Near-Optimal Spectral Filtering and Error Estimation for Solving Ill-Posed Problems

Viktoria Taroudaki* Dianne P. O'Leary†

May 1, 2015

Abstract

We consider regularization methods for numerical solution of linear ill-posed problems, in particular, image deblurring, when the singular value decomposition (SVD) of the operator is available. We assume that the noise-free problem satisfies the discrete Picard condition and define the *Picard parameter*, the index beyond which the data, expressed in the coordinate system of the SVD, are dominated by noise. We propose estimating the Picard parameter graphically or using standard statistical tests. Having this parameter available allows us to estimate the mean and standard deviation of the noise and drop noisy components, thus making filtered solutions much more reliable. We show how to compute a near-optimal choice of filter parameters for *any* filter. This includes the truncated singular value decomposition (TSVD) filter, the truncated singular component method (TSCM) filter, and several new filters which we define, including a truncated Tikhonov filter, a Tikhonov-TSVD filter, a Heaviside filter, and a spline filter. We show how to estimate the error in any spectral filter, regardless of how the filter parameters are chosen. We demonstrate the usefulness of our new filters, our near-optimal choice of parameters, and our error estimates for restoring blurred images.

1 Introduction

In this work, we construct near-optimal filters for solving general linear ill-posed problems. Our examples, however, all concern image deblurring, so we focus on that particular application.

Images recorded by cameras or medical imaging devices are usually contaminated by noise and by blur that comes from factors such as the motion of the camera or the object, lens imperfections, or atmospheric turbulence. The restoration of such images is challenging since the problem is ill-posed. Even

*Applied Mathematics & Statistics and Scientific Computation Program, University of Maryland, College Park, MD 20742 (victtar@math.umd.edu).

†Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (oleary@cs.umd.edu).

if the blur is known (e.g., due to the motion or defocus), the noise is unknown and random.

A camera records an array of pixel values, corresponding to averages over square subsets (pixels) that partition the domain of the image. For definiteness, consider a grayscale image with an intensity in the interval $[0, 255]$ recorded for each pixel, 0 for a black pixel value and 255 for white. Blurring occurs when a pixel value is affected by the values of its neighbors. In this work, we assume that this is caused by a linear transformation. We can simplify the problem formulation by stacking the pixels of the image, column by column, to form a vector. The matrix $\boldsymbol{A}$ defining the linear transformation can be determined analytically or measured experimentally by determining point spread functions for each pixel of the original image. Experimentally, under conditions where the noise is negligible, the camera is used to capture an image of a scene containing a single white pixel with black elsewhere. Arranging the resulting image, called a *point spread function (PSF)*, as a vector gives the column of $\boldsymbol{A}$ corresponding to that pixel. If the blur is spatially invariant, then the other columns of the matrix are rearrangements of that one. Otherwise, the process is repeated for each pixel [19].

1.1 The Linear Model

We use the notation in Table 1. Note that matrices (uppercase) and column vectors (lowercase) are boldface.

Table 1: Notation for the Model

Symbol	Definition
$\boldsymbol{A}$	The $m \times n$ blurring matrix, possibly defined through PSFs.
$\boldsymbol{X}_{true}, \boldsymbol{x}_{true}$	Original (true) n-pixel image in matrix and vector form.
$\boldsymbol{B}_{true}, \boldsymbol{b}_{true}$	The m-pixel image resulting from blurring $\boldsymbol{X}_{true}$.
$\boldsymbol{E}$, $\boldsymbol{e}$	The m-pixel noise.
$\boldsymbol{B}, \boldsymbol{b}$	The m-pixel blurred image with added noise: $\boldsymbol{B} = \boldsymbol{B}_{true} + \boldsymbol{E}$ and $\boldsymbol{b} = \boldsymbol{b}_{true} + \boldsymbol{e}$.

With the above notation, our discrete linear model of image blurring is described by the equation $\boldsymbol{b}_{true} = \boldsymbol{A}\boldsymbol{x}_{true}$, or

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_{true} + \boldsymbol{e}, \tag{1}$$

and we know that $0 \leq (x_{true})_j \leq 255$, $j = 1, \ldots, n$.

Much research has been performed on ways to restore an image (e.g., [2, 6, 21]) and many different approaches and algorithms are now used to eliminate noise and blur. Our work focuses on the computation of a restored image using *spectral filters* that give weight to components of the image that are not so contaminated by noise. For each filter, we seek a near-optimal choice of the filter parameters that will give nearly the best approximation of the original image

$\boldsymbol{x}_{true}$. The near-optimal parameter for the Tikhonov filter has been determined in [20]. We build upon this work to devise general near-optimal filters.

1.2 Filtered Solutions

We assume that the noise $\boldsymbol{e}$ is sampled from a distribution with mean 0 and standard deviation s and that the matrix $\boldsymbol{A}$ is full-rank (rank=$n \leq m$) and generally ill-conditioned, since it is a faithful discretization of an ill-posed continuous operator. For simplicity, we will assume that $\boldsymbol{A}$ is a real matrix but the generalization to complex is straight-forward.

We use the coordinate system of the singular value decomposition (SVD) of $\boldsymbol{A}$ to represent the true solution and approximations to it. Let the SVD of $\boldsymbol{A}$ be $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal with size $m \times m$ and $n \times n$ respectively. The matrix $\boldsymbol{\Sigma}$ is zero except for its main diagonal elements, the singular values of the matrix $\boldsymbol{A}$ in non-increasing order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. We denote the ith column of $\boldsymbol{U}$ by $\boldsymbol{u}_i$ and the ith column of $\boldsymbol{V}$ by $\boldsymbol{v}_i$.

Multiplying (1) by $\boldsymbol{u}_i^T$ and using the SVD, we obtain [11, Sec 1.4]

$$\boldsymbol{x}_{true} = \sum_{i=1}^{n} \frac{\boldsymbol{u}_i^T(\boldsymbol{b} - \boldsymbol{e})}{\sigma_i} \boldsymbol{v}_i. \tag{2}$$

Since the noise vector $\boldsymbol{e}$ is unknown, we cannot compute this solution.

Instead, we compute a *filtered solution* $\boldsymbol{x}_{filt}$ using a *filter function* $\varphi_{\boldsymbol{\lambda}}$ and the information available to us:

$$\boldsymbol{x}_{filt} = \sum_{i=1}^{n} \phi_i(\boldsymbol{\lambda}) \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i, \tag{3}$$

where

$$\phi_i(\boldsymbol{\lambda}) = \varphi_{\boldsymbol{\lambda}}(\sigma_i).$$

Ideally, we want to minimize the norm of the error, the difference between x_{true} and our approximation $\boldsymbol{x}_{filt}$:

$$\min_{\boldsymbol{\lambda}} \|\boldsymbol{x}_{filt} - \boldsymbol{x}_{true}\|^2.$$

We use the 2-norm for convenience, but other choices are possible. Well-known examples of filters $\varphi_{\boldsymbol{\lambda}}(\sigma)$ include the truncated SVD (TSVD) filter

$$\phi_i(\lambda) = \begin{cases} 1, & \text{if } i \leq \lambda, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

with $\lambda \in \{1, \ldots, n\}$, and the Tikhonov Filter

$$\phi_i(\lambda) = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}, \tag{5}$$

with $\lambda \in \mathbb{R}^+$. Note that the parameter in the TSVD filter can also be made continuous by reparameterizing, defining, for $\lambda \in \mathbb{R}^+$,

$$\varphi_{\boldsymbol{\lambda}}(\sigma) = \begin{cases} 1, & \text{if } \sigma \geq \lambda, \\ 0, & \text{otherwise.} \end{cases}$$

O'Leary has determined a choice of parameter λ that aims to minimize the error $\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|$ for the Tikhonov filter [20].

1.3 Contributions and Outline

In this paper we make four main contributions:

- Section 2: We define a *Picard parameter*, related to the discrete Picard condition [12], that identifies the value k such that $\boldsymbol{u}_i^T \boldsymbol{b} \approx \boldsymbol{u}_i^T \boldsymbol{e}$ for $i \geq k$. The Picard parameter, implicit in some earlier work on regularization (see, for example, [3, Sec. 3.2], [10, Sec. 5], [11, Sec. 6.6], [23]), can be estimated graphically or, when the noise is Gaussian, by using standard statistical tests. We show that having k available to us makes the filtered solutions much more reliable, since we can use it to estimate the mean and standard deviation of the noise and to drop the noisy components $i \geq k$.
- Section 3: We show how to compute a near-optimal choice of $\boldsymbol{\lambda}$ for *any* spectral filter. This includes the TSVD filter and several new filters which we define, including a truncated Tikhonov filter, Heaviside filters, and a spline filter. We call this parameter choice the *Statistically Optimal Filtering Method (SOF).*
- Section 4: We propose an estimate for the error $\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|$ that can be used for *any* spectral filter, whether or not $\boldsymbol{\lambda}$ is determined using SOF.
- Section 5: We propose several new filters, including Heaviside, hybrid, and spline filters.

Numerical results are presented in Section 6, and conclusions are given in Section 7.

This work is a further development of that in Taroudaki's PhD dissertation [24].

2 Picard Parameter Estimation

In this section we define the Picard parameter and discuss its estimation, manually and then (for Gaussian noise) automatically, using the Lilliefors test.

2.1 The Discrete Picard Condition

Using the SVD of $\boldsymbol{A}$, define

$$\boldsymbol{\beta} = \boldsymbol{U}^T \boldsymbol{b}, \quad \boldsymbol{\epsilon} = \boldsymbol{U}^T \boldsymbol{e}.$$

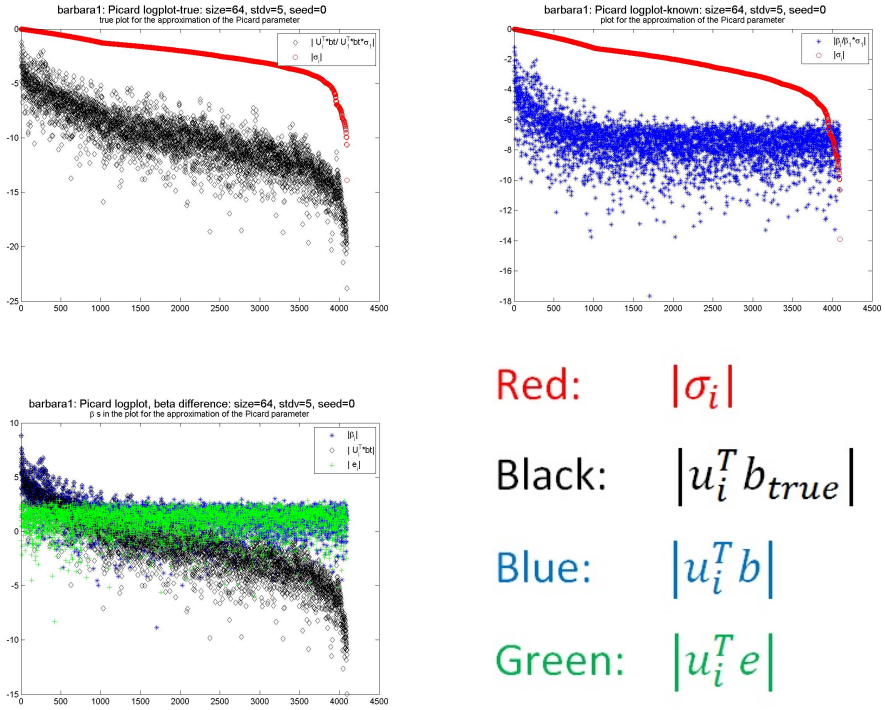


Figure 1: Picard plots. Top right: The data available in solving the problem, the singular values (red circles) and the observed image in the $\boldsymbol{U}$ coordinate system (blue stars). Top left: The true noise-free data (black diamonds) decay faster than the singular values, so the discrete Picard condition is satisfied. Bottom left: The noise (green plusses) violates the Picard condition and eventually dominates the true data. The blue data are the sum of the (unknown) black and green datapoints.

Then the $\boldsymbol{b}$ vector satisfies the *discrete Picard condition* if the sequence of true data values $\beta_i - \epsilon_i$ decay faster than the sequence of singular values σ_i. With the added noise, however, there is a value $k < n$ for which the β_i values are dominated by noise for $i \geq k$. For those values of i, $\beta_i \approx \epsilon_i$. So for a fixed n, we define the *Picard parameter* k as the smallest index such that $\beta_i \approx \epsilon_i$ for $i \geq k$.

Note that the Picard parameter is distinct from a regularization parameter that might be determined, for example, for the TSVD method. It is useful to use $k-1$ as an upper bound on indices of singular values to be considered when determining a regularization parameter, or use σ_{k-1} as a lower bound on singular values to be included. This eliminates consideration of contaminated data that might mislead an automatic parameter choice method. Setting the filter to zero for $\sigma \leq \sigma_k$ causes no loss of information and reduces the computational cost.

2.2 Manual estimation of the Picard parameter

The Picard parameter can be estimated graphically. The *Picard plot* displays $\{\log|\beta_i|\}$ and $\{\log\sigma_i\}$ vs. i. The value i for which the plot of $\{\log|\beta_i|\}$ levels off is an estimate of the Picard parameter for a noisy problem.

The Picard plot for a 64×64 Barbara image blurred by separable Gaussian blur is shown in the upper left in Figure 1. This noise-free problem satisfies the discrete Picard condition.

In contrast, the upper right Picard plot includes additive noise of standard deviation $s = 1$, shown in green in the lower left. The leveling off of the $|\beta_i|$ values is evident, at about $k = 1000$. If larger noise is added, then the leveling off occurs even sooner.

2.3 Automatic estimation of the Picard parameter when the noise is Gaussian

Estimating the Picard parameter from a Picard plot is subject to human judgement. In this subsection we discuss how the parameter might be estimated automaticially if the added noise is Gaussian with mean and standard deviation unknown. Other types of noise would require a different statistical test, but the methodology would be similar.

Our goal is to determine a value k beyond which the β_i are plausible samples from a normal distribution. This *normality testing* can be approached in many different ways, including examination of a histogram of the data and use of a test of whether a null hypothesis ("The sample is drawn from a normal distribution.") is valid or not. These null-hypothesis tests include the D'Agostino's K-squared test [4], the Jarque-Bera test [13], the Lilliefors test [15], the Kolmogorov-Smirnov test [17], and others [5]. For our purposes, we use the Lilliefors test, which is an adaptation of the Kolmogorov-Smirnov test.

The Liliefors test is performed on a sample of points in three major steps.

1. Estimate the population mean and the population variance using the sample mean and variance.

2. Compute the maximum discrepancy between the empirical distribution function $F_q(x)$ of the sample and the cumulative distribution function $F(x)$ of the normal distribution with the estimated mean and variance.

3. Assess whether the maximum discrepancy

$$D_q = \sup_x |F_q(x) - F(x)|$$

is large enough to be statistically significant.

The *empirical distribution function* F_q for q independent and identically distributed observations X_i is defined as

$$F_q(x) = \frac{1}{q}\sum_{i=1}^{q} I_{X_i \le x},$$

where

$$I_{X_i \le x}(x) = \begin{cases} 1, & \text{if } X_i \le x, \\ 0, & \text{otherwise.} \end{cases}$$

According to the Glivenko-Cantelli theorem [8, 1], if the sample that we examine comes from a distribution $F(x)$, then $F_q(x)$ will uniformly converge to $F(x)$ almost surely:

$$D_q = \|F_q - F\| = \sup_x |F_q(x) - F(x)| \to 0 \quad \text{almost surely.}$$

If D_q is less than a desired tolerance, then the null hypothesis is true with some probability. The smaller the tolerance, the larger the probability is.

We set the Picard parameter k by examining the sequences

$$\{\beta_i\}_{i=j}^{n},$$

for $j = n-3, n-4, \ldots, 1$. The first time that 10 successive sequences, for $j = \ell - 1, \ldots, \ell - 10$, fail the Lilliefors test at the 95% confidence level, we assume that the data are no longer noise-dominated and we set $k = \ell$.

We implement this using MATLAB's `lillietest`, which returns 0 if, with 95% confidence, the sample comes from a normal distribution and 1 if it does not. It requires sequences of at least 4 elements, so we need a discretization (pixelization) fine enough so that at least the trailing 14 β values are primarily due to noise. Once we have an estimate of k, we can also estimate the mean and standard deviation of the noise using the sample values.

In the case when there is no noise, we expect that the elements of $\boldsymbol{b}$ will decrease in average and that the Picard condition will be satisfied. That means that we do not expect the last entries in $\boldsymbol{b}$ to resemble a normal distribution, so the Lilliefors test will fail. In this unlikely event we would keep all the β values and set $k = n + 1$.

This method performed well in our tests, but other methods, such as the Mann-Kendall (nonparametric) test [14, 16] for monotonic trend should be considered, especially if the noise is not Gaussian.

3 Near-optimal parameter choice for general filters

Recall from (2) that the true solution to our problem is

$$\boldsymbol{x}_{true} = \sum_{i=1}^{n} \frac{\boldsymbol{u}_i^T(\boldsymbol{b} - \boldsymbol{e})}{\sigma_i} \boldsymbol{v}_i = \sum_{i=1}^{n} \frac{\beta_i - \epsilon_i}{\sigma_i} \boldsymbol{v}_i.$$

We hope to approximate it well using a filtered solution

$$\boldsymbol{x}_{filt} = \sum_{i=1}^{k} \phi_i(\boldsymbol{\lambda}) \frac{\beta_i}{\sigma_i} \boldsymbol{v}_i,$$

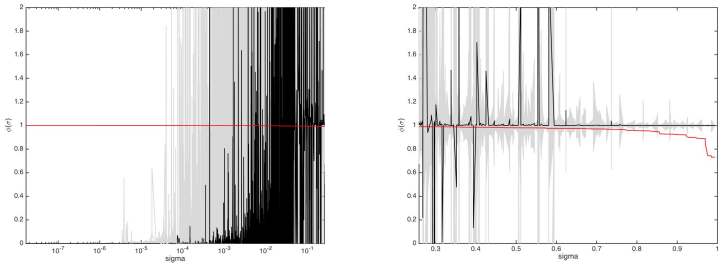


Figure 2: Truly optimal filters. The left plot is for $\sigma_i \in [0, 0.2587]$ and the right plot is for $\sigma_i \in [0.2587, 1]$. The gray curves are the optimal filters for 500 different noise samples. The black curve shows the average of these filters. The red curve shows the proportion of the 2-norm of the true image due to singular components to the left of the singular value.

where this expression differs from (3) in that it uses the Picard parameter to truncate the summation. We want to determine a nearly optimal filter $\boldsymbol{\phi}(\boldsymbol{\lambda})$.

From filters proposed in the literature, it is easy to conclude that an optimal filter is shaped something like the Tikhonov model: close to zero for small singular values, close to one for large singular values, and making a smooth transition for intermediate singular values.

We performed a simple experiment to determine how truly optimal filters really look. We used a 64×64 version of the Barbara image, blurring it with a separable Gaussian blur. We added noise that was normally distributed with mean zero and standard deviation one. We computed the truly optimal filter,

$$\phi_i = \frac{(\boldsymbol{\beta}_{true})_i}{\beta_i},$$

the one that recovers the true image, where

$$\boldsymbol{\beta}_{true} = \boldsymbol{\beta} - \boldsymbol{\epsilon}.$$

In Figure 2 we plot this filter in gray for 500 different noise samples. The black curve shows the average of these 500 filters. The optimal filter values are indeed near one for large singular values and near zero for small singular values, but even the average of the filters is wildly oscillating for intermediate singular values. A smoothly transitioning filter has no chance of capturing this behavior, so we might ask how important it is to have good filter values in the intermediate region. The answer is given in the red curve, which shows the proportion of the 2-norm of the true image due to singular components to the left of each value; i.e., the red value at σ_i is

$$\left(\sum_{j=i}^{n} (\boldsymbol{u}_j^T \boldsymbol{x}_{true})^2 / \sum_{j=1}^{n} (\boldsymbol{u}_j^T \boldsymbol{x}_{true})^2 \right)^{1/2}.$$

We see that 99% of the magnitude of the solution is in components corresponding to singular values of magnitude 0.2587 or larger, which explains why a smoothly transitioning filter might be effective.

Without full knowledge of the noise, we can't hope to reproduce the truly optimal filter for a given problem, but our aim in this work is to compute a *near-optimal* filter, one that approximately minimizes the error

$$\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|. \tag{6}$$

Our assumptions are:

- The entries in the noise vector $\boldsymbol{\epsilon}$ have mean 0 and standard deviation s, and an estimate of s is available.
- The Picard parameter k has already been determined.
- For $i \geq k$, we believe that $\beta_i - \epsilon_i$ is neglibly small, so we set $\phi_i(\boldsymbol{\lambda}) = 0$.

Algorithm 1 SOF algorithm to compute the near-optimal filter

1: **Input**: Blurring matrix $\boldsymbol{A}$, given image $\boldsymbol{b}$, the Picard parameter k, the standard deviation s or an estimate of it, and a function that evaluates the filter $\boldsymbol{\phi}(\boldsymbol{\lambda})$ for any given value of $\boldsymbol{\lambda}$.
2: **Output**: Optimal parameter $\boldsymbol{\lambda}^*$ of the filter and restored image $\boldsymbol{x}^*$.
3: Compute the SVD of $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$.
4: Let $\boldsymbol{\beta} = \boldsymbol{U}^T\boldsymbol{b}$.
5: Find the minimizer $\boldsymbol{\lambda}^*$ of

$$g(\boldsymbol{\lambda}) = \sum_{i=1}^{k-1} \frac{(1-\phi_i(\boldsymbol{\lambda}))^2\beta_i^2 - 2(1-\phi_i(\boldsymbol{\lambda}))s^2}{\sigma_i^2}.$$

6: Set $x_j^* = \sum_{i=1}^{k-1} \phi_i(\boldsymbol{\lambda}^*)\frac{\beta_i}{\sigma_i}\boldsymbol{v}_i$.
7: Estimate the error using (10).

We compute

$$\begin{aligned}
\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|^2 &= \sum_{i=1}^{n}\left[\left(\frac{\beta_i - \epsilon_i}{\sigma_i} - \phi_i(\boldsymbol{\lambda})\frac{\beta_i}{\sigma_i}\right)\right]^2 \\
&\approx \sum_{i=1}^{k-1}\left[\left(\frac{\beta_i - \epsilon_i}{\sigma_i} - \phi_i(\boldsymbol{\lambda})\frac{\beta_i}{\sigma_i}\right)\right]^2 \\
&= \sum_{i=1}^{k-1}\left(\frac{(1-\phi_i(\boldsymbol{\lambda}))\beta_i - \epsilon_i}{\sigma_i}\right)^2 \\
&= \sum_{i=1}^{k-1}\frac{\epsilon_i^2}{\sigma_i^2} + \sum_{i=1}^{k-1}\frac{(1-\phi_i(\boldsymbol{\lambda}))^2\beta_i^2 - 2(1-\phi_i(\boldsymbol{\lambda}))\beta_i\epsilon_i}{\sigma_i^2}.
\end{aligned}$$

Note that the first term does not depend on the filter. Since ϵ_i is unknown, we cannot evaluate this last expression. so we approximate it using expected values wherever necessary. Note that $\mathcal{E}(\beta_i\epsilon_i) = \mathcal{E}(((\boldsymbol{\beta}_{true})_i + \epsilon_i)\epsilon_i) = s^2$, so

$$\begin{aligned} \|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|^2 &\approx \sum_{i=1}^{k-1} \frac{\epsilon_i^2}{\sigma_i^2} + \sum_{i=1}^{k-1} \frac{(1-\phi_i(\boldsymbol{\lambda}))^2\beta_i^2 - 2(1-\phi_i(\boldsymbol{\lambda}))\mathcal{E}(\beta_i\epsilon_i)}{\sigma_i^2} \\ &= \sum_{i=1}^{k-1} \frac{\epsilon_i^2}{\sigma_i^2} + \sum_{i=1}^{k-1} \frac{(1-\phi_i(\boldsymbol{\lambda}))^2\beta_i^2 - 2(1-\phi_i(\boldsymbol{\lambda}))s^2}{\sigma_i^2}. \end{aligned} \tag{7}$$

We see that we need to minimize

$$g(\boldsymbol{\lambda}) = \sum_{i=1}^{k-1} \frac{(1-\phi_i(\boldsymbol{\lambda}))^2\beta_i^2 - 2(1-\phi_i(\boldsymbol{\lambda}))s^2}{\sigma_i^2}. \tag{8}$$

The resulting Statistically Optimal Filtering (SOF) algorithm is given in Algorithm 1. If some of the parameters in $\boldsymbol{\lambda}$ take on discrete values, then Step 6 in Algorithm 1 can be implemented as follows:

Set $g_{\min}$ to the largest floating point number.
for each possible choice of the discrete parameters in $\boldsymbol{\lambda}$ **do**
  Find the minimizer $\hat{\boldsymbol{\lambda}}$ of $g(\boldsymbol{\lambda})$ with respect to its continuous variables.
  if $g(\hat{\boldsymbol{\lambda}}) < g_{\min}$ **then**
    Set $g_{\min} = g(\hat{\boldsymbol{\lambda}})$
    Set $\boldsymbol{\lambda}^* = \hat{\boldsymbol{\lambda}}$.
  end if
end for

Minimizing equation (8) for the Tikhonov filter gives a solution similar to the one produced by O'Leary in [20]. The difference is that [20] includes terms for $i \geq k$ in the minimization.

We note that this procedure can be applied to *any* existing spectral filter, including the TSVD filter (4) the Tikhonov filter (5), and a filter of Rust [22] that is closely related to the TSVD filter. The difference is that it truncates components with small $|\beta_i|$, since these components are likely to be dominated by noise. In later work (private communication) Rust proposed also truncating components with small singular value, using a Picard plot, much as we did in determining the Picard parameter. The resulting truncated singular component (TSCM) filter is

$$\phi_i(\lambda) = \begin{cases} 1, & \text{if } 1 \leq i \leq \lambda \text{ and } |\boldsymbol{u}_i^T\boldsymbol{b}| > \tau s^2, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where τ is the truncation level (typically $\tau \in \{1, 2, 3\}$) and $\lambda \in \{1, \ldots, k\}$. Given the values of the Picard parameter k and the truncation level τ, the filter

depends on a single parameter λ. Of all the filters considered in this work, this is the only one that does not have a corresponding $\varphi_\lambda(\sigma)$ definition, since the filter values depend on the observed data.

4 Uncertainty Quantification

From our derivation in Section 3, we see that (7) is an estimate of $\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|$. We can make that estimate realistic by observing that if $\phi_i(\boldsymbol{\lambda}) < 1/2$, then we must conclude that $\beta_i \approx \epsilon_i$, so we propose the error estimate

$$\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|^2 \approx \sum_{i=1}^{k-1} \frac{r_i^2}{\sigma_i^2} + \sum_{i=1}^{k-1} \frac{(1-\phi_i)^2\beta_i^2 - 2(1-\phi_i)r_i^2}{\sigma_i^2}, \tag{10}$$

where

$$r_i = \begin{cases} s, & \text{if } \phi_i > 1/2, \\ \beta_i, & \text{otherwise.} \end{cases}$$

We note that this error estimate does not depend on how the filter $\boldsymbol{\phi}$ was derived, so it is of use even when the filter parameters are not determined by our SOF method but by other methods. The most commonly used parameter determination methods are generalized cross validation and the discrepancy principle.

Generalized cross validation (GCV) [9] determines the parameter $\boldsymbol{\lambda}$ so that if we leave one observation b_i out of the computation, it is best predicted by the parameters chosen for the remaining observations. This is equivalent to minimizing the GCV function

$$G(\boldsymbol{\lambda}) = \frac{\|(\boldsymbol{I} - \boldsymbol{A}\,\boldsymbol{V}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\,\boldsymbol{U}^T)\boldsymbol{b}\|_2^2}{\mathrm{trace}(\boldsymbol{I} - \boldsymbol{A}\,\boldsymbol{V}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\,\boldsymbol{U}^T))^2}, \tag{11}$$

where $\boldsymbol{\Phi}$ is the diagonal matrix of the filter factors ϕ_i.

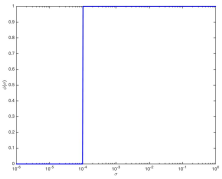
The Discrepancy Principle (DP) [18] computes the parameter $\boldsymbol{\lambda}$ for which the norm of the residual approximates the expected norm of the noise ($\delta = \mathcal{E}(\|\boldsymbol{e}\|_2)$):

$$\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{filt}\|_2 = \tau\delta, \tag{12}$$

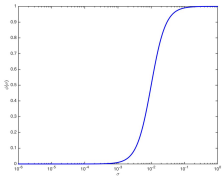
where τ is a factor commonly set to $\tau \in \{2, 3, 4, 5\}$. In our work, we used the parameter $\tau = 2$. This method relies on having a good estimate of the expected norm of the noise.

5 Some new regularizing filters

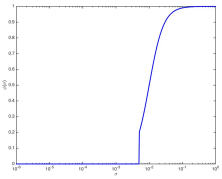
Next, we consider some useful new filters. See Figures 3 and 6 for illustrations of typical behavior of TSVD, Tikhonov, TSCM, and these new filters.



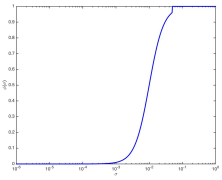
TSVD:
$\sigma_\lambda = 10^{-4}$



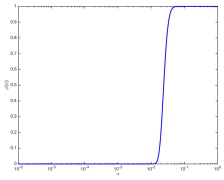
Tikhonov:
$\lambda = 10^{-4}$



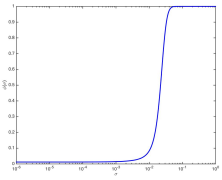
Tikhonov-k:
$\lambda = 10^{-4}$, $\sigma_k = 0.005$



Hybrid: $\sigma_{\lambda_1} = .05$
$\lambda_2 = 10^{-4}$, $\sigma_{\lambda_2} = .005$,



Heaviside-1:
$\tilde{\sigma} = 0.022$, $\lambda = 0.005$



Heaviside-2:
$\tilde{\sigma} = 0.022$, $\lambda = 0.005$

Figure 3: Typical behavior of various filters: plots of $\phi(\sigma)$ vs. σ

5.1 Truncated Tikhonov Filter

The Truncated Tikhonov filter is a truncated form of the Tikhonov filter:

$$\phi_i(\boldsymbol{\lambda}) = \begin{cases} \frac{\sigma_i^2}{\sigma_i^2+\lambda_1}, & \text{if } 1 \le i \le \lambda_2, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where $\lambda_1 \in \mathbb{R}^+$ and $\lambda_2 \in \{1, \ldots, k-1\}$.

Using this filter, we retain the good properties of the Tikhonov filter, i.e., the decrease in the weight of the terms that contain the small singular values, and also we decrease the computational cost since we do not add the terms that do not contribute any significant information but are dominated by noise.

Setting $k = n + 1$ recovers the Tikhonov filter.

5.2 The Hybrid (HYBR) Filter

We introduce a new, hybrid filter that combines the Tikhonov regularization filter with the TSVD filter:

$$\phi_i(\boldsymbol{\lambda}) = \begin{cases} 1, & \text{if } i \le \lambda_2, \\ \frac{\sigma_i^2}{\sigma_i^2+\lambda_1}, & \text{if } \lambda_2 < i \le \lambda_3, \\ 0, & \text{if } \lambda_3 < i \le n. \end{cases} \tag{14}$$

We have one continuous parameter $\lambda_1 \in \mathbb{R}^+$. and two discrete parameters $0 \le \lambda_2 \le \lambda_3 < k$. Alternatively, we can set $\lambda_3 = k - 1$, reducing the number of parameters to two.

This filter assumes that all of the singular values with an index less than λ_2 correspond to important information whereas those which have an index greater than λ_3 correspond to components overwhelmed by noise. For intermediate singular values, the Tikhonov filter is used.

Fuhry and Reichel [7] have created a similar filter called the regularized Tikhonov Filter:

$$\phi_i(\lambda) = \begin{cases} 1, & \text{if } \sigma_i > \lambda, \\ \frac{\sigma_i^2}{\lambda^2}, & \text{if } \lambda > \sigma_i. \end{cases}$$

5.3 Heaviside (HS) Filters

In general, the filters give more weight to the components of the solution that correspond to large singular values and less to those with smaller singular values. The TSVD, for example, is a shifted Heaviside function

$$\varphi_\lambda(x) = \begin{cases} 1, & \text{if } x > \lambda, \\ \frac{1}{2}, & \text{if } x = \lambda, \\ 0, & \text{if } x < \lambda, \end{cases} \tag{15}$$

where λ is between two singular values. We can use a continuous function that approximates the TSVD filter. For example, define the Heaviside-1 filter

$$\phi_i(\lambda) = e^{-e^{-(\sigma_i - \tilde{\sigma})/\lambda}},$$

where $\tilde{\sigma}$ is a centering parameter. Alternatively, define the Heaviside-2 filter as

$$\phi_i(\lambda) = \frac{1}{1 + e^{-(\sigma_i - \tilde{\sigma})/\lambda}}.$$

In both cases, the filters depend on the continuous parameter λ. The parameter $\tilde{\sigma} \in [\sigma_k, \sigma_1]$ can be specified by the user, chosen automatically based on the estimated noise, or used as an additional λ-parameter in the optimization.

5.4 A spline filter

Another idea is to create a filter

$$\phi_i(\lambda) = s(\sigma_i), \tag{16}$$

using a spline $s(\sigma)$ with knots $y_1 < y_2, \cdots < y_\ell$ in $[\sigma_k, \sigma_1]$, perhaps chosen equally spaced on a linear- or log-scale. As an example, we consider a cubic spline.

A convenient representation for a cubic spline $s(\sigma)$ on the interval $[y_j, y_{j+1}]$, for $j = 1, \cdots, \ell - 1$, is

$$s_{j+1}(\sigma) = m_j \frac{(y_{j+1} - \sigma)^3}{6h_{j+1}} + m_{j+1} \frac{(\sigma - y_j)^3}{6h_{j+1}} + a_j(\sigma - y_j) + b_j,$$

where $h_{j+1} = y_{j+1} - y_j$ and m_j, a_j, and b_j are parameters. The spline and its first two derivatives must be continuous at the knots, but it can have discontinuities in its third derivative at the knots. Imposing the continuity conditions results in constraints on the parameters. Our representation has been chosen so that the second derivative is continuous, since

$$s''_{j+1}(y_j) = m_j = s''_j(y_j),$$

but the other continuity conditions result in the linear constraints

$$\begin{aligned} s_{j+1}(y_j) = &\quad m_j \tfrac{h_{j+1}^2}{6} + b_j = m_j \tfrac{h_j^2}{6} + a_{j-1}(h_j) + b_{j-1} &&= s_j(y_j), \\ s'_{j+1}(y_j) = &\quad -m_j \tfrac{h_{j+1}}{2} + a_j = +m_j \tfrac{h_j}{2} + a_{j-1} &&= s'_j(y_j), \end{aligned}$$

for $j = 2, \ldots, \ell - 1$. This gives us $2\ell - 4$ conditions for the $3\ell - 2$ parameters, leaving $\ell + 2$ parameters to determine. We reduce this to ℓ parameters $\boldsymbol{\lambda}$ by specifying boundary conditions $s(\sigma_k) = 0$ and $s(\sigma_1) = 1$. Other choices of boundary conditions can be used.

6 Numerical Results

In this section we present results of using several filters:

- TSVDn: Standard TSVD (4), with no use of the Picard paramater.
- TIKn: Standard Tikhonov (5), with no use of the Picard paramater.
- TIKk: Tikhonov, truncated using the Picard parameter (13).
- TSVDk: TSVD with the restriction that $\lambda \leq k$.
- HYBRk: The new hybrid filter, using the Picard parameter (14).
- SPLlink: The new spline filter, 5 knots, using the Picard parameter (16).

We determine the parameters for the filters in four ways:

- SOF: our new Algorithm 1 (denoted by "S" in later figures).
- GCV: generalized cross-validation (11) (denoted by "G" in later figures).
- DP: the discrepancy principle (12) (denoted by "D" in later figures).
- OPT: the true optimal parameters for the given filter. Note that this cannot be determined without knowing the noise sample, so it is not a method that can be used on real problems.

This gives us 24 algorithms. MATLAB implementations of these algorithms are available at `http://www.cs.umd.edu/users/oleary/software`.

Figure 4: 128×128 Barbara: results

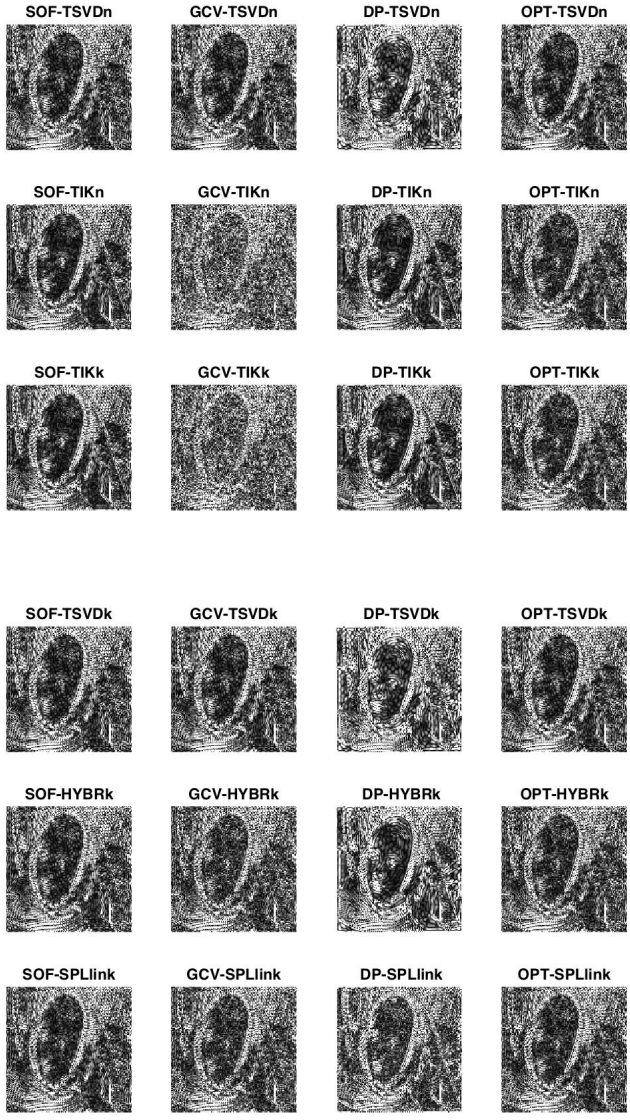


Figure 5: 128×128 Barbara: error images (black = 0)

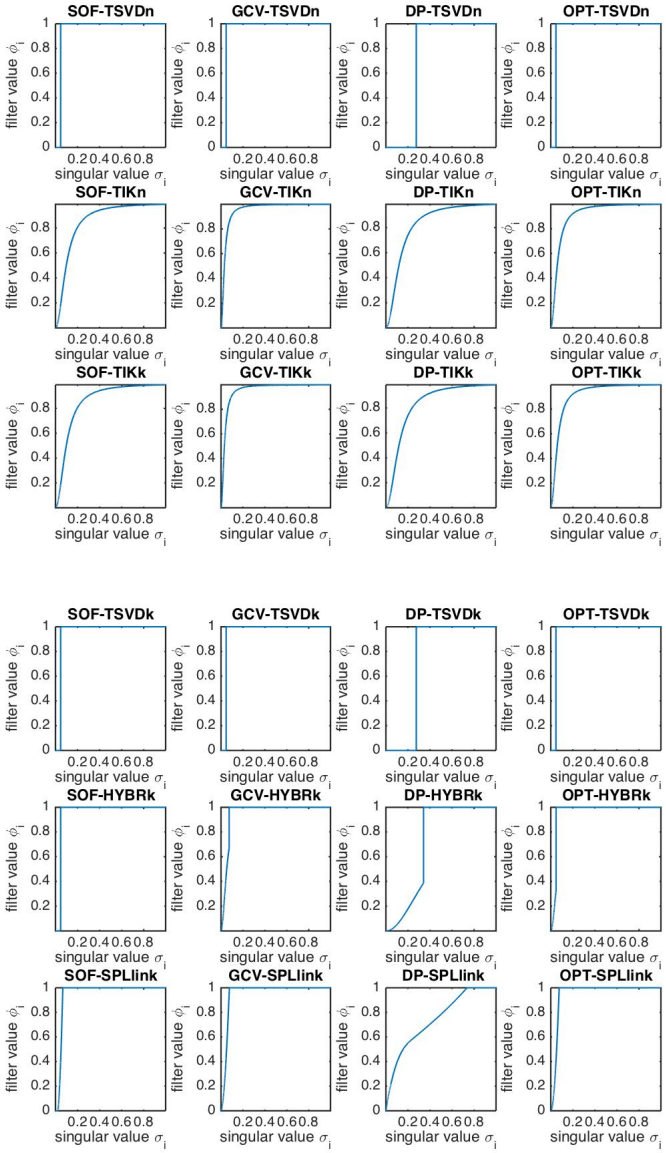


Figure 6: 128×128 Barbara: filters

Example 1

This experiment is performed using a 128×128 Barbara image with separable Gaussian blur and standard deviation of the noise $s = 1$. The resulting images for SOF, GCV, and DP, shown in Figure 4, are all quite similar to the optimal ones, shown in the last column. The error images, shown in Figure 5 do show some differences, though. In particular, the GCV error images, and sometimes the DP images, are whiter, indicating larger error. Figure 6 shows the filters that were used to produce these images. It is remarkable that such different filters can produce such similar results, but explained by the experiment in Figure 2.

Next we explore the reliability of these methods.

Example 2

We repeated the above experment 50 times, using a 64×64 Barbara image, to see how reliable the methods are. We used $s = 1$ as well as $s = 10$.

Figure 7 shows box plots of the relative errors for the methods. Along the horizontal axis are the 6 filters, with columns for SOF, GCV, and DP for each filter. Note that TSVDn produced a wide variety of relative errors. This was our motivation for developing reliable ways to compute the Picard parameter; the results for TSVDk are much more consistent. In general, DP produced larger errors, while the performance of SOF and GCV were generally comparable.

Figure 8 shows the error factors, the ratio of the error of these methods to the error produced by OPT, the true optimal parameter. Again DP is shown to often produce much larger errors. For the larger noise level, there is significant variation in the quality of the results for the SPL filter. We believe that there are multiple local minimizers for the error function and that more careful minimization from multiple starting points would be necessary to resolve this.

Figure 9 shows the quality of the error estimate (10) for the 24 methods. We note that regardless of method, the estimate is often within an order of magnitude of the true value.

7 Conclusions

We defined the Picard parameter and we described a manual and an automatic way of computing it. The manual approach is based on the Picard plot and the automatic on histograms and the Lilliefors method for normality testing. The Picard parameter proved to be quite useful for constraining the parameters in spectral filters. It would be very useful to develop better methods for determining the Picard parameter for Gaussian noise, and to develop methods for other noise distributions.

We derived a way to determine near-optimal parameters for spectral filters and compared this method with GCV and DP.

We defined several new filters and demonstrated their usefulness.

Finally, we developed an error estimate that can be used for any spectral filter, regardless of how the parameters are chosen.

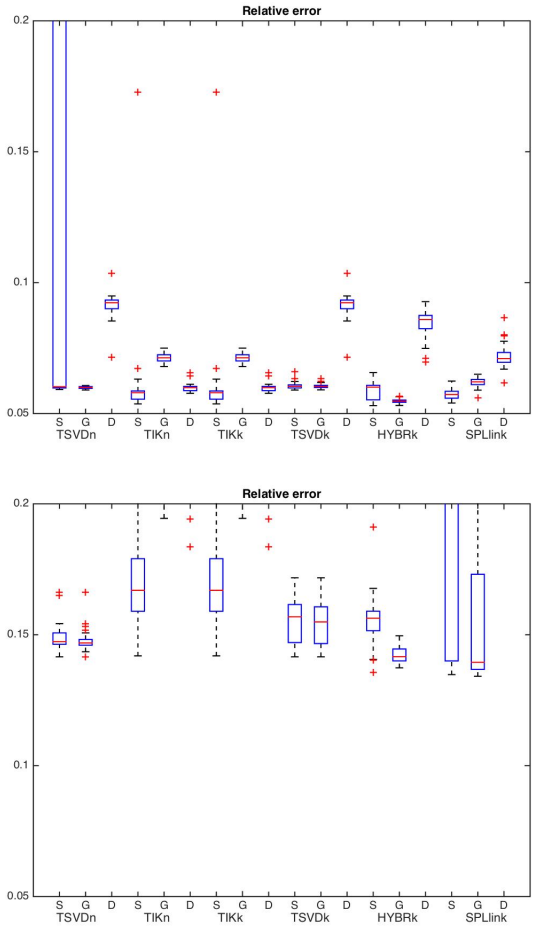


Figure 7: 64×64 Barbara: relative error $\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|/\|\boldsymbol{x}_{true}\|$. Top: $s = 1$. Bottom: $s = 10$.

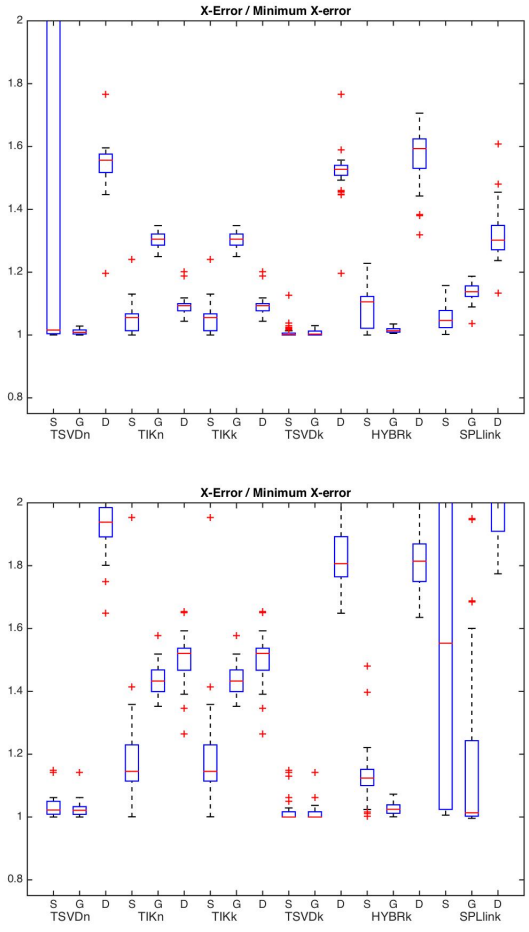


Figure 8: 64×64 Barbara: $\|\boldsymbol{x}_{true} - \boldsymbol{x}_{filt}\|/\|\boldsymbol{x}_{true} - \boldsymbol{x}_{opt}\|$. How close to optimal is the error for this kind of filter? Top: $s = 1$. Bottom: $s = 10$.

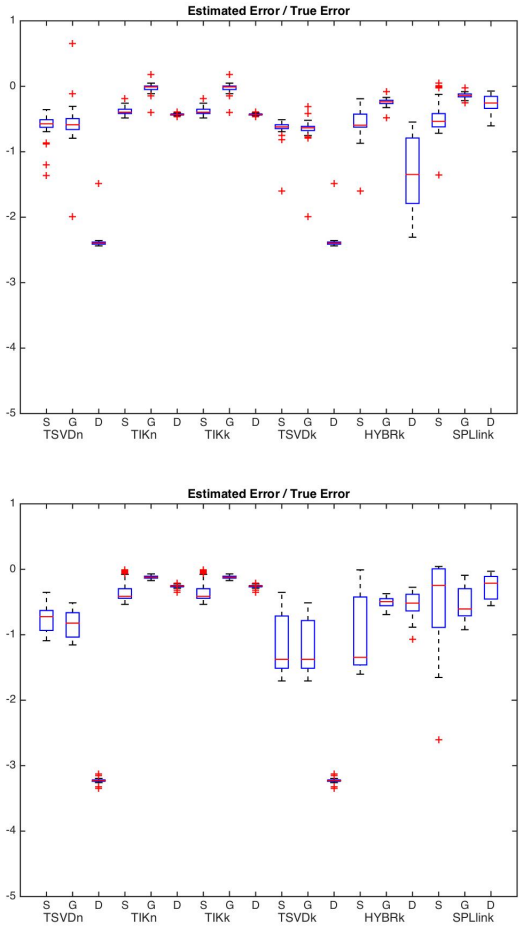


Figure 9: 64×64 Barbara: How good is the error estimate? Top: $s = 1$. Bottom: $s = 10$.

8 Acknowledgements

This work was supported by the US National Science Foundation under grant NSF DMS 1016266. The work of the first author was partially supported by a fellowship from the Onassis Foundation. We are grateful to David Darmon for helpful comments on this work.

References

[1] F.G. Cantelli, *Sulla determinazione empirica delle leggi di probabilita*, Giorn. Ist. Ital. Attuari, 4 (1933), pp. 421–424. Referenced by Kevin Ford, University of Illinois at Urbana-Champaign, (http://www.math.uiuc.edu/ ford/wwwpapers/kol_ engl2.pdf).

[2] S. Cho and S. Lee, *Fast motion deblurring*, ACM Transactions on Graphics (TOG), 28 (2009), p. 145.

[3] J.M. Chung, M.E. Kilmer, and D.P. O'Leary, *A framework for regularization via operator approximation*, SIAM J. Sci. Comput., 37 (2015), pp. B332–B359.

[4] R.B. D'Agostino, *An omnibus test of normality for moderate and large sample size*, Biometrika, 58 (1971), pp. 341–348.

[5] J. Devore, *Probability and Statistics for Engineering and the Sciences*, Cengage Learning, 2012.

[6] M. Elad and A. Feuer, *Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images*, IEEE Trans. on Image Processing, 6 (1997), pp. 1646–1658.

[7] M. Fuhry and L. Reichel, *A new Tikhonov regularization method*, Numerical Algorithms, 59 (2012), pp. 433–445.

[8] V. Glivenko, *Sulla determinazione empirica delle leggi di probabilita*, Giorn. Ist. Ital. Attuari, 4 (1933), pp. 92–99. Referenced by Kevin Ford, University of Illinois at Urbana-Champaign, (http://www.math.uiuc.edu/ ford/wwwpapers/kol_ engl2.pdf).

[9] G.H. Golub, M. Heath, and G. Wahba, *Generalized Cross Validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223.

[10] P.C. Hansen, *The discrete Picard condition for discrete ill-posed problems*, BIT Numerical Mathematics, 30 (1990), pp. 658–672.

[11] P.C. Hansen, J. Nagy, and D.P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.

[12] R.J. Hanson, *A numerical method for solving Fredholm integral equations of the first kind using singular values*, SIAM J. Numer. Anal., 8 (1971), pp. 616–622.

[13] C.M. Jargue and A.K. Bera, *A test for normality of observations and regression residuals*, International Statistical Review, 55 (1987), pp. 163–172.

[14] M.G. Kendall, *A new measure of rank correlation*, Biometrika, 30 (1938), pp. 81–93.

[15] H.W. Lilliefors, *On the Kolmogorov-Smirnov test for normality with mean and variance unknown*, Journal of the American Statistical Association, 62 (1967), pp. 399–402.

[16] H.B. Mann, *Non-parametric tests against trend*, Econometrica, 13 (1945), pp. 245–259.

[17] F.J. Massey, *The Kolmogorov-Smirnov test for goodness of fit*, Journal of the American Statistical Association, 46 (1951), pp. 68–78.

[18] V.A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7 (1966), pp. 414–417.

[19] J.G. Nagy and D. P. O'Leary, *Restoring images degraded by spatially-variant blur*, SIAM J. Sci. Comput., 19 (1998), pp. 1063–1082.

[20] D.P. O'Leary, *Near-optimal parameters for Tikhonov and other regularization methods*, SIAM J. Sci. Comput., 23 (2001), pp. 1161–1171.

[21] L.I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.

[22] B.W. Rust, *Truncating the singular value decomposition for ill-posed problems*, NISTIR 6131, U.S. Dept. of Commerce, (1998).

[23] B. W. Rust, *Parameter selection for constrained solutions to ill-posed problems*, Computing Science and Statistics, 32 (2000), pp. 333–347.

[24] V. Taroudaki, *Image Estimation and Uncertainty Quantification*, PhD thesis, Applied Mathematics & Statistics and Scientific Computation Program, University of Maryland, College Park, MD, 2015.