Technical Report DIKU-TR-2001-02
Institute of Computer Science
University of Copenhagen
Universitetsparken 1
DK-2100 Copenhagen

Denmark

The Problem of Sparse Image Coding*

Arthur E.C. Pece
aecp@diku.dk

April 10, 2001

Abstract

Linear expansions of images find many applications in image processing and computer vision. Overcomplete expansions are often desirable, as they are better models of the image generating process. Such expansions lead to the use of sparse codes. However, minimizing the number of non-zero coefficients of linear expansions is an unsolved problem. In this article, a generative-model framework is used to analyze the requirements, the difficulty, and current approaches to sparse image coding.

Keywords: ICA, atomic decomposition, adaptive representations, Gabor function, matching pursuit

1 Introduction

This article deals with a problem that can be formulated either in a generative-model framework or in an image-coding framework.

The generative-model formulation is as follows: we observe a multivariate random signal $\mathbf{x} \in \mathbb{R}^m$, which in the context of this article is an image consisting of m pixels. We suspect that this image is generated as a linear combination of n independent random sources $\mathbf{s} \in \mathbb{R}^n$ according to the equation:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\nu} \tag{1}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is called the mixing matrix and $\boldsymbol{\nu} \in \mathbb{R}^m$ is iid (independent, identically distributed) gaussian noise. We want to estimate the sources $\mathbf{s}$ on the basis of the observations $\mathbf{x}$. This framework is closely related to the classical Kalman filter [35, 62].

The coding formulation is as follows: we want to encode the image $\mathbf{x}$ into a linear expansion $\mathbf{c} \in \mathbb{R}^n$ such that

$$\mathbf{x} = \mathbf{B}\mathbf{c} + \mathbf{r} \tag{2}$$

*A shorter version of this report has been submitted for publication in the special issue on Statistics of Shapes and Textures of the *Journal of Mathematical Imaging and Vision.*

where the columns of $\mathbf{B} \in \mathbb{R}^{m \times n}$ are called the code vectors, and the residual $\mathbf{r} \in \mathbb{R}^m$ is small (in L_2 norm) relative to the image. The encoding must be done in such a way that the coefficients can be used more efficiently than the image grey-levels for the purposes of compression or further processing.

The equivalence of the two formulations is immediately apparent from information theory: the coefficients should be statistically independent for efficient compression [25, 47]; they should also be independent if they are to be used as statistical evidence for further inference [8, 5]. Therefore, assuming that the linear generative model of Eq.1 is physically valid, the code vectors should be the same as the columns of the mixing matrix and optimal coding is equivalent to source estimation.

If the linear model in Eq.1 is not valid, then, in general, *complete* independence of the coefficients cannot be achieved. However, independence can often be approached by methods based on a linear model with independent sources. Statistical dependencies do exist between coefficients of linear expansions of natural images [61, 68], but no method has yet been developed to exploit these dependencies for coefficient estimation, as opposed to coefficient compression [63].

Having established the equivalence of the two formulations, we shall use the term *coding* for the process of estimating the sources/coefficients. This terminology is adopted for convenience and does not imply that the generative-model framework will be ignored: on the contrary, this framework is adopted throughout the article. Therefore, the term "coefficients", and the symbol $\mathbf{c}$, should always be interpreted as equivalent to "source estimates". Similarly, the residual $\mathbf{r}$ is to be interpreted as an estimate of the noise $\boldsymbol{\nu}$. The term *codebook* will be used to indicate the estimate of (or approximation to) the mixing matrix. The term *code vectors* will be used for the columns of the codebook, while the term *mixing vectors* will be used for the columns of the mixing matrix. The term *mixing/code vectors* will be used to include both sets of vectors, when the mixing matrix and the codebook can be assumed to be identical.

Estimation of the mixing matrix from an ensemble of images constitutes *learning*. Another article in this issue [31] discusses some approaches to learning and contains references to the relevant literature.

In the present article, section 2 contains a brief review of linear models and of their limitations. Section 3 contains a discussion of some empirical constraints on generative models for natural images. Two implications of these constraints are that the codebook should be overcomplete and the prior density of the sources should be *super-gaussian* (*i.e.* long-tailed, having a large kurtosis). In the context of this paper, *sparse coding* is defined as coding based on a linear model with an overcomplete codebook and statistically independent sources with super-gaussian densities. This definition makes explicit the generative model that motivates coding with a sparse coefficient vector, *i.e.* a coefficient vector with a small number of non-zero coefficients[1].

Section 4 argues that the super-gaussian nature of the sources is a necessary working hypothesis, both for learning and for compression, when the codebook is overcomplete. This argument is independent of the empirical evidence for super-gaussian sources. To develop the argument, section 4 introduces the concept of cross-talk between sources. Section 5 uses the concept of cross-talk to show that a highly super-gaussian source density is a necessary working hypothesis also for coding. A simple mixture model for the source densities is introduced to develop the argument.

Section 6 reviews two approaches to sparse-coding. One approach is based on explicit generative models with smooth prior source densities, and uses gradient-based optimization methods. The other approach is based on combinatorial optimization. The concept of cross-talk is used to show some weaknesses of both classes of methods. Finally, section 7 concludes the article with a brief review of some applications of sparse coding and remarks on its biological relevance.

Sparse coding is very much an open problem. The aim of this paper is to illustrate the difficulty of the problem, to bring together some results revelant to the problem, and to review approaches that have been followed to date.

[1] One additional assumption that needs to be made, to motivate a sparse coefficient vector, is that the prior densities of the sources have zero mode. This assumption is discussed in section 3.

1.1 Related research

The principle of finding an economical description of data can be traced back to Occam's razor and has important applications in statistics [24, 41] The generative-model formulation outlined in section 2 has two advantages:

- following a Bayesian framework, the "economy" of a code can be conveniently expressed as the prior probability distribution of source values;
- the Bayesian (subjective) interpretation of probabilities is unnecessary, because all relevant prior probabilities can be (objectively) defined as relative frequencies and, at least in principle, estimated from image statistics.

The term *sparse coding* probably originates in the area of neural networks, when it was realised that the efficiency of auto-association of binary patterns could be improved if most of the patterns are sparse, *i.e.* if the probability of code bits being equal to zero is much larger[2] than the probability of bits being equal to one [52, 8].

Within the context of linear models, the methods developed for sparse coding (reviewed in section 6) originate from two lines of research: overcomplete ICA (independent-component analysis) and matching pursuit.

The first line of research can be traced back to the idea that an important task of sensory perception consists in reducing the redundancy of sensory signals [2, 3, 4]. More recently, a distinction has been made [20] between compact codes (in which the dimensionality of the input is reduced, as in principal-component analysis) and sparse codes. This paradigm has emphasized the *need* for sparse codes, rather than specific methods for sparse coding. In the last decade, this paradigm has led to learning algorithms for the mixing matrix of Eq.1 (*e.g.* [9, 49, 31]). However, the difficulty of coding, *i.e.* of using the mixing matrix, has generally, though not always, been underestimated.

The second line of research originates within the wavelet community and goes back at least to the development of orthogonal-basis selection [14], but it is primarily matching pursuit [40] that stimulated interest in this paradigm. Algorithms for finding sparse linear expansions have been developed in this framework, but without a statistical motivation for their use.

One of the motivations for this article is helping to bridge the gap between the ICA paradigm, which has provided much of the motivation for sparse coding through the generative-model framework (*e.g.* [49]) and the matching-pursuit paradigm, which has provided the most effective methods.

Research in image statistics [28, 61] is closely related to the ICA paradigm. The implications of image statistics which are most important for sparse coding will be reviewed in subsection 3.2.

The theory of shiftable multiscale transforms [66], which is an extension of the sampling theorem, provides the motivation for overcomplete codenooks and is at the basis of subsection 3.3.

2 The generative-model framework for image coding

In this section, the statistical framework is introduced and used to derive some well-established coding methods for the cases of non-overcomplete mixing matrix and/or gaussian prior densities of the sources. In this way, methods for sparse coding are put in context. The limitations of this entire class of models are outlined at the end of the section.

The most important subsections are 2.1, 2.2, 2.6. Subsections 2.3, 2.4, 2.5 can be skipped as long as subsection 2.6 is clear. Subsections 2.7, 2.8 are not essential to follow the rest of the paper.

[2] or, equivalently, much smaller

2.1 Some terminology and assumptions

Images are represented as vectors for simplicity. To avoid confusion, the term *image space* defines the two-dimensional image plane; the term *grey-level space* defines the m-dimensional space of image pixel values; the term *coefficient space* defines the n-dimensional space of coefficient values. The term *projection* will always indicate a projection onto a vector, or set of vectors, in grey-level space (*e.g.* onto a code vector).

In this paper, it is assumed that the mixing matrix is known and the error in its estimate can be ignored. We also assume, without loss of generality, that mixing/code vectors are normalized to unit length. An alternative (and incompatible) assumption is that all source densities have unit variance.

2.2 Bayesian formulation

The formulation that will be followed is similar to that found in [49]. For a given mixing matrix, the posterior density of the sources $\mathbf{s}$, given the image $\mathbf{x}$, can be obtained from Bayes' theorem:

$$p(\mathbf{s}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{s})p(\mathbf{s})}{p(\mathbf{x})} \tag{3}$$

As usual, it is convenient to apply a negative-logarithm transformation to all densities. We begin by defining the posterior negative log-density:

$$H(\mathbf{s}|\mathbf{x}) = -\log p(\mathbf{s}|\mathbf{x}) \tag{4}$$

As pointed out in the Introduction, we take the view that coefficients are source estimates, and we write $H(\mathbf{c}|\mathbf{x})$ for the posterior log-density of the estimate. A commonly-used estimation criterion is *MAP* (maximum *a posteriori*) estimation, by which the posterior negative log-density H is the *objective function* to be minimized. Denoting the *MAP* estimate of $\mathbf{s}$ by $\hat{\mathbf{s}}$, the *MAP* criterion can be expressed as

$$\hat{\mathbf{s}} = \mathrm{argmax}_{\mathbf{c}} H(\mathbf{c}|\mathbf{x}) \tag{5}$$

The *MAP* criterion is appropriate for the purpose of image compression, because the *MAP* coefficient vector can be encoded with shortest length[3]. For other purposes, it would be desirable to estimate the dispersion of the posterior log-density, as well. Unfortunately, this is difficult if the prior densities of the sources are not gaussian.

For convenience, we define the squared norm of the residual as the *reconstruction error*:

$$\begin{aligned} R &= ||\mathbf{r}||^2 \\ &= ||\mathbf{x} - \mathbf{A}\mathbf{c}||^2 \end{aligned} \tag{6}$$

As pointed out, the noise is iid gaussian and zero-mean. This assumption is implicit in most sparse-coding methods. Using Eq.1, Bayes' theorem, the assumptions of gaussian noise and independent sources, and the above definitions, we can write an expression for the objective function:

$$H(\mathbf{s}|\mathbf{x}) = \frac{R}{2\sigma_\nu^2} + \sum_i h_s(s_i) + C_0 \tag{7}$$

where $h_{\mathbf{s}}$ is the prior log-density of the sources; σ_ν^2 is the variance of the noise density; and the constant C_0 is not a function of the sources[4]. The first term on the right-hand side of Eq.7 is the *data term* or likelihood: it expresses the dependence of the objective function on the image data. The second term is the *penalty term*: it expresses the influence of the prior density.

[3] Strictly speaking, it is the coefficients plus residual that can be encoded with shortest length.

[4] For simplicity, we assume that the variance of the noise σ_ν^2 is known *a priori*; this assumption is implicit in most sparse coding algorithms. If the variance were to be estimated together with coefficients and residual, the term C_0 could not be neglected.

2.3 The low-noise limit

If the variance of the noise is very small compared to the variance of the sources, then the dominant term on the right-hand side of Eq.7 is proportional to the reconstruction error R. In this case, the first requirement on the coefficients is that they minimize R, *i.e.* the *MAP* estimate is a solution of the linear system

$$\nabla_{\mathbf{c}} R = \mathbf{0} \tag{8}$$

where $\mathbf{0}$ is a vector of n zero-valued elements.

If the mixing/code vectors are linearly independent [$n = \text{rank}(\mathbf{A})$], then there is a unique solution with minimum reconstruction error (*MRE*). In the low-noise limit, this is also the *MAP* estimate. This solution can be obtained by standard methods for solving linear systems if the matrix is square ($m = n$), standard least-squares methods otherwise [26].

Note the paradox that, in the low-noise limit, the estimated squared norm of the noise (*i.e.* the reconstruction error) is the dominant term in the objective function.

2.4 Underconstrained systems

The linear system of Eq.8 is said to be underconstrained if the mixing/code vectors are not linearly independent, *i.e.* if $\text{rank}(\mathbf{A}) < n$. If the system is underconstrained, there is a linear sub-space of the coefficient space, with dimension $n - \text{rank}(\mathbf{A})$, for which R is minimal. We call this the *MRE* subspace. In the low-noise limit, the *MAP* coefficient vector is the vector that minimizes the penalty term within the *MRE* sub-space.

Note that no assumption has yet been made about the number of pixels, m. If the system is full-rank [$m = \text{rank}(\mathbf{A})$] then the residual is identically zero in the *MRE* subspace.

A mixing matrix with $m < n$ is defined as overcomplete. This definition is somewhat confusing, since it can be the case that $\text{rank}(\mathbf{A}) < m < n$, in which case the matrix is overcomplete, but not complete, *i.e.* the codebook does not span the grey-level space, This can happen if there are no mixing/code vectors corresponding to high spatial frequencies or to the DC component of images[5]. An overcomplete matrix is always underconstrained, but the reverse does not hold in general.

2.4.1 A special case: gaussian source densities

We shall briefly consider the case of gaussian sources, to illustrate the hidden gaussian assumption behind some coding methods. In the gaussian case, the objective function to be minimized is quadratic:

$$H = \frac{R}{2\sigma_\nu^2} + \sum_i \frac{c_i^2}{2\sigma_i^2} + C_1 \tag{9}$$

where σ_i^2 is the variance of source i and the constant C_1 is independent of the coefficients. For a quadratic penalty, the *MAP* coefficients can be obtained by standard methods [26].

In the low-noise limit, the *MAP* estimate is obtained by minimizing the (quadratic) penalty

$$\sum_i \frac{c_i^2}{2\sigma_i^2}$$

within the *MRE* subspace. If we further assume that the variance is the same for all sources, *i.e.* $\sigma_i^2 = \sigma_{\mathbf{s}}^2$, then the *MAP* estimate is the minimum-coefficient-norm solution in the *MRE* subspace. This is given by

$$\hat{\mathbf{s}}_{MCN} = \mathbf{W}\mathbf{x} \tag{10}$$

[5] Note that a mixing matrix which does not span the grey-level space does not generate problems, because the set of natural images is constrained to the subspace spanned by the mixing matrix (except for the noise) and therefore can be represented by the codebook (except for a residual corresponding to the noise).

where the matrix

$$\mathbf{W} = \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T\right)^{-1}$$

is the pseudo-inverse of the mixing matrix. The columns of the pseudo-inverse are the *dual vectors* of the mixing/code vectors and can be obtained by the method of frames [15]. In general, the dual vectors of unit code vectors are not themselves unit vectors.

If the prior density of the sources is not gaussian, then H is not quadratic. A non-quadratic objective function cannot be minimized by linear methods and usually requires iterative optimization methods. Note the paradox of nonlinear methods being required for inference with a linear generative model.

2.5 Noise suppression

When the source and noise densities are both gaussian, the optimal method of noise removal is the Wiener filter [59]. If the sources are super-gaussian, then a better distinction between signal and noise is possible because of their different densities.

A system with significant noise can be considered equivalent to an underconstrained system with zero noise: each element of the noise vector becomes another source, with the corresponding mixing vector having only one non-zero element. The number of sources then becomes $n + m > n$. This would suggest that iterative optimization methods are required even when the mixing matrix is not overcomplete, if noise is present and the source density is not gaussian.

However, if the mixing/code vectors are linearly independent, the projections of the image onto the dual vectors

$$\mathbf{g}_D = \mathbf{A}^{-1}\mathbf{x}$$

are the sums of the corresponding sources plus noise:

$$\mathbf{g}_D = \mathbf{s} + \mathbf{A}^{-1}\boldsymbol{\nu} \tag{11}$$

Given that the elements of the vector $\boldsymbol{\nu}$ are statistically independent, it can be easily seen that correlations between the elements of the vector $(\mathbf{A}^{-1}\boldsymbol{\nu}) \in \mathbb{R}^n$ are induced by non-orthogonality between the dual vectors. Nonetheless, it is usually possible to ignore such correlations and use the approximation

$$H(\mathbf{s}|\mathbf{x}) \approx \sum_j h(s_j|g_j) \tag{12}$$

This approximation leads to *pointwise* estimation of the sources, *i.e.* the coefficients can be obtained by applying a soft threshold to the projections. This method is known as coring or wavelet shrinkage [18, 65].

In principle, if the code vectors (and hence the dual vectors) are not mutually orthogonal, then a better estimate of the noise could be obtained by taking into account correlations between the noise components of the projections. However, this is not necessary if the signal-to-noise ratio in each projection is large.

Coring is also used with overcomplete codebooks [65], in which case the coefficients are estimated by the method of frames. There is an apparent inconsistency between the super-gaussian assumption implicit in coring, and the gaussian assumption implicit in the method of frames. This is not a problem in practice when the coefficients are are not of interest as source estimates, but only used to reconstruct the image with increased signal-to-noise ratio.

In conclusion, we can consider coring as a coding method if the codebook is not underconstrained, but only as a denoising method when the codebook is underconstrained. Coring based on a model of non-independent sources (*e.g.* [68]) is beyond the scope of this paper.

2.6 Summary

The coding methods discussed in this section can be schematized in a table that puts sparse coding into context:

Table 1: **Coding Methods**

	constrained low-noise (subsection 2.3)	constrained noisy (subsection 2.5)	under-constrained
sub-gaussian	*MRE* [a]	?[b]	(none)[c]
gaussian	*MRE* [a]	Wiener filter [59]	frames, SVD, CG[d] (subsubsection 2.4.1)
super-gaussian	*MRE* [a]	coring [18, 65]	sparse coding (section 6)

[a] The *MRE* solution can be obtained by standard linear algebra [26, 59].
[b] No method has been developed for this case.
[c] Section 5 will argue that no method can be developed for this case.
[d] The method of frames can be used in the low-noise limit if the codebook constitutes a frame [15], otherwise SVD (singular-value decomposition) or CG (conjugate-gradient method) are applicable [26].

As can be seen, there is more similarity between the rows, rather than between the columns, of the table:

- for constrained, low-noise systems, the prior source densities are irrelevant (see subsection 2.3);
- for constrained, noisy systems, the estimate is non-iterative and pointwise (see subsection 2.5).

It is interesting to contrast Table 1 with a similar table of learning methods:

Table 2: **Learning Methods**

	constrained low-noise	constrained noisy	under-constrained
sub-gaussian	ICA[a]	IFA[b]	(none)[c]
gaussian	PCA, ZCA, *etc.* [d]	factor analysis (*e.g.* [62])	(none)[c]
super-gaussian	ICA[a]	IFA[b]	ICA extensions [31]

[a] ICA: independent component analysis [9, 32].
[b] IFA: independent factor analysis [1].
[c] No methods are possible for this case (see section 4).
[d] PCA: principal component analysis (see *e.g.* [62]); ZCA: zero-phase component analysis [9]; see also [21, 58].

Given that factor analysis is closely related to PCA, and IFA is closely related to ICA, it emerges that in Table 2 there is more similarity between rows than between columns: the main difference is between gaussian and non-gaussian source densities, leading to PCA-type or ICA-type learning algorithms, respectively.

2.7 A note on image compression

Note that the mapping from sources plus noise to images is many-to-one, while, in a deterministic coding algorithm, the mapping from images to *MAP* estimates plus residual is one-to-one. This implies that the density of the coefficients will be more concentrated in coefficient space (and will have lower entropy) than the density of the sources. As a consequence, if the coefficients are the true *MAP* estimate, then the coefficients can be encoded more efficiently than the sources. This is because there has been some information loss in the mixing of the sources, and this lost information is not encoded.

In general, we must distinguish between four densities: the *actual* densities and the *estimated* densities of both sources and coefficients. The estimated source density plays a role in some coding methods and will be of interest in Section 6. The estimated coefficient density is used in entropy coding for image compression [25, 47]. In the context of this paper, the coefficient densities (actual or estimated) do not require further attention.

2.8 Limitations of the approach as applied to natural images

One advantage of the generative-model formulation is that the limitations of this class of methods become evident: these limitations are the assumptions on which the generative model is based, specifically:

- linearity of the mapping from sources to image;
- statistical independence of the sources;
- no dynamics, *i.e.* statistical independence of source values from one image to the next in a sequence;
- gaussian iid noise.

Another assumption, which is not immediately evident, is that a continuous operator (mapping sources at any location, orientation and scale in a continuum, to images) can be well-approximated by matrix-vector multiplication. This assumption will be discussed in a forthcoming paper [54].

2.8.1 Linearity

Images are certainly not generated as linear superpositions of objects: one needs only to think of partial occlusion and cast shadows. Nonetheless, linear models have proved very useful in image coding, low-level vision (*e.g.* motion, colour, texture), pattern recognition (*e.g.* face recognition and medical imaging), and modelling of low-level biological vision. Even though linear models are not appropriate for all visual tasks, there is no need to abandon them altogether.

A less fundamental objection to linear models is that a simple nonlinear transformation can account for asymmetrical grey-level densities. Specifically, a good empirical model of image generation would be given by a system of nonlinear equations of the following form:

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \boldsymbol{\nu} \tag{13}$$

$$x_i = x_0 \exp{(y_i - \bar{y})} \qquad (0 < i \leq m) \tag{14}$$

where x_0 is a scaling factor common to all grey levels in a given image (usually of no interest in itself) and $\bar{y}$ is the average value of the elements of $\mathbf{y}$. One advantage of this model is that it restricts grey levels to positive values.

The assumption of linearity is equivalent to truncation of the Taylor expansion of Eq.14 around $\bar{y}$. In practice, the gamma correction applied by most cameras partially compensates for the exponential transformation of Eq.14.

2.8.2 Independence of sources

Statistical independence of the sources is also limited by occlusion: templates generated from different sources should be able to occlude each other in a sound generative model. Models based on statistically-independet sources will break down when occlusion is significant.

A different form of dependency has received much interest (*e.g.* [61, 68]) and has been exploited in the JPEG2000 compression standard [63]. This is *dependence through local variance*: the variances of sources, corresponding to mixing vectors close in image space and spatial frequency, are significantly correlated, even when all correlations between these sources are eliminated. This model provides very good fits for the densities of image projections onto wavelet codebooks [68]. However, this model does not eliminate the need for sparse coding: this is because, as mentioned in the Introduction, current methods allow the local variance to be estimated only after the sources themselves have been estimated.

2.8.3 Lack of dynamics

The lack of dynamical equations is not, properly speaking, a limitation: the model is an *observation model* and therefore independent from any dynamical model. The independence between dynamical model and observation model is well illustrated by the Kalman filter [35, 62]. Dynamical equations carry over information from one image to the next and therefore should simplify the coding problem, provided of course that the equations are a good model of the underlying dynamics. Dynamical equations specify a dynamic modification of the source prior densities: posterior densities at one frame in an image sequence become the prior densities at the next frame.

2.8.4 Gaussian iid noise

The first step in defining a noise model is to decide what is the signal and what is the noise. One possible definition of noise is that noise includes all sources corresponding to mixing vectors with only one non-zero element. This definition implies that the noise on each pixel is independent, and therefore, given the translational invariance of natural images, iid.

From the property of scale invariance (see section 3.3), one would infer that mixing vectors with only one non-zero element are generated by scaling of larger mixing vectors. (A closely related hypothesis is formulated in [42].) In this case, the observed noise arises from the corresponding sources multiplied by the DC component, and possibly from aliasing, of the scaled mixing vectors. If this inference is valid, then the averaging of several sources into a single pixel would make the noise density more gaussian than the densities of any of the sources.

The above paragraph is only meant to show that gaussian iid noise is a reasonable approximation. A detailed model is outside the scope of this paper. We note only that, if noise arises from sources at small spatial scales, then the model of dependence through local variance [68] becomes applicable to these sources and leads to the prediction that the local noise variance is proportional to the local source variance.

Whatever the limitations of the generative model, the reason to investigate it further is simple and compelling: the model is implicit in many methods in low-level vision.

3 Constraints on models of natural images

This section shall briefly review evidence for three basic constraints on generative models for natural images:

- the prior density of the sources should be *super-gaussian*, *i.e.* it should have a larger kurtosis than a gaussian density;

- any translation, rotation or scaling in the image plane of a mixing vector (within bounds) should still be a mixing vector;
- mixing/code vectors should be localized in the image plane.

The second and third constraints, taken together, imply that the codebook should be overcomplete. Linear models that do not satisfy these constraints can be useful for some applications; for instance, a non-overcomplete gaussian model is implicitly assumed by the JPEG image-compression standard. However, better models (and therefore better compression) can be achieved by satisfying all constraints.

3.1 Assumptions about source densities

Constraints arising from empirical evidence should be distinguished from *assumptions* which are made either because there is no loss of generality or because they are useful or even essential (*working assumptions*).

The basic working assumption about the sources, already mentioned, is statistical independence. Two other common assumptions are that the source densities are symmetrical and zero-mean. The zero-mean assumption can be made without loss of generality: subtracting the mean image from all images is equivalent to subtracting the mean value from all sources. Translational invariance requires that all grey levels be identical in the mean image[6]. Since the DC component is usually encoded separately (*i.e.* all code vectors have elements with zero mean), the mean grey level can be ignored.

As mentioned in the Introduction, the equivalence between coding based on an overcomplete codebook and statistically-independent, super-gaussian sources is equivalent to sparse coding only under the zero-mean assumption (or, in the case of non-symmetrical densities, a zero-mode assumption).

3.2 Super-gaussianity

The super-gaussian constraint for source densities is required because grey levels themselves are not gaussian, at least not in the tails of the density [61, 28]. Differences between neighbouring pixel grey levels have even more long-tailed densities [28]. These differences, as well as image projections onto wavelets [20, 38], have super-gaussian densities of the generalized laplacian form:

$$p(y) = \frac{1}{Z} \exp\left(- \left| \frac{y}{\sigma} \right|^{\beta} \right) \tag{15}$$

where y is a random variable with a generalized laplacian density, Z is a normalization constant, σ is a scale parameter, and $\beta > 0$ is a parameter determining the shape of the density: the smaller β, the greater the kurtosis, *i.e.* the more super-gaussian the density of y. Observed values of β are generally in the range $[0.5, 1]$. Note that this density has finite variance.

Some of these results have been obtained with log-transformed grey levels, but this transformation, by itself, would not turn a gaussian into a generalized laplacian.

This observation explains why wavelets are more efficient than Fourier transforms for image compression (see *e.g.* [39]): if the grey-level density were jointly gaussian, then shift-invariance prescribes that Fourier coefficients would be statistically independent, and therefore it would not be possible to achieve higher compression by a different linear transform.

If the sources were gaussian, then the grey levels, and any linear combination of grey levels, being sums of zero-mean gaussian variables, would also be gaussian. Therefore, the implication of image statistics is that the sources are super-gaussian. An informal explanation of these results is as follows: images are mostly made up of edges, and these edges can be at any image location, but only in a small

[6] Note however that the illumination usually comes from the top; this limitation of shift invariance is usually not taken into account.

percentage of locations in any one image: the statistical distribution of edges is super-gaussian because its density is concentrated around zero.

3.3 Invariances, localization and over-completeness

The hypothesis that images are mostly made up of edges can be also used to motivate the second and third constraints: edges, or rather, the objects that give rise to image edges, can be arbitrarily translated or rotated relative to the observer. Translation in depth corresponds to scaling. Hence, a large set of code vectors is needed to encode edges appearing at any location, orientation and scale[7]. Note that we are not assuming that all locations, orientations and scales of objects are equally likely, only that all of them can be realized. In addition, edges are localized in the image plane, leading to the localization constraint.

Shift invariance leads to a discussion of Fourier transforms. The amplitudes of the (complex) Fourier coefficients are invariant with respect to rigid translations of the image, while the phases of the coefficients, or equivalently the values of the sine and cosine coefficients, are changed by an amount that is frequency-dependent. However, coefficients for all phases of sinusoidal modulation can be obtained by linear combinations of the sine and cosine coefficients.

More generally, a mixing matrix that shows shift invariance will consist of subsets of mixing vectors, each subset being obtained by shifting a *filter* on the image plane. If the shifts can assume any value in a continuum, the concept of mixing matrix must be replaced by that of a mixing transform [54]. However, for many applications, a codebook representing a finite set of discrete shifts is desirable. A set of coefficients for such a codebook can be obtained by discrete convolution of the image with the filter. If the sampling of the convolution satisfies the sampling theorem, then coefficients for any arbitrary image location of the filter can be obtained by linear interpolation of the sampled values [66].

Objects can also be translated or rotated independently of each other. The representation should ideally be invariant with respect to these transformations: shifting one object in image coordinates, with other objects remaining at their locations, should result in a representation with the same coefficients, except for a shift of the coefficients representing the shifted object and for the effects of occlusion, which cannot be captured by a linear model. The requirement that the representation be invariant with respect to *localized* shifts means that the filters must themselves be localized in the image plane. This limits the localization of the filters in the frequency domain, and therefore the sampling of the convolutions. Obviously, the Fourier codebook violates the requirement of localization. For other codebooks, convolutions with the corresponding filters can, in practice, be sampled with only a small interpolation error in the representation of sources located between sampling points.

To satisfy invariances with respect to rotation and scaling, the filters generating the mixing/code vectors must themselves be arranged into subsets, within which all filters are rotated and/or scaled versions of a basic kernel. Sampling of scale and orientation, needed to generate a finite codebook, is subject to constraints analogous to the sampling theorem [66].

Given that the convolution with the smallest-scale filter cannot be downsampled, and a set of rotations must be represented at this smallest scale, it follows that the codebook must be overcomplete even if generated by a single basic kernel. Many codebooks that find practical application are generated from quadrature-pair kernels, *e.g.* Gabor functions with even and odd symmetry. This means that the degree of over-completeness (*i.e.* the ratio n/m) is doubled.

This subsection can be summarized as follows: natural images are generated by sources which can be realized in a continuum of locations, orientations and scales; approximating the mixing process by a finite codebook (representing a finite number of discrete locations, orientations and scales) is feasible, but only if the codebook is overcomplete.

[7] In the context of this paper, “scale” should always be interpreted as the scale of a code vector on the image plane; *e.g.* , for Gabor code vectors, the scale is defined as the parameter λ (Appendix A). This concept of scale should not be confused with the concept defined by scale-space theory.

The arguments in this subsection are informal and non-mathematical. A forthcoming paper [54] will start from a generative model without discretization in image space, rotation or scale and use the theory of shiftable multiscale transforms [23, 66] and the requirement of sparse coding to derive limits to the discretization of codebooks.

3.4 Other arguments for overcomplete codebooks

Independently from invariances, two other arguments can be made for overcomplete codebooks. These arguments are not specific to natural images.

The first argument is that the dual vectors of localized code vectors change with the degree of downsampling: the less downsampling, the more the dual vectors are localized in both image space and frequency. However, the dual vectors are not used in most of the sparse-coding methods reviewed in section 6, so this argument might not be relevant to sparse coding.

The second argument is based on a result obtained for binary coding of discrete input states [7]. Given a number N of states, the smallest number of bits required for encoding these states is, of course, the smallest integer M such that $M \geq \log_2 N$. However, such a code does not lead to statistically independent bits in general, and therefore the probability of a state cannot be computed from the product of the probabilities of its code bits. In some cases, greater statistical independence between the bits of the code can be achieved by an *expansive code*, *i.e.* a code in which there is one bit for each state, except the state with highest prior probability: $M = N - 1$. In this case, the advantage of statistical independence is that the state probabilities can be easily obtained as the products of bits probabilities.

4 Densities of image projections

In this section, we shall examine the nature of the density in grey-level space, for the case of an overcomplete mixing matrix. The aim is to show that a super-gaussian density is a necessary working assumption for an overcomplete codebook to be used in practice. Given that a density with infinite kurtosis (*e.g.* Cauchy) is always super-gaussian, it is only necessary to consider the case of source densities with finite kurtosis, and hence finite variance. Of course, grey levels do have finite variance, but that could be a consequence of optical sensors having a finite range and does not imply a finite variance for the sources.

Briefly stated, the connection between overcompleteness and prior source densities is the following: projections on any axis in grey-level space are sums of very many random variables, for any significant degree of overcompleteness and localization. If the source density were gaussian (and zero-mean), any projection would have a gaussian density, and therefore:

1. the code vectors could not be estimated from image data, because ICA-type learning methods rely on finding projections with non-gaussian densities [31];

2. even if the *MAP* coefficients were available, these coefficients would be useless for image compression or density estimation, since the density in grey-level space could be completely characterized by the axes of the gaussian ellipsoid.

Sums of random variables with finite variance converge to a gaussian density under mild conditions[8] [56]. Therefore, provided that the source variance is finite, any projection in grey-level space, being a linear combination of sources and noise, will have a density closer to a gaussian (*e.g.* by the χ^2 measure) than the densities of the sources.

The rate of convergence of sums depends on the kurtosis of the original densities: sums of sub-gaussian random variables (*i.e.* having a density with smaller kurtosis than a gaussian density) converge

[8] The basic condition is *infinite smallness*; for our purposes, this roughly means that, other things being equal, convergence is faster if the variances of the sources are all approximately the same.

very quickly to a gaussian density [56]. Therefore, if the sources were sub-gaussian, it would be impossible to find projections with other than gaussian densities (still under the assumption that the mixing matrix is overcomplete), and therefore (as in the case of gaussian source densities) it would be impossible to estimate the mixing matrix and useless (for compression) to estimate the sources.

In conclusion, over-completeness by itself leads us to assume super-gaussian source densities as a working hypothesis, independently of the empirical evidence.

In the following, we shall first develop a method to estimate the convergence of the joint density in grey-level space to a multivariate gaussian, for a given source density and mixing matrix. Reversing the reasoning, we shall estimate the kurtosis of the sources from the observed kurtosis of projections.

4.1 Cross-talk between sources

In the rest of this section, we shall consider projections of the image vector onto the code vectors:

$$\begin{aligned} \mathbf{g} &= \mathbf{A}^T\mathbf{x} \\ &= \mathbf{A}^T\mathbf{A}\mathbf{s} \end{aligned} \tag{16}$$

The reason for this interest is that these projections play a role in both learning algorithms [31] and coding algorithms (see section 6).

As can be seen from the above equation, the projections are linear combinations of source values. We define the *hidden signal* as the projection of a source onto the corresponding code (or dual) vector, and the *cross-talk* as the projection, onto a code (or dual) vector, of sources other than the corresponding source.

The matrix $\mathbf{A}^T\mathbf{A}$ can be decomposed into the identity matrix $\mathbf{I}_n$ and a symmetrical matrix $\mathbf{Q}$ with zero diagonal elements:

$$\mathbf{A}^T\mathbf{A} = \mathbf{I}_n + \mathbf{Q} \tag{17}$$

where

$$\mathbf{Q}_{ij} = \begin{cases} \mathbf{A}_i^T\mathbf{A}_j & (i \neq j) \\ 0 & (i = j) \end{cases} \tag{18}$$

Therefore, the hidden signal is equal to the source itself, while the cross-talk $\boldsymbol{\xi}$ is given by

$$\begin{aligned} \boldsymbol{\xi} &= \mathbf{Q}\mathbf{s} \\ &= \mathbf{g} - \mathbf{s} \end{aligned} \tag{19}$$

4.2 Cross-talk with a Gabor mixing matrix

Inner products between randomly-chosen pairs of random unit vectors in $\mathbb{R}^m$, with uniform density on the hypersphere, have zero mean and variance equal to $1/m$ [36]. The off-diagonal elements of $\mathbf{Q}$ are inner products between (different) mixing/code vectors and therefore, if the mixing/code vectors were randomly and uniformly distributed on the m-dimensional hypersphere in grey-level space, the expected value of $\mathbf{Q}_{ij}^2$ would be $1/m$.

A more realistic mixing matrix would consist of Gabor-like functions. These are the kind of code vectors that are recovered both by conventional ICA [10] and by overcomplete ICA [49, 31]. They are also similar to receptive fields of simple cells in the primary visual cortex [34]. Gabor functions have the smallest joint spread in space and frequency[16], and therefore the average inner product of two Gabor functions might be expected to be minimal for a given sampling of the space and frequency domains.

An example of a vastly overcomplete Gabor codebook can be found in [55]. The codebook consists of two-dimensional real-valued Gabor functions of unit norm and zero mean. The Gabor codebook is designed for images of size $m = 128^2$ and is 162 times overcomplete ($n = 162 \cdot 128^2$): Gabor functions at (logarithmic) scales 1 to 10, each with 8 orientations and 2 phases (*i.e.* even and odd symmetry), are centered on each pixel (no downsampling), resulting in 160 Gabor functions per pixel; in addition, at scale 0, even-symmetry Gabor functions with horizontal and vertical orientations provide an additional 2 Gabor functions per pixel. More details are given in Appendix A. Some coding results obtained with this codebook are in [55].

In this section and the next, we assume that this Gabor codebook is a good approximation to the mixing process for natural images (in spite of the discretization of locations, orientations and scales) and examine the amount of cross-talk between sources arising from this mixing matrix. The distributions of inner products between the Gabor code vectors are shown in Fig.4.2. For comparison, the distribution expected for random code vectors [36] is also shown.

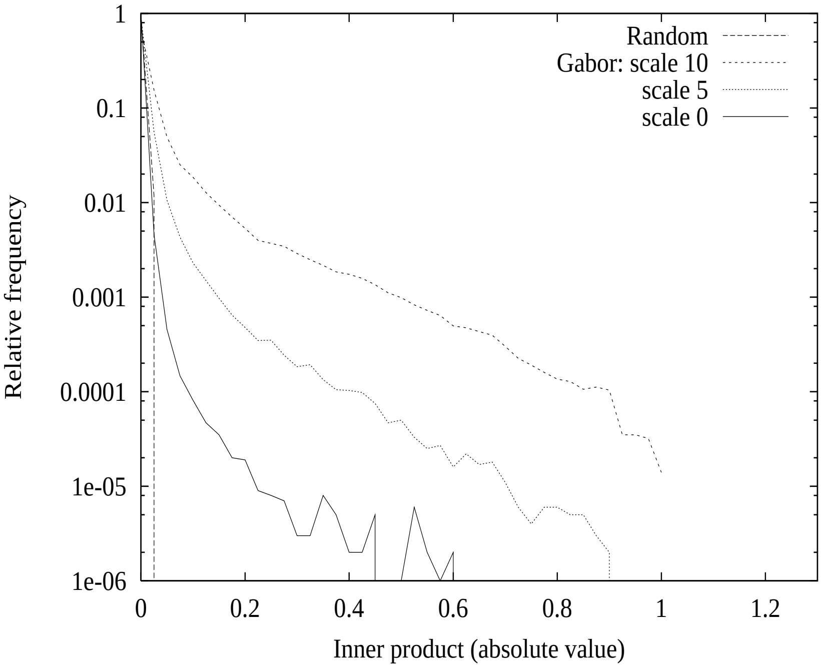

Figure 1: Relative frequencies of inner products (absolute values) between a reference code vector and all other code vectors in the Gabor codebook, compared to the relative frequencies expected for random code vectors (leftmost, solid curve). The relative frequencies are plotted for three different scales of a reference Gabor code vector with even phase: the relative frequencies for code vectors with odd phase are very similar. The (logarithmic) scale parameter is the parameter l defined in Appendix A (Eq.47).

It might seem surprising that the Gabor codebook, hand-crafted to minimize cross-correlations between code vectors, has much greater (in absolute value) inner products than an equivalent random codebook. However, this is a consequence of both localization and shift-invariance. Localization means that the directions of the code vectors in the m-dimensional grey-level space are not uniformly distributed. Shift-invariance means that Gabor functions are centered on all pixels: large-scale Gabor functions centered on nearby pixels have large inner products.

A related paper in this issue [31] reports on overcomplete codebooks learned from natural images with ICA learning rules. The learned code vectors resemble Gabor functions and have larger cross-correlations than would be expected for random vectors, but smaller than can be seen in Fig.4.2 (see also Fig.5.1.1 in section 5). This is probably because the learned codebooks were overcomplete by a factor of little more than 2, as compared to the factor of 162 for the codebook considered here.

4.2.1 Implications of invariances for source densities

Before going further, some more constraints are needed for the statistics of the source. These constraints can be derived from empirical invariances of image statistics with respect to translations, rotations or scalings of natural images (except for the effects of the sky being usually at the top, which we assume can be neglected).

Invariances with respect to translation and rotation require that all sources at the same scale have identical densities.

The implications of scaling invariance are less straighforward. Appendix B shows that scaling invariance of image statistics implies that the variances of the sources are constant over scales, but only if the number of sources is the same at all scales. Since this is the case for the mixing matrix under consideration, we proceed on the assumption that all sources have the same variance, though not necessarily the same kurtosis.

4.3 Convergence to a gaussian density

In the following, the kth cumulant of a random variable y is written as $\kappa_k(y)$ and the kurtosis $K(y)$ is defined as:

$$\begin{aligned} K(y) &= \frac{\langle y^4 \rangle}{\langle y^2 \rangle^2} - 3 \\ &= \frac{\kappa_4(y)}{\kappa_2^2(y)} \end{aligned} \tag{20}$$

The second equality follows from the equations for the second and fourth cumulants ([67], pp.86-91).

The kurtosis will be used as a measure of non-gaussianity: a kurtosis close to zero indicates an almost-gaussian density. A fundamental result of probability theory is relevant: the kth cumulant of a sum of independent random variables is equal to the sum of the kth cumulants of the variables ([67], p.156). From this result and Eq.20, it follows that the kurtosis of a projection is given by:

$$\begin{aligned} K(g_i) &= \frac{\kappa_4\left(s_i + \sum_j \mathbf{Q}_{ij} s_j\right)}{\kappa_2^2\left(s_i + \sum_j \mathbf{Q}_{ij} s_j\right)} \\ &= \frac{\kappa_4(s_i) + \sum_j \mathbf{Q}_{ij}^4 \kappa_4(s_j)}{\left[\kappa_2(s_i) + \sum_j \mathbf{Q}_{ij}^2 \kappa_2(s_j)\right]^2} \end{aligned} \tag{21}$$

On the basis of the assumption of source variance constant over scales (from subsection 4.2.1), the above equation can be re-written as:

$$K(g_i) = \frac{\kappa_4(s_i)\left[1 + \sum_j \rho_{ij} \mathbf{Q}_{ij}^4\right]}{\kappa_2^2(s_i)\left[1 + \sum_j \mathbf{Q}_{ij}^2\right]^2} \tag{22}$$

where ρ_{ij} is the (dimensionless) ratio between the 4th cumulant of source j and the 4th cumulant of source i.

From Eq.22, we obtain the ratio between the kurtosis of a projection onto code vector i and the kurtosis of the corresponding source:

$$\frac{K(g_i)}{K(s_i)} = \frac{1 + \sum_j \rho_{ij} \mathbf{Q}_{ij}^4}{\left[1 + \sum_j \mathbf{Q}_{ij}^2\right]^2} \tag{23}$$

We define this ratio as the *kurtosis-attenuation factor*, because it is the factor by which the kurtosis is decreased by the mixing of several sources into a projection.

We use again the constraint that image statistics are scale-invariant, to deduce that

- the projection kurtosis $K(g_i)$ (being an image statistic) is a constant $K_{\mathbf{g}}$ independent of the scale of g_i;
- the source kurtosis is a function of scale only.

Therefore:

$$\begin{aligned}\rho_{ij} &= \frac{\kappa_4(s_j)}{\kappa_4(s_i)} \\ &= \frac{K(s_j)}{K(s_i)} \\ &= \frac{K_{\mathbf{g}}/K(s_i)}{K_{\mathbf{g}}/K(s_j)} \qquad (24)\end{aligned}$$

This equation suggests an iterative procedure to estimate the ratios ρ_{ij}: start from an initial guess for ρ_{ij} and iterate to convergence:

- compute the ratios $K(g_i)/K(s_i) = K_{\mathbf{g}}/K(s_i)$ using Eq.23;
- refine the estimates of ρ_{ij} using Eq.24.

Fig. 4.3 shows the estimates of the kurtosis-attenuation ratios at different scales, obtained by repeating this procedure to convergence. It can be seen that the attenuation ratios are very small and decreasing with scale. Clearly, the kurtosis of a projection is always much smaller than the kurtosis of the corresponding source.

The lowest possible value for the kurtosis of a distribution is -2 ([67], p.109; for a unimodal symmetrical distribution, the lowest possible kurtosis is -1.2) Therefore, sub-gaussian sources have a kurtosis bounded between 0 and -2. Such sources, in conjunction with the Gabor mixing matrix under consideration, would give rise to projection densities undistinguishable from gaussian densities.

4.3.1 Source kurtosis inferred from empirical results

So far, the case for super-gaussian sources has been made without taking into consideration empirical results. However, having derived the relationship between source kurtosis and projection kurtosis is known, this relationship can be used to infer the source kurtosis from empirical observations of projection kurtosis.

As mentioned in subsection 3.2, densities of image projections onto Gabor-like vectors are well-approximated by a generalized laplacian with $\beta \approx 0.5$. The kurtosis of this density is approximately equal to 22. From this finding and the attenuation ratios obtained numerically, we obtain estimates for the kurtosis of the sources, shown in Fig.4.3.1.

As discussed in subsection 4.2.1, scale invariance implies that the expected number of non-zero sources decreases with scale. This constraint, by itself, would be sufficient to deduce that the kurtosis of the sources increases with scale, if the total number of sources is constant over scales. In reality, as mentioned in subsection 3.3, there is not a finite number of sources at a finite number of scales, but a continuum of source locations, orientations and scales. A model with a continuum of sources and mixing vectors will be analyzed in [54]. In such a model, the notions of source density, source variance and source kurtosis are no longer meaningful.

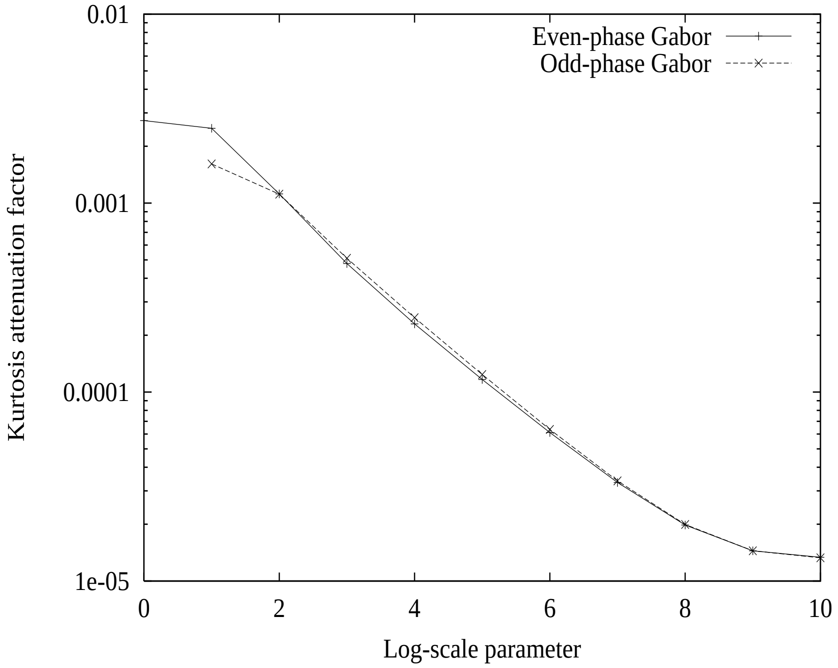

Figure 2: Ratios between the kurtosis of a source and the kurtosis of the corresponding projection, plotted as a function of the scale of the corresponding mixing/code vector.

5 Masking of the signal in an overcomplete codebook

In the previous section, it was seen that the cross-talk is much larger than the hidden signal when the mixing matrix is overcomplete and the mixing vectors are localized. Intuitively, it would seem that, under such circumstances, not only learning, but also coding should become problematic. This section will show that this is indeed the case and that, once more, the working assumption of super-gaussian sources can simplify the problem.

5.1 Signal-to-cross-talk ratio

We are interested in the ratio of the variance of the hidden signal to the variance of the cross-talk. The ratio of these variances will be defined as *SXR* , by analogy with *SNR* (signal-to-noise ratio),

The variance of each element of the cross-talk is equal to the weighted sum of the variances of the sources:

$$\mathrm{Var}(\xi_i) = \sum_j \mathbf{Q}_{ij}^2 \mathrm{Var}(s_i) \tag{25}$$

Under the assumption that all sources have equal variance (subsection 4.2.1), the *SXR* for the i-th projection is given by the inverse of the squared norm of the i-th row (or column) of $\mathbf{Q}$:

$$SXR_{\ i} = ||\mathbf{Q}_i||^{-2} \tag{26}$$

The *SXR* increases with both the average cross-correlation between mixing vectors and the number of mixing vectors.

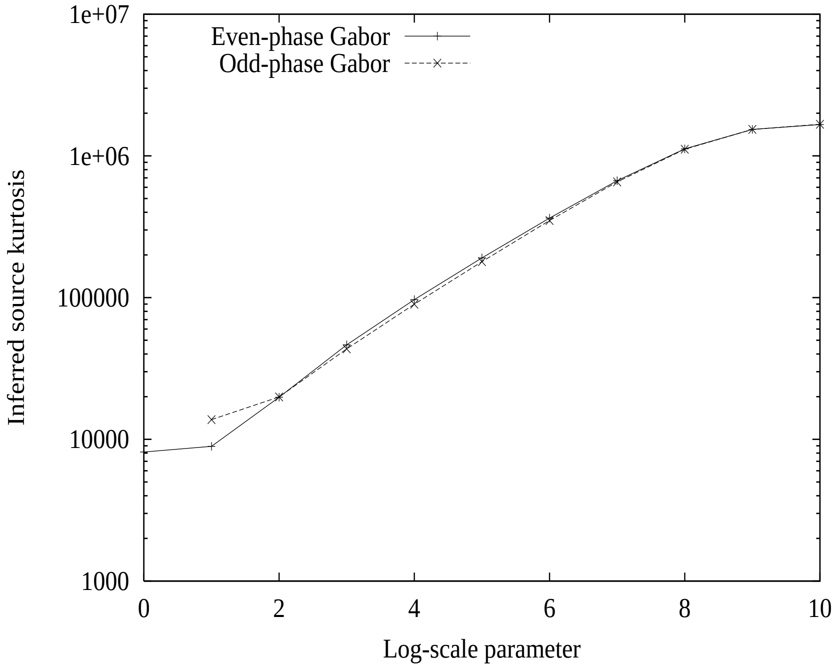

Figure 3: Estimated kurtosis of sources plotted as functions of the scale of the corresponding mixing vectors. The kurtosis estimates were computed from the numerical results shown in Fig.4.3 and the empirical kurtosis of projections onto code vectors.

5.1.1 Downsampling the codebook

The *SXR* for a downsampled codebook will also be considered. This codebook is obtained by downsampling the location parameters of the Gabor kernels with a sampling period proportional to the (linear) scale of the kernels. Downsampling produces a codebook that is about 29 times overcomplete; this codebook is described in Appendix A. It is important to keep in mind that, even though the codebook has been downsampled, the mixing matrix has not changed: sources can still be found at any image location and at any scale. Using a downsampled codebook $\mathbf{A}_{\downarrow}$ means substituting Eq.16 by the following:

$$\begin{aligned} \mathbf{g}_{\downarrow} &= \mathbf{A}_{\downarrow}^{T}\mathbf{x} \\ &= \mathbf{A}_{\downarrow}^{T}\mathbf{A}\mathbf{s} \end{aligned} \tag{27}$$

The definitions of hidden signal and of cross-talk must also be modified: suppose that the sampling period at a given scale is k pixels; by the definition of hidden signal adopted for the full codebook, at this scale only one out of k^2 sources would be hidden signal. Instead, we would like all the sources to be represented by the coefficients. Several interpolation schemes could be adopted for this representation. For our purpose, which is to illustrate the effect of downsampling, it is sufficient to adopt the simple convention that the hidden signal onto code vector i consists of all sources whose location parameters are closer to the location parameters of code vector i than to any other code vector. The cross-talk consists of the sum of projections of all other sources.

The inner products obtained with the downsampled Gabor codebook are shown in Fig.5.1.1. It can be seen that downsampling does not make much difference to the distribution of inner products, except for the largest inner products at the largest scales. The reason for this discrepancy is that these largest inner products are due to mixing vectors at the same scale and close on the image plane to the reference code vector: the corresponding sources become part of the hidden signal and therefore are not included in the cross-talk.

5.1.2 Projections onto dual vectors

The concepts of cross-talk and *SXR* are easily extended to projections onto the dual vectors:

$$\begin{aligned} \mathbf{g}^{(\mathbf{W})} &= \mathbf{W}\mathbf{x} \\ &= \mathbf{W}\mathbf{A}\mathbf{s} \end{aligned} \tag{28}$$

In this case, the hidden signal in projection i is equal to the source scaled by the inner product $(\mathbf{W}^T)_i\mathbf{A}_i$ and the cross-talk is obtained by a simple modification of Eq.18:

$$\mathbf{Q}_{ij}^{(\mathbf{W})} = \begin{cases} (\mathbf{W}^T)_i\mathbf{A}_j & (i \neq j) \\ 0 & (i = j) \end{cases} \tag{29}$$

Eqs. 19, 25, and 26 remain unchanged, except for the substitution of $\mathbf{Q}$ by $\mathbf{Q}^{(\mathbf{W})}$. Distributions of inner products between mixing vectors and dual vectors are shown in Fig.5.1.2.

5.1.3 Results

The *SXR* was estimated on the basis of the assumptions about source variances from subsection 4.2.1 and the measured distributions of inner products between mixing/code vectors (Figs. 4.2,5.1.1,5.1.2). The results are shown in Fig.5.1.3. It seems that the hidden signal from the "correct" source is always swamped by the cross-talk from other sources. This statement will be qualified in the next subsection.

The *SXR* for the downsampled codebook is increased roughly in proportion to the downsampling factor at each scale, keeping the *SXR* approximately constant over scales. Given that the cross-talk

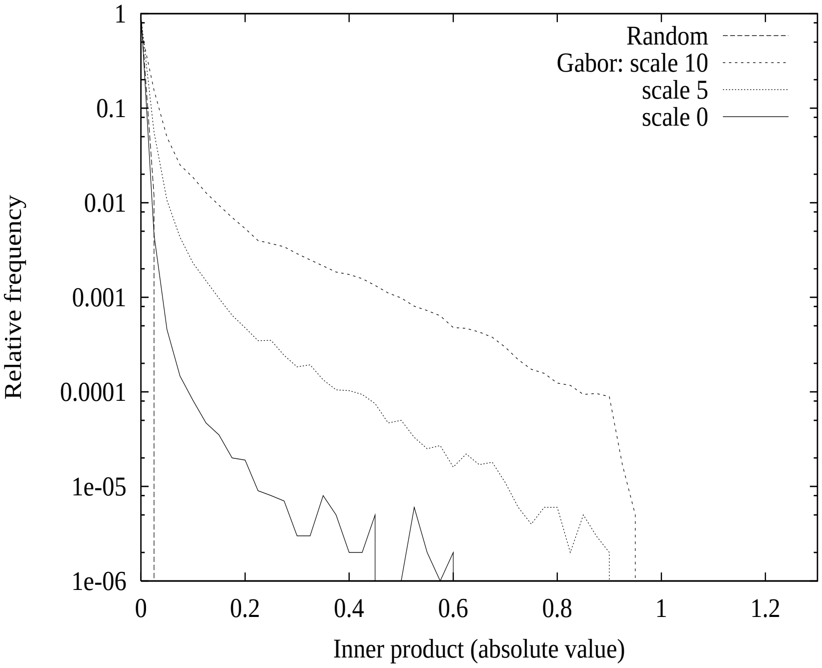

Figure 4: Relative frequencies of inner products between a reference code vector (with even symmetry) in the downsampled Gabor codebook and the mixing vectors which contribute to the cross-talk.

is not significantly decreased (as can be seen in Fig.5.1.1), the increase is due almost enitrely to the lumping together of several sources into the hidden signal. Therefore, the price to pay for the increase of SXR is a decrease in the ability to discriminate between sources.

The cross-talk for projections onto the dual vectors (of the full codebook) is also shown in Fig.5.1.3. This cross-talk is only a little less than for the code vectors. This is the consequence of the gaussian assumption, implicit in using the dual vectors: the image energy, arising from a single source, is spread across as many coefficients as possible by projection onto the dual vectors (compatibly with minimum reconstruction error). This is because, on the gaussian assumption, it is highly unlikely that a single source is active: ,inimizing the coefficient norm does not produce a sparse code.

5.2 The effect of the prior source density

At this point, another advantage of the super-gaussian assumption for the prior source densities becomes apparent: the cross-talk is a sum of random variables with finite variance, and therefore it is closer than the hidden signal to a gaussian density. As a consequence, the cross-talk will be relatively less important on the tails of the density of the projections.

To illustrate this concept, we introduce a density model that will also be of interest in subsection 6.2. Suppose that each source can be in one of two states:

- active, with probability P_α, in which case the density is gaussian or super-gaussian, with variance $\sigma_\alpha^2 = \langle s_\alpha^2 \rangle$ and (finite) kurtosis $K_\alpha = \langle s_\alpha^4 \rangle / \langle s_\alpha^2 \rangle^2 - 3$;
- inactive, with probability $1 - P_\alpha$, in which case the source value is always zero.

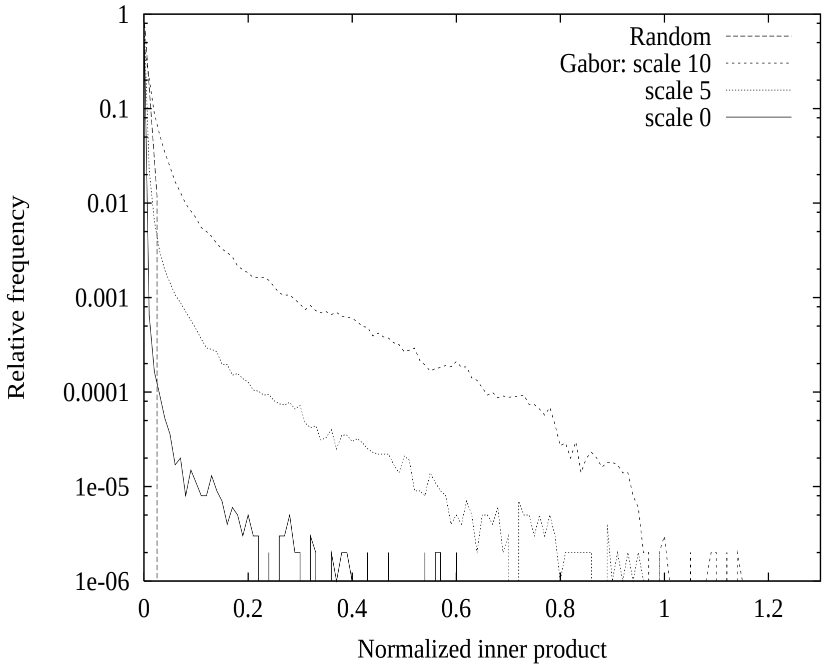

Figure 5: Relative frequencies of inner products between a reference dual vector (with even symmetry) and the mixing vectors which contribute to the cross-talk.

Note that it is desirable to have fewer active coefficients than image pixels, implying $P_\alpha < m/n$. Note also that the density of inactive sources can be considered a delta function scaled by the factor $1 - P_\alpha$ (a very narrow rectangular or gaussian density with zero mean would be equivalent for practical purposes).

Let us consider the density of the *average* source: its second and fourth moments are weighted sums of the corresponding moments for the two mixture components, *i.e.* they are equal to the scaled moments of the "active density":

$$\begin{aligned} \langle s^2 \rangle &= P_\alpha \langle s_\alpha^2 \rangle & (30) \\ \langle s^4 \rangle &= P_\alpha \langle s_\alpha^4 \rangle & (31) \end{aligned}$$

It follows that the variance of the average source is decreased with respect to the variance of an active source:

$$\sigma_{\mathbf{s}}^2 = P_\alpha \sigma_\alpha^2 \tag{32}$$

On the other hand, the kurtosis of the average source is increased with respect to K_α:

$$\begin{aligned} K_{\mathbf{s}} &= \frac{P_\alpha \langle s_\alpha^4 \rangle}{(P_\alpha \langle s_\alpha^2 \rangle)^2} - 3 \\ &= \frac{K_\alpha}{P_\alpha} + \frac{3}{P_\alpha}(1 - P_\alpha) & (33) \end{aligned}$$

Even though this result has no implications for the SXR , it is shown to illustrate the exact relationship between the relative frequency P_α of active sources and the source kurtosis.

Assuming that the variance of the cross-talk is a scaled version of the variance of the average source, it follows from Eq.32 that the SXR for an active source is increased by a factor of $1/P_\alpha$. It might seem

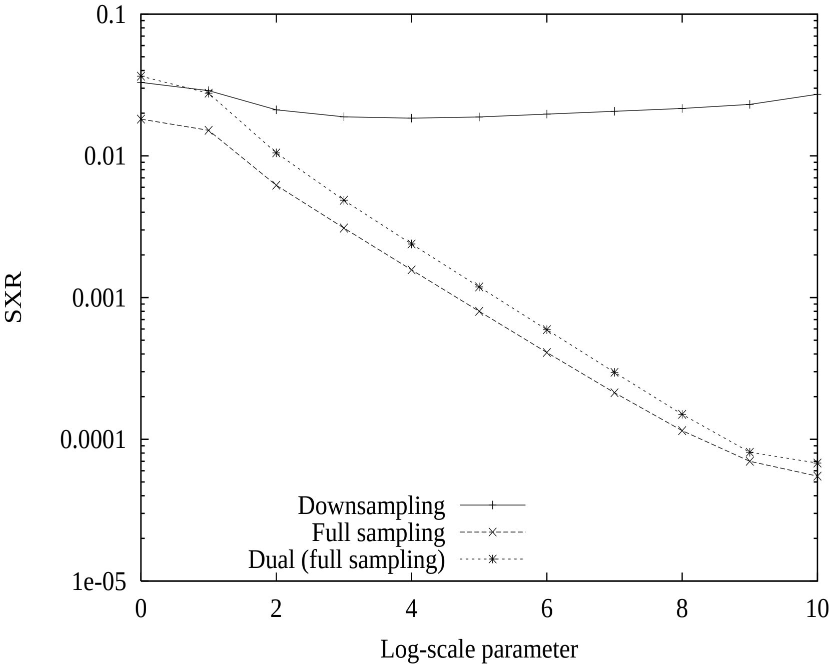

Figure 6: *SXR* as a function of the Gabor scale parameter (as defined in [55]) for the full and downsampled codebooks. The *SXR* for the dual vectors is also shown for comparison.

that we have obtained something for nothing, until we consider that the *SXR* for the *inactive* sources is zero. This is not a problem if we can distinguish active from inactive sources and set all coefficients corresponding to inactive sources to zero. In this case, the dimensionality of model is effectively reduced to the number of active sources. Most codebooks do not have any subset of m or fewer code vectors which are not linearly independent, so that, if $nP_\alpha < m$, the active sources are likely to correspond to linearly independent mixing/code vectors. This leads to a further simplification.

The main problem is to determine which sources are active. As we shall see in subsection 6.2, this problem, too, is complicated by the cross-talk.

An additional problem is that some of the sources making up the cross-talk are weighted very heavily, as can be seen in Figs. 4.2,5.1.1,5.1.2. If these sources happen to be active, then the *SXR* is effectively very much smaller than on average.

5.3 A laplacian kurtosis is not sufficient for sparse coding

We conclude this section with a simple example showing that sparse coding can only be obtained if the sources have a kurtosis larger than the kurtosis of a laplacian.

Consider the case of two identical mixing/code vectors $\mathbf{A}_i$ and $\mathbf{A}_j$ and only one of the corresponding sources being active:

$$s_i \neq 0, \quad s_j = 0$$

Zero reconstruction error can be obtained with any combination

$$c_i = qs_i, \quad c_j = (1-q)s_i \quad (0 \leq q \leq 1)$$

Assuming a laplacian prior, any such combination would have equal prior probability. A gaussian prior would favour $q = 1/2$. A generalized laplacian prior (Eq.15) would favour the sparse solution (with $q = 1$, or equivalently $q = 0$) if $0 < \beta < 1$, the minimum-norm solution (with $q = 1/2$) if $\beta > 1$.

Of course, the case of two identical code vectors would not be realized in practice; however, the same reasoning applies by substituting any linear combination of mutually orthogonal mixing/code vectors for $\mathbf{A}_j$. Therefore, it would seem that a laplacian prior can fail to produce a sparse code when a code vector is a linear combinations of other, mutually orthogonal code vectors.

This result has important implications for optimization: it means that the penalty term in the objective function must be non-convex at least for some coefficient values. The non-convexity can be inferred from the fact that the penalty reaches a maximum for some value of q intermediate between 0 and 1.

6 Two approaches to sparse coding

This section will describe two classes of sparse-coding methods. The first class has received most attention within the ICA community; it assumes a smooth prior log-density for the sources and uses standard methods for continuous optimization. The second class originates within the wavelet community; it implicitly assumes a mixture model for the sources, of the kind described in subsection 5.2.

It is fair to say that the efficiency of sparse coding algorithms is determined more by the optimization method than by the underlying generative model. Nonetheless, the pitfalls of the optimization methods can often be best understood by considering the optimization task, which is determined by the underlying generative model.

The term “approach” in the section title is used to emphasize that most methods developed to date do not seem suitable for natural images, due to their computational cost. Comparisons between the methods have mostly been made on simpler signals.

There is not yet an accepted standard of performance evaluation for sparse coding. The ideal measure would be the total entropy of coefficients and residual, averaged over several natural images. Comparison of different methods requires that all methods are applied with the same codebooks. In practice, the most used performance measure is the number of non-zero coefficients as a function of relative reconstruction error. This measure implicitly assumes a mixture model, as will be seen in subsection 6.2.

6.1 Gradient-based coding algorithms

If the source densities are smooth, then the objective function can be optimized by gradient-based methods. Following standard terminology in constrained continuous optimization, we define algorithms based on this approach as *penalty methods*. In practice, the exact form of the prior density is not known, and it is an open question how close the hypothetical density must be to the actual source density for the method to work. The penalty term used by the algorithms will be defined as $\hat{h}(\mathbf{c})$, to distinguish it from the actual log-prior density of the sources.

The gradient of the objective function is:

$$\nabla_{\mathbf{c}} H = \frac{1}{\sigma_\nu^2}\mathbf{g} + \nabla_{\mathbf{c}} \sum_i \hat{h}(c_i) \tag{34}$$

where $\mathbf{g}$ is the gradient of the reconstruction error (scaled by 1/2):

$$\begin{aligned} \mathbf{g} &= \frac{1}{2}\nabla_{\mathbf{c}} E \\ &= \mathbf{A}^T(\mathbf{x} - \mathbf{A}\mathbf{c}) \\ &= \mathbf{A}^T\mathbf{A}(\mathbf{s} - \mathbf{c}) \end{aligned} \tag{35}$$

A likely problem with penalty methods is apparent in the above equation: the error in the source estimate, $\mathbf{s}-\mathbf{c}$, is pre-multiplied by the matrix product $\mathbf{A}^T\mathbf{A}$ before being added to the penalty. As was seen in section 4, the result of this multiplication is that the error is "spread" over many projections, rather than contributing to the correction of the appropriate coefficient.

6.1.1 Basis Pursuit

A penalty method introduced within the wavelet community is based on iid laplacian prior densities for the sources [13]. This method has been called *basis pursuit.* The laplacian assumption leads to the penalty

$$\hat{h}_L(c) = \left|\frac{c}{\sigma}\right| + \log Z \tag{36}$$

where σ is a scale parameter and Z is a normalization factor. For zero noise, the objective function to be minimized (within the *MRE* subspace) is a weighted sum of the absolute values of the coefficients. As a consequence, linear programming can be used for the optimization. However, the scale of the problems in image processing would seem to make linear programming unsuitable.

Two problems with the laplacian assumption have been pointed out in section 5.3: sources of natural images have kurtosis that is larger than for a laplacian; and the assumption of a laplacian prior density can fail to produce a sparse code when a code vector is a linear combination of other, mutually orthogonal code vectors. An example of this failure is shown in [44] with a simple artificial signal.

6.1.2 Applications to learning

The penalty method was independently developed within the neural-network community by two research groups [27, 48, 49]. Various forms of the penalty term have been tried, mostly corresponding to generalized laplacian or Cauchy densities. Objective functions including these penalties have been optimized by the conjugate-gradient method.

A penalty term which cannot be interpreted as a log-density was also used [48]:

$$\hat{h}_G(c) = \exp\left(-\frac{c^2}{2\sigma_G^2}\right) \tag{37}$$

This penalty is a *gaussian moment* [29] and minimizing it is an effective way of maximizing the kurtosis of the coefficients, without commitment to any particular form of the coefficient density. It cannot be interpreted as a log-density because $\exp[-\hat{h}_G(c)]$ is not integrable over $\mathbb{R}$.

These variations of the penalty method have been used as components of ICA-type learning algorithms. The code vectors learned by the algorithms have been evaluated for image compression, but the performance of the penalty method itself has not been quantitatively evaluated on natural images. One of the learning algorithms was applied to images of size 512^2 pixels [51]. From the published results, it seems that satisfactory image compression was only achieved with codebooks overcomplete by factors of $n/m < 2$, suggesting that the penalty method itself, being part of the learning algorithm, was not effective for $n/m = 2$ or higher.

These results agree with the experience of the author, who has tried (without success) to apply the method to natural images with codebooks overcomplete by factors between 12 and 162, the conjugate-gradient method, and various penalties: the image energy was spread over too many coefficients at convergence, suggesting either convergence to local minima or an extremely slow rate of convergence. The problem is likely to be due to the cross-talk between sources, as discussed in relation to Eq.35. However, it is possible that better optimization methods and/or a better choice of penalty term would lead to better results.

6.2 Algorithms based on mixture models

Most of the methods in this class have been developed without an explicit formulation of a generative model; indeed, the objective function being minimized is often ill-defined.

Nonetheless, a motivation for these methods can be found within the generative-model framework, and specifically by the mixture model introduced in subsection 5.2. In that model, sources are either active, with probability P_α, or inactive, with probability $1 - P_\alpha$. Since the density of the sources in this model is concentrated at zero, it follows that the penalty term (Eq.7) increases with the number of active (*i.e.* non-zero) coefficients.

Such an objective function is unsuitable for gradient-based optimization: the problem becomes combinatorial. In fact, encoding a signal with the smallest error for a given number of active coefficients is NP-hard [17]. Note that this holds for the general case: if the codebook is not underconstrained, it is easy to minimize the error for any given number of active coefficients. Similarly, it might be possible to exploit the structure of a Gabor codebook.

Two methods that rely on a partition between active and inactive coefficients should be mentioned at this point:

- orthogonal-basis selection [14], in which the codebook consists of a set of orthogonal bases, and only one basis is selected to encode any single image;
- shape-gain vector quantization [25], in which only one coefficient is selected for each image block.

These methods are outside the scope of this paper because they impose statistical dependendencies between coefficients: an active coefficient prevents a subset of other coefficients from being active.

6.2.1 Matching Pursuit

The most influential method for sparse coding [40, 17] can be described as an extension of shape-gain vector quantization. This, and some of the following methods, are based on iterative greedy strategies. The iteration for the basic version of matching pursuit is as follows:

1. the coefficient is selected that corresponds to the largest element of the projection vector

$$\begin{aligned} \mathbf{g} &= \mathbf{A}^T\mathbf{r} && (38)\\ &= \mathbf{A}^T\mathbf{A}(\mathbf{s}-\mathbf{c}) && (39) \end{aligned}$$

   this coefficient is updated by an amount equal to the corresponding projection;

2. the residual is updated on the basis of the change of the selected coefficient.

The first step is equivalent to vector quantization, but the iterative nature of the method makes it applicable to the entire image, rather than image blocks.

One interesting feature of this method is that, for each iteration t, the relationship holds:

$$R(t-1) = g_t^2 + R(t) \qquad (40)$$

where $R(t)$ is the reconstruction error, and g_t is the coefficient update, at iteration t. This results in the sum of squared coefficient updates being equal to the difference between the squared norms of the image and the residual:

$$\sum_{t=1}^{T} g_t^2 = ||\mathbf{x}||^2 - R(T) \qquad (41)$$

where T is the total number of iterations. The same relationship does not hold in general for the sum of squared coefficients:

$$\sum_{i=1}^{m} c_i^2 \neq \|\mathbf{x}\|^2 - R(T) \tag{42}$$

This is because a coefficient can be updated more than once. Therefore, it is difficult to define what Eq.40 implies for the final coefficients, but it would seem to favour a small coefficient norm.

Another version of the method has been developed, in which, for each iteration:

1. a coefficient is selected as in basic matching pursuit its state is changed from inactive to active;

2. the reconstruction error is minimized in the subspace of active coefficients.

Note that the second step of the iteration ensures that the residual is always orthogonal to the active code vectors. This variation is called orthogonal matching pursuit The iteration is more complex, but the objective function is fairly well defined (by step 2 of the iteration) as a weighted sum of reconstruction error and number of active coefficients.

By comparison, the basic version of matching pursuit can be said to add the additional "soft" constraint of a small coefficient norm. As a consequence, basic matching pursuit results in a larger reconstruction error for a given number of active coefficients.

6.2.2 Inhibition Method

Matching pursuit can itself be generalized by updating more than one coefficient per iteration, under the constraint that the updated coefficients correspond to orthogonal or near-orthogonal code vectors. This is the inhibition method [55]. A version analogous to orthogonal matching pursuit also exists (manuscript in preparation).

The inhibition method offers much faster convergence than matching pursuit, in terms of both number of iterations and total computational cost, as demonstrated on natural images. Since the two methods are different and both are sub-optimal, they will not converge to the same active set of coefficients. However, the *numbers* of active coefficients for a given reconstruction error are almost identical for the two methods [55].

Note that matching pursuit and the inhibition method are the only two algorithms for sparse coding that have been compared on natural images with the same codebook [55].

6.2.3 Limitations of greedy methods

Greedy methods fail when two mixing/code vectors, corresponding to active sources, are both close to one or more other code vectors. As a concrete example, suppose that two active sources correspond to orthogonal code vectors $\mathbf{A}_i$ and $\mathbf{A}_j$, and that there is a third code vector $\mathbf{A}_k$ at an angle of $\pi/4$ to both $\mathbf{A}_i$ and $\mathbf{A}_j$:

$$\begin{aligned} \mathbf{Q}_{ij} &= 0 \\ \mathbf{Q}_{ik} = \mathbf{Q}_{jk} &= \sqrt{2} \end{aligned}$$

Then the cross-talk onto $\mathbf{A}_k$ is equal to

$$\xi_k = \frac{1}{\sqrt{2}}(s_i + s_j)$$

If $s_i \approx s_j$, then the cross-talk onto $\mathbf{A}_k$ is larger than either of the sources; therefore, a greedy algorithm will update c_k rather than c_i or c_j. Once such a mistake is made, a greedy algorithm never recovers. A similar example (the TwinSine signal) can be found in [13].

6.2.4 High-Resolution Pursuit

One modification of matching pursuit, which goes some way towards solving the above problems, is High-Resolution Pursuit (HRP) [33]. The modification consists in a different criterion for the selection of the coefficient to be updated: HRP selects the coefficient corresponding to the code vector which best matches the fine structure of the residual. Matching of the fine structure is measured by the smallest (in absolute value) of the inner products between the residual and the components of a code vector. The problem with this method is that it is not clear how one should divide a code vector into components: for instance, in the case of a Gabor codebook, one could decide that each lobe (positive or negative) constitutes a separate component, but one could also insist on a finer subdivision.

6.2.5 Optimal-Subset Selection

As pointed out above, orthogonal matching pursuit minimizes the norm of the residual for a given set of active coefficients, and uses a greedy strategy to select the active coefficients. Optimal-subset selection [44] can be seen as the result of two modifications of orthogonal matching pursuit:

- Instead of minimizing the norm of the residual, what is minimized is the norm of the residual's projection onto the dual vectors of the inactive set of coefficients[9].
- Similarly to matching pursuit, the code vector corresponding to the largest projection is selected and added to the active set. However, the number of active coefficients is kept fixed by simultaneously removing a code vector from the active set; specifically, the code vector corresponding to the smallest coefficient in the active set.

As a result of the second modification, this method must be applied with different numbers of active coefficients to find the smallest active set giving the desired reconstruction error.

Given its combinatorial nature, this method is very slow and has only been applied to small signals.

6.2.6 Mixtures of gaussians and applications to learning

Explicit formulations of mixture models have been proposed in the ICA literature and used to derive learning and coding methods. These models are more general than the two-component mixture described in subsection 5.2, because more than two states are allowed for each source and all states are "active", *i.e.* the sources can have non-zero value in any of the states. Two specific methods have been proposed for coding and learning.

In the first method [1] (independent-factor analysis) coding is simplified by the assumption that the sources are not only *a priori* independent, but also conditionally independent given the image. This assumption allows pointwise source estimation, as in wavelet shrinkage [18, 65]. This method has not been applied on images and the assumption of conditional independence might prove unworkable in such an application.

In the second method [50], the *MAP* active coefficients were obtained by Gibbs sampling with simulated annealing. This method seems suitable only for small image blocks, such as the 8×8 blocks used in [50]. Note that Gibbs sampling allows, at least in principle, to obtain not only the *MAP* estimate, but an estimate for the entire posterior source density (and therefore also the Bayes' Least Squares estimate).

Other overcomplete-ICA methods are not of interest in the context of this paper because they do not involve coefficient estimation (*e.g.* [31]); or else are based on assumptions which are not valid for natural images, *e.g.* a gaussian prior source density [60] or a convex log-prior density [72].

[9] The two are equivalent only if the codebook is a tight frame.

7 Discussion

7.1 Applications of sparse coding

Most applications of sparse coding are based on matching pursuit and related algorithms. The main exceptions are applications to learning algorithms, which have been discussed in the previous section.

Matching pursuit has proved effective in video compression [46]. In visual pattern recognition, high-resolution pursuit has been applied to the recognition of silhouettes [33]; matching pursuit has been applied to face recognition [57] and to the detection of micro-calcifications in mammograms [70]. In these last two applications, matching pursuit resulted in detection efficiency significantly higher than obtained with competing methods.

The application to mammography [70] is a good example to illustrate the potential of matching pursuit in pattern recognition. The problem is essentially one of template matching. The templates that must be recognized are the wavelet components typical of images of micro-calcifications. These wavelets can appear with different translations and rotations, and in a range of scales. In simple template-matching, the mammogram images would be projected onto the templates (code vectors) most likely to arise from micro-calcifications. Since mammogram images have a complex background, this method would generate many false positives. Matching pursuit reduces the number of false positives by removing the image components likely to arise from the background.

Matching pursuit has also been applied to the processing of seismic data [37]. The problem of finding sparse linear expansions is common in numerical linear algebra [45], statistics [24, 41], system identification [12] and control [64]. Matching pursuit and related methods have found applications in these fields.

7.2 Biological implications

Responses of simple cells in visual cortical area V1 can be described, to a first approximation, as (half-wave rectified) projections of the retinal image (pre-filtered by retinal neurons) onto Gabor-like code vectors. The degree of overcompleteness of this Gabor representation is difficult to estimate: the number of neurons in V1 exceeds the number of inputs by a factor of about one thousand, but these neurons also serve purposes of processing, not only of representation. Nonetheless, considering that many orientations and scales of the Gabor-like kernel can be found for any image location, one can safely assume that the V1 representation is overcomplete.

The idea that sparse coding plays an essential role in sensory perception, and particularly visual perception, has a long history in psychophysics and brain research [3, 6, 69, 71, 20, 22]. It seems likely that a sparse-coding principle is used by V1:

- Cortical cells are much less active than their retinal inputs, *i.e.* the distribution of their activity is super-gaussian [71, 20].
- Simple cells in V1 are nonlinearly tuned to their optimal stimuli.

Examples of the latter phenomenon are end-stopping (*i.e.* a decrease of cell response when the size of the gaussian envelope of the Gabor stimulus is increased beyond the optimum) and cross-orientation inhibition (*i.e.* the nonlinear tuning of simple cells to the orientation of Gabor stimuli). It must be emphasized that the representation in V1 is not ideally sparse, in the sense that the tuning is not so sharp that only one simple cell responds to a Gabor stimulus.

The question arises of whether sparse coding is performed iteratively in V1, as in the algorithms reviewed in this paper. This hypothesis has been formulated in [53]. The experimental evidence is inconclusive: end-stopping does seem to arise from feedback loops within the brain [43]; on the other hand, orientation tuning of simple cells seems to arise too quickly to allow feedback loops to play a role [11]. However, the simple stimuli used in these experiments would produce convergence of any sparse-coding method within a single iteration: such convergence might be too fast to be monitored in the brain. Experiments with natural images might show measurably slow convergence.

7.3 Coding is more difficult than learning

In the last decade, several methods have been developed to learn codebooks for natural images (see *e.g.* [31] in this issue). All of these methods produce similar results, *i.e.* Gabor-like code vectors. It seems unlikely that further research will improve the type of code vectors that are learned. Within the linear-model framework, a more important problem is what sampling of the Gabor parameters is optimal for natural images. Even more interesting problems lie beyond the linear framework [30].

By contrast to learning, the problem of sparse coding is still unsolved even with linear models, and this is the main point of this article. It remains true even though the sub-optimal solutions, produced by the methods developed to date, are suitable for some applications to natural or medical images. Better methods for sparse coding should also lead to better learning algorithms, capable of addressing the issue of optimal sampling of the Gabor parameters.

Acknowledgements

This review was motivated by the summer school/workshop on Statistics of Shapes and Textures in Copenhagen. I am grateful to the organizers and the participants for critical comments and suggestions, to Thomas Mikosch for help with probability theory, to my colleagues in the Image Group for critical discussions and in particular to Peter Johansen, Søren Olsen and Jon Sporring for comments on earlier versions of the manuscript. This work was supported by the Danish Research Agency (Forskningsstyrelsen) through the Natural Shape project.

Appendix A: the Gabor codebook

The Gabor kernel A_G has the form:

$$A_G(u, v; U, V, \theta, \lambda, \phi) = \frac{1}{Z} \exp\left(-\frac{1}{2}\mathbf{\Delta u}^T \cdot \mathbf{\Sigma}^{-2} \cdot \mathbf{\Delta u}\right) \cos(2\pi \mathbf{\Delta u}_1 + \phi) \tag{44}$$

where Z is a normalization factor and we define for convenience

$$\mathbf{\Delta u}^T = \frac{1}{\lambda}\begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix}\begin{pmatrix} u - U \\ v - V \end{pmatrix}$$

and

$$\mathbf{\Sigma}^2 = \frac{1}{2\pi}\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

A.1 Sampling of parameter space

The following discretization applies to square images of size $m = 128^2$ pixels. We define the linear image size $L = \sqrt{m} = 128$.

The scale parameter λ ranges over 11 different scales:

$$\lambda = 2 \cdot \varepsilon^l \text{ pixels}, \qquad 0 \leq l \leq \log_\varepsilon(L/4) \tag{47}$$

where $\varepsilon = \sqrt{2}$.

For all scales except the smallest, the orientation parameter ranges over 8 different orientations:

$$\theta = (k/8)\pi \text{ radians}, \qquad 0 \leq k < 8$$

and the phase parameter ranges over 2 different phases:

$$\varphi \in \{0, \pi/2\} \tag{48}$$

At the smallest spatial scale ($\lambda = 2$ pixels), only Gabor functions in cosine phase ($\varphi = 0$) and with horizontal or vertical orientations ($\theta \in \{0, \pi/2\}$) are used, to avoid aliasing.

In the codebook with no downsampling, the parameters U and V assume all integer values between 1 and L, *i.e.* the Gabor functions are centered on all image pixels. The codebook is thus overcomplete by a factor of $10 \cdot 8 \cdot 2 + 2 = 162$, resulting in over 2.65 million code vectors.

A.2 Subsampling of location parameters

In the downsampled Gabor codebook, the parameters U and V assume values in a range which is scale-dependent:

$$U = i\,[\lambda/2] \text{ pixels}, \qquad 0 \leq i < 2L/\lambda$$
$$V = j\,[\lambda/2] \text{ pixels}, \qquad 0 \leq j < 2L/\lambda$$

where $[\lambda/2]$ is the largest integer such that $[\lambda/2] \leq \lambda/2$. This scheme results in 461440 code vectors.

This subsampling scheme is not ideal: for instance, it does not take into account the anisotropy of the Gaussian envelopes of the Gabor functions. However, it is sufficient to obtain a rough estimate of the cross-talk in a subsampled codebook.

Appendix B: scaling invariance and source densities

Scaling invariance implies an image power spectrum $\sigma_{\mathbf{x}}^2(f)$ decreasing in proportion to the square of spatial frequency f: (see *e.g.* [19]):

$$\sigma_{\mathbf{x}}^2(f) = \frac{K}{f^2} \tag{49}$$

where C is a constant. This prediction is in agreement with empirical results.

The contribution of a source i to the image power spectrum is equal to the variance of the source σ_i^2 times the square of the amplitude of the mixing vector. Summing over all sources:

$$\sigma_{\mathbf{x}}^2(f) = \sum_i \mathbf{a}_i^2(f)\sigma_i^2 \tag{50}$$

Due to the normalization to unit norm, the Fourier transforms $\mathbf{a}_i(f)$ of the mixing/code vectors have peak amplitudes inversely proportional to peak frequency f_i:

$$\mathbf{a}_i(f_i) = \frac{c_p}{f_i} \tag{51}$$

where c_p is a constant. Assuming that the sum of squares over all mixing vectors is proportional to the envelope of the squared peak amplitudes of the mixing vectors:

$$\sum_i \mathbf{a}_i^2(f) \propto \frac{c_p^2}{f^2} \tag{52}$$

it follows that the sum of source variances must be a constant over scales, for Eqs. 49, 50 and 52 to hold.

Given that

- all sources at the same scale must have the same variance, irrespective of location and orientation (due to location and orientation invariance) and
- there is an equal number of sources at every scale;

it follows that the variances of all sources must be constant also over scales:

$$\sigma_i^2 = \sigma_{\mathbf{s}}^2 \qquad (1 \leq i \leq n) \tag{53}$$

To summarize: the relationship between peak amplitudes and peak frequencies of mixing vectors already accounts for the observed scaling of the power spectrum and therefore the sum of squared source values must be constant over scales. Given that the mixing matrix under consideration implies an equal number of sources at all scales, we conclude that the variance of all sources must be the same.

It must be emphasized that this result is based on the assumption that there is an equal number of sources at all scales, and therefore it is not a trivial consequence of scaling invariance.

References

[1] H. Attias, "Independent factor analysis", Neural Comp. vol.11, no.4, pp.803-851, 1999.

[2] F. Attneave, "Informational aspects of visual perception", Psychol. Review vol.61, pp.183-193, 1954.

[3] H.B. Barlow, "Possible principles underlying the transformation of sensory messages" in: *Sensory Communication*, W. Rosenblith, Ed. Cambridge: MIT Press. pp. 217-234, 1961

[4] H.B. Barlow, "Single units and sensation: a neuron doctrine for perceptual psychology?", Perception vol.1, pp.371-394, 1972.

[5] H.B. Barlow, "Unsupervised Learning" Neural Comp., vol.1, pp.295-311, 1989.

[6] H.B. Barlow, "What is the computational goal of the neocortex?" in *Large Scale Neuronal Theories of the Brain*, C. Koch, Ed. Cambridge, MA: MIT Press, pp.1-22, 1994.

[7] H.B. Barlow, T.P. Kaushal, G.J. Mitchison, "Finding minimum entropy codes" *Neural Comp.*, vol.1, 1989, pp.412-423.

[8] E.B. Baum, J. Moody, F. Wilczeck, "Internal representations for associative memory", Biol. Cybern. vol.59, pp.217-228, 1988.

[9] A.J. Bell, T.J. Sejnowski "An information-maximization approach to blind separation and blind deconvolution" *Neural Computation* vol.7, 1995, pp.1129-1159.

[10] A.J. Bell, T.J. Sejnowski "The 'independent components' of natural scenes are edge filters" Vision Res. vol.37, pp.3327-3338, 1997.

[11] S. Celebrini, S. Thorpe, Y. Trotter, M. Imbert, "Dynamics of orientation coding in area V1 of the awake monkey", Visual Neurosci. vol.10, pp.811-825, 1993.

[12] S. Chen, S.A. Billings, W. Luo, "Orthogonal least squares methods and their application to non-linear system identification" Int. J. Control vol.50, no.5 , pp.1873-1896, 1989.

[13] S. Chen, D.L. Donoho "Examples of basis pursuit" Proc. of the SPIE vol.2569, no.2, pp.564-574, 1995.

[14] R. Coifman, V. Wickerhauser "Entropy-based algorithms for best basis selection" IEEE Trans. Info. Theory vol.38 no.2, pp.713-718, 1992.

[15] I. Daubechies "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," IEEE Trans. Info. Theory vol.36, pp.961-1005, 1990.

[16] J.G. Daugman, "Uncertainty relations for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," J. of the Opt. Soc. of Am. A, vol.2, 1985, pp.1160-1169.

[17] G. Davis, S. Mallat, M. Avellaneda, "Greedy adaptive approximation" J. of Constructive Approximation vol.13, pp.57-98, 1997.

[18] D.L. Donoho, "De-noising by soft thresholding", IEEE Trans. Info. Theory vol.41, pp.613-627, 1995.

[19] D.J. Field, "Relations between the statistics of natural images and the response properties of cortical cells" J. Opt. Soc. Am. A vol.4, pp.2379-2394, 1987.

[20] D.J. Field, "What is the goal of sensory coding?" Neural Comp. vol.6, pp.559-601, 1994.

[21] P. Foldiak, "Adaptive network for optimal linear feature extraction", Proc. Int. Joint Conf. Neural Net., Washington, DC, June 18-22, 1989, pp.401-405.

[22] P. Foldiak, M Young, "Sparse coding in the primate cortex", in *The Handbook of Brain Theory and Neural Networks*, ed. Michael A. Arbib, pp. 895-898, 1995.

[23] W.T. Freeman, E.H. Adelson, "The design and use of steerable filters", IEEE Trans. Pat. Anal. Mach. Intell. vol.13,no.9,pp.891-906, 1991.

[24] J.H. Friedman, W. Stuetzle, "Projection pursuit regression" J. of the Am. Stat. Assoc. vol.76 no.376, pp.817-823, 1981.

[25] A. Gersho, R.M. Gray, *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[26] G. Golub, C. van Loan, *Matrix Computations.* (3rd edition) Baltimore: Johns Hopkins University Press, 1996.

[27] G.F. Harpur, R.W. Prager, "Development of low entropy coding in a recurrent network" Network vol.7 no.2, pp.277-284, 1996.

[28] J. Huang, D. Mumford, "Statistics of natural images and models" In: Proc. IEEE Conf. Computer Vis. Pattern Rec., Fort Collins, CO, 2000, vol.1, pp541-547.

[29] A. Hyvärinen, "Gaussian moments for noisy independent component analysis", IEEE Signal Proc. Letters, vol.6 no.6, pp.145-147, 1999.

[30] A. Hyvärinen, "Beyond Independent Components", Proc. Int. Conf on Artificial Neural Networks, Edinburgh, UK, 1999, pp. 809–814.

[31] A. Hyvärinen, M. Inki, "Estimating overcomplete independent component bases for image windows", in this issue.

[32] A. Hyvärinen, E. Oja, "Independent Component Analysis: Algorithms and Applications", Neural Networks vol.13, pp.411-430, 2000.

[33] S. Jaggi, W. Karl, S. Mallat, and A. Willsky. "Silhouette recognition using high-resolution pursuit," Pattern Recognition, vol.32 no.5, pp.753–771, 1999.

[34] J. Jones, L. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex", J. Neurophysiol. vol.58, no.6, pp.1233-1258, 1987.

[35] R.E. Kalman, R.S. Bucy, "New results in linear filtering and prediction theory", Trans. Am. Soc. Mech. Eng. D, J. of Basic Eng. vol.83, pp.95-108, 1961.

[36] S. Kaski, "Dimensionality reduction by random mapping: fast similarity computation for clustering", Proc. of IEEE Int. Joint Conf. Neural Net.: IJCNN'98, Anchorage, Alaska 1998, vol.1, pp.413-418.

[37] F.P. Kourouniotis, R.F. Kubichek, N. Boyd III, A.K. Majumdar, "Application of the wavelet transform and matching pursuit algorithm in seismic data processing for the development of new noise reduction techniques", Int. Symposium on Optical Sci. Eng. and Instrumentation, Denver, CO, August 1996.

[38] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", IEEE Trans. Pat. Anal. Mach. Intell. vol.11, pp.674-693, 1989.

[39] S. Mallat, *A Wavelet Tour of Image Processing*, Academic Press, 1999.

[40] S. Mallat, Z. Zhang, "Matching pursuit with time-frequency dictionaries" *IEEE Trans. Signal Processing* vol.41, no.12, pp.3397-3415, 1993.

[41] A.J. Miller, *Subset selection in regression.* London: Chapman and Hall, 1990.

[42] D. Mumford, B. Gidas, "Stochastic models for generic images". Submitted.
(http://www.dam.brown.edu/people/mumford/Papers/Generic5.ps)

[43] P.C. Murphy, A.M. Sillito, "Corticofugal feedback influences the generation of length tuning in the visual pathway," Nature vol.329, pp.727-729, 1987.

[44] M. Nafie, M. Ali, A.H. Tewfik, "Optimal subset selection for adaptive signal representation" Proc. of the IEEE Int. Conf. on Acoust. Speech and Signal Proc., 1996, pp.2511-2514.

[45] B.K. Natarajan, "Sparse approximate solutions to linear systems" SIAM J. Computing vol.24 no.2, pp.227-234, 1995.

[46] R. Neff and A. Zakhor, "Very low bit-rate video coding based on matching pursuits," IEEE Trans. Circuits and Syst. for Video Technology, vol.7, no.1, pp.158-171, 1997.

[47] A.N. Netravali, B.G. Haskell, *Digital Pictures: Representation and Compression.* Plenum Press 1988.

[48] B.A. Olshausen, D.J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images" Nature vol.381, pp.607-609, 1996.

[49] B.A. Olshausen, D.J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" Vision Res., vol.37, pp.3311-3325, 1997.

[50] B.A. Olshausen, K.J. Millman, "Learning sparse codes with a mixture-of-Gaussians prior," Advances in Neural Information Processing Systems, 12, S.A. Solla, T.K. Leen, and K.R. Muller, Eds. MIT Press, 2000, pp. 841-847.

[51] B.A. Olshausen, P. Sallee, M.S. Lewicki, "Learning sparse image codes using a wavelet pyramid architecture," Advances in Neural Information Processing Systems, 13, T.K. Leen, T.G. Dietterich, V. Tresp, eds. MIT Press (in press).

[52] G. Palm, "On associative memory", Biol. Cybern. vol.36, pp.19-31, 1980.

[53] A.E.C. Pece, "Redundancy reduction of a Gabor representation: a possible computational role for feedback from primary visual cortex to lateral geniculate nucleus" in: *Artificial Neural Networks* 2 I. Aleksander, J. Taylor, Eds. Amsterdam: Elsevier Science Publishers, 1992, pp.865-868.

[54] A.E.C. Pece, "A linear-operator model for the generation of natural images and its implications for sparse image coding", in preparation.

[55] A.E.C. Pece, N. Petkov, Fast atomic decomposition by the inhibition method. Proceedings of the 15th International Conference on Pattern Recognition: ICPR 2000, Barcelona, Spain, September 2-8, 2000, pp.215-218.

[56] V.V. Petrov, *Limit Theorems of Probability Theory*, Clarendon, Oxford, 1995.

[57] P.J. Phillips, "Matching pursuit filters applied to face identification" IEEE Trans. Image Proc. vol.7, no.8, pp.1150-1164, 1998.

[58] M. D. Plumbley. "Information processing in negative feedback neural networks" Network, vol.7, no.2, pp.301-305, 1996.

[59] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C* (2nd edition). Cambridge: Cambridge University Press, 1992.

[60] D.B. Rowe, "Bayesian blind source separation", submitted to IEEE Trans. Signal Proc. (http://www.hss.caltech.edu/ drowe/BBSS.ps)

[61] D.L. Ruderman, "The statistics of natural images", Network vol.5, pp.517-548, 1994.

[62] S. Roweis, Z. Ghahramani, "A Unifying Review of Linear Gaussian Models", Neural Computation vol.11, no.2, pp.305-345, 1999.

[63] A. Said, W.A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees", IEEE Trans. Circuits and Syst. for Video Technol. vol.6 no.3 pp.243-250, 1996.

[64] A. Shmilovici and O. Maimon. "Application of adaptive matching pursuit to adaptive control of nonlinear dynamic systems," IEE Proceedings - Control Theory and Application vol.145 no.6, pp.575–582, 1998.

[65] E.P. Simoncelli, E.H. Adelson, "Noise removal via bayesian wavelet coring", Proc. Int. Conf. Im. Proc., Lausanne (CH), Sept. 1996, pp.379-383.

[66] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, D.J. Heeger, "Shiftable multiscale transforms", IEEE Trans. Info. Theory, vol.38 no.2, pp.587-607, 1992.

[67] A. Stuart, K. Ord, *Kendall's Advanced Theory of Statistics, Vol.I: Distribution Theory* (6th Edition). London: Edward Arnold, 1994.

[68] M.J. Wainwright, E.P. Simoncelli, A.S. Willsky, "Random cascades on wavelet trees and their use in analyzing and modelling natural images", Proc. 45th Annual Meeting of SPIE, San Diego (CA), July 30-Aug.4, 2000.

[69] S. Watanabe *Pattern Recognition: Human and Mechanical.* Wiley, New York, 1985.

[70] H. Yoshida. "Matching pursuit with optimally weighted wavelet packets for extraction of microcalcifications in mammograms." Applied Signal Processing vol.5 no.3, pp.127–141, 1999.

[71] C. Zetzsche, "Sparse coding: the link between low level vision and associative memory", In *Parallel processing in neural systems and computers*, R. Eckmiller, G. Hartmann, G. Hauske, eds. North-Holland, Amsterdam, 1990, pp.273-276.

[72] M. Zibulevsky. B.A. Pearlmutter, "Blind Source Separation by Sparse Decomposition in a Signal Dictionary", Neural Computation, in press.