Quasi Maximum Likelihood Blind Deconvolution of Images Using Optimal Sparse Representations

Alexander Bronstein   Michael Bronstein   Michael Zibulevsky

Yehoshua Y. Zeevi

Department of Electrical Engineering,
Technion–Israel Institute of Technology,
Haifa 32000, Israel.

December 8, 2003

Abstract

A quasi maximum likelihood framework for blind deconvolution of images is presented. We generalize the relative Newton algorithm, previously proposed for quasi maximum likelihood blind source separation and blind deconvolution of time signals, and provide asymptotic analysis of its performance. Smooth approximation of the absolute value is used to model the log probability density function, which is suitable for sparse sources. In addition, we propose a method of sparsification, which allows to perform blind deconvolution of sources with arbitrary distribution, and show how to find optimal sparsifying transformations by training.

1   Introduction

Two-dimensional *blind deconvolution* (BD) is a special case of a more general problem of *image restoration*. The goal of BD is to reconstruct the original scene from an observation degraded by a linear shift invariant (LSI) system, when no or very little *a priori* information about the scene and the degradation process is available, hence the term "blind". BD is critical in many fields, such as astronomy [4, 7, 8, 40], remote sensing [37], biological and medical imaging [1, 34, 35], microscopy [25, 28, 44], etc. Typically, the image degradation is a result of imperfections of an optical system, and can be presented in terms of convolution of the source image with some *blurring kernel* or *point spread function* (PSF); in such applications, the term *deblurring* is synonymous to deconvolution.

1

A more difficult problem of deblurring is encountered in cases where the blur is caused by a spatially-varying system. However, in this case too, as long as the blurring operator can be considered to be "locally-invariant", deblurring can also be addressed by the presented methods.

1.1 Problem formulation

The general setup of 2D BD is presented in Figure 1. The observed sensor image X is created from the *source image* S passing through a convolutive system described by the impulse response W,

$$X_{mn} = \sum_{k,l} W_{kl} S_{m-k,n-l} + U_{mn},$$ (1)

and is possibly contaminated by additive sensor noise U. We assume that the action of W is invertible, at least approximately. The aim of BD is to find such *deconvolution* (*restoration*) kernel H that produces an estimate $\tilde{S}$ of S up to integer shift and scaling factor c:

$$\tilde{S}_{mn} = \sum_{k,l} H_{kl} X_{m-k,n-l} \approx c \cdot S_{m-\Delta_M, n-\Delta_N},$$ (2)

or equivalently, the *global system response* should be

$$G_{mn} = (W * H)_{mn} \approx c \cdot \delta_{m-\Delta_M, n-\Delta_N}.$$ (3)

The quality of deconvolution operation can be quantified by comparing the obtained global system response with the desired one (for this purpose, H can be assumed to be known) in sense of some norm. We use the *signal-to-interference ratio* (SIR) criteria, defined by

$$SIR = \frac{|c|^2}{\|G_{m+\Delta_M, n+\Delta_N} - c \cdot \delta_{mn}\|_2^2}$$ (4)

$$SIR_\infty = \frac{|c|^2}{\|G_{m+\Delta_M, n+\Delta_N} - c \cdot \delta_{mn}\|_\infty^2}$$ (5)

as a measure of image restoration quality. Another restoration quality measure, especially in use in image processing is the *peak signal-to-interference ratio* (PSIR) defined as

$$PSIR = \frac{\|S\|_\infty^2}{\|G_{m+\Delta_M, n+\Delta_N} - c \cdot \delta_{mn}\|_2^2}.$$ (6)
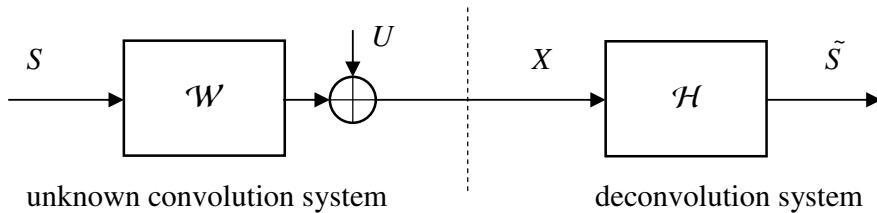
Figure 1: Schematic representation of the blind deconvolution problem. The source S passes is degraded by convolution with W and contaminated by additive noise U. The resulting observation X is then restored by the kernel H.

1.2 Previous work

Various approaches to BD and practical algorithm have been previously proposed. We will only briefly, and exhaustively, outline the basic ones here; for a comprehensive comparison see e.g. [29, 30].

A stand alone approach is the zero-sheet separation method, which is based on properties of complex functions of several variables [31]. This method is mathematically beautiful, but extremely sensitive to noise and computationally expensive.

Most of the approaches can be divided into *parametric* and *non-parametric*. In applications where the form of the PSF can be assumed in advance (e.g. motion blur or defocus), it is possible to use a parametric model of the PSF. Thus, instead of finding the PSF itself, one can try to estimate the parameters of its model. Parametric deconvolution is convenient when the PSF has large support; however, in real applications it is often difficult to derive a good model for the PSF, hence the results can be inferior compared to non-parametric methods. The advantages are, obviously, in smaller number of variables.

Secondly, blind deconvolution approaches can be divided into those trying to estimate the blurring kernel; those trying to estimate the source image and the blurring kernel simulataneously; and those trying to estimate the restoration kernel. The first class includes the so-called *a priory* blur identification methods, which first estimate the blurring kernel and then employ a non-blind deconvolution algorithm to find the source estimate [16, 18, 22, 24]. These methods usually employ a parametric model of the PSF, and suffer from mediocre performance.

The second class includes methods based on statistical or deterministic priors of the source image, the blurring kernel and the noise [20, 43]. Estimation of the source image is performed by maximizing some optimality criterion, which includes these priors. Since, the variables in this problem are both the source image and the blurring kernel, the computational complexity is a major problem.

The third class of methods usually employs maximum likelihood estimators of the restoration kernel; such estimators can incorporate priors on the image and the kernel. Since

3

there is no need to estimate the source image, these approaches demand the solution of more modest optimization problems and, consequently, are much more efficient. However, the exact source distribution, required for the ML approach is often unknown or leads to numerically untractable problems. A possible remedy is to use an approximate probability density function; such a modified ML approach is usually referred to as *quasi ML*. Quasi ML estimation techniques were successfully used in blind source separation (BSS) [13, 38, 47]. In [12], the relative Newton quasi ML framework for BD of 1D signals was introduced. In this work, we extend it for BD of images. In addition, we present a novel approach of using optimal sparse representation, which can be used for blind deconvolution of source images with arbitrary distributions.

In Section 2, we present the quasi ML approach for image deconvolution. The relative optimization and the fast relative Newton method as its particular instance is discussed in Section 3, and then, we present asymptotic performance analysis of quasi ML image deconvolution in Section 4. Section 5 is dedicated to optimal sparse representations, and we also address the possibility of obtaining optimal "sparsification" transformations by training. Simulation results on natural images are presented in Section 6.

2 Quasi maximum likelihood blind deconvolution

The likelihood of the observed signal X, given the restoration kernel $\mathcal{H}$, is

$$p(X|\mathcal{H}) \;=\; p(S = \mathcal{H}X) = p_S(\mathcal{H}X) \cdot |\det \mathcal{H}| . \tag{7}$$

in the zero-noise case. Denoting the source estimate by $Y = \mathcal{H}X$ and assuming that S is i.i.d., the likelihood $p(X|\mathcal{H})$ reduces to

$$p(X|\mathcal{H}) \;=\; \prod_{m,n} p_S(Y_{mn}) \cdot |\det \mathcal{H}| , \tag{8}$$

where $p_S(\cdot)$ stands for the source probability density function (PDF). Taking logarithm and assuming that $\mathcal{H}$ is an infinite Toeplitz block-Toeplitz operator [11] defined by the impulse response H_{mn}, the following minus-log likelihood function is obtained:

$$L(\mathcal{H}; X) \;=\; -\frac{M_X N_X}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log \left| \hat{H}(\xi, \eta) \right| \, d\xi d\eta + \sum_{m,n} \varphi(Y_{mn}), \tag{9}$$

where $M_X N_X$ is the observation sample size, $\varphi(\cdot) = -\log p_S(\cdot)$, and

$$\hat{H}(\xi, \eta) \;=\; \sum_{m,n} H_{mn} \, e^{-i(m\xi + n\eta)} \tag{10}$$

denotes the Fourier transform of H_{mn}. We will henceforth assume that H_{mn} is a FIR, supported on $[-M, ..., M] \times [-N, ..., N]$.

In the 1D case, cost functions similar to (9) were also obtained using negative joint entropy [2, 3] and information maximization [5] considerations.

4

2.1 The choice of $\varphi(\cdot)$

Natural images encountered in most applications are usually characterized by non-log-concave, multi-modal distributions, which are difficult to model and are not well-suited for optimization. However, consistent[1] estimator of S can be obtained by minimizing $L(H;X)$ even when $\varphi(\cdot)$ is not exactly equal to $-\log p_S(\cdot)$. Such *quasi ML estimation* has been shown to be practical in instantaneous BSS [27, 38, 48, 50] and BD of 1D signals [3, 12]. For example, when the source is super-Gaussian (e.g. it is sparse or sparsely representable), a smooth approximation of the absolute value function is a good choice for $\varphi(\cdot)$ [12, 23, 48, 50]. Although natural images are usually far from being sparse, they can be projected by a proper transformation into a space of sparse representation [14, 27, 48]. in Section 5 we will show how to transform general classes of natural images into sparse ones. We therefore focus our attention on modelling super-Gaussian distributions using a family of convex smooth functions

$$\varphi_\lambda(t) \;=\; |t| - \lambda \log\left(1 + \frac{|t|}{\lambda}\right), \tag{11}$$

where λ is a positive smoothing parameter [47]; $\varphi_\lambda(t) \to |t|$ as $\lambda \to 0^+$ (see Figure 2). The derivatives of $\varphi_\lambda(t)$ are

$$\varphi_\lambda'(t) \;=\; \mathrm{sign}(t) \cdot \left(1 - \frac{1}{1 + \frac{|t|}{\lambda}}\right) \tag{12}$$

$$\varphi_\lambda''(t) \;=\; \left(1 + \frac{|t|}{\lambda}\right)^{-2}, \tag{13}$$

and it can be shown that $\varphi_\lambda'(t) \to \mathrm{sign}(t)$ and $\varphi_\lambda''(t) \to \frac{1}{2\lambda}\delta(t)$ as $\lambda \to 0^+$. For convenience, we will henceforth omit λ from our notation and whenever possible, and refer to $\varphi_\lambda(\cdot)$ without using the subscript, i.e. by using $\varphi(\cdot)$.

Yet another important advantage of working with super-Gaussian sources is the fact that the asymptotic restoration error variance is significantly smaller compared to sub-Gaussian sources, i.e., deconvolution is more accurate. This issue will be addressed in Section 4.

2.2 Approximation of the log-likelihood function using the FFT

In practice, the first term of $L(H;X)$, containing the integral, is difficult to evaluate. It can, however, be approximated with any desired accuracy by [12]

$$\frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \log\left|\hat{H}(\xi,\eta)\right| d\xi d\eta \;\approx\; \frac{1}{M_F N_F} \sum_{k=0}^{M_F} \sum_{l=0}^{N_F} \log\left|\hat{H}_{kl}\right|, \tag{14}$$

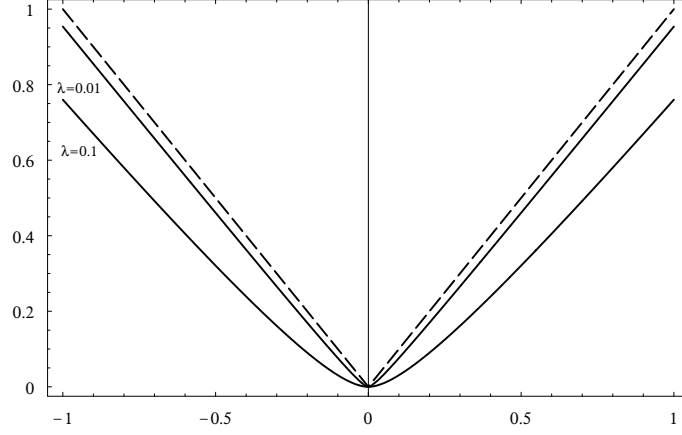[1]Consistency is discussed in Section 4.2

Figure 2: The smooth approximation of the absolute value φ_λ for different values of λ. Dashed lines show the limit $\lambda \to 0^+$.

where

$$\hat{H}_{kl} = \mathcal{F}_{M_F, N_F} \{H_{mn}\}_{kl} = \hat{H}\left(\frac{2\pi k}{M_F}, \frac{2\pi l}{N_F}\right) \tag{15}$$

are the 2D DFT coefficients of H_{mn}, zero-padded to $M_F \times N_F$. The approximation error vanishes as M_F, N_F grow to infinity. Choosing M_F and N_F as integer powers of 2, allows to use 2D FFT. For convenience, we denote

$$f_1 = \sum_{k=0}^{M_F}\sum_{l=0}^{N_F}\log\left|\hat{H}_{kl}\right|^2 \tag{16}$$

$$f_2 = \sum_{m,n}\varphi(Y_{mn}), \tag{17}$$

and define the approximate normalized minus-log likelihood function as

$$L(H; X) = -\frac{1}{2M_F N_F} \cdot f_1(H) + \frac{1}{M_X N_X} \cdot f_2(Y), \tag{18}$$

where $Y_{mn} = (H * X)_{mn}$.

2.3   Gradient and Hessian of $L(H; X)$

Optimization algorithms discussed in Section 3.3 require the knowledge of the gradient and the Hessian of $L(H; X)$. Since the optimization variable H is a $(2M + 1) \times (2N + 1)$ matrix, the gradient ∇L is also a $(2M + 1) \times (2N + 1)$ matrix, whereas the Hessian $\nabla^2 L$ is

a $(2M + 1) \times (2N + 1) \times (2M + 1) \times (2N + 1)$ fourth-order tensor. For convenience, we parse the variables column-wise into a $(2M + 1)(2N + 1) \times 1$ vector

$$h \;=\; \text{vec}(H) = \left[\; H_{-M,-N}, H_{-M+1,-N}, ..., H_{M,-N}, H_{-M,-N+1}, ..., H_{M,N} \;\right]^T, \quad (19)$$

and define the gradient and the Hessian of $L(H; X)$ as a $(2M + 1)(2N + 1) \times 1$ vector and a $(2M + 1)(2N + 1) \times (2M + 1)(2N + 1)$ matrix, respectively.

The gradient of f_1 is given by

$$\nabla f_1 \;=\; \text{vec}\left(Q'_{kl} + Q'^{*}_{kl}\right), \quad (20)$$

and the i-th row of the Hessian of f_1 is given by

$$\left(\nabla^2 f_1\right)_i \;=\; \text{vec}\left(Q''_{k+k',l+l'} + Q''^{*}_{k+k',l+l'}\right), \quad (21)$$

where

$$Q'_{kl} \;=\; \mathcal{F}_{M_F, N_F}\left\{\hat{H}^{-1}_{mn}\right\}_{kl}$$

$$Q''_{k+k',l+l'} \;=\; -\mathcal{F}_{M_F, N_F}\left\{\hat{H}^{-2}_{mn}\right\}_{k+k',l+l'}, \quad (22)$$

and $k' = (i - 1) \mod (2M + 1) - M$ and $l' = \lfloor\frac{i-1}{2M+1}\rfloor - N$. The gradient of f_2 is given by

$$\nabla f_2 \;=\; \text{vec}\left((\Phi' * \mathcal{J}X)_{kl}\right) \quad (23)$$

and the i-th row of the Hessian of f_2 is given by

$$\left(\nabla^2 f_2\right)_i \;=\; \text{vec}\left(\left(A^{k'l'} * \mathcal{J}X\right)_{kl}\right), \quad (24)$$

where $\Phi'_{mn} = \varphi'(Y_{mn})$, $A^{k'l'}_{mn} = \varphi''(Y_{mn}) \cdot X_{m-k',n-l'}$, $(\mathcal{J}X)_{mn} = X_{M_X-m,N_X-n}$, $k' = (i - 1) \mod (2M + 1) - M$, and $l' = \lfloor\frac{i-1}{2M+1}\rfloor - N$. For derivation see Appendices A.1–A.2. Computational complexities of the functions f_1 and f_2, their gradients and Hessians are given in Table 1.

3   Relative Newton algorithm

A fast relative optimization algorithm for BSS based on the Newton method, was introduced in [47] as a modification of the approach in [38]. This method was extended in [12] to BD of time series. Here, we extend these results for blind deconvolution of images. We first present the general relative optimization algorithm. Next, the *relative Newton* algorithm, using a Newton step in the relative optimization framework is introduced. A fast version of the relative Newton step is addressed in Section 3.3.

Term	Computational complexity
f_1	$M_F N_F \log_2 M_F N_F + k_L M_F N_F$
f_2	$(k+1) M_X N_X + 4 M_X N_X \log_2 M_X N_X$
∇f_1	$M_F N_F \log_2 M_F N_F$
∇f_2	$(k'+1) M_X N_X + 4 M_X N_X \log_2 M_X N_X$
$\nabla^2 f_1$	$M_F N_F \log_2 M_F N_F$
$\nabla^2 f_2$	$k'' M_X N_X + (2M+1)(2N+1) \left[4 M_X N_X \log_2 M_X N_X + M_X N_X \right]$

Table 1: Computational complexity of f_1 (16), f_2 (17), and of their gradients and Hessians. The constants k_L, k, k' and k'' stand for the complexity of $log| \cdot |^2$, $\varphi(\cdot)$, $\varphi'(\cdot)$ and $\varphi''(\cdot)$, respectively.

3.1 Relative optimization algorithm

The main idea of relative optimization is to iteratively produce source estimate and use it as the current observation. This yields the following algorithm:

Relative optimization algorithm

1. Start with an initial estimate of the restoration kernel $H^{(0)}$, and with $X^{(0)} = X$.

2. **For** $k = 1, 2, ...$, until convergence

3. Compute current source estimate: $X^{(k)} = H^{(k-1)} * X$.

4. Starting with $V_{mn}^{(k)} = \delta_{mn}$, compute coefficients of the restoration kernel, producing one or few steps of a conventional optimization method, which sufficiently decrease the objective function $L(H = V^{(k)}; X^{(k)})$.

5. Update current restoration kernel estimate: $H^{(k)} = V^{(k)} * H^{(k-1)}$.

6. **End For**

This method allows to construct large restoration kernels of the form

$$H = H^{(0)} * H^{(1)} * ... * H^{(K-1)} \tag{25}$$

using a set of relatively low-order factors. The algorithm assumes infinite memory and produce a restoration kernel of order growing at each iteration. In real applications it might be necessary to limit the support of the restoration kernel. This can be done by cropping the kernel obtained in Step 5.

Unfortunately, unlike in the 1D case, reducible kernels of the form (25) constitute a less general class of kernels than all the bivariate polynomials of order $K(2M + 1) \times K(2N + 1)$ of the form

$$H(z_1, z_2) = \sum_{m=-KM}^{KM} \sum_{n=-KN}^{KN} H_{mn} \, z_1^{-m} z_2^{-n}. \tag{26}$$

A general bivariate polynomial is "almost surely" *irreducible*, i.e. cannot be factorized into simpler factors. However, empirical observations show that kernels occurring in real applications are far from being arbitrary bivariate polynomials and can be approximated as a product of several relatively simple factors, as in (25).

Another remarkable property of the relative optimization algorithm is its equivariance, stated in the following proposition:

Proposition 1 *The relative optimization algorithm is equivariant, i.e. its step at iteration k depends only on $G^{(k-1)} = W * H^{(k-1)}$.*

Proof: Straightforward, since Step 4 and the update in Step 5 do not depend explicitly on W, but on the currents global system response [3, 12]. □

Equivariance implies that for any invertible kernel A, the estimator $\tilde{H}(X)$ of the restoration kernel H given the observation X, obtained by minimization of the target function $L(H; X)$ obeys [17]

$$\tilde{H}(A * X) = A^{-1} * \tilde{H}(X), \tag{27}$$

i.e., the parameters to be estimated (in our case, coefficients H_{mn} of the restoration kernel) form a group. This is indeed the case for invertible kernels with the convolution operation. It must be noted, however, that when the restoration filter support is limited by cropping, equivariance holds only approximately.

Proposition 2 *The sequence $L(H^{(k)}; X)$ of function values, produced by the relative optimization algorithm, is monotonically decreasing.*

Proof: Let us first observe that for $H = A * B$, the DFT coefficients in (15) are $\hat{H}_{kl} = \hat{A}_{kl} \cdot \hat{B}_{kl}$. Consequently,

$$f_1(A * B) = \sum_{k,l} \log \left| \hat{A}_{kl} \right|^2 + \log \left| \hat{B}_{kl} \right|^2 = f_1(A) + f_1(B). \tag{28}$$

Next, let us observe that for $H_{mn} = \delta_{mn}$, $\hat{H}_{kl} = 1$, hence $f_1(\delta_{mn}) = 0$. Therefore, for two consecutive iterations of the relative optimization algorithm, we have

$$
\begin{aligned}
L(H^{(k)}; X) &= -\frac{f_1(H^{(k)})}{2M_F N_F} + \frac{f_2(H^{(k)} * X)}{M_X N_X} \\
&= -\frac{f_1(H^{(k)})}{2M_F N_F} + L(\delta_{mn}; X^{(k+1)}) \qquad (29) \\
L(H^{(k+1)}; X) &= -\frac{f_1(H^{(k+1)})}{2M_F N_F} + \frac{f_2(H^{(k+1)} * X)}{M_X N_X} \\
&= -\frac{f_1(V^{(k+1)} * H^{(k)})}{2M_F N_F} + \frac{f_2(V^{(k+1)} * H^{(k)} * X)}{M_X N_X} \\
&= -\frac{f_1(H^{(k)})}{2M_F N_F} + L(V^{(k+1)}; X^{(k+1)}). \qquad (30)
\end{aligned}
$$

Since Step 4 decreases the value of L,

$$
L(H^{(k)}; X) - L(H^{(k+1)}; X) = L(\delta_{mn}; X^{(k+1)}) - L(V^{(k+1)}; X^{(k+1)}) > 0, \qquad (31)
$$

which means that the sequence $L(H^{(k)}; X)$ is monotonically decreasing. ∎

3.2 Newton method

Newton method often provides very fast (quadratic) rate of convergence. We will first consider the standard Newton method; later we will see how its use in the relative optimization framework allows to overcome the difficulty resulting from the computationally expensive Newton iterations. In the standard Newton approach, the direction d at each iteration is given by solution of the linear system [9]

$$
\nabla^2 L \cdot d = -\nabla L. \qquad (32)
$$

Since the objective function is non-convex, in order to guarantee descent direction, positive definiteness of the Hessian is forced by using modified Cholessky factorization, which finds such a diagonal matrix R, that the matrix $\nabla^2 L + R$ is positive definite, and provides a solution to the modified system

$$
(\nabla^2 L + R)\, d = -\nabla L. \qquad (33)
$$

This requires about $\frac{1}{6}(2M + 1)^3(2N + 1)^3 + (2M + 1)^2(2N + 1)^2$ operations [9]. Having the direction d, the new iterate $h^{(k+1)}$ is given by

$$
h^{(k+1)} = h^{(k)} + \alpha^{(k)} d, \qquad (34)
$$

10

where $\alpha^{(k)}$ is the step size determined by either exact line search

$$\alpha = \operatorname{argmin} L(h^{(k)} + \alpha^{(k)} d; X), \tag{35}$$

or by backtracking line search:

Backtracking line search algorithm

1. $\alpha = 1$

2. **While** $L(h + \alpha d; X) > L(h; X) + \beta \alpha \nabla L(h; X)^T d$ **or** $\log |\det \mathcal{H}| > -\infty$

3. $\alpha = \gamma \alpha$

4. **End While**,

where β and γ are constants, and $\mathcal{H}$ is the cropped Toeplitz-block Toeplitz matrix corresponding to the restoration kernel coefficients h. The use of line search guarantees monotonic decrease of the objective function at every iteration. The condition on $\log |\det \mathcal{H}|$ guarantees that no search will be performed beyond the subspace, where the inverse of the restoration kernel is stable. In our implementation, we used the backtracking line search with $\beta = \gamma = 0.3$. It should be noted that when the gradient norm becomes very small (say, below 10^{-5}), computational inaccuracies may make the line search inefficient. For this reason, we used the Newton direction as is (i.e. chose $\alpha = 1$) when the gradient norm fell below 10^{-5}.

Newton method can be used as is to find the restoration kernel that minimizes $L(H; X)$. Another possibility is to use Newton iteration in Step 4 of the relative optimization algorithm [12, 13, 47]. The latter possibility is advantageous, since it allows to construct a large restoration kernel using relatively small factors. This, in turn, implies solution of smaller optimization problems. Relative optimization algorithm using the Newton step will be termed henceforth as *relative Newton* method.

3.3   Fast relative Newton step

Practical use of the relative Newton step is limited to small values of M, N and M_X, N_X, due to the complexity of Hessian construction, and solution of the Newton system. This complexity can be significantly reduced if special Hessian structure at the minimum is exploited.

Proposition 3 *For $X = c \cdot S$, the Hessian of $L(H; X)$ at $H = \delta_{mn}$ is approximately given by*

$$\nabla^2 L(\delta_{mn}; c \cdot S) \approx \begin{pmatrix} \ddots & & & & & \cdot^{\cdot^{\cdot}} \\ & \gamma \sigma'^2 & & & 1 & \\ & & \alpha c^2 + 1 & & \\ & 1 & & \gamma \sigma'^2 & \\ \cdot^{\cdot^{\cdot}} & & & & & \ddots \end{pmatrix}, \tag{36}$$

where c is the restoration kernel scaling factor, $\sigma^2 = I\!E S^2$, $\sigma'^2 = (c\sigma)^2$, $\alpha = c^2 \cdot I\!E \varphi''(c \cdot S) S^2$, $\gamma = I\!E \varphi''(c \cdot S)$. When $\gamma \sigma'^2 \gg 1$, $\nabla^2 L(H = \delta_{mn}; c \cdot X)$ is approximately diagonal. When $\gamma \sigma'^2 \ll 1$, $\nabla^2 L(H = \delta_{mn}; c \cdot X)$ has an approximate anti-diagonal form.

Proof: See Appendix A.3.

Hessian structure is visualized in Figure 3 for different ranges of $\gamma \sigma'^2$. When $\gamma^2 \sigma'^4 < 1$, the Hessian at $X = c \cdot S$, $H = \delta_{mn}$ is no more positive semi-definite. This case is addressed in Section 4.2, where we show that the obtained estimator is asymptotically unstable.

Using the diagonal approximation, which is valid for $\gamma \sigma'^2 \gg 1$, the Newton system (32) can be solved as a set of $(2M + 1)(2N + 1)$ independent linear equations

$$d_k = -\frac{(\nabla L)_k}{(\nabla^2 L)_{kk}}, \tag{37}$$

for $k = 1, ..., (2M + 1)(2N + 1)$. In order to guarantee decent direction and avoid saddle points, we force positive definiteness of the Hessian by forcing small diagonal elements to be above some positive threshold, say, $\epsilon = 10^{-8} \cdot \max \{|(\nabla^2 L)_{kk}|\}$:

$$d_k = -\frac{(\nabla L)_k}{\max \{|(\nabla^2 L)_{kk}|, \epsilon\}}. \tag{38}$$

For $\gamma \sigma'^2 \sim 1$, the diagonal-anti-diagonal approximation of the Hessian should be used, which allows to reduce Newton system solution to regularized solution of a set of 2×2 systems of the form

$$D_k \cdot d_k = \begin{pmatrix} (\nabla^2 L)_{kk} & 1 \\ 1 & (\nabla^2 L)_{K-k, K-k} \end{pmatrix} \cdot \begin{pmatrix} d_k \\ d_{K-k} \end{pmatrix} = \begin{pmatrix} -(\nabla L)_k \\ -(\nabla L)_{K-k} \end{pmatrix}, \tag{39}$$

and an additional 1×1 system

$$(\nabla^2 L)_{\frac{K}{2}} \cdot d_{\frac{K}{2}} = -(\nabla L)_{\frac{K}{2}}. \tag{40}$$

Regularization is performed by forcing positive definiteness of each of the 2×2 submatrices D_k in (39) by inverting the sign of negative eigenvalues and forcing small eigenvalues to be larger than some positive threshold. This can be done by using either analytic expressions for eigendecomposition of D_k [12, 47], or numerical procedures.

When the diagonal or the diagonal-anti-diagonal approximations are used, fast relative Newton algorithm requires about $(k'' + 1) M_X N_X + 4 M_X N_X \log_2 M_X N_X$ operations for approximate Hessian construction, which is of the same order as gradient computation. Additional $(2M + 1)(2N + 1)$ operations are required for approximate Hessian inversion in case of diagonal approximation, and slightly more in case of the diagonal-anti-diagonal approximation. This is compared to $k'' M_X N_X + (2M + 1)(2N + 1) [4 M_X N_X \log_2 M_X N_X + M_X N_X]$ operations for exact Hessian evaluation and additional $\frac{1}{6}(2M + 1)^3 (2N + 1)^3 + (2M + 1)^2 (2N + 1)^2$ computations for exact Newton system solution required for the full relative Newton step.
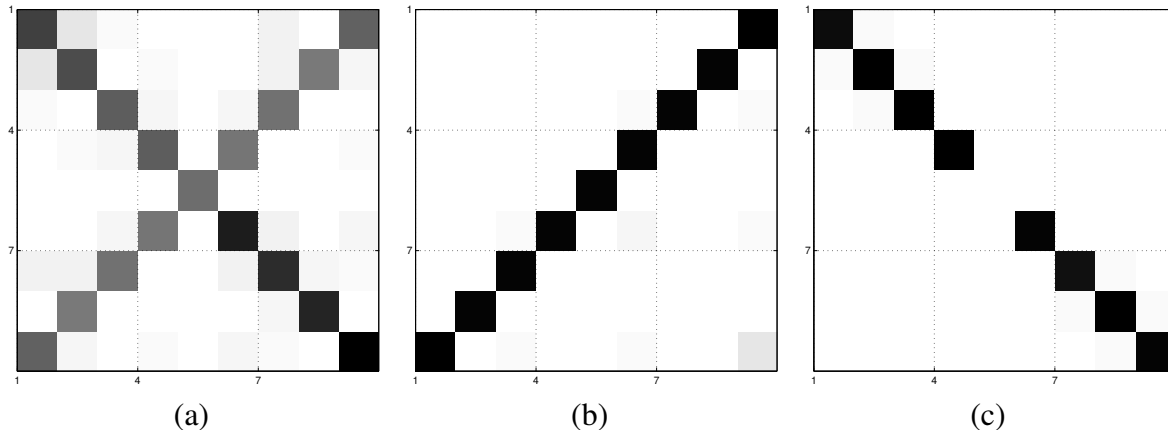
Figure 3: Hessian structure at $H_{mn} = \delta_{mm}$ for $M = N = 1$: (a) diagonal-anti-diagonal form for $\gamma\sigma'^2 \approx 10$; (b) anti-diagonal form for $\gamma\sigma'^2 \approx 10^{-6}$; (c) diagonal form for $\gamma\sigma'^2 \approx 10^6$. White stands for near-zero elements of the matrix.

3.4 Simulation results

Performance of the fast relative Newton algorithm was compared to the full Newton method in a simulation. A 101×101 Gauss-Bernoully (sparse normally) distributed i.i.d. image with $\rho = 0.2$ (see Appendix B) was used as the source. The image was convolved with a 3×3 FIR kernel with slowly-decaying inverse (see Figure 5). Full Newton and fast relative Newton (with the diagonal Hessian approximation) were used to estimate the inverse kernel. 3×3, 5×5, 7×7, and 9×9 restoration kernels were used. The smoothing parameter was set to $\lambda = 10^{-2}$. Optimization was terminated when $\|\nabla L\|$ reached 10^{-10}. Gradient norms, SIR and SIR$_\infty$ were measured as a function of CPU time[2] and iteration number.

Convergence of both algorithms can be seen in Figure 4. In the specific experiment, fast relative Newton showed about 10 times faster convergence in terms of SIR, compared with the full Newton step. For the same values of M, N, the obtained restoration quality of the fast relative Newton algorithm, compared to the full Newton step, was better by about 2–5 dB (in terms of SIR and SIR$_\infty$), since the effective restoration kernel was of higher order. Figures 6– 7 depict the restoration kernels of different sizes obtained by both algorithms.

4 Asymptotic performance analysis

Asymptotic performance of maximum-likelihood parameter estimation in blind system identification and deconvolution problems was studied in many previous studies (see, for exam-

---

[2]All algorithms were implemented in MATLAB and executed on an ASUS portable computer with Intel Pentium IV Mobile processor and 640MB RAM.
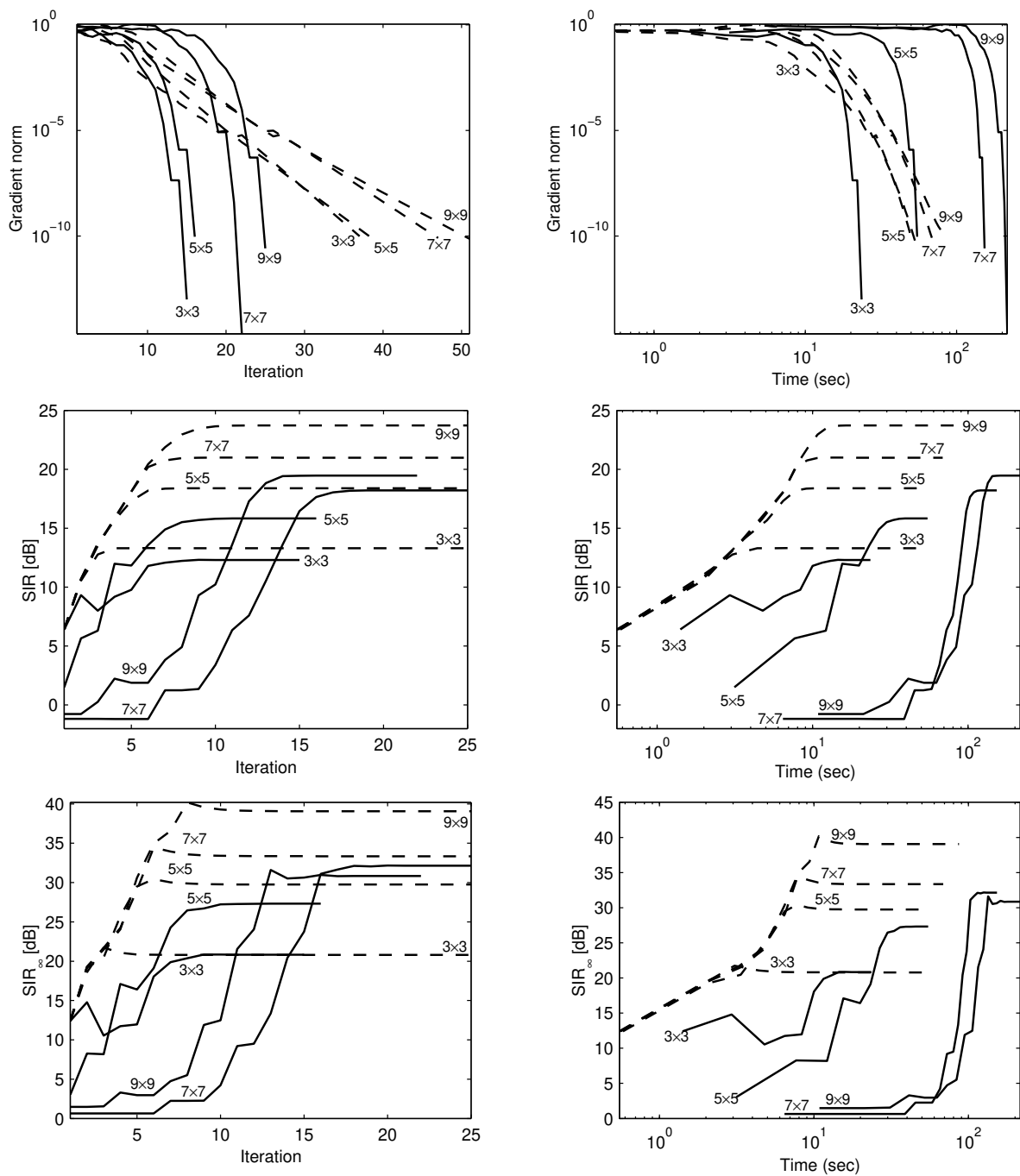
Figure 4: Convergence of the Newton method (solid) and the fast relative Newton method (dashed) for different restoration kernel sizes (indicated on the plots).
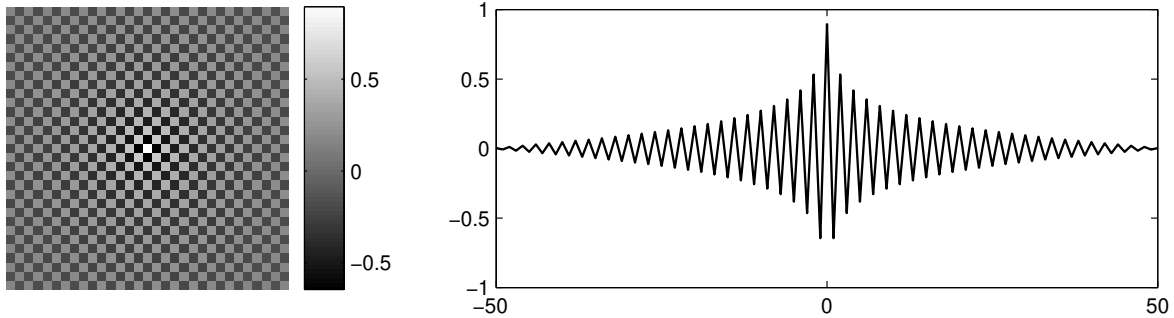
14

Figure 5: True restoration kernel impulse response (cropped) and its central 1D slice.

ple, [6, 17, 41, 42, 45, 46]). In all these studies, the Cramér-Rao lower bound (CRLB) for the system parameters are found, and lower bounds on signal reconstruction quality are derived. However, the presented quasi ML approach does not achieve the CRLB, since $\varphi(S) \neq -\log p_S(S)$. In this section, we derive an asymptotic restoration error estimate for the quasi ML BD method. We will first find the asymptotic error covariance matrix and use it in order to estimate the restoration SIR. We will show that in case of the true ML estimator, these results coincide with the previously reported CRLB. We also address the asymptotic stability conditions for the quasi ML estimator and present numerical examples of the generalized Laplacian distribution. Finally, we compare the predicted restoration SIR with the true one, obtained from minimization of $L(H; X)$.

4.1 Asymptotic error covariance matrix

Let us assume that the source S is zero-mean i.i.d, and the restoration kernel $\mathcal{H}$ is estimated by minimizing the minus log likelihood function $L(H; X)$ defined in (9), where the real minus log PDF is replaced by some other function $\varphi(S)$. We will denote the vector of parameters of $\mathcal{H}$, i.e. the kernel coefficients H_{mn} as $h = \text{vec}(H)$, and assume that $\mathcal{H}$ has enough degrees of freedom to accurately approximate the inverse of $\mathcal{W}$. We also assume that $\gamma = \mathbf{E}\varphi''(c \cdot S)$, $\sigma^2 = \mathbf{E}S^2$, $\sigma'^2 = (c\sigma)^2$, $\alpha = \mathbf{E}\varphi''(c \cdot S)S^2$, $\beta = \mathbf{E}\varphi'^2(c \cdot S)$, $\zeta = \mathbf{E}\varphi'(c \cdot S)S$ and $\theta = \mathbf{E}\varphi'^2(c \cdot S)S^2$ exist and are bounded. [3]

Let $H^* = c \cdot W^{-1}$ be the exact restoration kernel (up to a scaling factor). It can be shown that H^* satisfies [27]

$$H^* = \text{argmin } \mathbf{E}_X L(H; X). \tag{41}$$

Let $\bar{H}$ be the estimate of the exact restoration kernel H^* based on the finite realization of the

---

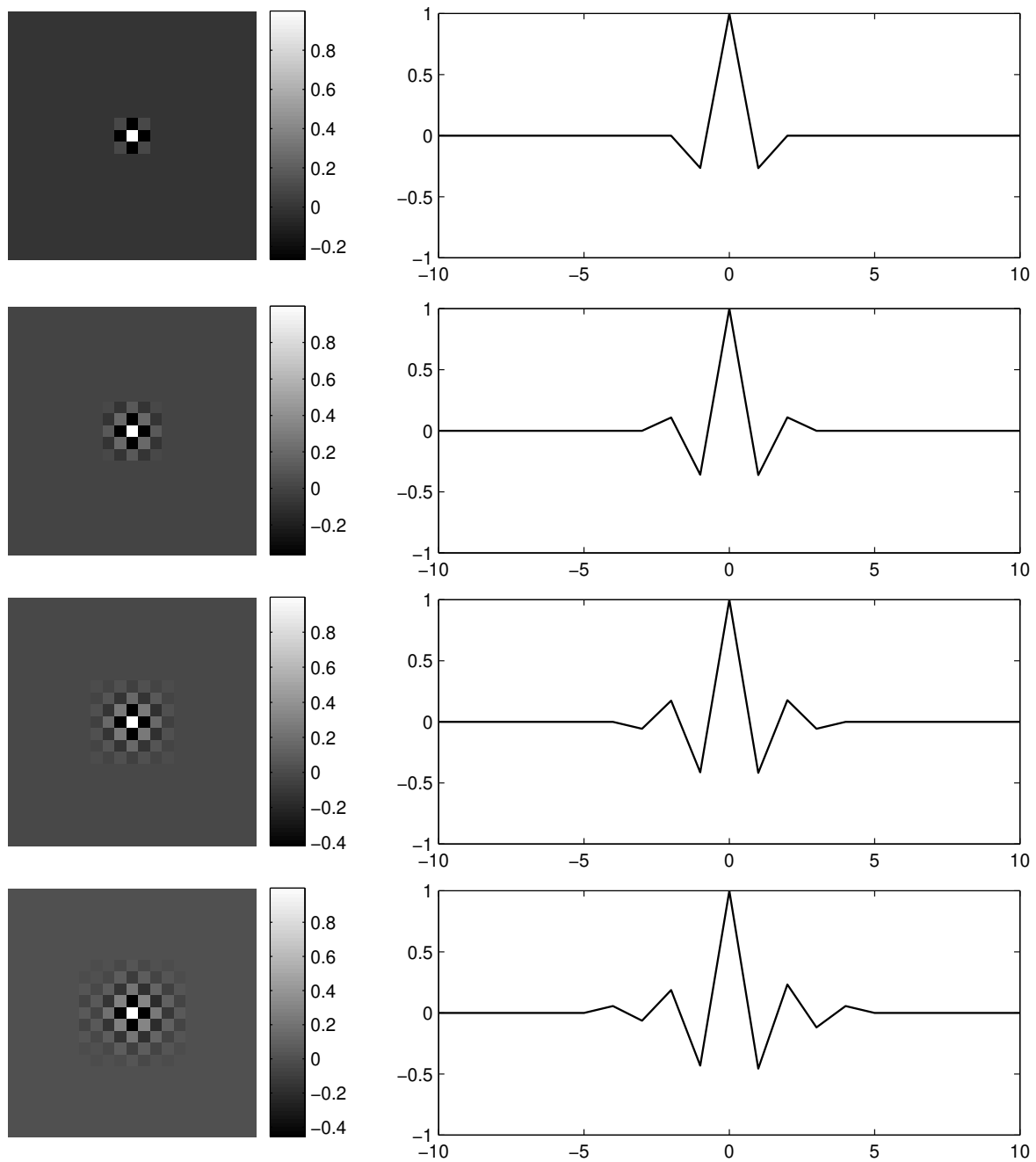[3]Note that the expected values are computed with respect to the real PDF os S.

Figure 6: Restoration kernels and their one-dimensional central slices obtained by the Newton algorithm with different values of M, N. From top to bottom: 3×3, 5×5, 7×7, and 9×9 estimated restoration kernels.
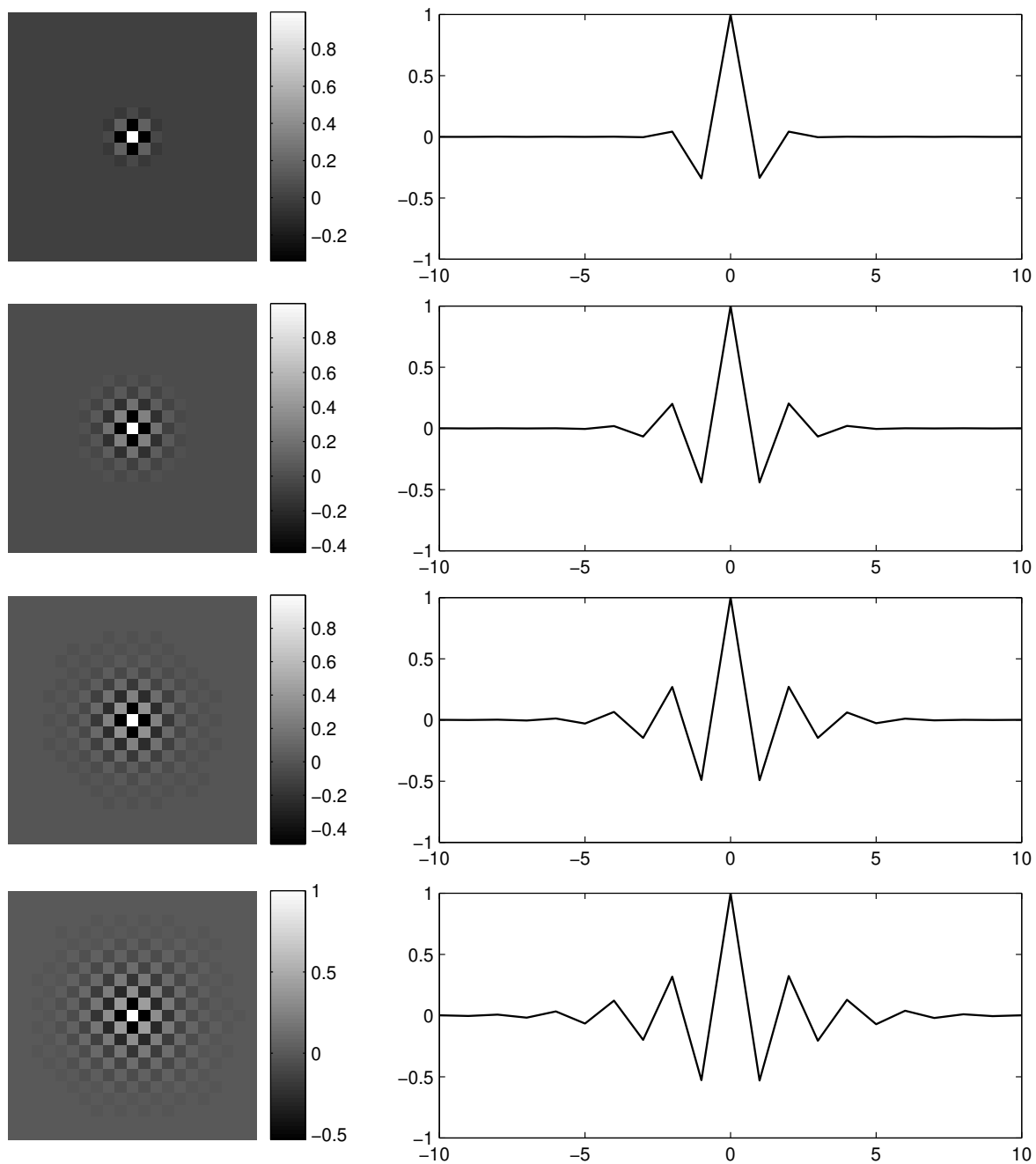
Figure 7: Restoration kernels and their one-dimensional central slices obtained by the fast relative Newton algorithm with different values of M, N. From top to bottom: 3×3, 5×5, 7×7, and 9×9 restoration kernels.

17

data X,

$$\bar{H} = \operatorname{argmin} L(H; X). \tag{42}$$

Note that $\nabla L(\bar{H}; X) = 0$, whereas $\nabla L(H^*; X) \neq 0$; yet $\mathbf{E}\{\nabla L(H^*; X)\} = 0$. Denote the estimation error as $\Delta H = H^* - \bar{H}$, and $h^* = \operatorname{vec}(H^*)$, $\tilde{h} = \operatorname{vec}(\tilde{H})$, $\Delta h = \operatorname{vec}(\Delta H)$. Then, assuming $\|\Delta H\|$ is small, second-order Taylor expansion yields

$$\begin{aligned} \nabla L(H^*; X) &\approx \nabla^2 L(H^*; X) \cdot \left(h^* - \bar{h}\right) \\ &= \nabla^2 L(H; X)\big|_{H = c \cdot W^{-1}} \cdot \left(h^* - \bar{h}\right). \end{aligned} \tag{43}$$

Due to equivariance, the former relation can be rewritten as

$$\nabla L(\delta_{mn}; H^* * X) \approx \nabla^2 L(\delta_{mn}; H^* * X) \cdot \left(h^* - \bar{h}\right). \tag{44}$$

Since $H^* = c \cdot W^{-1}$, we can substitute $H^* * X = c \cdot S$, and obtain

$$\nabla L(\delta_{mn}; c \cdot S) \approx \nabla^2 L(\delta_{mn}; c \cdot S) \cdot \left(h^* - \bar{h}\right), \tag{45}$$

or, alternatively,

$$\Delta h = h^* - \bar{h} \approx \left[\nabla^2 L(\delta_{mn}; c \cdot S)\right]^{-1} \cdot \nabla L(\delta_{mn}; c \cdot S). \tag{46}$$

For convenience, we will denote $\nabla^2 L(\delta_{mn}; c \cdot S)$ as $\nabla^2 L$. The covariance matrix of Δh is therefore given by

$$\begin{aligned} \Sigma_{\Delta h} &= \mathbf{E}(h^* - \bar{h})(h^* - \bar{h})^T \\ &\approx \left[\nabla^2 L\right]^{-1} \cdot \mathbf{E}\left\{\nabla L \nabla L^T\right\} \cdot \left[\nabla^2 L\right]^{-T} \\ &= \left[\nabla^2 L\right]^{-1} \cdot \Sigma_{\nabla L} \cdot \left[\nabla^2 L\right]^{-1}. \end{aligned} \tag{47}$$

Using the asymptotic Hessian structure (36) from Proposition 3, allows to split the problem of finding $\Sigma_{\Delta h}$ into a set of symmetric 2×2 problems of the form

$$\Sigma_{\nabla L}^{(kl)} = \begin{pmatrix} \mathbf{E}G_{-k,-l}^2 & \mathbf{E}\{G_{-k,-l} \cdot G_{kl}\} \\ \mathbf{E}\{G_{-k,-l} \cdot G_{kl}\} & \mathbf{E}G_{kl}^2 \end{pmatrix} \tag{48}$$

$$\begin{aligned} \Sigma_{\Delta H}^{(kl)} &= \begin{pmatrix} \mathbf{E}\Delta H_{-k,-l}^2 & \mathbf{E}\{\Delta H_{-k,-l} \cdot \Delta H_{kl}\} \\ \mathbf{E}\{\Delta H_{-k,-l} \cdot \Delta H_{kl}\} & \mathbf{E}\Delta H_{kl}^2 \end{pmatrix} \\ &\approx \begin{pmatrix} \gamma\sigma'^2 & 1 \\ 1 & \gamma\sigma'^2 \end{pmatrix}^{-1} \Sigma_{\nabla L}^{(kl)} \begin{pmatrix} \gamma\sigma'^2 & 1 \\ 1 & \gamma\sigma'^2 \end{pmatrix}^{-1} \\ &= \frac{1}{(\gamma^2\sigma'^4 - 1)^2} \begin{pmatrix} \gamma\sigma'^2 & -1 \\ -1 & \gamma\sigma'^2 \end{pmatrix} \Sigma_{\nabla L}^{(kl)} \begin{pmatrix} \gamma\sigma'^2 & -1 \\ -1 & \gamma\sigma'^2 \end{pmatrix}, \end{aligned} \tag{49}$$

18

where

$$G_{kl} = \frac{\partial L}{\partial H_{kl}}, \tag{50}$$

and an additional 1×1 problem

$$
\begin{aligned}
\Sigma_{\nabla L}^{(00)} &= \mathbb{E} G_{00}^2 \\
\Sigma_{\Delta H}^{(00)} &= \frac{\Sigma_{\nabla L}^{(00)}}{\left(\alpha c^2 + 1\right)^2}.
\end{aligned}
\tag{51}
$$

That is, the asymptotic error covariance matrix has a digaonal-anti-diagonal form.

Let us evaluate the 2×2 gradient covariance matrix $\Sigma_{\nabla L}^{(kl)}$ for $(k, l) \neq (0, 0)$. From Appendices A.1–A.2,

$$G_{kl} = \frac{\partial L(\delta_{mn}; c \cdot S)}{\partial H_{kl}} = -\delta_{kl} + \frac{c}{M_X N_X} \sum_{m,n} \varphi'(c \cdot S_{mn}) S_{m-k,n-l}, \tag{52}$$

which for $(k, l) \neq (0, 0)$ reduces to

$$G_{kl} = \frac{c}{M_X N_X} \sum_{m,n} \varphi'(c \cdot S_{mn}) S_{m-k,n-l}. \tag{53}$$

Taking the expectation w.r.t. S and neglecting second-order terms, we obtain

$$
\begin{aligned}
\mathbb{E}\left\{G_{kl}^2\right\} &= \frac{c^2}{(M_X N_X)^2} \sum_{m,n,m',n'} \mathbb{E}\left\{\varphi'(c \cdot S_{mn})\varphi'(c \cdot S_{m'n'})\, S_{m-k,n-l} S_{m'-k,n'-l}\right\} \\
&\approx \frac{c^2}{M_X N_X} \cdot \mathbb{E}\varphi'^{\,2}(c \cdot S) \cdot \mathbb{E} S^2 = \frac{1}{M_X N_X} \cdot \beta(\sigma c)^2 = \frac{1}{M_X N_X} \cdot \beta\sigma'^2 \\
\mathbb{E}\left\{G_{-k,-l} \cdot G_{kl}\right\} &= \frac{c^2}{(M_X N_X)^2} \sum_{m,n,m',n'} \mathbb{E}\left\{\varphi'(c \cdot S_{mn})\varphi'(c \cdot S_{m'n'})\, S_{m+k,n+l} S_{m'-k,n'-l}\right\} \\
&\approx \frac{c^2}{M_X N_X} \cdot \mathbb{E}^2 \varphi'(c \cdot S)S = \frac{1}{M_X N_X} \cdot c^2 \zeta^2,
\end{aligned}
\tag{54}
$$

that is,

$$\Sigma_{\nabla L}^{(kl)} \approx \frac{1}{M_X N_X} \cdot \begin{pmatrix} \beta\sigma'^2 & c^2\zeta^2 \\ c^2\zeta^2 & \beta\sigma'^2 \end{pmatrix}, \tag{55}$$

where $\beta = \mathbb{E}\varphi'^{\,2}(c \cdot S)$, $\zeta = \mathbb{E}\varphi'(c \cdot S)S$.

19

Substituting the former result to (49) yields

$$
\Sigma_{\Delta H}^{(kl)} \approx \frac{1}{M_X N_X \, (\gamma^2 \sigma'^4 - 1)^2} \begin{pmatrix} \gamma\sigma'^2 & -1 \\ -1 & \gamma\sigma'^2 \end{pmatrix} \begin{pmatrix} \beta\sigma'^2 & c^2\zeta^2 \\ c^2\zeta^2 & \beta\sigma'^2 \end{pmatrix} \begin{pmatrix} \gamma\sigma'^2 & -1 \\ -1 & \gamma\sigma'^2 \end{pmatrix}
$$

$$
= \frac{1}{M_X N_X} \begin{pmatrix} \frac{\beta\sigma'^2 \left(\gamma^2\sigma'^4+1\right)-2\gamma\sigma'^2 c^2\zeta^2}{\left(\gamma^2\sigma'^4-1\right)^2} & \frac{c^2\zeta^2+\gamma\sigma'^2\left(\gamma\sigma'^2 c^2\zeta^2-2\beta\sigma'^2\right)}{\left(\gamma^2\sigma'^4-1\right)^2} \\ \frac{c^2\zeta^2+\gamma\sigma'^2\left(\gamma\sigma'^2 c^2\zeta^2-2\beta\sigma'^2\right)}{\left(\gamma^2\sigma'^4-1\right)^2} & \frac{\beta\sigma'^2\left(\gamma^2\sigma'^4+1\right)-2\gamma\sigma'^2 c^2\zeta^2}{\left(\gamma^2\sigma'^4-1\right)^2} \end{pmatrix}, \tag{56}
$$

that is, asymptotically

$$
\operatorname{var}\{\Delta H_{kl}\} \approx \frac{\beta\sigma'^2 \left(\gamma^2\sigma'^4 + 1\right) - 2\gamma\sigma'^2 c^2\zeta^2}{M_X N_X \left(\gamma^2\sigma'^4 - 1\right)^2} \tag{57}
$$

for $(k, l) \neq (0, 0)$. Note that the asymptotic variance depends on the sample size $M_X N_X$ and on parameters β, γ, ζ, c and σ'^2, which depend on the source distribution and $\varphi(\cdot)$ only.

For $k = l = 0$, one has

$$
\mathbb{E}G_{00} = -1 + c \cdot \mathbb{E}\varphi'(c \cdot S)S = c \cdot \zeta - 1. \tag{58}
$$

Demanding $\mathbb{E}G_{00} = 0$, we obtain the following condition:

$$
c \cdot \mathbb{E}\varphi'(c \cdot S)S = 1, \tag{59}
$$

from where the scaling factor c can be found. Neglecting second-order terms, the second moment of G_{00} is given by

$$
\begin{aligned}
\mathbb{E}G_{00}^2 &\approx -1 - 2c \cdot \mathbb{E}\varphi'(c \cdot S)S + c^2 \cdot \mathbb{E}^2\varphi'(c \cdot S)S \\
&\quad + \frac{c^2}{M_X N_X} \cdot \left[\mathbb{E}\varphi'^{\,2}(c \cdot S)S^2 - \mathbb{E}^2\varphi'(c \cdot S)S\right] \\
&= (c \cdot \zeta - 1)^2 + \frac{c^2(\vartheta - \zeta^2)}{M_X N_X} = \frac{c^2(\vartheta - \zeta^2)}{M_X N_X},
\end{aligned} \tag{60}
$$

where $\vartheta = \mathbb{E}\varphi'^{\,2}(c \cdot S)S^2$. Hence,

$$
\Sigma_{\nabla L}^{(00)} \approx \frac{c^2(\vartheta - \zeta^2)}{M_X N_X}. \tag{61}
$$

Substituting $\Sigma_{\nabla L}^{(00)}$ into (51) yields

$$
\operatorname{var}\{\Delta H_{00}\} \approx \frac{1}{M_X N_X} \cdot \frac{c^2(\vartheta - \zeta^2)}{(\alpha c^2 + 1)^2}. \tag{62}
$$

Asymptotic variance estimate can be used to find the best (at least asymptotically) quasi ML estimator among a family of estimators. For example, consider that the source is exponentially distributed and contaminated by i.i.d. zero-mean Gaussian noise, independent on the source. The best choice of $\varphi(\cdot)$ in this case is

$$\varphi(S) = -\log\{p_S(S) * p_G(S)\},$$

where $p_G(S)$ is the PDF of the Gaussian noise. The resulting function is a smoothed version of the absolute value and leads to true ML estimation. Yet, it is computationally expensive and one could replace it with a family of sub-optimal smoothed absolute values $\varphi_\lambda(\cdot)$. The optimal value of the smoothing parameter λ is chosen to minimize the asymptotic estimation error variance.

4.2 Consistency and asymptotic stability

It is a widely known fact that an ML estimator is *consistent*, i.e., the estimated parameters converge in probability to their real values, as the sample size grows to infinity. In the BD problem, even if $\varphi(S)$ is proportional to $\log p_S(S)$, the estimator obtained by minimization of $L(H; X)$ is generally not truly maximum likelihood due to *scaling ambiguity*, i.e. the assumed source variance does not correspond generally to the true one. Because of the scaling ambiguity, a "good" deconvolution procedure needs to produce $W * H$ close to δ_{mn} up to a scaling factor c only. This factor can be found from relation (59).

We say that the estimator $\tilde{H}(X)$ of H, obtained by minimization of $L(H; X)$, is *asymptotically stable* if $H = H^*$ is a local minimizer of $L(H; X)$ for infinitely large sample size. Any consistent estimator is asymptotically stable.

Proposition 4 *Let $\tilde{H}^{QML}(X)$ be the quasi ML estimator of H. Then the following holds:*

1. *If $\gamma^2\sigma'^4 > 1$ and $\alpha c^2 > -1$, then $\tilde{H}^{QML}(X)$ is asymptotically stable.*

2. *If $\gamma^2\sigma'^4 < 1$ or $\alpha c^2 < -1$, then $\tilde{H}^{QML}(X)$ is asymptotically unstable.*

Proof: The quasi ML estimator is asymptotically stable if in the limit $M_X N_X \to \infty$, $H = H^*$ is a local minimizer of $L(H; X)$, or due to equivariance, $H = \delta_{mn}$ is a local minimizer of $L(H; c \cdot S)$. The sufficient conditions for this are the Karush-Kuhn-Tucker conditions [9]

$$\underset{M_X N_X \to \infty}{\text{plim}} \nabla L(\delta_{mn}; c \cdot S) = 0 \tag{63}$$

$$\underset{M_X N_X \to \infty}{\text{plim}} \nabla^2 L(\delta_{mn}; c \cdot S) \succ 0, \tag{64}$$

where plim denotes the probability limit

$$\underset{M_X N_X \to \infty}{\text{plim}} \nabla L = 0 \iff \lim_{M_X N_X \to \infty} P\left(\forall \epsilon > 0 : |\nabla L - 0| > \epsilon\right) = 1.$$

The first order necessary condition (64) requires that $\nabla L = 0$ as the sample size goes to infinity. For $(k, l) \neq (0, 0)$ we obtain from (52)

$$\plim_{M_X N_X \to \infty} G_{kl} = \mathbb{E}\varphi'(c \cdot S) \cdot \mathbb{E} \, c \cdot S = 0, \tag{65}$$

and for $k = l = 0$, by choice of c,

$$\plim_{M_X N_X \to \infty} G_{00} = c \cdot \mathbb{E}\varphi'(c \cdot S)S - 1 = 0. \tag{66}$$

The second-order sufficient condition (64) requires that $\nabla^2 L \succ 0$ as the sample size goes to infinity. Using the asymptotic Hessian given in (36), this condition can be rewritten as

$$\begin{pmatrix} \gamma\sigma'^2 & 1 \\ 1 & \gamma\sigma'^2 \end{pmatrix} \succ 0 \tag{67}$$

$$\alpha c^2 + 1 \succ 0. \tag{68}$$

This implies that $\tilde{H}^{QML}(X)$ is asymptotically stable if $\gamma^2 \sigma'^4 > 1$ and $\alpha c^2 > -1$. The required second-order condition for existence of local minimum of L is

$$\plim_{M_X N_X \to \infty} \nabla^2 L(\delta_{mn}; c \cdot S) \succeq 0. \tag{69}$$

Hence, if $\gamma^2 \sigma'^4 < 1$ or $\alpha c^2 < -1$, $\tilde{H}^{QML}(X)$ is asymptotically unstable. $\blacksquare$

It can be observed that when $\varphi(\cdot)$ is chosen proportional to $-\log p_S(S)$, $\tilde{H}^{QML}(X)$ is always asymptotically stable.

4.3   Super-efficiency

Let us now consider the particular case of truly sparse sources, i.e. such sources that take the value of zero with some non-zero probability $P(S = 0) > 0$. An example of such sources are the Gauss-Bernoully (sparse normally) distributed i.i.d. images; in the latter case, $P(S = 0) = 1 - \rho$ (see Appendix B.1 for details). When $\varphi(\cdot)$ is chosen according to (11), $\varphi'_\lambda(\cdot) \to \text{sign}(\cdot)$ and $\varphi''_\lambda(\cdot) \to \frac{1}{2\lambda}\delta(\cdot)$ as $\lambda \to 0^+$. Hence, for a sufficiently small λ,

$$\gamma = \mathbb{E}\varphi''(c \cdot S) = \frac{1}{2\lambda} \int_{-\infty}^{\infty} \delta(cs) \, p_S(s) \, ds = \frac{1}{2\lambda c} \cdot P(S = 0).$$

whereas β is bounded. Hence,

$$\plim_{M_X N_X \to \infty} M_X N_X \cdot \text{var}\{\Delta H_{kl}\} = \frac{\beta}{\gamma^2 \sigma'^2} \leq \text{const} \cdot \lambda^2 \tag{70}$$

for $(k, l) \neq (0, 0)$. Observe that

$$\lim_{\lambda \to 0^+} \plim_{M_X N_X \to \infty} M_X N_X \cdot \text{var}\{\Delta H_{kl}\} \leq \lim_{\lambda \to 0^+} \text{const} \cdot \lambda^2 = 0, \tag{71}$$

that is, the estimator $\tilde{H}_{kl}$ of H_{kl} is *super-efficient* in the limit $\lambda \to 0^+$.

4.4 Cramér-Rao Lower Bounds

We now demonstrate that asymptotic error variance in (57), (62) matches the CRLB on the asymptotic variance of $\tilde{H}_{kl}$, when the true MLE procedure is used, i.e., when $\varphi(S) = -\log p_S(S)$. In this case, $c = 1$, $\sigma'^2 = \sigma^2$, and under the assumption that $\lim_{S \to \pm\infty} p'_S(S)S = 0$,

$$\zeta = \mathbf{E}\varphi'(S)S = -\int_{-\infty}^{\infty} \frac{p'_S(S)}{p_S(S)} p_S(S) S \, dS = -\left. p'_S(S)S \right|_{-\infty}^{\infty} + \int_{-\infty}^{\infty} p_S(S) \, dS = 1.$$

Similarly,

$$\begin{aligned}
\gamma &= \mathbf{E}\varphi''(S) = -\int_{-\infty}^{\infty} \frac{p'_S(S)p_S(S) - p'^2_S(S)}{p_S^2(S)} p_S(S) \, dS \\
&= -\left. p'_S(S) \right|_{-\infty}^{\infty} + \int_{-\infty}^{\infty} \frac{p'^2_S(S)}{p_S^2(S)} p_S(S) \, dS = \mathbf{E}\varphi'^2(S) = \beta.
\end{aligned}$$

Substituting c, σ'^2, γ and ζ into (57), we obtain for $(k,l) \neq (0,0)$

$$\operatorname{var}\{\Delta H_{kl}\} \approx \frac{\beta\sigma^2}{M_X N_X (\beta^2 \sigma^4 - 1)} = \frac{1}{M_X N_X} \cdot \frac{\mathcal{L}}{\mathcal{L}^2 - 1}, \tag{72}$$

where $\mathcal{L} = \sigma^2 \cdot \mathbf{E}\varphi'^2(S)$ is known as Fisher's information for location parameter. This result coincides with the CRLB on H_{kl} developed in [45].

Observe that differentiating the equation

$$\int_{-\infty}^{\infty} p_S(S) S^2 dS = \sigma^2$$

w.r.t. S yields

$$\begin{aligned}
0 &= \int_{-\infty}^{\infty} p'_S(S)S^2 dS + 2\int_{-\infty}^{\infty} p_S(S)S dS \\
&= \int_{-\infty}^{\infty} \frac{p'_S(S)}{p_S(S)} S^2 p_S(S) dS + 2\mathbf{E}S \\
&= -\int_{-\infty}^{\infty} \varphi'(S)S^2 p_S(S) dS.
\end{aligned}$$

Differentiating again w.r.t. S, we obtain

$$\begin{aligned}
0 &= \int_{-\infty}^{\infty} \varphi''(S)S^2 p_S(S) dS + 2\int_{-\infty}^{\infty} \varphi'(S)S p_S(S) dS + \int_{-\infty}^{\infty} \varphi'(S)S^2 p'_S(S) dS \\
&= \mathbf{E}\varphi''(S)S^2 - 2\int_{-\infty}^{\infty} p'_S(S)S dS - \int_{-\infty}^{\infty} \left(\frac{p'_S(S)}{p_S(S)}\right)^2 S^2 p_S(S) dS \\
&= \mathbf{E}\varphi''(S)S^2 - \mathbf{E}\varphi'^2(S)S^2 - 2\left. p_S(S)S \right|_{-\infty}^{\infty} + 2\int_{-\infty}^{\infty} p_S(S) dS \\
&= \mathbf{E}\varphi''(S)S^2 - \mathbf{E}\varphi'^2(S)S^2 + 2.
\end{aligned}$$

23

Hence, $\theta = \alpha + 2$. Substituting $c = 1$, $\zeta = 1$ and the latter result into (62) yields

$$
\begin{aligned}
\text{var}\,\{\Delta H_{00}\} &\approx \frac{1}{M_X N_X} \cdot \frac{\vartheta - 1}{(\alpha + 1)^2} = \frac{1}{M_X N_X} \cdot \frac{\alpha + 1}{(\alpha + 1)^2} = \frac{1}{M_X N_X} \cdot \frac{1}{\alpha + 1} \\
&= \frac{1}{M_X N_X} \cdot \frac{1}{\text{cum}\,\{\varphi'(S), \varphi'(S), S, S\} + \beta\sigma^2 + 1} \\
&= \frac{1}{M_X N_X} \cdot \frac{1}{\mathcal{S}},
\end{aligned} \tag{73}
$$

where $\mathcal{S} = \text{cum}\,\{\varphi'(S), \varphi'(S), S, S\} + \mathcal{L} + 1$ is the Fisher information for the scale parameter. This result coincides with the CRLB on H_{00} in [45].

4.5 Asymptotic restoration quality

An asymptotical estimate of restoration quality in terms of signal-to-interference ratio (SIR) can be expressed as

$$
\begin{aligned}
SIR &= \frac{\mathbb{E}\|S\|_2^2}{\mathbb{E}\|H * X - S\|_2^2} = \frac{|H_{00}^*|^2}{\mathbb{E}\|\Delta H\|_2^2} \approx \frac{1}{\sum_{(k,l)\neq(0,0)} \mathbb{E}(\Delta H_{kl})^2} \\
&\approx \frac{1}{[(2M+1)(2N+1)-1] \cdot \text{var}\,\{\Delta H_{11}\}} \\
&\approx \frac{M_X N_X \,(\gamma^2\sigma'^4 - 1)^2}{[(2M+1)(2N+1)-1]\,(\beta\sigma'^2\,(\gamma^2\sigma'^4 + 1) - 2\gamma\sigma'^2 c^2\zeta^2)}.
\end{aligned} \tag{74}
$$

In particular, when $\varphi(S) = -\log p_S(S)$,

$$
SIR \approx \frac{M_X N_X \,(\mathcal{L}^2 - 1)}{[(2M+1)(2N+1)-1] \cdot \mathcal{L}} \leq \frac{M_X N_X \cdot \mathcal{L}}{(2M+1)(2N+1)-1}. \tag{75}
$$

This result coincides with the bound on the SIR derived in [45].

4.6 Simulation results

In the first simulation, we examine the derived asymptotic estimates in the particular case of the generalized Laplacian distribution (see Appendix B) with the parameters $a, b > 0$, the minus-log PDF is

$$
-\log p_S(S) = \frac{|S|^a}{b}. \tag{76}
$$

This family of probability density functions makes analytical computations possible on one hand, and covers a wide range of distributions (nearly uniform for $a \gg 1$, "short-tale" sub-Gaussian for $a > 2$, normal for $a = 2$, "long-tale" super-Gaussian for $a < 2$, exponential for $a = 1$, and nearly sparse for $a \ll 1$).

The best choice of $\varphi(S)$ is $\varphi(S) = \frac{1}{b}|S|^a$, which yields the true ML estimator and minimizes $\Sigma_{\Delta H}$. The derivatives of $\varphi(S)$ are

$$
\begin{aligned}
\varphi'(S) &= \frac{a}{b} \cdot \text{sign}(S)\,|S|^{a-1} \\
\varphi''(S) &= \frac{a(a-1)}{b} \cdot \text{sign}(S)\,|S|^{a-2}
\end{aligned}
\tag{77}
$$

for $a \neq 1$, and $\varphi''(S) = \frac{2a}{b}\,\delta(S)$ for $a = 1$. In this case, as well as in the case $\varphi(\cdot) = |\cdot|$, there exist closed-form analytic expressions for the distribution-dependent parameters $\alpha, \beta, \gamma, \zeta, \theta, \sigma^2$ and c. For other choices of $\varphi(\cdot)$ (e.g., the smoothed absolute value), these parameters were evaluated by numerical integration.

Figure 8 depicts var $\{\Delta H_{11}\} \cdot M_X N_X$ versus the distribution parameter a for the true ML estimator. The parameter b was fixed to 1. It can be seen that the variance grows very fast to infinity as a approaches 2, which implies the well-known fact of impossibility to deconvolve Gaussian sources. The variance drops dramatically for sparse sources, i.e. when $a \leq 1$. This observation stresses the reason why one should use sparse representations of source images.

In Figure 9, var $\{\Delta H_{11}\} \cdot M_X N_X$ for the true ML estimator, and two quasi ML estimators: one using $\varphi(\cdot) = |\cdot|$, and another using the smooth approximation of the absolute value with $\lambda = 10^{-3}$, are plotted as a function of the distribution parameter a. Figure 10 depicts $\gamma^2 \sigma'^4 - 1$ as function of the distribution parameter a for the true ML estimator and the quasi ML estimator with $\varphi(\cdot) = |\cdot|$. Note that the true ML estimator always obeys the asymptotic stability condition, whereas the quasi ML is asymptotically stable for $a < 2$ and unstable for $a > 2$, since $\alpha + 1 \equiv 1$ and $\gamma^2 \sigma'^4 - 1$ becomes negative for $a > 2$. This fact justifies the choice of $\varphi(\cdot) = |\cdot|$ for super-Gaussian sources. The scaling factor c is depicted in in Figure 11 for different values of a for the true ML estimator and the quasi ML estimator with $\lambda = 0$. The parameter b was fixed to 1.

An additional simulation was performed to assess the asymptotic SIR estimation in (74). A set of 25 i.i.d. source images of size 100×100 was generated from the Gauss-Bernoully distribution with different parameter values (see Appendix B), and convolved with a stable recursive filter, ideally invertible by a 3×3 FIR filter. Restoration kernel was estimated by minimizing $L(H; X)$, using the Newton method, which was terminated when the gradient norm reached the value of 10^{-10}. Smoothing parameter $\lambda = 10^{-2}$ was used. Asymptotic SIR estimates were calculated according to (74), where the values of β, γ, σ'^2 and c were approximated by empirical averages on a set of 10^6 i.i.d. samples.

The obtained results are presented in Figure 12, indicating that the asymptotic estimate is very close to experimentally obtained SIR. Restoration quality increases dramatically for sparser distributions, i.e. when ρ is small [27].
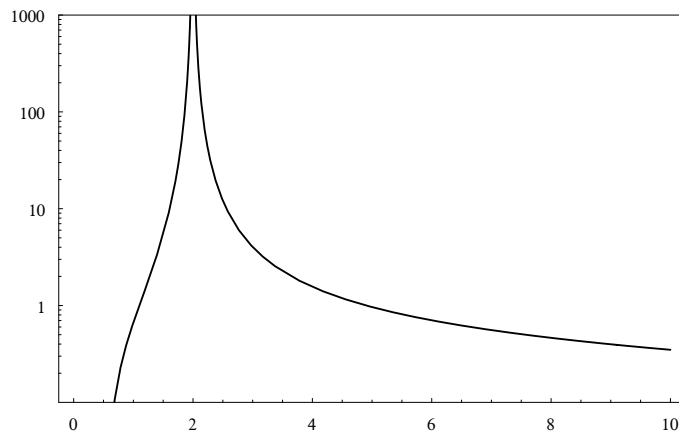
Figure 8: $\mathrm{Var}\{\Delta H_{11}\}$ normalized by the sample size, plotted as a function of the distribution parameter a for the true ML estimator.
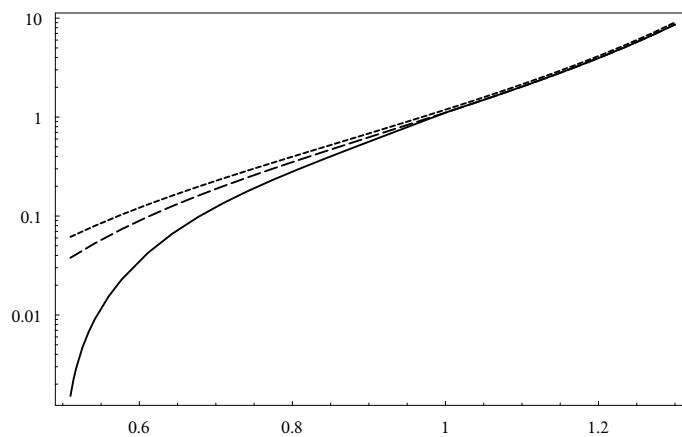

Figure 9: $\mathrm{Var}\{\Delta H_{11}\}$ normalized by the sample size, plotted as a function of the distribution parameter a for the true ML estimator (solid), and the quasi ML estimators with $\lambda = 0$ (dashed), and $\lambda = 10^{-3}$ (dotted).
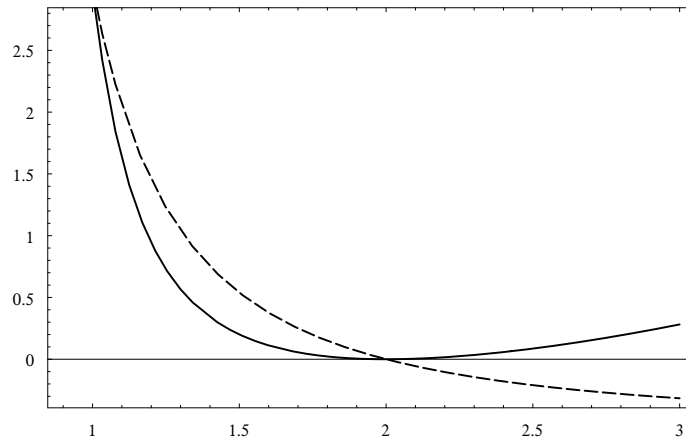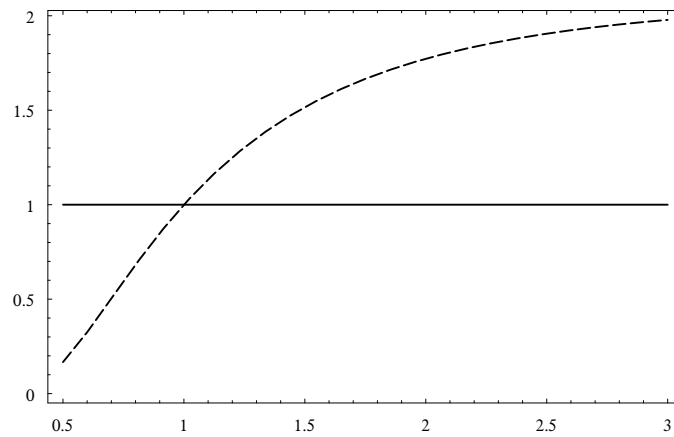
26

Figure 10: The value of $\gamma^2\sigma'^4 - 1$, plotted as a function of the distribution parameter a for the true ML estimator (solid) and the quasi ML estimator with $\lambda = 0$ (dashed). When $\gamma^2\sigma^4 - 1 < 0$, the estimator is asymptotically unstable.

Figure 11: The scaling factor c, plotted as a function of the distribution parameter a for the true ML estimator (solid) and the quasi ML estimator with $\lambda = 0$ (dashed).

27
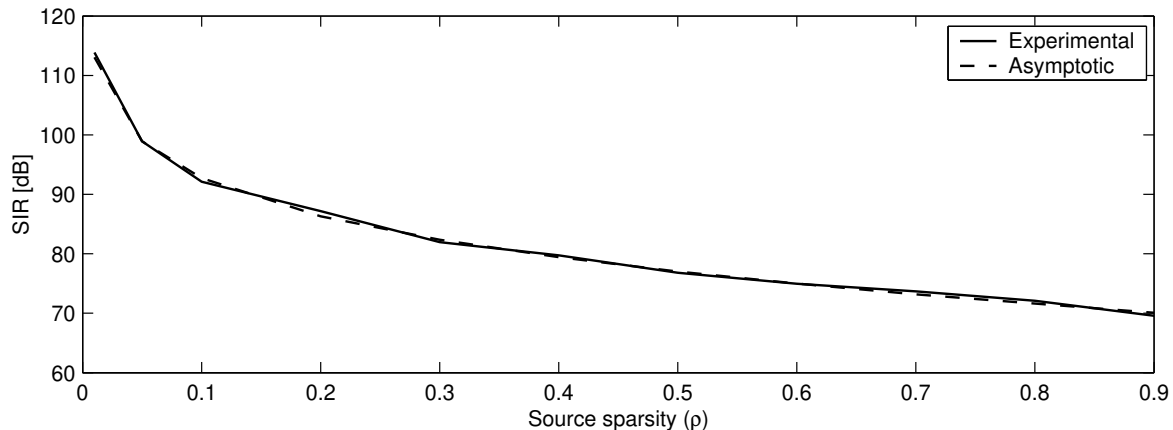
Figure 12: Experimental (solid) and asymptotic (dashed) restoration SIR obtained for Gauss-Bernoulli i.i.d. sources as a function of the sparseness parameter ρ.

5 Optimal sparse representations of images

The quasi-ML framework presented in Section 2 is valid for sparse sources; this type of a prior of source distribution is especially convenient for the underlying optimization problem due to its convexity, and results in very accurate deconvolution. However, natural images arising in the majority of BD applications can by no means be considered to be sparse in their native space of representation (usually, they are *sub-Gaussian*, i.e. have negative kurtosis), and thus such a prior is not valid for "real-life" sources. On the other hand, it is very difficult to model actual distributions of natural images, which are often multimodal and non-convex. This apparent gap between a simple model and the real world calls for an alternative approach. In this section, we show how to overcome this problem using sparse representation.

5.1 Sparsification

While it is difficult to derive a prior suitable for natural images, it is much easier to transform an image is such a way that it fits some universal prior. Hence, we need some kind of a transformation from the native space of natural images with arbitrary distributions to the space of images with distribution defined by a simple prior, which makes the quasi ML function convenient for optimization. In this study, we limit our attention to the sparsity prior, and thus discuss sparsifying transformations, though the idea is general and is suitable for other priors as well.

The idea of *sparsification* was successfully exploited in BSS [14,27,32,33,48,49]. In [14] it was shown that even such simple transformation as a discrete derivative can make the
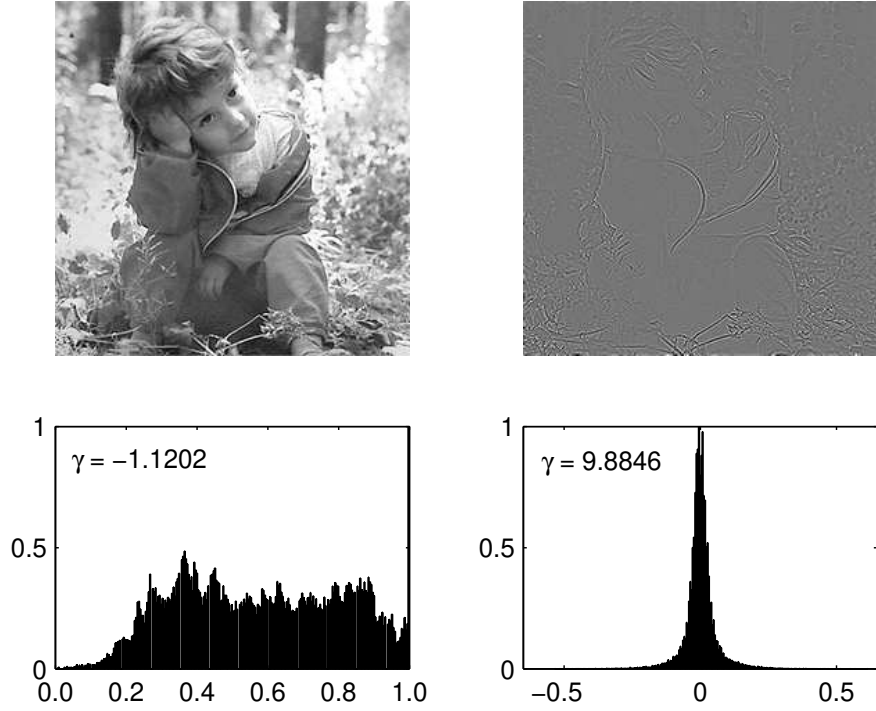
28

Figure 13: Natural image (left top) with sub-Gaussian distribution (negative kurtosis, left bottom). After sparsification by convolution with a second-order discrete derivative kernel (right top), the distribution becomes super-Gaussian (positive kurtosis, right bottom).

image sparse (Figure 13). Let assume that there exists a *sparsifying transformation* $\mathcal{T}_S$, which makes the source S sparse. In this case, our algorithm is likely to produce a good estimate of the restoration kernel H since the source meets the sparsity prior. The problem is that in the BD setting, S is not available, and we can apply $\mathcal{T}_S$ only to the observation X. Hence, it is necessary that the sparsifying transformation commute with the convolution operation, i.e.

$$(\mathcal{T}_S S) * W = \mathcal{T}_S(S * W) = \mathcal{T}_S X, \tag{78}$$

such that applying $\mathcal{T}_S$ to X is equivalent to applying it to S. Obviously, $\mathcal{T}_S$ must be a shift-invariant (SI) transformation.[4] We use X', S' to denote $\mathcal{T}_S X$ and $\mathcal{T}_S S$, respectively; the subscript S in $\mathcal{T}_S$ will be omitted for brevity.

Using the most general nonlinear form of $\mathcal{T}$, we have a wide class of sparsifying trans-

---

[4]In BSS problems, the sparsifying transformation needs to be linear and not necessarily shift-invariant, e.g. wavelet packets were used for sparsification in [27, 48, 49].

formations. An important example is a family of SI transformations of the following form:

$$(\mathcal{T}_S S)_{mn} = \sqrt{(T_1 * S)_{mn}^2 + (T_2 * S)_{mn}^2}, \tag{79}$$

where T_1, T_2 are some convolution kernels. After sparsification with $\mathcal{T}$, the prior term f_2 of the likelihood function becomes

$$\sum_{m,n} |Y'_{mn}| = \sum_n \sqrt{(T_1 * Y)_{mn}^2 + (T_2 * Y)_{mn}^2}, \tag{80}$$

which is a generalization of the 2D *total-variation* (TV) norm. The TV norm, which has been found to be a successful prior in numerous studies related to signal restoration and denoising [10, 15, 19, 26, 39], and was also used by Chan and Wong as a regularization in BD [21], is obtained when T_1, T_2 are chosen as discrete x- and y-directional derivatives.

For the sake of simplicity, we limit our attention in this study to linear shift-invariant (LSI) transformations, i.e. $\mathcal{T}$ that can be represented by convolution with a *sparsifying kernel*

$$\mathcal{T}S = T * S. \tag{81}$$

The class of images which can be sparsified by an LSI transformation (*LSI sparsely representable* images) is obviously smaller than the class of SI sparsely representable images; yet, as will be shown, even in such a limited setting, it is possible to perform deconvolution of some classes of natural images using the sparsification approach.

Thus, we obtain a general BD algorithm, which is not limited to sparse sources. We first sparsify the observation data X by convolving it with T (which has to be found in a way described in Section 5.5), and then apply the sparse BD algorithm on X'. The obtained restoration kernel H is then applied to Y to produce the source estimate.

5.2 The sparsifying kernel

An important practical issue is how to find the kernel T. By definition T must produce a sparse representation of the source; it is obvious that T would usually depend on S (e.g. if the source is sparse in its native representation, the optimal sparsifying kernel is likely to be the Kroenecker delta-function). It should be also noted that T does not necessarily have to be stable, since we use it as a pre-processing of the data and hence never need its inverse.

Let assume for simplicity of the presentation that the source S is given (this is, of course, impossible in reality; the issue of what to use instead of S will be addressed in Section 5.5). It is desired that the unity restoration kernel δ_{mn} be a local minimizer of the quasi-ML function (18) given the transformed source $S * T$ as an observation, i.e.:

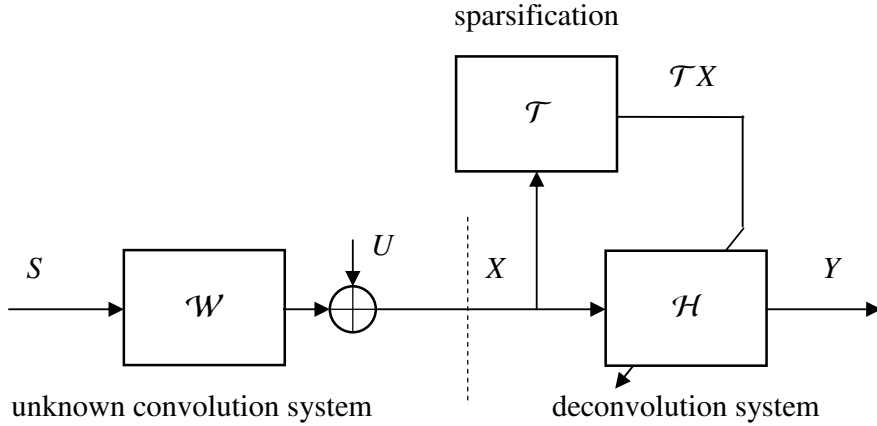$$\nabla L(\delta_{mn}; S * T) = 0. \tag{82}$$

Figure 14: Scheme of blind deconvolution using sparsification. Arbitrarily distributed source S undergoes convolution with unknown kernel W and possibly contaminated by additive noise U. The observed signal X is sparsified using the kernel T, and the restoration kernel H is estimated. Then, H is applied on X to produce an estimate Y of the source.

Informally, this means that $S * T$ optimally fits the sparsity prior (at least in local sense). Due to equivariance, (82) is equivalent to

$$\nabla L(T; S) = 0. \tag{83}$$

In other words, we can define the following optimization problem:

$$\min_T L(T; S), \tag{84}$$

whose solution is the "most sparsifying kernel" for S. This problem is equivalent to the problem

$$\min_H L(H; S) \quad s.t. \ H \ is \ stable, \tag{85}$$

solved for deconvolution itself, with the exception of the stability condition, which is not needed here since T is not necessarily invertible. The term $f_1(T)$ in $L(T; S)$ defined in (16) eliminates the trivial solution $T = 0$. We will start from the one-dimensional case as a motivating example, and then show sparsifying kernels in the two-dimensional case.

5.3 Sparsifying kernel in 1D

First, consider the sparsification problem of one-dimensional signals. One-dimensional quasi-ML BD was treated extensively in our previous study [12]; we adhere here to the

31

same notation. In the current context, we can consider 1D signals as rows or columns of images. It is common to model natural images as piecewise-constant functions; consequently, each row of a natural image can be considered a block signal.

Here, in order to illustrate the computation of the sparsification kernel, we take two 1D signals. The first signal is a synthetic block signal resulting from a Poisson process; the length of each step is exponentially distributed with $\lambda = 10$ and the amplitude of each step is normally distributed (Figure 15, left). The second signal is row number 140 in the Baby image (Figure 15, right). A 3-tap sparsifying kernel t is computed for both signals by solution of the 1D version optimization problem (84),

$$\min_t L(t; s) = \min_t -\frac{1}{2N_F} \cdot f_1(t) + \frac{1}{T} \cdot f_2(t; s). \qquad (86)$$

(see details on the 1D quasi ML function in [12]). The results are shown in Figure 15 and in Table 2.

	Block signal	Row from Baby image
Source Size	504	256
t	$[-0.0124, 0.9909, -1.0000]$	$[0.0040, -0.9902, 1.0000]$

Table 2: Normalized 1D sparsifying kernels.

In both cases, the normalized sparsifying kernel is very close to the 2-tap 1D discrete derivative kernel $t_{dif} = [1, -1]$ (as can be seen in Table 2, the rest of the coefficients in both sparsifying kernels are close to zero). After sparsification with t_{dif}, the prior term f_2 of the likelihood function becomes

$$\sum_n |y'_n| = \sum_n |(t_{dif} * y)_n|, \qquad (87)$$

which coincides with the 1D TV norm. Unlike the 2D case, here the TV norm stems from a linear sparsifying transformation.

5.4   Sparsifying kernel in 2D

The same approach is used to find optimal sparsifying kernels for images. Unlike 1D signals, the sparsifying kernels in 2D have much more degrees of freedom; e.g. the discrete derivative kernel can have different orientations.

To account for these phenomena, we consider three synthetic piecewise-constant images at different orientations ($0°$, $30°$ and $45°$). Three 5×5 sparsifying kernels T are computed for each image by solution of the optimization problem (84). The results are presented in Figures 16–17.
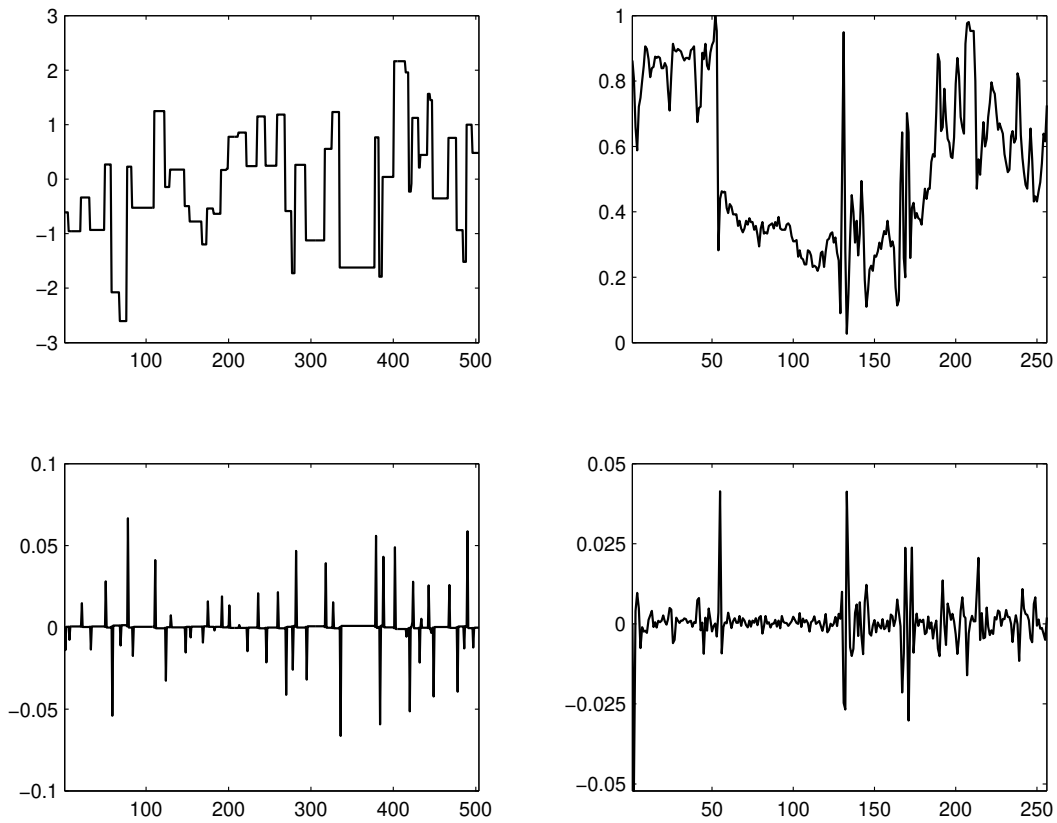
Figure 15: Sparsification results obtained in 1D: synthetic block signal before (left top) and after (left bottom) convolution with sparsifying kernel $[-0.0124, 0.9909, -1.0000]$. Row number 140 from the Baby image before (right top) and after (right bottom) convolution with sparsifying kernel $[0.0040, -0.9902, 1.0000]$.
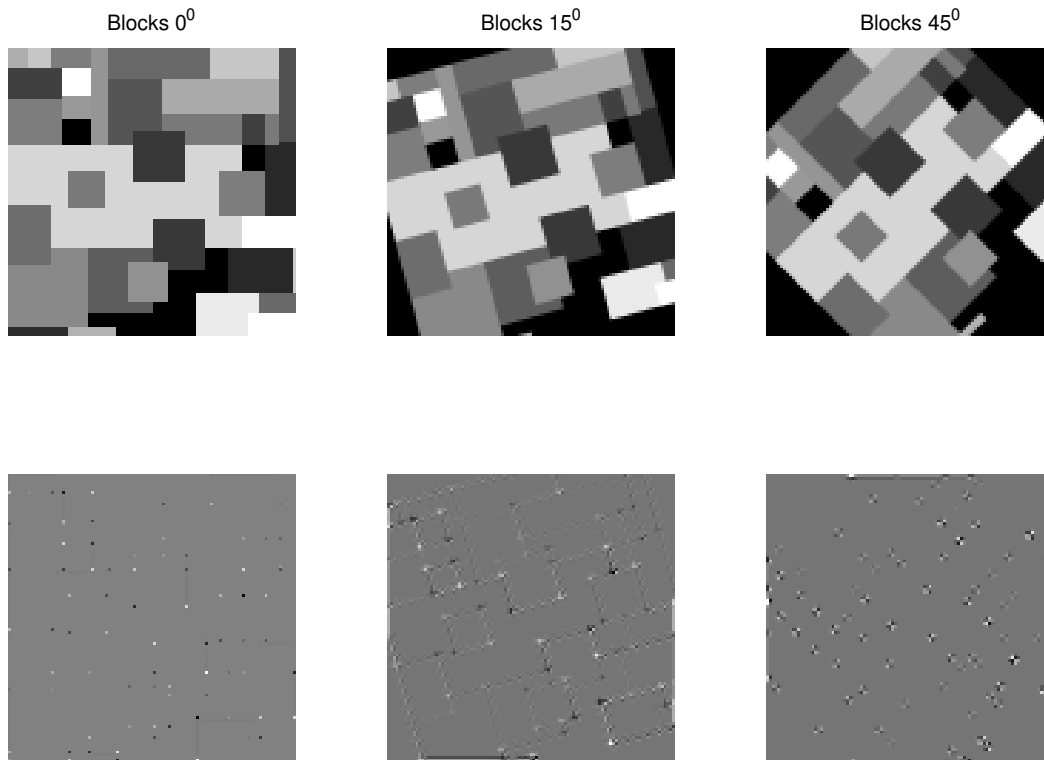
33

Figure 16: Synthetic block images (orientations at $0°$, $15°$ and $45°$, left to right) before (top) and after (bottom) their sparsification.

For the block image at $0°$, the optimal sparsifying kernel is very close (neglecting near-zero elements) to the 2×2 *corner detector*:

$$T_{cor} = \begin{pmatrix} 1 & \text{-1} \\ \text{-1} & 1 \end{pmatrix}. \tag{88}$$

For the block image at $15°$ and $45°$, the sparsifying kernels have the form of diagonal corner detectors (see Figure 17).

5.5   Finding the sparsifying kernel by training

Unfortunately, since the source image S is not available, computation of the sparsifying kernel by the procedure described above is possible only theoretically. However, empirical results show that for images belonging to the same class, the sparsifying kernels are also sufficiently similar. Though we cannot give a formal mathematical definition of such classes of similarity, we will assume that given an image in a class, it is possible to find other images belonging to the same class.
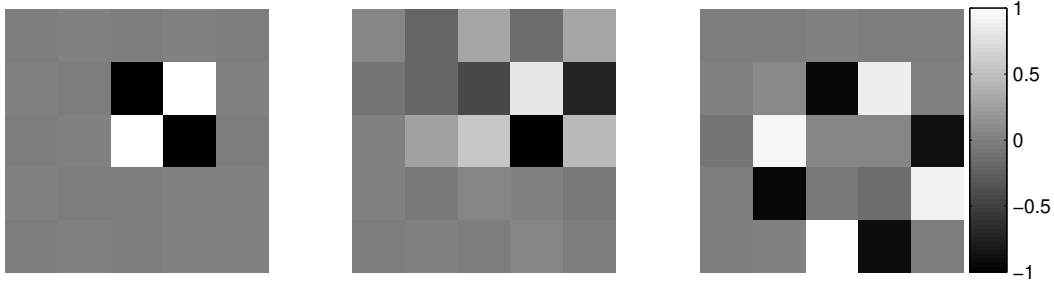
Figure 17: Normalized optimal sparsifying kernels for synthetic block images (orientations at $0°$, $15°$ and $45°$, left to right).

Let $\mathcal{C}_1$ denote a class of images, e.g. of human faces, and assume that the unknown source S belongs to $\mathcal{C}_1$. We can find find images $S^{(1)}, S^{(2)}, ..., S^{(N_T)} \in \mathcal{C}_1$ and use them to find the optimal sparsifying kernel of S. Optimization problem (84) becomes in this case

$$\min_T -\frac{1}{2M_F N_F} \cdot f_1(T) + \frac{1}{M_X N_X} \cdot \frac{1}{N_T} \cdot \sum_{n=1}^{N_T} f_2(S^{(i)} * T), \qquad (89)$$

i.e. T is required to be the optimal sparsifying kernel for all $S^{(1)}, S^{(2)}, ..., S^{(N_T)}$ simultaneously. We term the images $S^{(1)}, S^{(2)}, ..., S^{(N_T)}$ as the *training set*, and the process of finding such T as *training*. Given that the images in the training set are "sufficiently similar" to S, the optimal sparsifying kernel obtained from (89) is similar enough to T_S.

For experimental assessment of this idea, we take three classes of images: piecewise constant (block) images, text images and aerial photos. A 3×3 sparsifying kernel is found by training on a single image, then the same kernel is used as a pre-processing for BD applied to a different blurred source image from the same class of images. All the source images are convolved with a symmetric FIR 31×31 Lorenzian-shaped blurring kernel. Deconvolution kernel is of size 3×3.

In the class of block images, both the training and the source image are synthetic (Figure 18); In the class of text images, the training and the source images contain anti-aliased text of two different citations from Hamlet (Figure 20); in the case of aerial photos, the training image is synthetic (drawn using PhotoShop) and the source image is a real aerial photo of a factory (Figure 22). The results are presented in Figures 18–23 and Table 3.

The sparsifying kernels obtained for all the three classes of images are very close to a corner detector. The best results (in terms of SNR) were obtained on block images, which are optimally sparsified using a corner detector; the worst results are obtained on aerial photos. However, even in the worst case the SNR is sufficiently high (20.1561 dB) and the image quality is plausible.

	Blocks	Text	Aerial
SIR	62.672	44.9952	20.1561
SIR_∞	74.7353	47.8011	25.7228

Table 3: Results of BD with training: reconstruction quality assessed in terms of signal-to-interference ratio [dB].
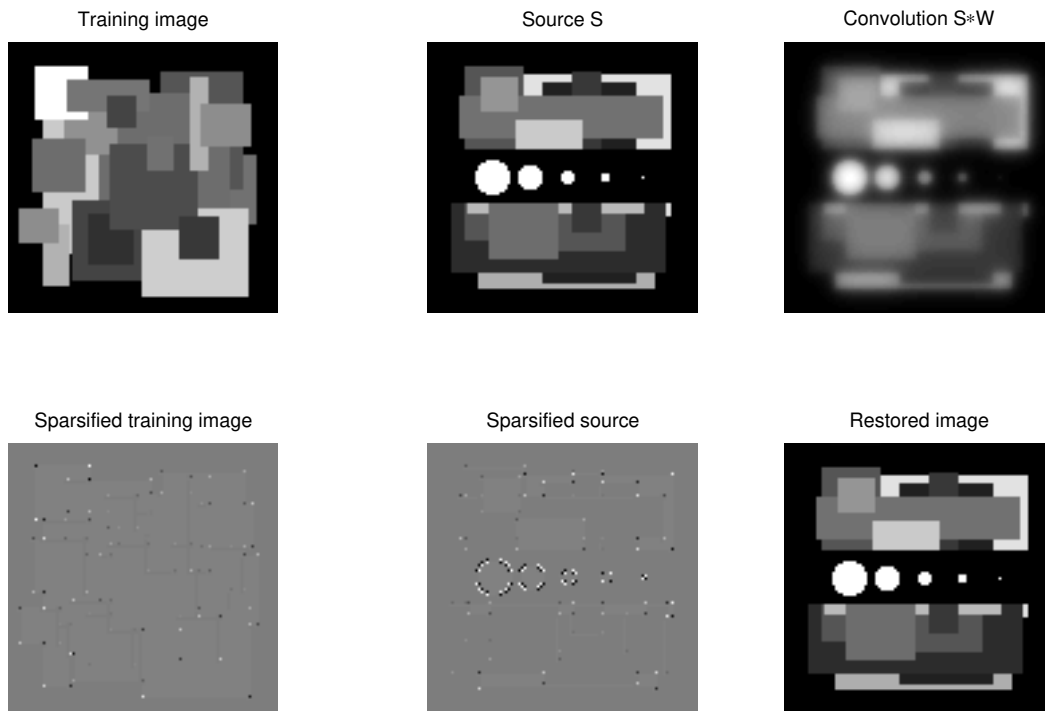

Figure 18: Deconvolution results on synthetic block images: training block image (left top) and its optimal sparsification (left bottom); source block image (middle top) and its sparsification, using the kernel obtained for the training image (middle bottom); convolution result (right top) and the restored image (right bottom).

Convolution kernel W

Sparsifying kernel T

Deconvolution kernel H

Global system response G
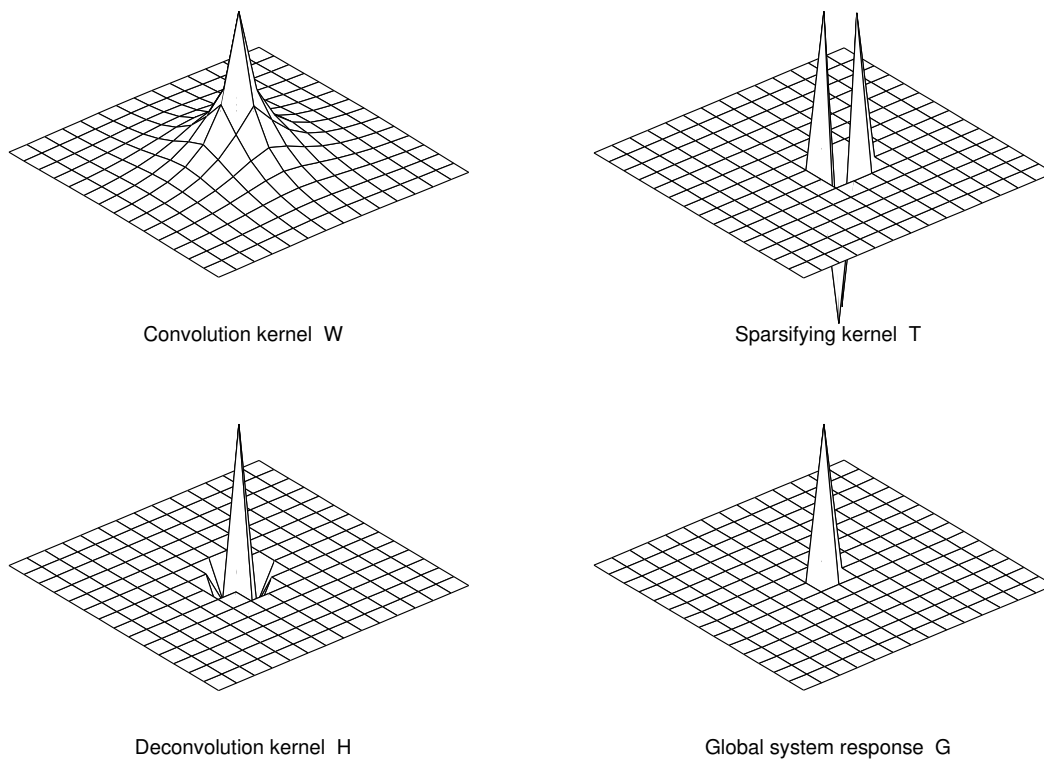
Figure 19: Kernels used in the deconvolution experiment on synthetic block images: convolution kernel (left top), sparsifying kernel obtained by training (right top), estimated deconvolution kernel (left bottom) and the global system response (right bottom).
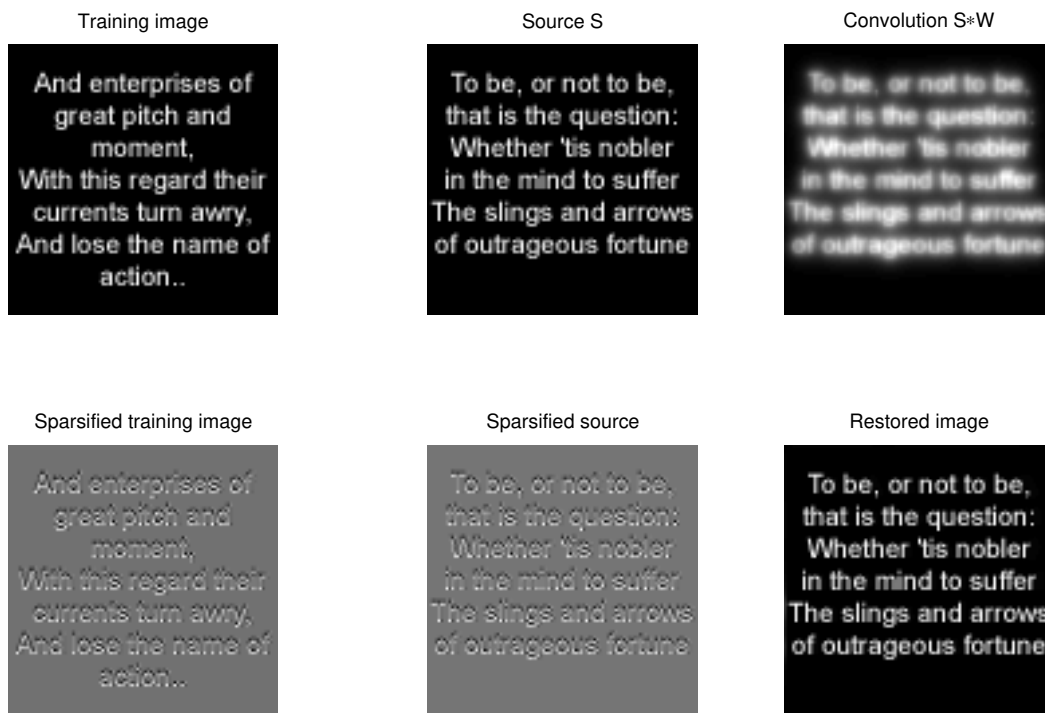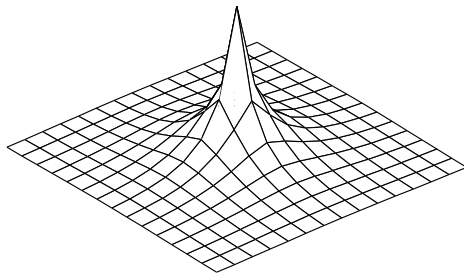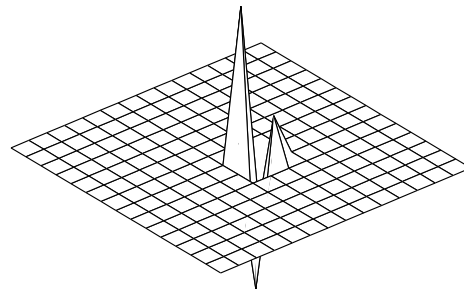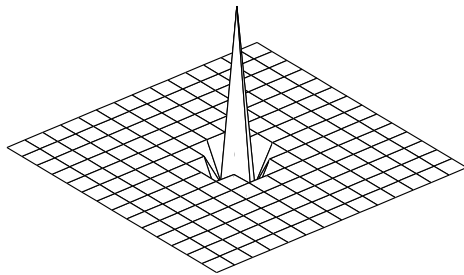
Figure 20: Deconvolution results obtained with text images: training text image (left top) and its optimal sparsification (left bottom); source text image (middle top) and its sparsified version after applying the kernel obtained by training (middle bottom); convolution result (right top) and the restored image (right bottom).
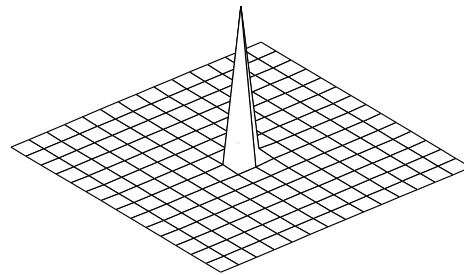
Convolution kernel W

Sparsifying kernel T

Deconvolution kernel H

Global system response G

Figure 21: Kernels used in the deconvolution of text images: convolution kernel (left top), sparsifying kernel obtained by training (right top), estimated deconvolution kernel (left bottom) and the global system response (right bottom).
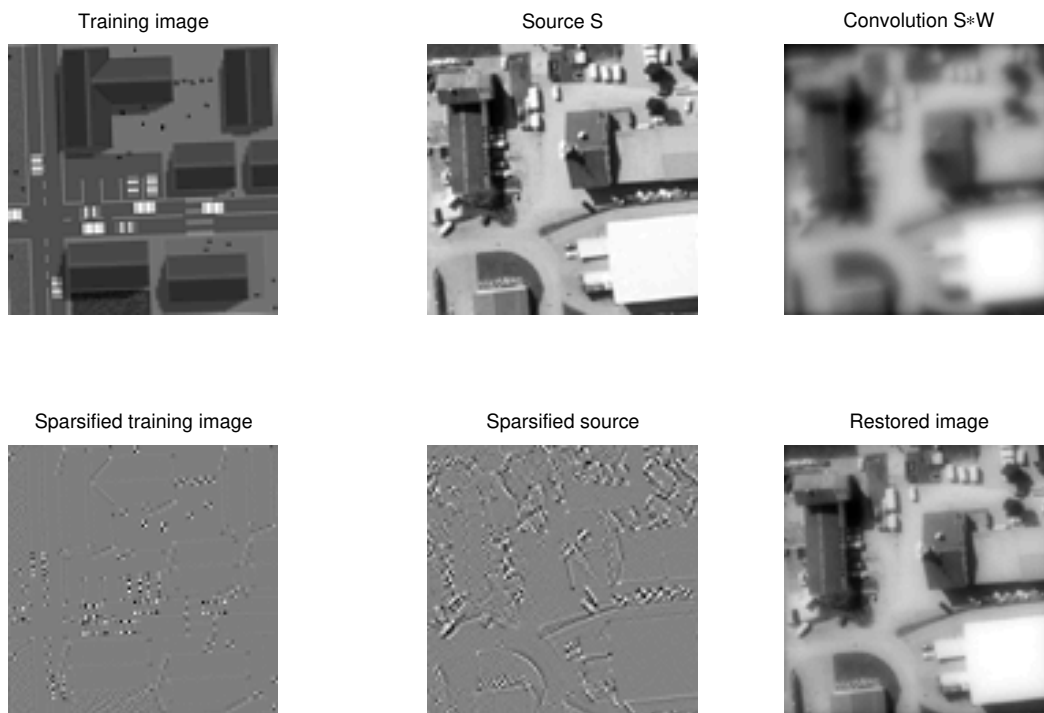
Figure 22: Deconvolution of aerial images: training aerial image drawn in PhotoShop (left top) and its optimal sparsification (left bottom); source aerial image (middle top) and its sparsified version after applying the kernel obtained by training (middle bottom); convolution result (right top) and the restored image (right bottom).

Convolution kernel W

Sparsifying kernel T

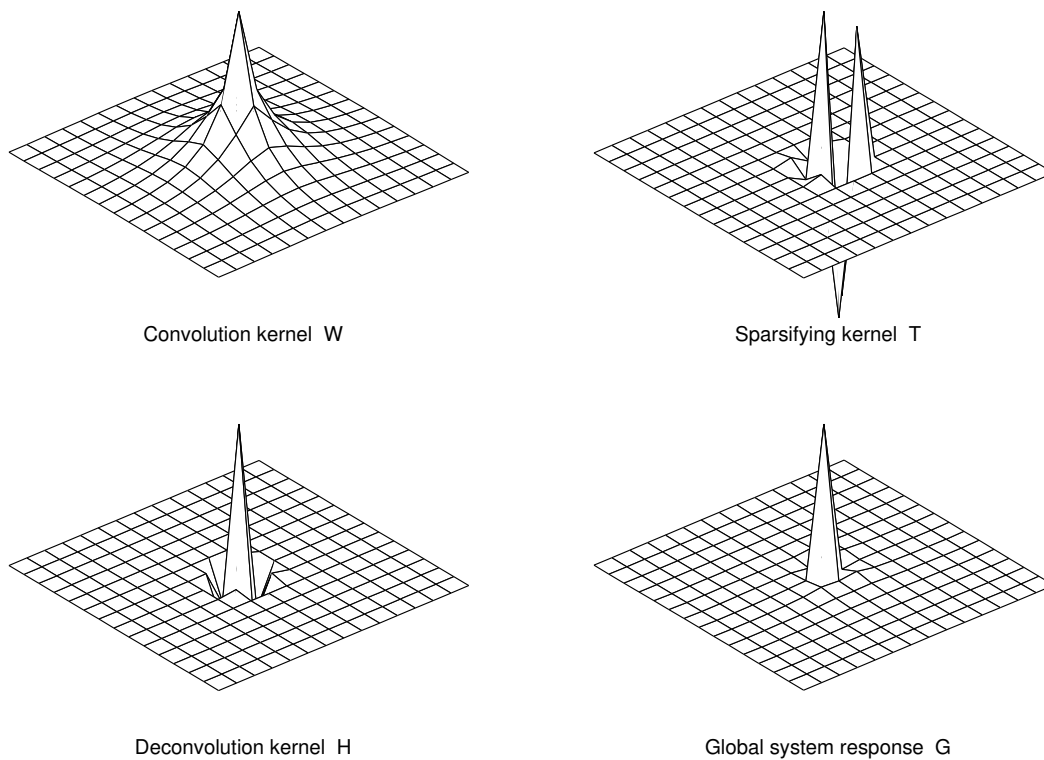Deconvolution kernel H

Global system response G

Figure 23: Kernels used in the deconvolution experiment with aerial images: convolution kernel (left top), sparsifying kernel obtained by training (right top), estimated deconvolution kernel (left bottom) and the global system response (right bottom).

5.6 Alternatives to the optimum sparsifying kernel problem

In some cases, it may be inconvenient to solve the optimum sparsifying kernel problem (84). We present here an alternative way to find sparsifying kernels; it is derived from different considerations, but produces similar results.

The sum of absolute values (f_2 in the quasi ML function) can be used as the objective function to find a kernel T which yields the sparsest image, i.e.

$$\min_T \sum_{m,n} |(T * S)_{mn}| \quad s.t. \quad \|T\|_2^2 = 1, \tag{90}$$

where the constant energy constraint is posed on T to avoid the trivial (zero kernel) solution. This is an optimization problem with nonlinear convex objective and a non-convex quadratic constraint; one of the most common methods to solve such a problem is by using the penalty method [9]. According to this approach, the constrained problem (90) is replaced by the penalty aggregate

$$F_p = \sum_{m,n} \varphi((T * S)_{mn}) + \mu \psi_p(\|T\|_2^2 - 1), \tag{91}$$

which is minimized by an unconstrained optimization algorithm. Here as previously, $\varphi(t)$ denotes a smooth approximation of the absolute value; $\psi_p(t)$ is an equality constraint penalty function (e.g. a quadratic penalty $\psi_p(t) = t^2$), and μ is a parameter determining the tradeoff between the objective function and the penalty, and ideally should equal the Lagrange multiplier of the constraint. A variation of this approach is the Augmented Lagrangian method, in which the Lagrange multiplier is introduced as a parameter into the penalty function ψ [9].

Note the similarity between problems (84) and (91): in both cases we have (up to a factor) f_2 and a penalty requiring T to be a non-trivial solution. Problem (91) is more convenient from an optimization viewpoint; using artificial variables, it can be rewritten as a quadratically-constrained optimization problem, in which case no approximation of the absolute value function is needed. The feasible domain $\{T : \|T\|_2^2 = 1\}$ can be approximated as a polytope using linear constrains, and thus the problem can be approximately presented as a linear program (LP).

6 Simulation results

The presented quasi ML deconvolution approach was tested on simulated data in zero-noise conditions. Four natural source images were used: S_1 (Susy), S_2 (Aerial), S_3 (Gabby) and S_4 (Hubble). The images are presented in Figure 24. Four nearly-stable Lorenzian-shaped, kernels W_1, W_2, W_3 and W_4 (Figure 25) were applied to the corresponding sources. This type of kernels characterizes scattering media, such as biological fluids, aerosoles found in

the atmosphere, etc [36]. The observed images are depicted in Figure 26. Quality of the degraded images in terms of SIR, SIR_∞ and PSIR is presented in Table 4.

Fast relative Newton step with kernel size set to 3×3 was used. The smoothing parameter was set to $\lambda = 10^{-2}$. Corner detector was used as the sparsifying kernel. Optimization was terminated when the gradient norm reached 10^{-10}. Convergence was achieved in $10 - 20$ iterations taking about 10 sec on an ASUS portable computer with Intel Pentium IV Mobile processor and 640MB RAM. The restored images are depicted in Figure 27. Restoration quality results in terms of SIR, SIR_∞ and PSIR are presented in Table 5.

Source		SIR [dB]	SIR_∞ [dB]	PSIR [dB]
S_1	Susy	-1.4648	7.8416	-16.1491
S_2	Aerial	-1.4648	7.8416	-19.9403
S_3	Gabby	4.9018	11.5504	-1.6315
S_4	Hubble	3.3969	10.6454	-0.7940

Table 4: SIR, SIR_∞ and PSIR of the observed images.

Source		SIR [dB]	SIR_∞ [dB]	PSIR [dB]
S_1	Susy	17.7994	22.2092	22.6132
S_2	Aerial	17.0368	23.5482	9.6673
S_3	Gabby	19.3249	23.8109	29.8316
S_4	Hubble	14.5152	17.1552	19.8083

Table 5: SIR, SIR_∞ and PSIR of the restored images.

7   Conclusion

We have extended the 1D quasi ML BD framework studied in [12] for BD of images. We have also presented an extension of the relative optimization approach to quasi ML BD in the 2D case and studied the relative Newton method as its special case. Sparse approximation of the Hessian in a neighbourhood of the solution allows the derivation of a fast version of the relative Newton algorithm, with iteration complexity compared to that of gradient methods. Although in the 2D case restoration kernels obtained from the relative optimization algorithm constitute a more limited class of filters compared to the 1D case, good restoration accuracy is observed in simulations.
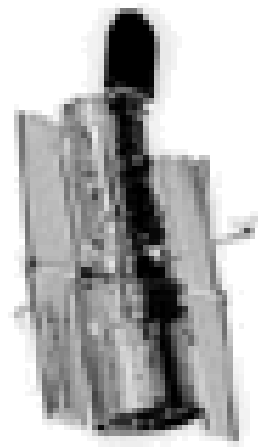
S_1 (Susy)

S_2 (Aerial)

S_3 (Gabby)

S_4 (Hubble)

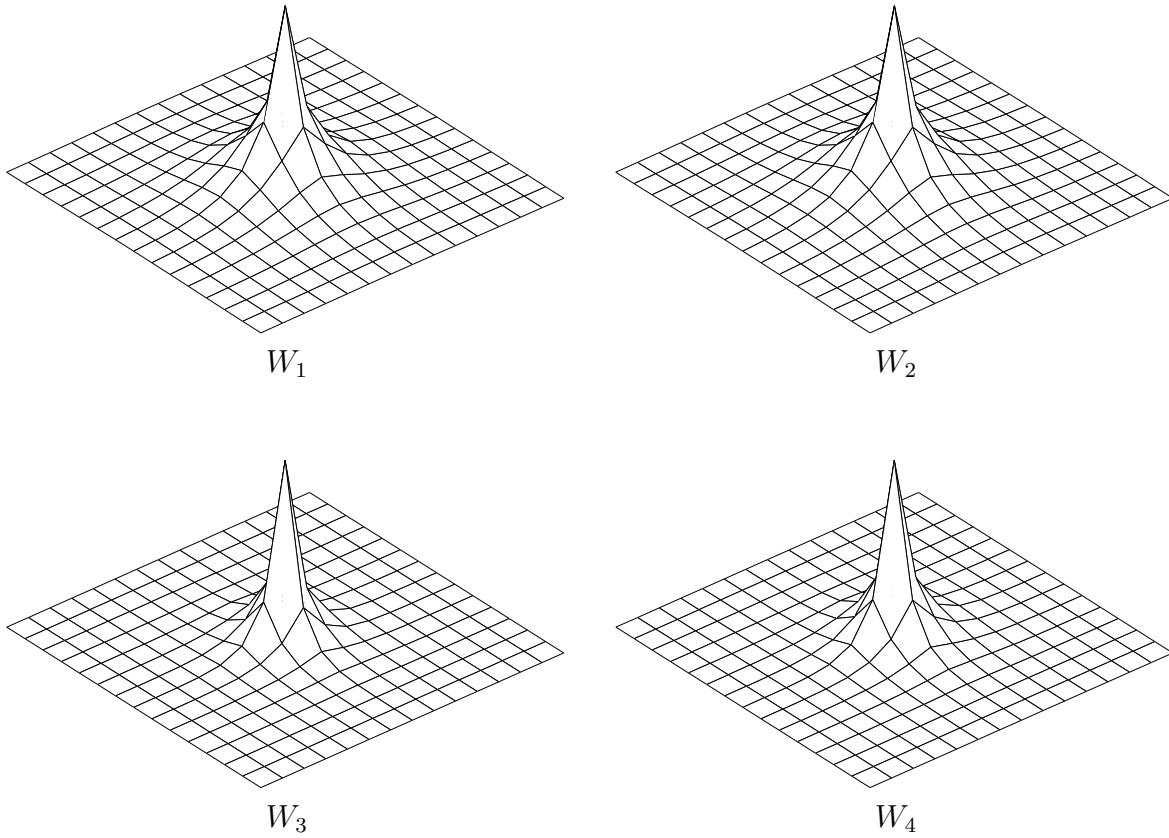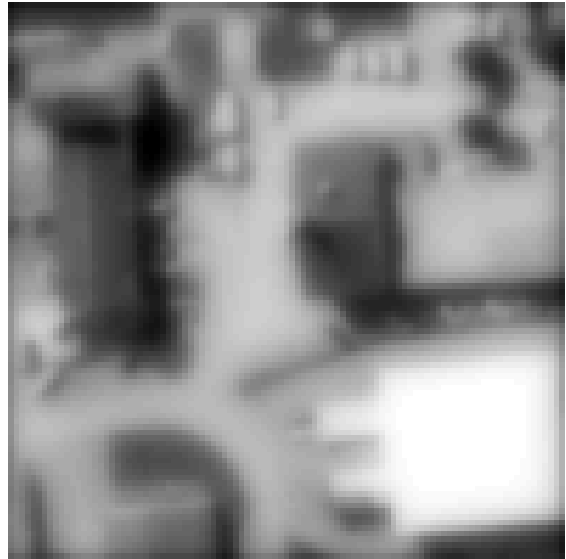Figure 24: Source images S_1, S_2, S_3 and S_4 used in the simulations.

Figure 25: Convolution kernels used in the simulation. Kernels W_1, W_2, W_3, W_4 were applied to the source images: Susy, Aerial, Gabby and Hubble, respectively.
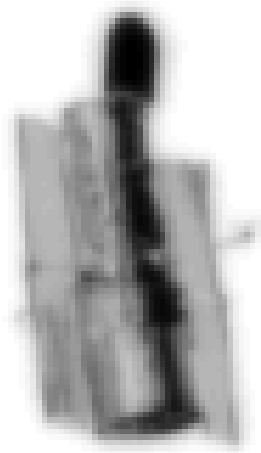
X_1 (Susy)

X_2 (Aerial)

X_3 (Gabby)

X_4 (Hubble)

Figure 26: Observed (blurred) images obtained by convolving S_k with W_k, for $k = 1, ..., 4$.
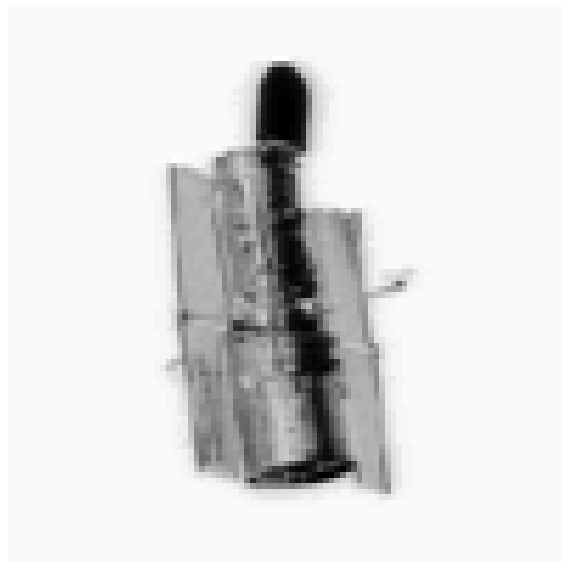
$\tilde{S}_1$ (Susy)

$\tilde{S}_2$ (Aerial)

$\tilde{S}_3$ (Gabby)

$\tilde{S}_4$ (Hubble)

Figure 27: Restoration results using the quasi ML deconvolution approach.

Additionally, asymptotic performance analysis of quasi ML deconvolution and asymptotic stability conditions were presented and assessed by simulations. This analysis revealed some important insights, which justify the particular choice of the smoothed absolute value as the minus-log PDF, $\varphi(\cdot)$, and explain where the high accuracy of BD of sparse sources stems from. It was also demonstrated that for the true ML estimator, the obtained asymptotic performance bounds coincide with the Cramér-Rao bounds, previously reported in literature.

In this study, we have also presented a novel approach of optimal sparse representations, which allows various classes of images with different distributions to be transformed in a way that they fit the sparsity prior used in our quasi ML deconvolution framework. As the result, a general-purpose BD method is obtained. We discussed a particular class of LSI sparsifying transformations and showed that they generalize some previously used approaches such as the total variation prior. We have also shown how optimal sparsifying transformations can be found by training. In our current research, non-linear transformations are addressed explicitly as well.

Simulation results demonstrated the efficiency of the proposed methods. Though in this work we limit our attention to noiseless blind deconvolution, it is important to emphasize that the sparsification framework is applicable to the noisy case as well. Sparsifying kernels are typically high-pass filters, since by their very nature sparse signals have high-frequency components. Such kernels have the property of amplifying noise - thus in case when the signal is contaminated by additive noise, using such kernels in undesired. Intuitively, to cope with the problem of noise, the signal should be smoothed with a low-pass filter F and afterwards the sparsifying kernel T should be applied. Due to commutativity of the convolution, it is equivalent to carrying out the sparsification with a smoothed kernel $T * F$. Noise can be incorporated into the quasi ML deconvolution framework as well. In future works, we intend to study this issue in depth.

Potential powerful applications of our approach and results are in optics, remote sensing, microscopy and biomedical imaging, especially where the amount of noise is moderate. Particularly, this approach is especially accurate and efficient in problems involving slowly-decaying (e.g. Lorenzian-shaped) kernels, which can be approximately inverted using a kernel with small support. Such kernels are typical for imaging through scattering media.

Acknowledgement

A   Proofs and derivations

A.1   Gradient and Hessian of f_1

Let us first find the first- and second-order derivatives of $\hat{H}_{mn}$. Differentiating

$$\hat{H}_{mn} \;=\; \sum_{k=-M}^{M} \sum_{l=-N}^{N} H_{kl}\, e^{-i\left(\frac{2\pi k}{M_F} + \frac{2\pi l}{N_F}\right)} \tag{92}$$

with respect to H_{kl} yields

$$\frac{\partial \hat{H}_{mn}}{\partial H_{kl}} \;=\; e^{-i\left(\frac{2\pi k}{M_F} + \frac{2\pi l}{N_F}\right)}. \tag{93}$$

Let us define

$$q \;=\; \sum_{m=0}^{M_F} \sum_{n=0}^{N_F} \log \hat{H}_{mn}, \tag{94}$$

so that $f_1 = q + q^*$. Differentiating q with respect to H_{kl}, we obtain

$$\frac{\partial q}{\partial H_{kl}} \;=\; \sum_{m=0}^{M_F} \sum_{n=0}^{N_F} \hat{H}_{mn}^{-1} \cdot \frac{\partial \hat{H}_{mn}}{\partial H_{kl}} = \sum_{m=0}^{M_F} \sum_{n=0}^{N_F} \hat{H}_{mn}^{-1} \cdot e^{-i\left(\frac{2\pi km}{M_F} + \frac{2\pi ln}{N_F}\right)}$$

$$\;=\; \mathcal{F}_{M_F,N_F}\left\{\hat{H}_{mn}^{-1}\right\}_{kl} \;=\; Q'_{kl}, \tag{95}$$

where $\mathcal{F}_{M_F,N_F}$ stands for the DFT of size $M_F \times N_F$. Differentiating again with respect to $H_{k'l'}$ yields

$$\frac{\partial^2 q}{\partial H_{kl} \partial H_{k'l'}} \;=\; -\sum_{m=0}^{M_F} \sum_{n=0}^{N_F} \hat{H}_{mn}^{-2} \cdot \frac{\partial \hat{H}_{mn}}{\partial H_{kl}} \frac{\partial \hat{H}_{mn}}{\partial H_{k'l'}}$$

$$\;=\; -\sum_{m=0}^{M_F} \sum_{n=0}^{N_F} \hat{H}_{mn}^{-2} \cdot e^{-i\frac{2\pi}{M_F}(k+k') + i\frac{2\pi}{N_F}(l+l')}$$

$$\;=\; -\mathcal{F}_{M_F,N_F}\left\{\hat{H}_{mn}^{-2}\right\}_{k+k',l+l'} \;=\; Q''_{k+k',l+l'}. \tag{96}$$

Hence, the gradient of f_1 is given by

$$\nabla f_1 \;=\; \operatorname{vec}\left(Q'_{kl} + Q'^{*}_{kl}\right), \tag{97}$$

and the i-th row of the Hessian of f_1 is given by

$$\left(\nabla^2 f_1\right)_i \;=\; \operatorname{vec}\left(Q''_{k+k',l+l'} + Q''^{*}_{k+k',l+l'}\right), \tag{98}$$

where $k' = (i-1)\ \operatorname{mod}\ (2M+1) - M$ and $l' = \lfloor \frac{i-1}{2M+1} \rfloor - N$.

A.2 Gradient and Hessian of f_2

In order to derive ∇f_2 and $\nabla^2 f_2$, let us first express the first- and second-order derivatives of Y_{mn}. The source estimate Y is given by

$$Y_{mn} = (H * X)_{mn} = \sum_{k,l} H_{kl} \, X_{m-k,n-l}. \tag{99}$$

Differentiating (99) with respect to H_{kl} from both sides, we obtain

$$\frac{\partial Y_{mn}}{\partial H_{kl}} = X_{m-k,n-l}, \tag{100}$$

which due to commutativity of the derivative operator with convolution, can be expressed as

$$\frac{\partial Y_{mn}}{\partial H_{kl}} = \frac{\partial Y_{m-k,n-l}}{\partial H_{00}}. \tag{101}$$

Differentiating (100) again with respect to $H_{k'l'}$ yields

$$\frac{\partial^2 Y_{mn}}{\partial H_{kl} \partial H_{k'l'}} = 0. \tag{102}$$

Differentiating $f_2(Y)$ defined in (17) with respect to H_{kl} yields

$$\frac{\partial f_2}{\partial H_{kl}} = \sum_{m,n} \varphi'(Y_{mn}) \frac{\partial Y_{mn}}{\partial H_{kl}} = \sum_{m,n} \varphi'(Y_{mn}) X_{m-k,n-l} = (\Phi' * \mathcal{J}X)_{kl}, \tag{103}$$

where $\Phi'_{mn} = \varphi'(Y_{mn})$ and $(\mathcal{J}X)_{mn} = X_{M_X-m,N_X-n}$. Hence,

$$\nabla f_2 = \mathrm{vec}\left(\frac{\partial f_2}{\partial H_{kl}}\right). \tag{104}$$

Differentiating again with respect to $H_{k'l'}$ yields

$$\begin{aligned}
\frac{\partial^2 f_2}{\partial H_{kl} \partial H_{k'l'}} &= \sum_{m,n} \varphi''(Y_{mn}) \frac{\partial Y_{mn}}{\partial H_{kl}} \frac{\partial Y_{mn}}{\partial H_{k'l'}} \\
&= \sum_{m,n} \varphi''(Y_{mn}) \, X_{m-k,n-l} \, X_{m-k',n-l'} \\
&= \left(A^{k'l'} * \mathcal{J}X\right)_{kl},
\end{aligned} \tag{105}$$

where $A^{k'l'}_{mn} = \varphi''(Y_{mn}) \, X_{m-k',n-l'}$. Hence, the i-th row of the Hessian of f_2 is given by

$$\left(\nabla^2 f_2\right)_i = \mathrm{vec}\left(\frac{\partial^2 f_2}{\partial H_{kl} \partial H_{k'l'}}\right), \tag{106}$$

where $k' = (i-1) \mod (2M+1) - M$ and $l' = \lfloor \frac{i-1}{2M+1} \rfloor - N$.

A.3 Approximate Hessian structure

We consider the Hessian of $f_1(H)$ and $f_2(H; X)$ defined in (21) and (24) for $H_{mn} = \delta_{mn}$ and $X = c \cdot S$. Substituting $H_{mn} = \delta_{mn}$ yields $\hat{H} \equiv 1$, from where by (95)

$$
\begin{aligned}
\frac{\partial^2 f_1}{\partial H_{kl} \partial H_{k'l'}}\bigg|_{H_{mn}=\delta_{mn}} &= -\mathcal{F}_{M_F, N_F}\left\{\hat{H}_{mn}^{-2}\right\}_{k+k', l+l'} - \mathcal{F}_{M_F, N_F}^*\left\{\hat{H}_{mn}^{-2}\right\}_{k+k', l+l'} \\
&= -2 M_F N_F \, \delta_{k+k', l+l'}.
\end{aligned}
\tag{107}
$$

It can be easily seen that $\nabla^2 f_1$ is a constant anti-diagonal matrix with $-2M_F N_F$ on the secondary diagonal. Therefore, f_1 contributes a unit anti-diagonal to $\nabla^2 L$.

For $H_{mn} = \delta_{mn}$ and $X = c \cdot S$, one obtains $Y = c \cdot S$. Substituting to (105) yields

$$
\frac{\partial^2 f_2}{\partial H_{kl} \partial H_{k'l'}} = c^2 \cdot \sum_{m,n} \varphi''(c \cdot S_{mn}) \, S_{m-k, n-l} \, S_{m-k', n-l'}.
\tag{108}
$$

For sufficiently large M_X, N_X,

$$
\frac{\partial^2 f_2}{\partial H_{kl} \partial H_{k'l'}} \approx c^2 M_X N_X \cdot \mathbb{E}\left\{\varphi''(c \cdot S_{mn}) \, S_{m-k, n-l} \, S_{m-k', n-l'}\right\}.
\tag{109}
$$

Without loss of generality, let us assume that S_{mn} is zero-mean. Since S is i.i.d.,

$$
\frac{\partial^2 f_2}{\partial H_{kl} \partial H_{k'l'}} \approx M_X N_X \cdot \begin{cases} \alpha c^2 & : \quad k = k' = l = l' = 0 \\ \gamma(c\sigma)^2 & : \quad k = k' \neq 0, l = l' \neq 0 \\ 0 & : \quad \text{otherwise,} \end{cases}
\tag{110}
$$

where

$$
\alpha = c^2 \cdot \mathbb{E}\varphi''(c \cdot S) S^2
\tag{111}
$$
$$
\gamma = \mathbb{E}\varphi''(c \cdot S)
\tag{112}
$$
$$
\sigma^2 = \mathbb{E}S^2.
\tag{113}
$$

B Probability density functions (PDFs)

B.1 Gauss-Bernoully distribution

A random variable s is said to obey the Gauss-Bernoully distribution with sparsity ρ and variance σ^2 if its probability function is given by

$$
p(s) = (1-\rho)\delta(s) + \frac{\rho}{\sqrt{2\pi\rho\sigma^2}} \exp\left\{-\frac{s^2}{2\rho\sigma^2}\right\},
\tag{114}
$$

where $\rho \in [0, 1]$ and $\sigma^2 > 0$. PDFs and cumulative distribution function (CDFs) of the Gauss-Bernoully distribution are depicted in Figure 28.
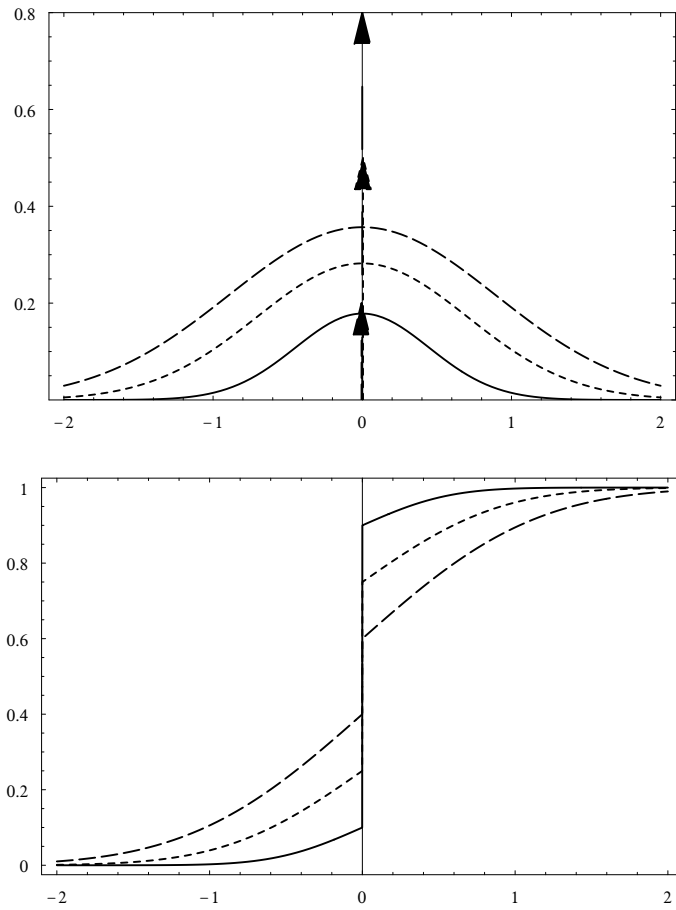
Figure 28: PDF (top) and CDF (bottom) of the Gauss-Bernoully distribution for $\sigma^2 = 1$ and $\rho = 0.2$ (solid), 0.5 (dotted), and 0.8 (dashed). Vertical arrows denote delta functions.

B.2   Generalized Laplacian distribution

A random variable s is said to obey the generalized Laplacian distribution with parameters $a, b > 0$ if its probability function is given by

$$p(s) = \frac{1}{2\Gamma(1 + \frac{1}{a})b^{\frac{1}{a}}} \, \exp\left\{ -\frac{|s|^a}{b} \right\},\tag{115}$$

where $\Gamma(z)$ is the Euler Gamma function defined by

$$\Gamma(z) = \int_0^\infty t^{z-1} \exp\left\{ -t \right\} dt,\tag{116}$$

Distribution (115) can be interpreted as a generalization of the Laplace distribution obtained for $a = 1$, and of the normal distribution obtained for $a = 2$. For $a < 2$, the distribution is super-Gaussian. PDFs and CDFs of the generalized Laplace distribution are depicted in Figure 29.
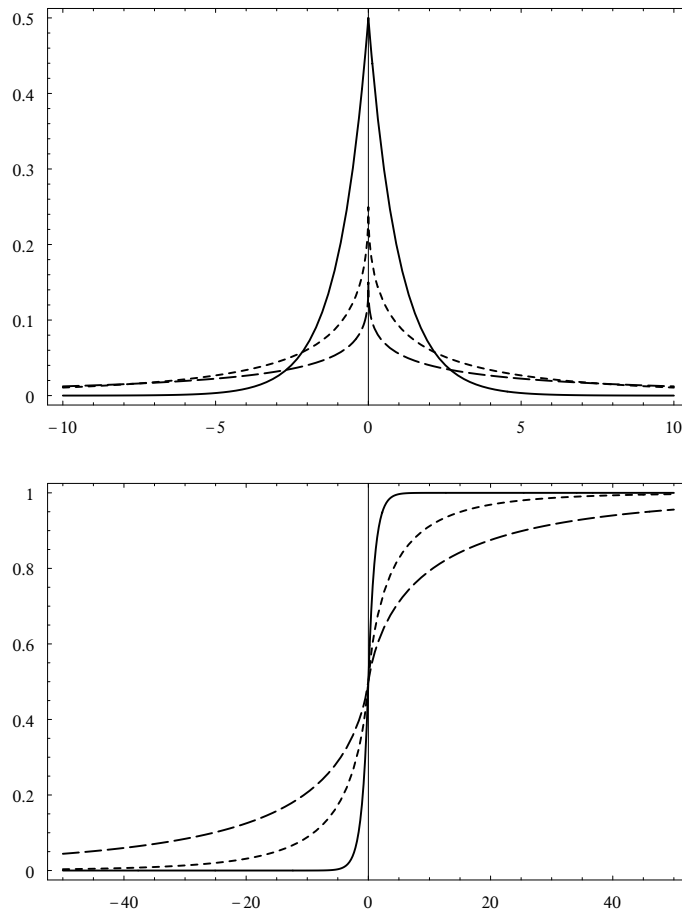
Figure 29: PDF (top) and CDF (bottom) of the generalized Laplace distribution for $b = 1$ and $a = 1$ (solid), 0.5 (dotted), and 0.4 (dashed).

C Notation

I, I_N	Identity matrix of size $N \times N$.
A^T	Transpose of a matrix A.
A^*	Complex conjugate of a matrix A.
$A^H = (A^*)^T$	Hermitian transpose of a matrix A.
$\text{vec}(X)$	Column-stack representation of matrix X.
$f'(t),\ \frac{df(t)}{dt}$	First-order derivative of $f(t)$ with respect to t.
$f''(t),\ \frac{d^2 f(t)}{dt^2}$	Second-order derivative of $f(t)$ with respect to t.
$\partial_x f,\ \frac{\partial f}{\partial x}$	Partial derivative of f with respect to x.
$\partial_{xy}^2 f,\ \frac{\partial^2 f}{\partial x \partial y}$	Second-order partial derivative of f with respect to x and y.
$\nabla_x f(x_0),\ \nabla f$	Gradient of f with respect to x at $x = x_0$.
$\nabla_{xx}^2 f(x_0),\ \nabla^2 f$	Hessian of f with respect to x at $x = x_0$.
$\mathcal{W},\ W_{mn}$	Blurring kernel.
$\mathcal{H},\ H_{mn}$	Restoration kernel.
$\mathcal{G} = \mathcal{HW},$	
$G = H * W$	Global system response.
$\mathcal{T}_S, \mathcal{T},\ T_{mn}$	Sparsifying kernel for image S.
S	Source image.
S'	Sparsified source image.
X	Observed blurred image.
$\mathcal{F}_{MN}$	$M \times N$ DFT-2D operator.
$\mathcal{J} X_{mn} = X_{M-m,N-n}$	Mirror operator of image X_{mn}, $m = 0, ..., M-1; n = 0, ..., N-1$.
δ_{mn}	Kroenecker delta image.
δ_t	Dirac delta function.
$\mathbb{E}\{x\}$	Expectation value of a random variable x.
$\text{var}\{x\}$	Variance of a random variable x.
$\tilde{Y}(X)$	Estimator of the parameters Y given the observation X.
$\hat{X}$	Fourier transform of X.

D   Abbreviations

BD	Blind Deconvolution
BSS	Blind Source Separation
CDF	Cumulative Distribution Function
CRLB	Cramér-Rao Lower Bounds
DFT	Discrete Fourier Transform
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
ICA	Independent Component Analysis
i.i.d.	independent identically distributed
LP	Linear Programming
LSI	Linear Shift-Invariant
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
PDF	Probability Density Function
PSF	Point Spread Function
PSIR	Peak Signal-to-Interference Ratio
QML	Quasi Maximum Likelihood
SI	Shift Invariant
SIR	Signal-to-Interference Ratio
SNR	Signal-to-Noise Ratio
TV	Total Variation

References

[1] D. Adam and O. Michailovich. Blind deconvolution of ultrasound sequences using non-parametric local polynomial estimates of the pulse. *IEEE Trans. on Biomedical Eng.*, 42(2):118–131, 2002.

[2] S.-I. Amari, A. Cichocki, and H. H. Yang. Novel online adaptive learning algorithms for blind deconvolution using the natural gradient approach. In *Proc. SYSID*, pages 1057–1062, July 1997.

[3] S.-I. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *Proc. SPAWC*, pages 101–104, April 1997.

[4] R. H. T. Bates. Astronomical speckle imaging. *Phys. Rep.*, 90(4):203–297, 1982.

[5] A.J. Bell and T.J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[6] S. Bellini and F. Rocca. Near optimal blind deconvolution. In *Proc. of IEEE Conf. Acoust., Speech, Sig. Proc.*, 1988.

[7] M. Bertero and P. Boccacci. Application of the OS-EM method to the restoration of LBT images. *Astron. Astrophys. Suppl. Ser.*, 144:181–186, 2000.

[8] M. Bertero and P. Boccacci. Image restoration methods for the large binocular telescope. *Astron. Astrophys. Suppl. Ser*, 147:323–333, 2000.

[9] D. P. Bertsekas. *Nonlinear Programming (2nd edition)*. Athena Scientific, 1999.

[10] P. Blomgren, T. F. Chan, P. Mulet, and C.K. Wong. Total variation image restoration: numerical methods and extensons. In *Proc. IEEE ICIP*, 1997.

[11] A. Bottcher and S. M. Grudsky. *Toeplitz Matrices, Asymptotic Linear Algebra and Functional Analysis*. Hindustan Book Agency, New Delhi, 2000.

[12] A. M. Bronstein, M. M. Bronstein, and M. Zibulevsky. Blind deconvolution with relative Newton method. Technical Report 444, Technion, Israel, October 2003.

[13] A. M. Bronstein, M. M. Bronstein, and M. Zibulevsky. Block-coordinate relative Newton method for blind source separation. Technical Report 445, Technion, Israel, October 2003.

[14] A. M. Bronstein, M. M. Bronstein, M. Zibulevsky, and Y. Y. Zeevi. Separation of reflections via sparse ICA. In *Proc. IEEE ICIP*, 2003.

[15] M. Bronstein, A. Bronstein, M. Zibulevsky, and H. Azhari. Reconstruction in ultrasound diffraction tomography using non-uniform FFT. *IEEE Trans. Medical Imaging*, 21(11):1395–1401, November 2002.

[16] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Trans. Acoust. , Speech, Signal Process.*, 24(1):58–63, February 1976.

[17] J.-F. Cardoso and B. Laheld. Equivariant adaptive source separation. *IEEE Trans. Sig. Proc.*, 44(12):3017–3030, 1996.

[18] B. Chalmond. PSF estimation for image deblurring. *CVGIP: Graphical Models and Image Processing*, 53(4):364–372, July 1991.

[19] T. F. Chan and P. Mulet. Iterative methods for total variation image restoration. *SIAM J. Num. Anal*, 36, 1999.

[20] T. F. Chan and C. K. Wong. Total variation blind deconvolution. Technical report, 1996.

[21] T. F. Chan and C. K. Wong. Total variation blind deconvolution. In *Proc. ONR Workshop*, 1996.

[22] M. M. Chang, A. M. Tekalp, and A. T. Erdem. Blur identification using the bispectrum. *IEEE Trans. Signal Processing*, 39(10):2323–2325, October 1991.

[23] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[24] R. Fabian and D. Malah. Robust identification of motion and outoffocus blur parameters from blurred and noisy images. *CVGIP: Graphical Models and Image Processing*, 53(5):403–412, July 1991.

[25] T. J. Holmes, S. Bhattacharyya, J. A. Cooper, D. Hanzel, V. Krishnamurthi, W. Lin, B. Roysam, D. H. Szarowski, and J. N. Turner. Light microscopic images reconstructed by maximum likelihood deconvolution. In J. B. Pawley, editor, *Handbook of Biological and Confocal Microscopy, 2nd ed.* Plenum Press, New York, 1995.

[26] R. Kaftori, N. Sochen, and Y.Y. Zeevi. ??? *???*, 2003.

[27] P. Kisilev, M. Zibulevsky, and Y.Y. Zeevi. Multiscale framework for blind source separation. *JMLR*, 2003. In press.

[28] V. Krishnamurthi, YiHwa Liu, and Timothy J. Holmes. Blind deconvolution of 2D and 3D fluorescent micrographs. In *Proc. SPIE 1660*, pages 95–102, 1992.

[29] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Sig. Proc. Magazine*, pages 43–64, May 1996.

[30] D. Kundur and D. Hatzinakos. Blind image deconvolution revisited. *IEEE Sig. Proc. Magazine*, pages 61–63, November 1996.

[31] R. G. Lane and R. H. T. Bates. Automatic multidimensional deconvolution. *J. Opt. Soc. Am. A*, 4(1):180–188, January 1987.

[32] M. S. Lewicki and B. A. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. Am. A*, 16(7):1587–1601, 1999.

[33] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Comp.*, 12:337–365, 2000.

[34] O. Michailovich and D. Adam. Blind deconvolution of ultrasound images using partial spectral information and sparsity constraints. In *Proc. IEEE ISBI*, 2002.

[35] M. Mignotte and J. Meunier. Three-dimensional blind deconvolution of SPECT images. *IEEE Trans. on Biomedical Eng.*, 47(2):274–280, 2000.

[36] M. Moscoso, J. B. Keller, and G. Papanicolaou. Depolarization and blurring of optical images by biological tissue. *J. Opt. Soc. Am. A*, 18(4):948–960, 2001.

[37] J. P. Muller, editor. *Digital Image Processing in Remote Sensing*. Taylor & Francis, Philadelphia, 1988.

[38] D. Pham and P. Garrat. Blind separation of a mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. Sig. Proc.*, 45:1712–1725, 1997.

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

[40] T. J. Schulz. Multiframe blind deconvolution of astronomical images. *J. Opt. Soc. Am. A*, 10(5):1064–1073, 1993.

[41] D. Sengupta and S. Kay. Efficient estimation of parameters for non-Gaussian autoregressive processes. *IEEE Trans. Acoust., Speech, and Sig. Proc.*, 37(6):785–794, 1989.

[42] O. Shalvi and E. Weinstein. Maximum likelihood and lower bounds in system identification with non-Gaussian inputs. *IEEE Trans. Information Theory*, 40(2):328–339, 1994.

[43] E. Thiébaut and J.-M. Conan. Strict a priori constraints for maximum likelihood blind deconvolution. *J. Opt. Soc. Am. A*, 12(3):485–492, 1995.

[44] T. Wilson and S. J. Hewlett. Imaging strategies in threedimensional confocal microscopy. In *Proc. SPIE 1245*, pages 35–45, 1991.

[45] D. Yellin and B. Friedlander. Multichannel system identification and deconvolution: performance bounds. *IEEE Trans. Sig. Proc.*, 47(5):1410–1414, 1999.

[46] D. Yellin and E. Weinstein. Multichannel signal separation: methods and analysis. *IEEE Trans. Sig. Proc.*, 44(1):106–118, 1996.

[47] M. Zibulevsky. Sparse source separation with relative Newton method. In *Proc. ICA2003*, pages 897–902, April 2003.

[48] M. Zibulevsky, P. Kisilev, Y. Y. Zeevi, and B. A. Pearlmutter. Blind source separation via multinode sparse representation. In *Proc. NIPS*. MIT Press, 2002.

[49] M. Zibulevsky and B. A. Pearlmutter. Blind source separation by sparse decomposition. *Neural Computation*, 13(4), 2001.

[50] M. Zibulevsky, B. A. Pearlmutter, P. Bofill, and P. Kisilev. Blind source separation by sparse decomposition. In S. J. Roberts and R. M. Everson, editors, *Independent Components Analysis: Principles and Practice*. Cambridge University Press, 2001.