariate normal distribution, Ann. Statist. 9 (6) (1981) 1135–1151.
[19] D. Donoho, I. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, J. Amer. Statist. Assoc. 90 (432) (1995) 1200–1224.
[20] T. Blu, F. Luisier, The SURE-LET approach to image denoising, IEEE Trans. Image Process. 16 (11) (2007) 2778–2786.
[21] J.-C. Pesquet, D. Leporini, A new wavelet estimator for image denoising, in: Sixth International Conference on Image Processing and Its Applications, 1997, vol. 1, 1997, pp. 249–253.
[22] A. Benazza-Benyahia, J.-C. Pesquet, Building robust wavelet estimators for multicomponent images using Stein's principle, IEEE Trans. Image Process. 14 (11) (2005) 1814–1830.
[23] Y. Eldar, Generalized SURE for exponential families: Applications to regularization, IEEE Trans. Signal Process. 57 (2) (2009) 471–481.
[24] M. Elad, B. Matalon, M. Zibulevsky, Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization, Appl. Comput. Harmon. Anal. 23 (2007) 346–367.
[25] M. Figueiredo, R. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Trans. Image Process. 12 (8) (2003) 906–916.
[26] C. Vonesch, S. Ramani, M. Unser, Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint, in: Proceedings of the 2008 IEEE International Conference on Image Processing (ICIP'08), 2008, pp. 665–668.
[27] Z. Wang, A.C. Bovik, Mean squared error: love it or leave it? A new look at signal fidelity measures, IEEE Signal Process. Mag. 26 (2009) 98–117.
[28] A. Bruckstein, M. Elad, D. Donoho, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Rev. 51 (1) (2009) 34–81.
[29] M. Elad, Why simple shrinkage is still relevant for redundant representations? IEEE Trans. Inform. Theory 52 (2006) 5559–5569.
[30] M. Hutchinson, A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines, Comm. Statist. Simulation Comput. 19 (2) (1990) 433–450.
[31] W.J. Fu, Nonlinear GCV and quasi-GCV for shrinkage models, J. Statist. Plann. Inference 131 (2) (2005) 333–347.
[32] E. Haber, D. Oldenburg, A GCV based method for nonlinear ill-posed problems, Comput. Geosci. 4 (1) (2000) 41–63, doi:10.1023/A:1011599530422.
[33] S. O'Sullivan, G. Whaba, A cross validated Bayesian retrieval algorithm for non-linear remote sensing experiments, J. Comput. Phys. 59 (3) (1985) 441–455.
[34] M. Hanke, Limitations of the L-curve method in ill-posed problems, BIT Numerical Mathematics 36 (2) (1996) 287–301.
[35] J. Kiefer, Sequential minimax search for a maximum, Proc. Amer. Math. Soc. 4 (1953) 502–506.
[36] R. Giryes, M. Elad, Y. Eldar, Automatic parameter setting for iterative shrinkage methods, in: Proceedings of the IEEE 25th Convention of Electrical and Electronics Engineers in Israel (IEEEI'08), 2008, pp. 820–824.