Image Denoising by Sparse Code Shrinkage

Aapo Hyvärinen, Patrik Hoyer and Erkki Oja
Neural Networks Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland
Email: http://www.cis.hut.fi/projects/ica/

November 15, 1999

Abstract

Sparse coding is a method for finding a representation of data in which each of the components of the representation is only rarely significantly active. Such a representation is closely related to independent component analysis (ICA), and has some neurophysiological plausibility. In this paper, we show how sparse coding can be used for image denoising. We model the noise-free image data by independent component analysis, and denoise a noisy image by maximum likelihood estimation of the noisy version of the ICA model. This leads to the application of a soft-thresholding (shrinkage) operator on the components of sparse coding. Our method is closely related to the method of wavelet shrinkage and coring methods, but it has the important benefit that the representation is determined solely by the statistical properties of the data. In fact, our method can be seen as a simple re-derivation of the wavelet shrinkage method for image data, using just the basic principle of maximum likelihood estimation. On the other hand, it allows the method to adapt to different kinds of data sets.

1 Introduction

Sparse coding [2, 13, 30, 31] is a method for finding a neural network representation of multidimensional data in which only a small number of neurons is significantly activated at the same time. Equivalently, this means that a given neuron is activated only rarely. In this paper, we assume that the representation is linear. Denote by $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ the observed n-dimensional random vector that is input to a neural network, and by $\mathbf{s} = (s_1, s_2, ..., s_n)^T$ the vector of the transformed component variables, which are the n linear outputs of the network. Denoting further the weight vectors of the neurons by $\mathbf{w}_i, i = 1, ..., n$, and by $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_n)^T$ the weight matrix whose rows are the weight vectors, the linear relationship is given by

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{1}$$

We assume here that that the number of sparse components, i.e., the number of neurons, equals the number of observed variables, but this need not be the case in general [31, 20]. Sparse coding can now be formulated as a search for a weight matrix $\mathbf{W}$ such that the components s_i are as "sparse" as possible. A zero-mean random variable s_i is called sparse when it has a probability density function with a peak at zero, and heavy tails; for all practical purposes, sparsity is equivalent to supergaussianity [21] or leptokurtosis (positive kurtosis) [24].

Sparse coding is closely related to independent component analysis (ICA) [1, 3, 6, 21, 17, 23, 22, 29]. In the data model used in ICA, one postulates that $\mathbf{x}$ is a linear transform of independent components: $\mathbf{x} = \mathbf{A}\mathbf{s}$. Inverting the relation, one obtains (1), with $\mathbf{W}$ being the (pseudo)inverse of $\mathbf{A}$. Moreover, it has been proven that the estimation of the ICA data model can be reduced to the search for uncorrelated directions in which the components are as nongaussian as possible [6, 17]. If the independent components are sparse (more precisely, supergaussian), this amounts to the search for uncorrelated projections which have as sparse distributions as possible. Thus, estimation of the ICA model for sparse data is roughly equivalent to sparse coding if the components are constrained to be uncorrelated. This connection to ICA also shows clearly that sparse coding may be considered as a method for redundancy reduction, which was indeed one of the primary objectives of sparse coding in the first place [2, 13].

Sparse coding of sensory data has been shown to have advantages from both physiological and information processing viewpoints [2, 13]. In this paper, we present and analyze a denoising method [19] based on sparse coding, thus increasing the evidence in favor of such a coding strategy. Given a signal corrupted by additive gaussian noise, we attempt to *reduce gaussian noise* by soft thresholding ("shrinkage") of the sparse components. Intuitively, because only a few of the neurons are active (i.e. significantly non-zero) simultaneously in a sparse code, one may assume that the activities of neurons with small absolute values are purely noise and set them to zero, retaining just a few components with large activities. This method is then shown to be very closely connected to the wavelet shrinkage method [10], as well as bayesian wavelet coring [34]. In fact, sparse coding may be viewed as a principled, adaptive way for determining an orthogonal wavelet-like basis based on data alone. Another advantage of our method is that the shrinkage nonlinearities can be adapted to the data as well.

This paper is organized as follows. In Section 2, the basic problem is formulated as maximum likelihood estimation of a nongaussian variable corrupted by gaussian noise. In Section 3, the optimal sparse coding transformation is derived using maximum likelihood estimation of a linear generative model (ICA). Section 4 discusses the alternative approach of minimum mean-square estimation. The resulting algorithm of sparse code shrinkage is summarized in Section 5, and connections to other methods are discussed in Section 6. Extensions of the basic theory are discussed in Section 7. Section 8 contains experimental results, and some conclusions are drawn in Section 9.

2 Maximum Likelihood Denoising of Nongaussian Random Variables

2.1 Maximum likelihood denoising

The starting point of a rigorous derivation of our denoising method is the fact that the distributions of the sparse components are nongaussian. Therefore, we shall begin by developing a general theory that shows how to remove gaussian noise from (scalar) nongaussian variables, making minimal assumptions on the data. Our method is based on maximum likelihood (ML) estimation of nongaussian variables which are corrupted by Gaussian noise.

Denote by s the original (scalar) nongaussian random variable, and by ν gaussian noise of zero mean and variance σ^2. Assume that we only observe the random variable y:

$$y = s + \nu \tag{2}$$

and we want to estimate the original s. Denoting by p the probability density of s, and by $f = -\log p$ its negative log-density, the maximum likelihood method gives the following estimator[1] for s:

$$\hat{s} = \arg\min_u \frac{1}{2\sigma^2}(y-u)^2 + f(u). \tag{3}$$

Assuming f to be strictly convex and differentiable, this minimization is equivalent to solving the following equation:

$$\frac{1}{\sigma^2}(\hat{s}-y) + f'(\hat{s}) = 0 \tag{4}$$

which gives

$$\hat{s} = g(y) \tag{5}$$

where the inverse of the function g is given by

$$g^{-1}(u) = u + \sigma^2 f'(u). \tag{6}$$

Thus, the ML estimator is obtained by inverting a certain function involving f', or the score function [33] of the density of s. For nongaussian variables, the score function is nonlinear, and so is g.

In the general case, even if (6) cannot be inverted, the following first-order approximation of the ML estimator (with respect to noise level) is always possible:

$$\hat{s}^* = y - \sigma^2 f'(y), \tag{7}$$

still assuming f to be convex and differentiable. This estimator is derived from (4) simply by replacing $f'(\hat{s})$, which cannot be observed, by the observed quantity $f'(y)$; these two quantitites are equal to first order. The problem with the estimator in (7) is that the sign of $\hat{s}^*$ is often different from the sign of y even for symmetrical zero-mean densities. Such counterintuitive estimates are possible because f' is often discontinuous or even singular at 0, which implies that the first-order approximation is quite inaccurate near 0. To alleviate this problem of "overshrinkage" [12], one may use the following modification:

$$\hat{s}^o = \text{sign}(y)\max(0, |y| - \sigma^2|f'(y)|). \tag{8}$$

Thus we have obtained the exact maximum likelihood estimator (5) of a nongaussian random variable corrupted by gaussian noise, and its two approximations in (7) and (8).

2.2 Modeling sparse densities

To use the estimator defined by (5) in practice, the densities of the s_i need to be modelled with a parameterization that is rich enough. We have developed two parameterizations that seem to describe very well most of the densities encountered in image denoising. Moreover, the parameters are easy to estimate, and the inversion in (6) can be performed analytically. Both models use two parameters and are thus able to model different degrees of supergaussianity, in addition to different scales, i.e. variances. The densities are here assumed to be symmetric and of zero mean.

[1] This might also be called a maximum a posteriori estimator.

2.2.1 Laplace density

First we review the classical Laplace (or double exponential) density, which is the classical sparse density. The density of a Laplace distribution of unit variance [13, 27] is given by

$$p(s) = \frac{1}{\sqrt{2}} \exp(-\sqrt{2}|s|). \tag{9}$$

The Laplace density is plotted in Fig. 2. For this density, the ML denoising nonlinearity g takes the form[2]

$$g(y) = \operatorname{sign}(y) \max(0, |y| - \sqrt{2}\sigma^2). \tag{10}$$

The function in (10) is a *shrinkage* function that reduces the absolute value of its argument by a fixed amount, as depicted in Fig 1. Intuitively, the utility of such a function can be seen as follows. Since the density of a supergaussian random variable (e.g., a Laplace random variable) has a sharp peak at zero, it can be assumed that small values of y correspond to pure noise, i.e., to $s = 0$. Thresholding such values to zero should thus reduce noise, and the shrinkage function can indeed be considered a soft thresholding operator.

2.2.2 Mildly sparse densities

Our first density model is suitable for supergaussian densities that are not sparser than the Laplace distribution [19], and is given by the family of densities

$$p(s) = C \exp(-as^2/2 - b|s|), \tag{11}$$

where $a, b > 0$ are parameters to be estimated, and C is an irrelevant scaling constant. The classical Laplace density is obtained when $a = 0$, and gaussian densities correspond to $b = 0$. Indeed, since the score function (i.e., f') of the gaussian distribution is a linear function, and the score function of the typical supergaussian distribution, the Laplace density, is the sign function, it seems reasonable to approximate the score function of a symmetric, mildly supergaussian density of zero mean as a linear combination of these two functions. Figure 2 shows a typical density in the family.

A simple method for estimating a and b was derived in [19]. This was based on projecting (using a suitable metric) the score function of the observed data on the subspace spanned by the two functions (linear and sign). Thus we obtain [19]

$$\begin{aligned} b &= \frac{2p_s(0)E\{s^2\} - E\{|s|\}}{E\{s^2\} - [E\{|s|\}]^2} \\ a &= \frac{1}{E\{s^2\}}[1 - E\{|s|\}b] \end{aligned} \tag{12}$$

where $p_s(0)$ is the value of the density function of s at zero. Corresponding estimators of a and b can be then obtained by replacing the expectations in (12) by sample averages; $p_s(0)$ can be estimated, e.g., using a single kernel at 0.

For densities in the family (11), the nonlinearity g takes the form:

$$g(u) = \frac{1}{1 + \sigma^2 a} \operatorname{sign}(u) \max(0, |u| - b\sigma^2) \tag{13}$$

where σ^2 is the noise variance. This function is a shrinkage with additional scaling, as depicted in Fig. 1.

[2]Rigorously speaking, the function in (6) is not invertible in this case, but approximating it by a sequence of invertible functions, (10) is obtained as the limit

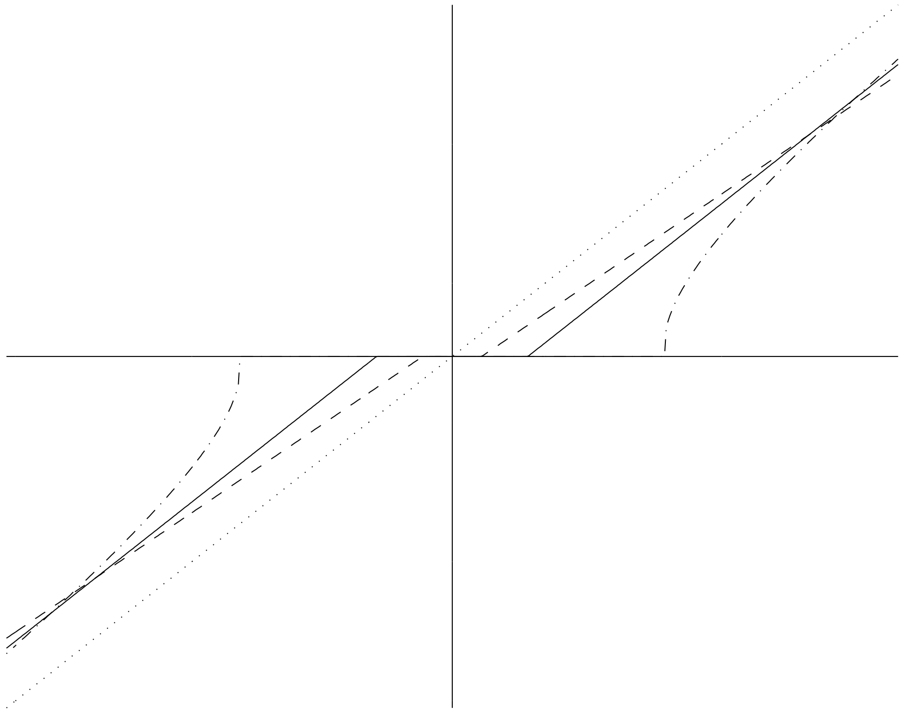

Figure 1: Plots of the shrinkage functions. The effect of the functions is to reduce the absolute value of its argument by a certain amount which depends on the noise level. Small arguments are set to zero. This reduces gaussian noise for sparse random variables. Solid line: shrinkage corresponding to Laplace density as in (10). Dashed line: typical shrinkage function obtained from (13). Dash-dotted line: typical shrinkage function obtained from (17). For comparison, the line $x = y$ is given by dotted line. All the densities were normalized to unit variance, For illustration purposes, the densities are normalized to unit variance, but these parameterizations allow the variance to be choosen freely. Noise variance was fixed to .3.

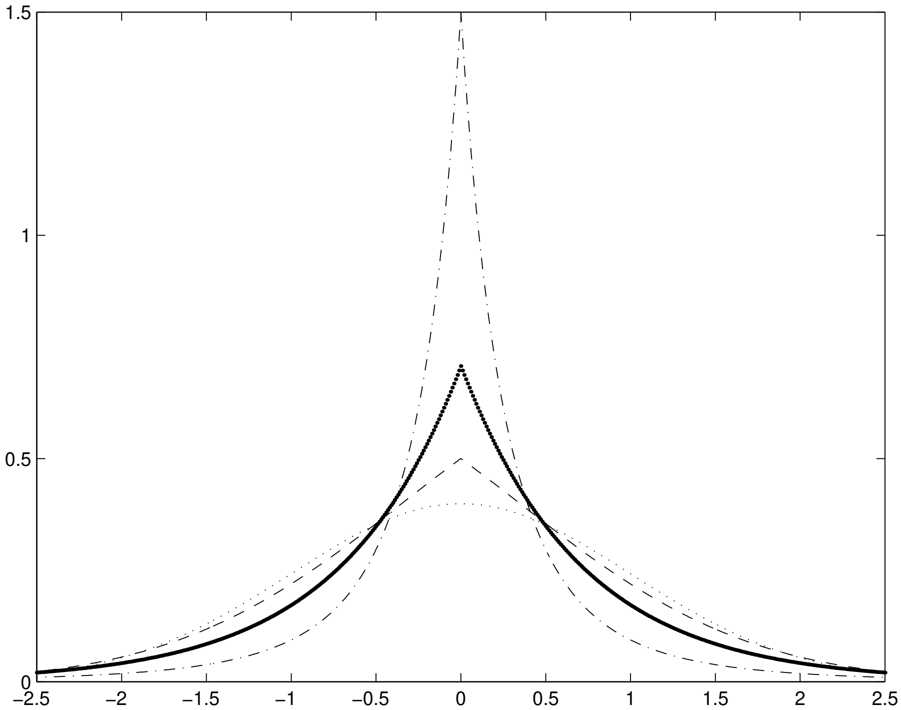

Figure 2: Plots of densities corresponding to different models of the sparse components. Solid line: Laplace density in (9). Dashed line: a typical moderately supergaussian density given by (11). Dash-dotted line: a typical strongly supergaussian density given by (14). For comparison, gaussian density is given by dotted line.

2.2.3 Strongly sparse densities

In fact, most densities encountered in image denoising are sparser than the Laplace density. Therefore, we have developed a second model that describes such very sparse densities:

$$p(s) = \frac{1}{2d} \frac{(\alpha+2)\left[\alpha\,(\alpha+1)/2\right]^{(\alpha/2+1)}}{\left[\sqrt{\alpha\,(\alpha+1)/2} + \mid s/d \mid\right]^{(\alpha+3)}}. \tag{14}$$

Here, d is a scale parameter, and α is a sparsity parameter. When $\alpha \to \infty$, the Laplace density is obtained as the limit. The strong sparsity of the densities given by this model can be seen e.g. from the fact that the kurtosis [13, 21] of these densities is always larger than the kurtosis of the Laplace density, and reaches infinity for $\alpha \leq 2$. Similarly, $p(0)$ reaches infinity as α goes to zero. Figure 2 shows a typical density in the family.

A simple consistent method for estimating the parameters $d, \alpha > 0$ in (14) can be obtained from the relations

$$d = \sqrt{E\{s^2\}}, \tag{15}$$

$$\alpha = \frac{2 - k + \sqrt{k(k+4)}}{2k-1} \tag{16}$$

with $k = d^2 p_s(0)^2$.

The resulting shrinkage function can be obtained as [3]

$$g(u) = \operatorname{sign}(u) \max(0, \frac{|u| - ad}{2} + \frac{1}{2}\sqrt{(|u| + ad)^2 - 4\sigma^2(\alpha+3)}\,) \tag{17}$$

where $a = \sqrt{\alpha(\alpha+1)/2}$, and $g(u)$ is set to zero in case the square root in (17) is imaginary. This is a shrinkage function that has a certain hard-thresholding flavor, as depicted in Fig. 1.

2.2.4 Choice of model

Given a sparse density, we thus model it using one of the above models. Choosing whether model (11) or (14) should be used can be based on moments of the distributions. We suggest that if

$$\sqrt{E\{s^2\}}p_s(0) < \frac{1}{\sqrt{2}}, \tag{18}$$

the first model in (11) be used; otherwise use the second model in (14). The limit case $\sqrt{E\{s^2\}}p_s(0) = \sqrt{1/2}$ corresponds to the Laplace density, which is contained in both models.

2.2.5 Some other models

For sake of completeness, we give here also two classical models for sparse densities. These models are not, however, well suited for our method. The first alternative is the *generalized Laplacian* density [34]

$$p(s) = C \exp(-\left|s/d\right|^{\alpha}) \tag{19}$$

[3] Strictly speaking, the negative log-density of (14) is not convex, and thus the minimum in (5) might be obtained in a point not given by (17): in this case, the point 0 might be the true minimum. To find the true minimum, the value of likelihood at $g(y)$ should be compared with its value at 0, which would lead to an additional thresholding operation. However, such a thresholding changes the estimate only very little for reasonable values of the parameters d and α, and therefore we omit it, using (17) as a simpler and very accurate approximation of the minimization in (3).

where d adjusts the scale, $\alpha \in (0, 2]$ is a sparsity parameter, and C is a normalizing constant. As $\alpha \to 0$ the density becomes sparser and sparser, while at the other end $p = 2$ gives the Gaussian density. Estimation of the parameters d and α can be based on the second and fourth moments of the data as in [34]. It should be noted that the fourth moment is quite sensitive to outliers and thus the estimate is not very robust. No closed form solution of the shrinkage function is available, so if one wants to use this parametrization then these estimators have to be calculated numerically.

A second alternative model for a sparse density can be obtained by using a Gaussian mixture model. A Gaussian mixture model is a parametrization

$$p(s) = \sum_{i=1}^{N} P_i G_i(s) \tag{20}$$

where $\sum_{i=1}^{N} P_i = 1$, and each G_i is a Gaussian kernel

$$G_i(s) = \frac{1}{\sqrt{2\pi} d_i} \exp(-(s - \mu_i)^2 / 2d_i^2). \tag{21}$$

To represent sparse densities, we can choose a mixture of two Gaussians ($N = 2$), both zero-mean ($\mu_i = 0$) but of different variance. Choosing the right parameters, such a mixture can be made sparse, so that a significant amount of data is found at or near zero, while much of the remaining data is far from zero. The estimation of the parameters (P_i and d_i) could be performed using the expectation maximization (EM) algorithm [9]. Such a parameterization does not, however, seem to describe image components very well for our purposes.

3 Finding the Sparse Coding Transformation

In the previous section, it was shown how to reduce additive gaussian noise in scalar nongaussian random variables by means of ML estimation. To denoise random vectors, we could apply such a shrinkage operation separately on each component. But before shrinkage, we would like to transform linearly the data so that the component-wise denoising is as efficient as possible. We shall here restrict ourselves to the class of linear, orthogonal transformations, for reasons that will be explained below. This restriction will be partly relaxed in Section 7.

Let us consider the estimation of the generative data model of independent component analysis (ICA) in the presence of noise. The noisy version of the conventional ICA model is given by

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\nu} \tag{22}$$

where the latent variables s_i are assumed to be independent and nongaussian (usually supergaussian), $\mathbf{A}$ is a constant square matrix, and $\boldsymbol{\nu}$ is a gaussian noise vector. The noise-free ICA model has been shown to describe some important aspects of the basic higher-order structure of image data [30, 31, 4].

Thus, modeling image data with the ICA model, an intuitively simple method for denoising the whole vector $\mathbf{x}$ could be obtained as follows: First, find estimates $\hat{s}_i$ of the (noise-free) independent components, and then reconstruct $\mathbf{x}$ as $\hat{\mathbf{x}} = \hat{\mathbf{A}}\hat{\mathbf{s}}$, as proposed in [16, 25]. Unfortunately, estimating $\mathbf{s}$ in this way is, in general, computationally very demanding. However, in [16] it was proven that if the covariance matrix of the noise and the mixing matrix $\mathbf{A}$ fulfill a certain relation, the estimate $\hat{\mathbf{s}}$ can be obtained simply by applying a shrinkage nonlinearity on the components of $\hat{\mathbf{A}}^{-1}\mathbf{x}$. This relation is fulfilled, e.g. if $\mathbf{A}$ is orthogonal, and noise covariance is proportional to identity.

Thus, restricting $\mathbf{W}$ to be orthogonal, we can find the optimal sparsifying transformation by *estimating the matrix* $\mathbf{A}$ *in the model (22)*, under the constraint of orthogonality of $\mathbf{A}$. The sparsifying transformation is then given by $\mathbf{W} = \mathbf{A}^{-1}$. We adopt this method in the rest of this paper.

To estimate $\mathbf{A}$ under the constraint of orthogonality, we could use the following approximative procedure. First we find an estimate $\tilde{\mathbf{A}}$ of $\mathbf{A}$ using any conventional ICA method, and then transform its inverse $\mathbf{W}_0 = \tilde{\mathbf{A}}^{-1}$ by

$$\mathbf{W} = \mathbf{W}_0(\mathbf{W}_0^T\mathbf{W}_0)^{-1/2} \tag{23}$$

to obtain an orthogonal transformation matrix. The utility of this approximative method resides in the fact that there exist algorithms for ICA that are computationally highly efficient [17, 21]. Therefore, the above procedure enables one to estimate the basis even for data sets of high dimensions. Empirically, we have found that this approximation does not significantly deteriorate the statistical properties of the obtained sparse coding transformation.

In practice, we can further simplify the estimation of $\mathbf{W}$ by assuming that we have access to a random variable $\mathbf{z}$ that has the same statistical properties as $\mathbf{x}$, and can be observed without noise. Thus we can estimate $\mathbf{W}$ using ordinary noise-free ICA algorithms. This assumption is not unrealistic on many applications: for example, in image denoising it simply means that we can observe noise-free images that are somewhat similar to the noisy image to be treated, i.e., they belong to the same environment or context. We make this assumption because estimating the sparsifying matrix from noisy data is too problematic: noisy ICA [11, 18, 16] has proven to be is an inherently difficult problem. However, in principle we could estimate $\mathbf{W}$ from noisy data directly by any method of noisy ICA estimation, as discussed in Section 7.

4 Mean-square error approach

4.1 Minimum mean-square estimator in scalar case

Above, we developed a denoising method based on maximum likelihood estimation of a generative model. It could also be interesting to derive a corresponding result using the criterion of minimum mean-square error. Let us consider the basic scalar denoising problem of Section 2. The mean-square error of the estimator would be minimized if we define $\hat{s}$ as the conditional expectation:

$$\hat{s} = E\{s|y\}. \tag{24}$$

Unfortunately, such a minimum mean-square error (MMSE) estimator cannot be obtained in closed form for any interesting nongaussian distributions. It could be obtained in closed form if the distribution of s would be modelled by a mixture of gaussians, but such models do not seem to be suitable here.

A first-order approximation of $\hat{s}$ can be obtained, however. This is in fact equal to the estimator above in (7), as proven in the Appendix. As a tractable first order approximation, we obtain thus the same estimator as with ML estimation.

4.2 Analysis of mean-square error

In this subsection, we analyze the denoising capability of the scalar MMSE estimator given in (24). We show that, roughly, the more nongaussian the variable s is, the better gaussian noise can be reduced. Nongaussianity is here measured by Fisher information. Due to the intractability of the general problem, we consider here the limit of infinitesimal noise, i.e., all the results are first-order approximations with respect to noise level. Due to the first-order equality between the ML and MMSE estimators, the analysis is also valid for the ML estimator.

To begin with, recall the definition of Fisher information [7] of a random variable s with density p:

$$I_F(s) = E\{[\frac{p'(s)}{p(s)}]^2\}. \tag{25}$$

The Fisher information of a random variable (or, strictly speaking, of its density) equals the conventional, "parametric" Fisher information [33] with respect to a hypothetical location parameter [7].

Fisher information can be considered as a measure of nongaussianity. It is well-known [14] that in the set of probability densities of unit variance, Fisher information is minimized by the gaussian density, and the minimum equals 1. Fisher information is not, however, invariant to scaling; for a constant a, we have

$$I_F(as) = \frac{1}{a^2} I_F(s). \tag{26}$$

The main result on the performance of the ML estimator is the following theorem, proven[4] in [19].

Theorem 1 *Define by (5) and (6), or alternatively by (24), the estimator $\hat{s} = g(y)$ of s in (2). For small σ, the mean-square error of the estimator $\hat{s}$ is given by*

$$E\{(s-\hat{s})^2\} = \sigma^2[1-\sigma^2 I_F(s)] + o(\sigma^4), \tag{27}$$

where σ^2 is the variance of the gaussian noise ν.

To get more insight into the Theorem, it is useful to compare the noise reduction of the ML estimator with the best *linear* estimator in the minimum mean square error sense. If s has unit variance, the best linear estimator is given by

$$\hat{s}_{lin} = \frac{y}{1+\sigma^2}. \tag{28}$$

This estimator has the following mean-square error:

$$E\{(s-\hat{s}_{lin})^2\} = \frac{\sigma^2}{1+\sigma^2}. \tag{29}$$

We can now consider the ratio of these two errors, thus obtaining an index that gives the percentage of additional noise reduction due to using the nonlinear estimator $\hat{s}$:

$$R_s = 1 - \frac{E\{(\hat{s}-s)^2\}}{E\{(\hat{s}_{lin}-s)^2\}}. \tag{30}$$

The following corollary follows immediately:

Corollary 1 *The relative improvement in noise reduction obtained by using the nonlinear ML estimator instead of the best linear estimator, as measured by R_s in (30), is given by*

$$R_s = (I_F(s)-1)\sigma^2 + o(\sigma^2), \tag{31}$$

for small noise variance σ^2, and for s of unit variance.

[4] Please note a notation error in the proof in [19]: what is denoted by $\mathbf{x}$ should be denoted by $\mathbf{y}$

Considering the above-mentioned properties of Fisher information, Theorem 1 thus means that the more nongaussian s is, the better we can reduce noise. In particular, for sparse variables, the sparser s is, the better the denoising works. If s is gaussian, $R = 0$, which follows from the fact that the ML estimator is then equal to the linear estimator $\hat{s}_{lin}$. This shows again that for gaussian variables, allowing nonlinearity in the estimation does not improve the performance, whereas for nongaussian (e.g. sparse) variables, it can lead to significant improvement[5].

4.3 Minimum mean squares approach to basis estimation

We could also use the mean square as a criterion for choosing the sparsifying basis. Assume that we observe a multivariate random vector $\tilde{\mathbf{x}}$ which is a noisy version of the nongaussian random vector $\mathbf{x}$:

$$\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\nu}. \tag{32}$$

where the noise $\boldsymbol{\nu}$ is gaussian and of covariance $\sigma^2\mathbf{I}$. Again, we would like find an orthogonal transformation of the data so that the shrinkage method reduces noise as much as possible. Given an orthogonal (weight) matrix $\mathbf{W}$, the transformed vector equals

$$\mathbf{W}\tilde{\mathbf{x}} = \mathbf{W}\mathbf{x} + \mathbf{W}\boldsymbol{\nu} = \mathbf{s} + \tilde{\boldsymbol{\nu}}. \tag{33}$$

The covariance matrix of $\tilde{\boldsymbol{\nu}}$ equals the covariance matrix of $\boldsymbol{\nu}$, which means that the noise remains essentially unchanged.

The noise reduction obtained by the scalar MMSE method is, according to Theorem 1, proportional to the sum of the Fisher informations of the components $s_i = \mathbf{w}_i^T\mathbf{x}$. Thus, the optimal orthogonal transformation $\mathbf{W}_{opt}$ can be obtained as

$$\mathbf{W}_{opt} = \arg\max_{\mathbf{W}} \sum_{i=1}^{n} I_F(\mathbf{w}_i^T\mathbf{x}) \tag{34}$$

where $\mathbf{W}$ is constrained to be orthogonal, and the $\mathbf{w}_i$ are the rows of $\mathbf{W}$. To estimate $\mathbf{W}$ using this method, one can use the approximation of Fisher information derived in [19].

To estimate the optimal orthogonal transform $\mathbf{W}_{opt}$, we need to assume that we have access to a random variable $\mathbf{z}$ that has the same statistical properties as $\mathbf{x}$, and can be observed without noise.

In image denoising, however, the above result needs to be slightly modified. These modifications are necessary because of the well-known fact that ordinary mean-square error is a rather inadequate measure of errors in images. Perceptually more adequate measures can be obtained e.g. by weighting the mean-square error so that components corresponding to lower frequencies have more weight. Since the variance of the sparse and principal components is larger for lower frequencies, such a perceptually motivated weighting can be approximated simply by the following objective function

$$J = \sum_{i=1}^{n} E\{(\mathbf{w}^T\mathbf{z})^2\} I_F(\mathbf{w}_i^T\mathbf{z}). \tag{35}$$

Using (26), this can be expressed as

$$J = \sum_{i=1}^{n} I_F\left(\frac{\mathbf{w}_i^T\mathbf{z}}{\sqrt{E\{(\mathbf{w}_i^T\mathbf{z})^2\}}}\right). \tag{36}$$

[5] For multivariate gaussian variables, however, improvement can be obtained by Stein estimators [12].

This is the normalized Fisher information, which is a scale-invariant measure of nongaussianity.

In fact, maximizing (36) is very closely related to the estimation of the ICA model, as in (3). Maximizing the sum of nongaussianities of $\mathbf{w}^T\mathbf{z}$ is one intuitive method of estimating the ICA data model [17]. Therefore, it can be seen that using the perceptually weighted mean-square error, we rediscover the basis estimation method of Section 3.

5 Sparse Code Shrinkage

Now we summarize the algorithm of sparse code shrinkage as developed in the preceding sections. In this method, the ML noise reduction is applied on sparse components, first choosing the orthogonal transformation so as to maximize the sparseness of the components. This restriction to sparse variables is justified by the fact that in many applications, such as image processing, the distributions encountered are sparse. The algorithm is as follows:

1. Using a representative noise-free set of data $\mathbf{z}$ that has the same statistical properties as the n-dimensional data $\mathbf{x}$ that we want to denoise, estimate the sparse coding transformation $\mathbf{W} = \mathbf{W}_{opt}$ by first estimating the ICA transform matrix and then orthogonalizing it. (See Section 3.)

2. For every $i = 1, ..., n$, estimate a density model for $s_i = \mathbf{w}_i^T\mathbf{z}$, using the models described in Section 2.2. (Choose by (18) whether model (11) or (14) is to be used for s_i. Estimate the relevant parameters e.g. by (12) or (15), respectively.) Denote by g_i the corresponding shrinkage function, given by (13) or by (17), respectively.

3. Observing $\tilde{\mathbf{x}}(t), t = 1, ..., T$, which are samples of a noisy version of $\mathbf{x}$ as in (32), compute the projections on the sparsifying basis:
$$\mathbf{y}(t) = \mathbf{W}\tilde{\mathbf{x}}(t). \tag{37}$$

4. Apply the shrinkage operator g_i corresponding to the density model of s_i on every component $y_i(t)$ of $\mathbf{y}(t)$, for every t, obtaining
$$\hat{s}_i(t) = g_i(y_i(t)); \tag{38}$$
where σ^2 is the noise variance (see below on estimating σ^2).

5. Transform back to original variables to obtain estimates of the noise-free data $\mathbf{x}(t)$:
$$\hat{\mathbf{x}}(t) = \mathbf{W}^T\hat{\mathbf{s}}(t). \tag{39}$$

If the noise variance σ^2 is not known, one might estimate it, following [10], by multiplying by 0.6475 the mean absolute deviation of the y_i corresponding to the very sparsest s_i.

6 Comparison with wavelet and coring methods

The resulting algorithm of sparse code shrinkage is closely related to wavelet shrinkage [10], with the following differences:

1. Our method assumes that one first estimates the orthogonal basis using noise-free training data that has similar statistical properties. Thus our method could be considered as a principled method of choosing the wavelet basis for a given class of data: instead of being limited to bases that have certain abstract mathematical properties (like self-similarity), we let the basis be determined by the data alone, under the sole constraint of orthogonality.

2. In sparse code shrinkage, the shrinkage nonlinearities are estimated separately for each component, using the same training data as for the basis. In wavelet shrinkage, the form of shrinkage nonlinearity is fixed, and the shrinkage coefficients are either constant for most of the components (and perhaps set to zero for certain components), or constant for each resolution level [10]. (More complex methods like cross-validation [28] are possible, though.) This difference stems from the fact that wavelet shrinkage uses minimax estimation theory, whereas our method uses ordinary ML estimation. Note that point 2 is conceptually independent from point 1, and further shows the adaptive nature of sparse code shrinkage.

3. Our method, though primarily intended for sparse data, could be directly modified to work for other kinds of nongaussian data.

4. An advantage of wavelet methods is that very fast algorithms have been developed to perform the transformation [27], avoiding multiplication of the data by the matrix $\mathbf{W}$ (or its transpose).

5. Of course, wavelet methods avoid the computational overhead, and especially the need for additional, noise-free data required for estimating the matrix $\mathbf{W}$ in the first place. The requirement for noise-free training data is, however, not an essential part of our method, as shown in Section 7.

The connection is especially clear if one assumes that both steps 1 and 2 of sparse code shrinkage in Section 5 are omitted, using a wavelet basis and the shrinkage function (13) with $a_i = 0$ and a b_i that is equal for all i (except perhaps some i for which it is zero). Such a method would be essentially equivalent to wavelet shrinkage.

A related method is Bayesian wavelet coring, introduced in [34]. In Bayesian wavelet coring, the shrinkage nonlinearity is estimated from the data to minimize mean-square error. Thus the method is more adaptive than wavelet shrinkage, but still uses a predetermined sparsifying transformation.

7 Extensions of the basic theory

In this section, we present some extensions to the basic theory given above.

7.1 Nongaussian noise

In fact, in the above derivation noise was always considered as noise in the sparse components s_i, i.e. after the transformation. Therefore, if the noise in the x_i is weakly nongaussian (e.g. Laplace noise), then the noise in s_i is a sum of independent noise components, and thus, due to the central limit theorem, more gaussian than the original noise. This means that the noise in the components may still be considered approximately gaussian, and the basic method presented above can still be expected to work satisfactorily even for weakly nongaussian noise. Modifications of the method are thus necessary only in the case of strongly nongaussian (e.g. impulsive) noise. With such noise, however, it may be more useful to use methods based on reconstructing missing pixels [32].

7.2 Estimation of parameters from noisy data

We have already seen that using the ML principle of basis estimation as in Section 3, it is possible to estimate the sparse code transformation directly from noisy data. We emphasize this here because this important point was missing in our earlier work. Basis estimation from noisy data can thus be accomplished by any method that can estimate the noisy ICA model. In practice, however, this may be very problematic, because the estimation of the noisy ICA model is still very much an open research problem, and it seems probable that the performance that can be obtained will necessarily be considerably lower than what can be obtained from noise-free data. Therefore, in many practical applications, it may not be useful to estimate the basis from noisy data. For the same reasons, estimation of the optimal nonlinearities from noisy data may be very problematic: this is also a future research problem. Nonlinearities that are fixed by prior knowledge may often give satisfactory results, however.

7.3 Non-orthogonal bases

No restriction to orthogonal bases is necessary, either, if the basis is estimated by the ML principle of Section 3. In fact, the transformation can be taken to be simply the inverse of the the mixing matrix $\mathbf{A}$, i.e. $\mathbf{W} = \mathbf{A}^{-1}$, where $\mathbf{A}$ is estimated by any ordinary ICA estimation method. On the other hand, this has the unpleasant side-effect that the noise in the transformed sparse components will be correlated. To be able to estimate the noise-free components without computationally complex operations, we must then approximate the noise structure by a diagonal matrix. Thus the shrinkage functions above can be simply adapted to the case of nonorthogonal transformations by replacing the constant noise covariance by an estimate of the noise variance in the direction of each of the sparse components. Thus we obtain a component-wise shrinkage as before, ignoring all correlations between noise components. However, the advantage gained by relaxing the restriction of orthogonality is diminished by the need to approximate noise covariance.

8 Experiments

8.1 Generation of image data

It would be interesting to test the sparse code shrinkage method on real data (i.e. data which is not available free of noise). However, evaluating such results are difficult, thus we decided to test the performance on images which were artificially corrupted with noise. We chose two separate datasets, so as to be able to compare the performance and see the differences of the results on different datasets.

The first set of images consisted of *natural scenes* previously used in [15] in which ICA was applied to image data. These images were hoped to reflect truly natural images, i.e. images void of human-imposed structure. An example of the images used are shown in Figure 3. These images may be obtained from our web pages.[6] The second set consisted of demo images from Kodak's PhotoCD system. These are royalty free and may be downloaded from the internet by FTP.[7] These are intended to represent images of the human-built world, which have quite different statistics from the natural scenes of the first set. Figure 4 displays an example of these images which we will call *man-made* scenes.

From both sets, ten images were picked at random for estimation of the transforms and the densities, and a separate image from both sets was picked for the actual denoising experiments.

[6] http://www.cis.hut.fi/projects/ica/data/images
[7] ftp://ipl.rpi.edu/pub/image/still/KodakImages/

Figure 3: A representative image from the first set of image data (natural scenes).

Figure 4: A representative image from the second set of image data (man-made scenes).

8.2 Remarks on image data

8.2.1 Windowing

Thus far, we have considered the problem of denoising a random vector of arbitrary dimension. When applying this framework to images, certain problems arise.

The simplest way to apply the method to images would be to simply divide the image into $N \times N$ windows, and denoise each such window separately. This approach, however, has a couple of drawbacks: statistical dependencies across the synthetic edges are ignored, resulting in a blocking artifact, and the resulting procedure is not translation invariant: The algorithm is sensitive to the precise position of the image with respect to the windowing grid.

We have solved this problem by taking a *sliding window* approach. This means that we do not divide the image into distinct windows, rather we denoise every possible $N \times N$ window of the image. We then effectively have N^2 different suggested values for each pixel, and select the final result as the mean of these values. Although originally chosen on rather heuristic grounds, the sliding window method can be justified by two interpretations.

The first interpretation is *spin-cycling.* The basic version of the recently introduced wavelet shrinkage method is not translation-invariant, because this is not a property of the wavelet decomposition in general. Thus, Coifman and Donoho [5] suggested performing wavelet shrinkage on all translated wavelet decompositions of the data, and taking the mean of these results as the final denoised signal, calling the obtained method spin-cycling. It is easily seen that our sliding window method is then precisely spin-cycling of the distinct window algorithm.

The second interpretation of sliding windows is due to the *method of frames.* Consider the case of decomposing a data vector into a linear combination of a set of given vectors, where the number of given vectors is larger than the dimensionality of the data, i.e. given $\mathbf{x}$ and $\mathbf{A}$, where $\mathbf{A}$ is an m by n matrix ($m < n$), find the vector $\mathbf{s}$, in $\mathbf{x} = \mathbf{A}\mathbf{s}$. This has an infinite number of solutions. The classical way is to select the solution $\mathbf{s}$ with minimum norm, given by $\hat{\mathbf{s}} = \mathbf{A}^{\dagger}\mathbf{x}$, where $\mathbf{A}^{\dagger}$ is the pseudo-inverse of $\mathbf{A}$. This solution is often referred to as the method of frames solution [8].

Now consider each basis window in each possible window position of the image as an overcomplete basis for the whole image. Then, if the transform we use is orthogonal, the sliding window algorithm is equivalent to calculating the method of frames decomposition, shrinking each component, and reconstructing the image.

8.2.2 The local mean

ICA applied to image data usually gives one component representing the local mean image intensity, or the DC component. This component normally has a distribution that is not sparse, often even subgaussian. Thus, it must be treated separately from the other, supergaussian components.

One could estimate a suitable density model for this component, and denoise it just as the others. However, since the component generally has a large variance it is relatively unaffected by the noise, and a simplification is to simply leave it alone. This is the approach we have chosen to take. This means that in all experiments we first subtract the local mean (and drop the dimension using PCA), and then estimate a suitable sparse coding basis for the rest of the components. In restructuring the image after denoising, we add again the local means.

8.2.3 Normalizing the local variance

Some image processing methods normalize the local variance in an image. We can considered such a variant of our method as well. We can incorporate normalization into our method by dividing each image window by its norm. This can be done in both the estimation of parameters, and in the denoising procedure.[8]

8.3 Transform estimation

8.3.1 Methods

Estimating an ICA transform from patches of natural image data has previously been shown to give a transform mainly consisting of local filters, resembling somewhat the so-called Gabor filters or wavelets [15, 30, 31, 4], and this is what we find here as well.

Each image was first linearly normalized so that pixels had zero mean and unit variance. A set of 10000 image patches (windows) were taken at random locations from the images. From each patch the local mean was subtracted as explained above. This resulted in a linear dependency between the pixels of each patch, and thus we reduced the dimension by one using standard PCA. The preprocessed dataset was used as the input to the FastICA algorithm [21, 17], using the hyperbolic tangent nonlinearity [17].

8.3.2 Results

Figure 5 shows the results on the first data set, which was patches from natural scenes with 8×8 window size. The PCA transform consists of global features, and resembles strongly the 2D-Fourier transform. The transform given by ICA, however, finds features which resemble local edges and bars. They can thus be said to be more representative of the image data.

There is an interesting difference between the ICA filters (comprising the separating matrix $\mathbf{W}$) and the basis vectors (making up the mixing matrix $\mathbf{A}$). The basis vectors constitute the 'building blocks' from which the data is thought to be generated. Thus they have the same type of features that we are used to seeing in images. Looking closely, it is easily seen that the filters are similar to the basis vectors, in that each filter has the same position and orientation as its corresponding basis vector. However, the separating filters are clearly more 'spiky'. This is in essence a consequence of the whitening, since whitening must amplify high frequencies (because these have the smallest variance).

In our basic method the ICA transform is orthogonalized, and the thus obtained transform is simply the ICA transform for zero-phase whitened data. This transform is also depicted in Figure 5, and can be seen as being "in-between" the separating matrix $\mathbf{W}$ and the mixing matrix $\mathbf{A}$. The features of the transform are not essentially changed by this orthogonalization.

In the second data set, the image patches were gathered from the man-made scenes. The results are in Figure 6. The difference to the ICA transforms from natural scenes is quite clear. In man-made scenes, there are much stronger continuous lines and edges, which shows up in the ICA decomposition as basis vectors which are more edge-like than the "Gabor-like" basis vectors from the natural scenes.

We also experimented with a larger window size of 16-by-16 pixels. The found ICA basis (not shown) was qualitatively similar to that estimated from the smaller windows, consisting of lines and edges at various locations and at various orientations.

For comparison, we show a Daubechies wavelet basis (filter length=4) in Fig. 7.

8.4 Component statistics

Since the denoising procedure is based on the property that individual components in the transform domain have sparse distributions, it is obvious that it must be tested how well this requirement holds. Also, the method

[8] When considering noisy windows, one must estimate the norm before the noise was added, and divide by this quantity. This estimation can be done reasonably well when the size of the image window is large enough, e.g. 8×8.

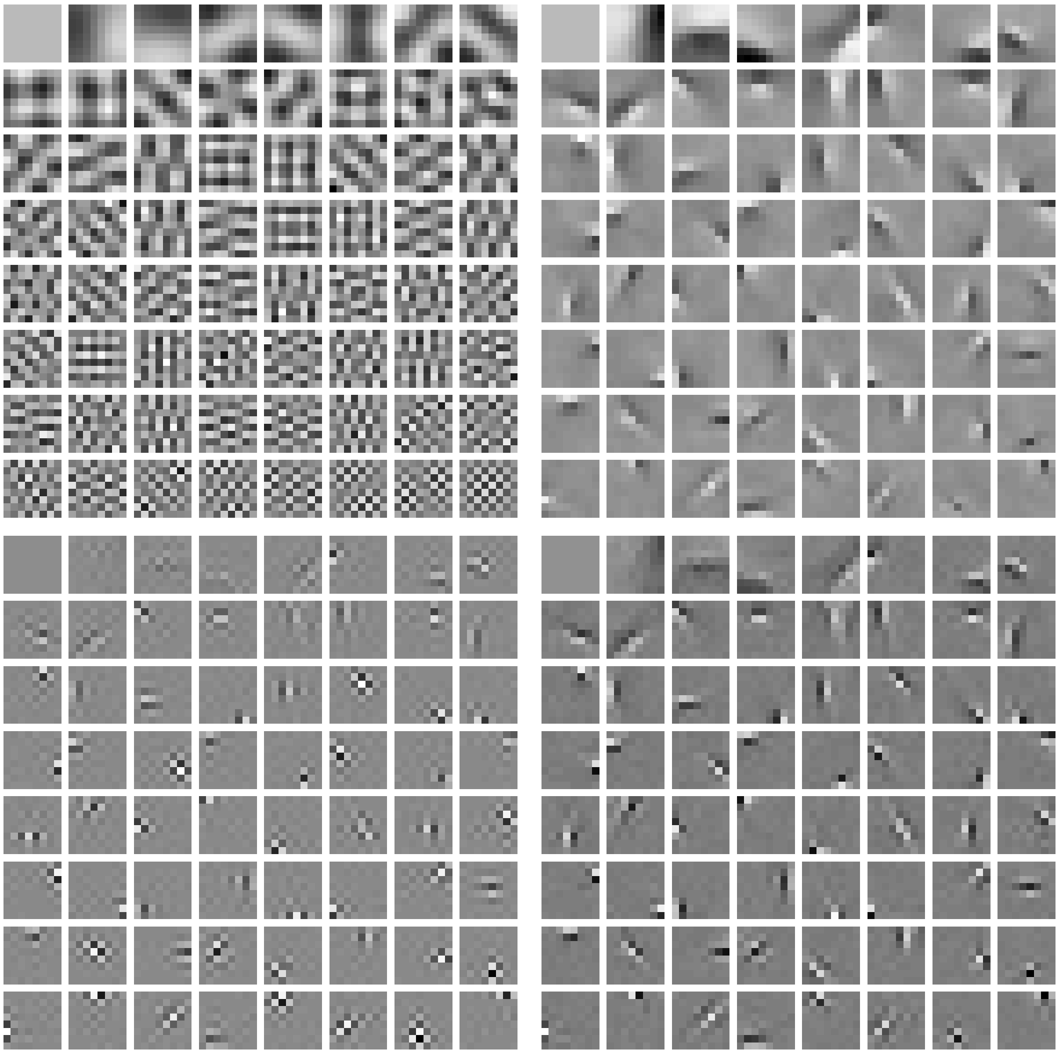

Figure 5: Transforms estimated for natural scene data (8-by-8 patches). The windows have been ordered according to mean frequency for visualization purposes. (This ordering is irrelevant in the denoising algorithm.) Top-left: PCA transform (orthogonal, so basis and filters are the same). Top-right: ICA basis. Bottom-left: ICA separating filters. Bottom-right: Orthogonalized ICA transform.

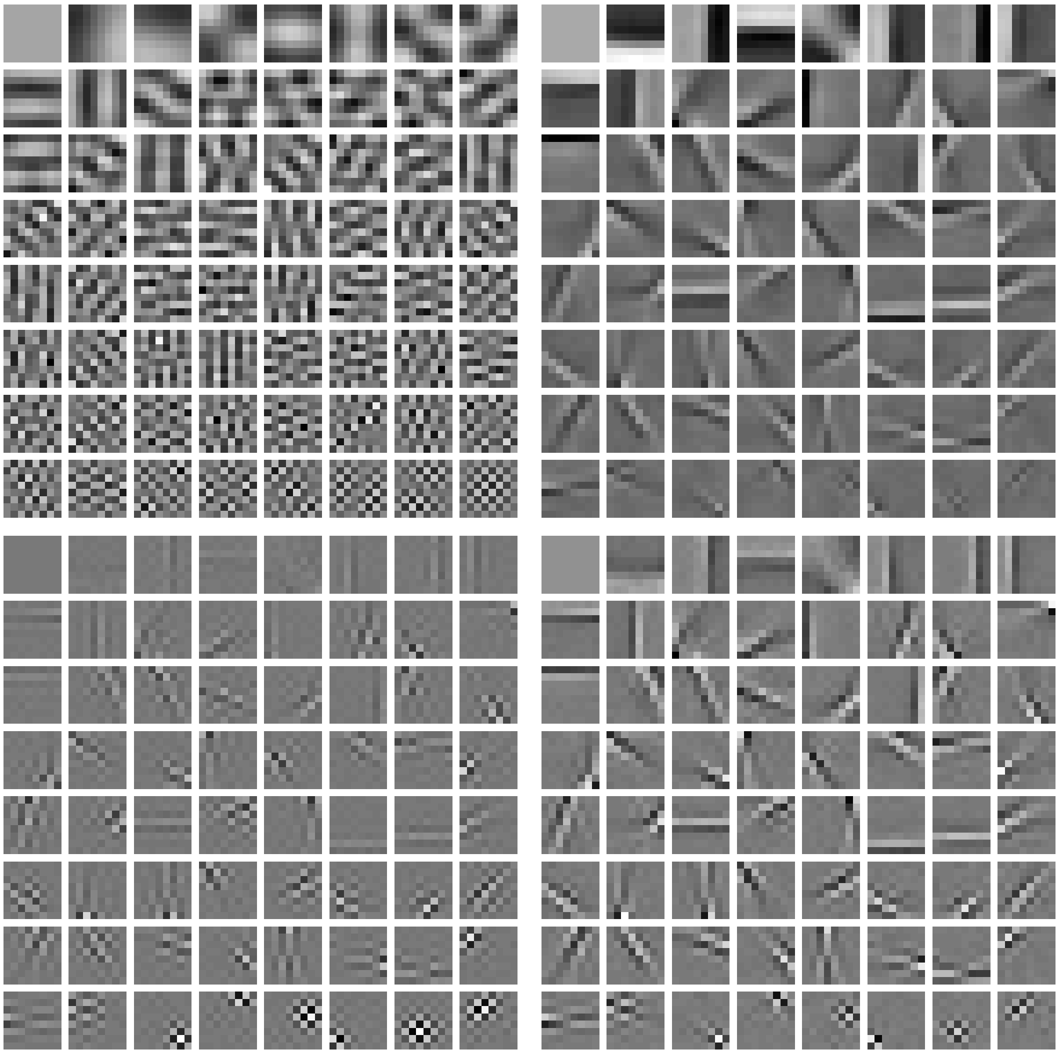

Figure 6: Transforms estimated for man-made scene data (8-by-8). The windows have been ordered according to mean frequency. Top-left: PCA transform. Top-right: ICA basis. Bottom-left: ICA separating filters. Bottom-right: Orthogonalized ICA transform.

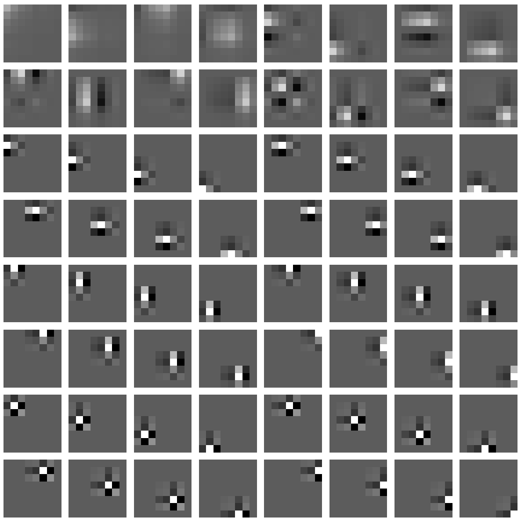

Figure 7: A wavelet basis, to be compared with the transforms in the preceding figures.

requires selecting suitable parametrizations and estimating the parameters; thus in these experiments we also evaluate how well the proposed parametrizations of Section 2.2 approximate the underlying densities.

Measuring the sparseness of the distributions can be done by almost any non-Gaussianity measure. We have chosen the most widely used measure, the normalized kurtosis. Recall that normalized kurtosis is defined as

$$\kappa(s) = \frac{E\{s^4\}}{(E\{s^2\})^2} - 3. \tag{40}$$

The average sparseness of each of the three transform sets (estimated in the previous section) was calculated the following way: First, we sampled 30000 image patches from the same dataset which was used for estimation of the transform. Then, we transformed these samples using the estimated PCA, ICA, and orthogonalized ICA transforms, and calculated the normalized kurtosis for each component of each basis separately. In Figure 8 the mean of these component kurtoses is displayed for each transform of each dataset. Because of the sparse structure in the images, all three transforms show supergaussian distributions, indeed even the individual pixel values show a mildly supergaussian distribution when the local mean has been subtracted. From the graph, it is seen that the ICA transform clearly finds a sparser representation than PCA. Also, note that the orthogonalized ICA representation is not quite as sparse as that given by standard ICA, but it is still far more supergaussian than PCA on average.

Next, we attempted to compare various parametrizations in the task of fitting the observed densities. We picked one component at random from the orthogonal 8×8 sparse coding transform for both natural scenes. First, using a non-parameteric histogram technique, we estimated the density of the component, and from this representation derived the log density and the shrinkage nonlinearity shown in Figure 9. Next, we fitted the parametrized densities discussed in Section 2.2 to the observed density. Note that in each case, the densities were sparser than the Laplace density, and thus the very sparse parametrization in (14) was used. In can be seen that the density and the shrinkage nonlinearity derived from the density model match quite well those given by nonparametric estimation. We did the corresponding experiments for a window size 16×16, shown in

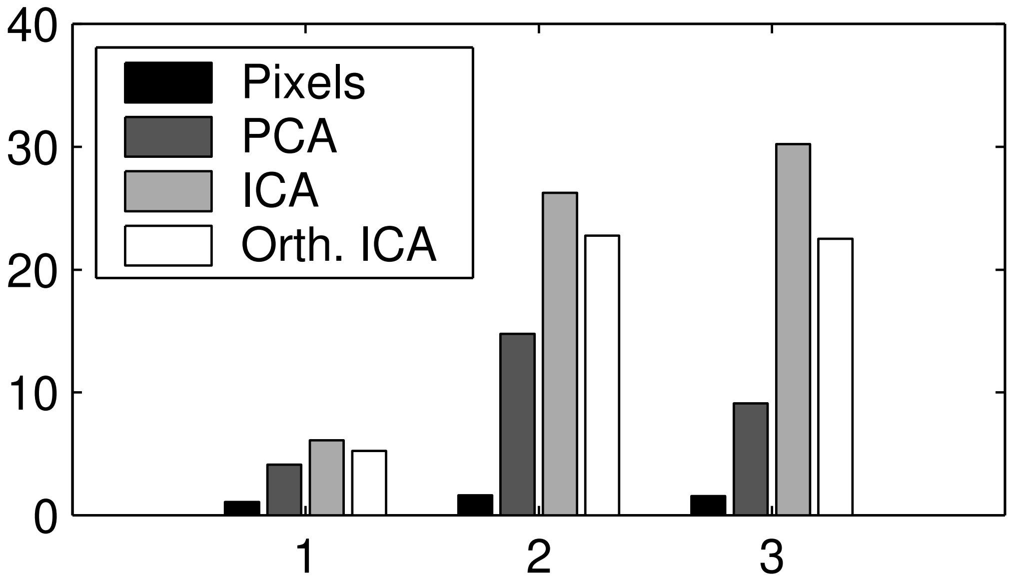

Figure 8: Mean of normalized kurtosis of components for the different datasets and different transforms within each dataset. (1) Natural Scenes, 8-by-8 patch bases. (2) Man-made scenes, 8-by-8 bases. (3) Man-made scenes, 16-by-16 bases.

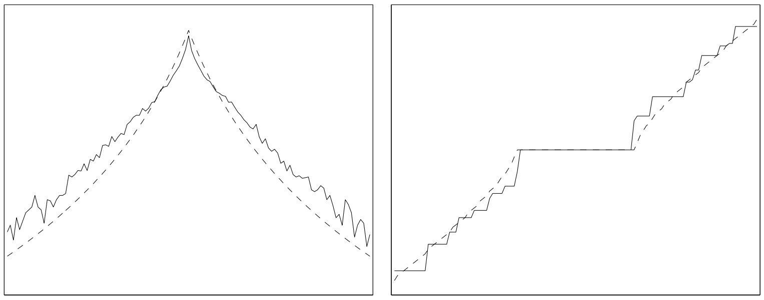

Figure 9: Analysis a randomly selected component from the orthogonalized ICA transforms of natural scenes, with window size 8×8. Top: Non-parametrically estimated log-densities (solid curve) vs. the best parametrization (dashed curve). Bottom: Non-parametric shrinkage nonlinearity (solid curve) vs. that given by our parametrization (dashed curve).

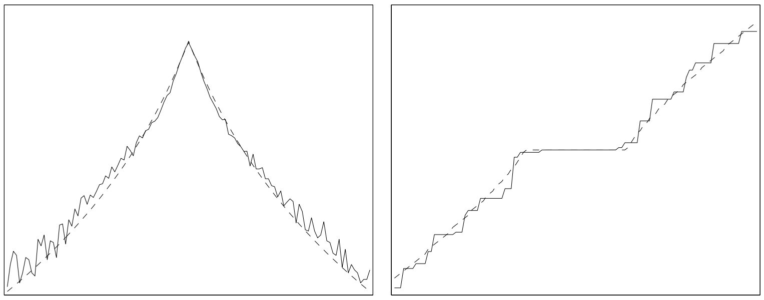

Figure 10: Analysis a randomly selected component from the orthogonalized ICA transforms of natural scenes, this time with window size 16×16. Top: Non-parametrically estimated log-densities (solid curve) vs. the best parametrization (dashed curve). Bottom: Non-parametric shrinkage nonlinearity (solid curve) vs. that given by our parametrization (dashed curve).

Figure 10; the results are not different in this case. The results for man-made scenes (not shown) were similar as well.

The Gaussian mixture model in (20) (not shown) gave very poor matches to the observed densities. The generalized Laplace density did give good matches, as observed by others [27, 34], but the drawback is that it does not allow for an analytical for form the shrinkage nonlinearity, which is why we prefer our parameterization.

As mentioned above, a possible variant of the method is to normalize the local image variance by dividing each input window by its norm. To experiment with such a denoising method, we had to estimate density parameters for such data. Thus, we proceeded as in the above experiments, but normalized each observed image patch to unit length before transforming it by the orthogonalized ICA transform. Now, the densities were not nearly as sparse, and the mildly sparse density model (11) was the appropriate parametrization, which can be seen from the figures.

In conclusion, we have seen that the components of the sparse coding bases found are highly supergaussian for natural image data in the basic case. This is true whether the images depict nature or man-made scenes.

8.5 Denoising results

To begin the actual denoising experiments, a random image from the natural scene collection was chosen for denoising, and Gaussian noise of standard deviation 0.3 was added (compared to a standard deviation of 1.0 for the original image). This noisy version was subsequently denoised using the Wiener filter to give a baseline comparison. Then, the sparse code shrinkage method was applied using the estimated orthogonalized ICA transform (8×8 windows), with the component nonlinearities as given by the appropriate estimated parametrization. Figure 11 shows the results of the first experiments. Visually, it seems that sparse code shrinkage gives the best noise reduction while retaining the features in the image. The Wiener filter does not really eliminate the noise. It seems as though our method is performing like a feature detector, in that it retains those features which are clearly visible in the noisy data but cuts out anything which is probably a result of the noise. Thus, it reduces noise effectively due to the nonlinear nature of the shrinkage operation.

In Figure 12, the exactly same parameters have been used with the exception that the noise level has been raised to 0.5. The results are qualitatively very similar those for the lower noise level.

The same experiments were then performed for data consisting of man-made scenes. We show the results for the noise levels of 0.5 in Fig. 13. This data seems to suit the method even better than the data from natural scenes. The image consists mainly of even areas with sharp edges. These are the type of features that the basis and density parameters have been adapted to, and the results are quite good.

To see the effect of window size, we made the same experiments with a larger window size. The results (not shown) did not really change at all.

Since the theory was derived under the strict assumption that the noise was Gaussian, it is interesting to see how the method performs when the noise is slightly non-Gaussian, e.g. Laplacian. The results of such an experiment are in Figure 14. Perhaps a bit surprisingly, the method performed quite well. For a possible explanation, see Section 7.1. Note, however, that when the noise becomes increasingly supergaussian (i.e. impulsive) the assumptions of the method begin to break down, and it would be better to use some regression-based methods as discussed in Section 7.1.

We also experimented with the variant of the method where normalization of the local variance is performed, see Section 8.2.3. The results are displayed in Figure 15. These results should be compared with those of Figures 11-12. This variant cuts noise in even areas very well, but leaves some noise in the area around edges.

We made the experiments with a median filter techique as well (not shown), which gave results that were qualitatively similar to the Wiener filter results. The results obtained by using the standard ICA transform (not shown) instead of the orthogonalized were clearly inferior to those obtained by the orthogonalized method.

To conclude, visual inspection of the results shows that the method performs very well. The denoising result is qualitatively quite different from those given by traditional filtering methods, and more along the lines of wavelet shrinkage and coring results [34, 35].[9]

9 Conclusion

We presented the method of sparse code shrinkage [19] using maximum likelihood estimation of nongaussian random variables corrupted by gaussian noise. In the method, we first determine an orthogonal basis in which the components of given multivariate data have the sparsest distributions possible. The sparseness of the components is utilized in ML estimation of the noise-free components; these estimates are then used to reconstruct the

[9] There is a large number of different variants of the wavelet shrinkage method, differing in choice of wavelet basis as well as the choice of shrinkage function. No one choice would have made a fair comparison, and thus we chose not to explicitly compare our method to wavelet shrinkage here.

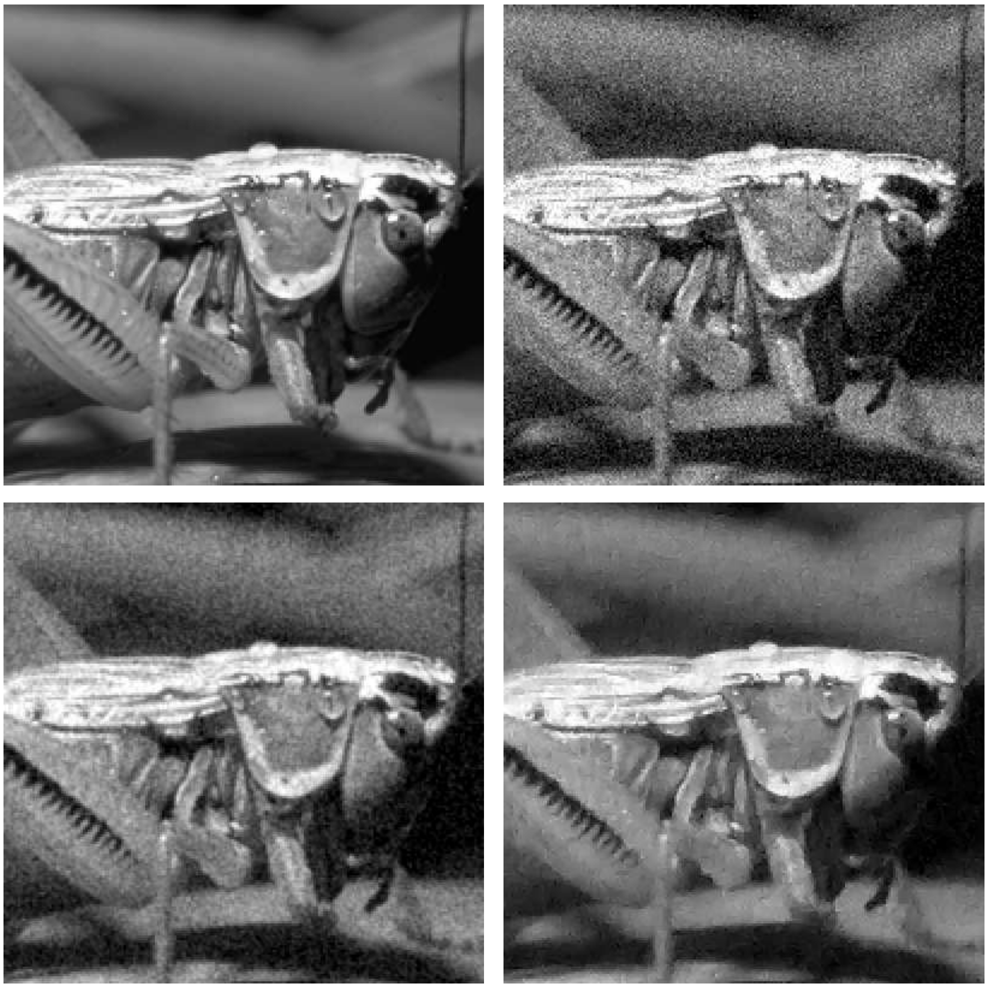

Figure 11: Denoising a natural scene (noise level 0.3). Top-left: The original image. Top-right: Noise added. Bottom-left: After wiener filtering. Bottom-right: Results after sparse code shrinkage.

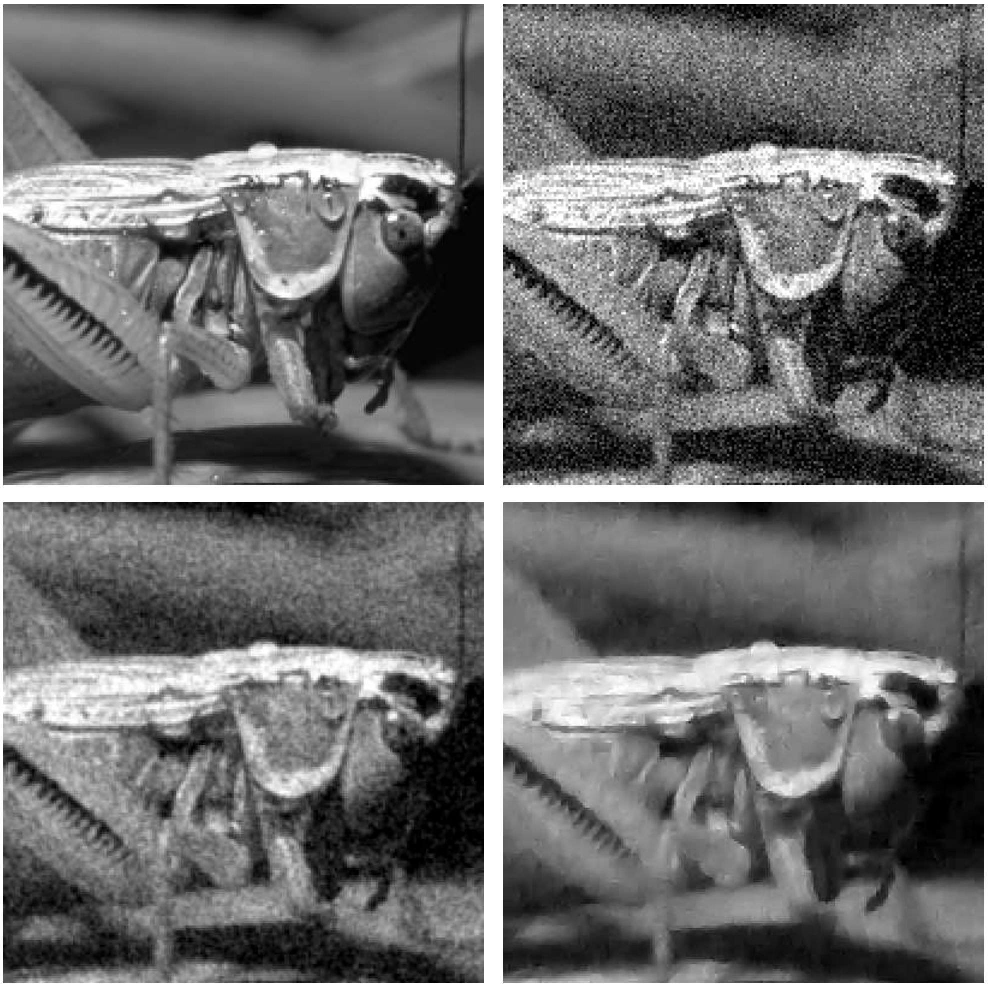

Figure 12: Denoising a natural scene (noise level 0.5). Top-left: The original image. Top-right: Noise added. Bottom-left: After wiener filtering. Bottom-right: Results after sparse code shrinkage.

Figure 13: Denoising a man-made scene (noise level 0.5). Top-left: The original image. Top-right: Noise added. Bottom: After wiener filtering. Bottom-right: Results after sparse code shrinkage.

Figure 14: Denoising a man-made scene with *Laplacian noise* (noise level 0.5). Top-left: The original image. Top-right: Noise added. Bottom-left: After wiener filtering. Bottom-right: Results after sparse code shrinkage.

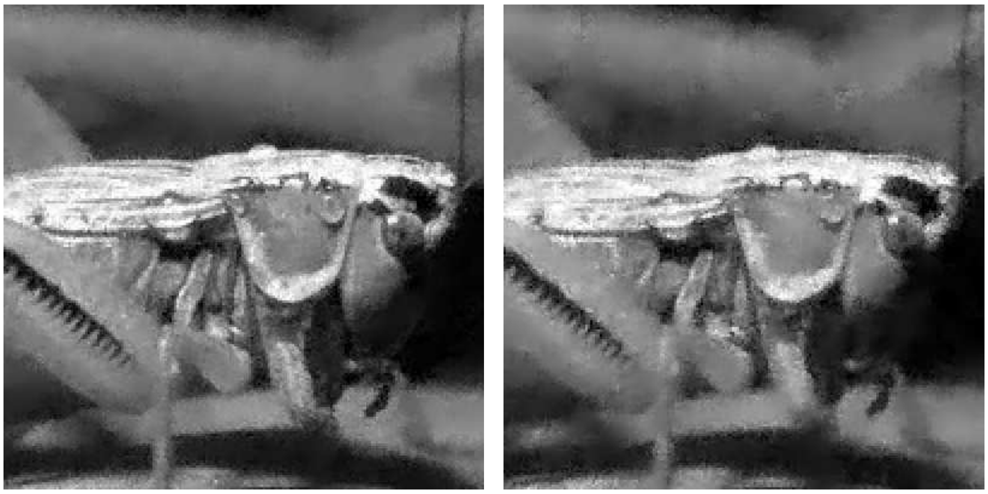

Figure 15: Left: Denoising using *local variance normalization*, with Gaussian noise at level 0.3 (compare with Figure 11). Right: Denoising using local variance normalization, when the noise level has been raised to 0.5 (compare with Figure 12).

original noise-free data by inverting the transformation. This is an approximation of the estimation of the noisy ICA model. The resulting method of sparse code shrinkage is closely connected to wavelet shrinkage and coring methods; in fact, it can be considered as a principled way of choosing the orthogonal wavelet-like basis based on data alone, as well as an alternative way of choosing the shrinkage nonlinearities.

References

[1] S.-I. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems 8*, pages 757–763. MIT Press, Cambridge, MA, 1996.

[2] H.B. Barlow. What is the computational goal of the neocortex ? In C. Koch and J.L. Davis, editors, *Large-scale neuronal theories of the brain*. MIT Press, Cambridge, MA, 1994.

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

[4] A.J. Bell and T.J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.

[5] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. Technical report, Department of Statistics, Stanford University, 1995.

[6] P. Comon. Independent component analysis – a new concept? *Signal Processing*, 36:287–314, 1994.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[8] Ingrid Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Math., Philadelphia, 1992.

[9] A. P. Dempster, N. Laird, and D. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *J. of the Royal Statistical Society, ser. B*, 39:1–38, 1977.

[10] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society ser. B*, 57:301–337, 1995.

[11] S.C. Douglas, A. Cichocki, , and S. Amari. A bias removal technique for blind source separation with noisy measurements. *Electronics Letters*, 34:1379–1380, 1998.

[12] B. Efron and C. Morris. Data analysis using Stein's estimator and its generalizations. *J. of the American Statistical Association*, 70:311–319, 1975.

[13] D.J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[14] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[15] J. Hurri, A. Hyvärinen, and E. Oja. Wavelets and natural image statistics. In *Proc. Scandinavian Conf. on Image Analysis '97*, Lappenranta, Finland, 1997.

[16] A. Hyvärinen. Independent component analysis in the presence of gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22:49–67, 1998.

[17] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3):626–634, 1999.

[18] A. Hyvärinen. Gaussian moments for noisy independent component analysis. *IEEE Signal Processing Letters*, 6(6):145–147, 1999.

[19] A. Hyvärinen. Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Computation*, 11(7):1739–1768, 1999.

[20] A. Hyvärinen, R. Cristescu, and E. Oja. A fast algorithm for estimating overcomplete ICA bases for image windows. In *Proc. Int. Joint Conf. on Neural Networks*, Washington, D.C., 1999.

[21] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[22] C. Jutten and J. Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[23] J. Karhunen, E. Oja, L. Wang, R. Vigário, and J. Joutsensalo. A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8(3):486–504, 1997.

[24] M. Kendall and A. Stuart. *The Advanced Theory of Statistics.* Charles Griffin & Company, 1958.

[25] M. Lewicki and B. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. Am. A: Optics, Image Science, and Vision*, 16(7):1587–1601, 1998.

[26] D. G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, 1969.

[27] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on PAMI*, 11:674–693, 1989.

[28] G. P. Nason. Wavelet shrinkage using cross-validation. *Journal of the Royal Statistical Society, Series B*, 58:463–479, 1996.

[29] E. Oja. The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25–46, 1997.

[30] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[31] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[32] J. J. K. O'Ruanaidh and W. J. Fitzgerald. *Numerical Bayesian Methods Applied to Signal Processing.* Springer, 1996.

[33] M. Schervish. *Theory of Statistics.* Springer, 1995.

[34] E. P. Simoncelli and E. H. Adelson. Noise removal via bayesian wavelet coring. In *Proc. Third IEEE Int. Conf. on Image Processing*, pages 379–382, Lausanne, Switzerland, 1996.

[35] T. Yu, A. Stoschek, and D. Donoho. Translation- and direction- invariant denoising of 2-d and 3-d images: Experience and algorithms. In *Proceedings of the SPIE, Wavelet Applications in Signal and Image Processing IV*, pages 608–619, 1996.

A The 1-D MMSE estimator

Here we derive the first-order approximation of the MMSE estimator in (24), showing that it is identical to the one in (7). Let us denote $\tilde{g}(s) = g(s) - s$. Clearly, $\tilde{g}$ is infinitesimal, so let us denote by $o(\tilde{g})$ terms that are of lower order than $E\{\tilde{g}(s)\}$. We have:

$$\begin{aligned} E\{(s-g(y))^2\} &= E\{[s-g(s)-g'(s)\nu - \frac{1}{2}g''(s)\nu^2]^2\} + o(\sigma^2) \\ &= E\{[-\tilde{g}(s)-(1+\tilde{g}'(s))\nu - \frac{1}{2}\tilde{g}''(s)\nu^2]^2\} + o(\sigma^2) \\ &= E\{\tilde{g}(s)^2\} + E\{(1+\tilde{g}'(s))^2\}\sigma^2 + E\{\tilde{g}(s)g''(s)\}\sigma^2 + o(\sigma^2). \end{aligned} \tag{41}$$

From the second line it can be seen that the approximation error occurred by this Taylor expansion is of order $o(\tilde{g})o(\sigma^2)$. Next we have

$$(1+\tilde{g}(s))^2 = 1 + 2\tilde{g}(s) + o(\tilde{g}) \tag{42}$$

and

$$\tilde{g}''(s) = g''(s) \tag{43}$$

which means that (41) can be written as

$$E\{(s-g(y))^2\} = \sigma^2 + E\{\tilde{g}(s)^2\} + 2E\{\tilde{g}'(s)\}\sigma^2 - \sigma^2 o(\tilde{g}) + o(\sigma^2). \tag{44}$$

We can now use variational calculus [26] to find the $\tilde{g}$ that minimizes the mean-square error. Neglecting terms of smaller order, and equating the variational derivative of the right-hand side of (44) with respect to $\tilde{g}$ to zero, we obtain

$$p(s)g(s) - p(s)s + \sigma^2 p'(s) = 0 \tag{45}$$

which gives

$$g(s) = s - \sigma^2 f'(s) \tag{46}$$

with $f = p'/p$, and we have indeed (7).