Probabilistic framework for the adaptation and comparison of image codes

Michael S. Lewicki

Howard Hughes Medical Institute, Computational Neurobiology Laboratory, The Salk Institute,
10010 North Torrey Pines Road, La Jolla, California 92037

Bruno A. Olshausen

Center for Neuroscience, University of California, Davis, 1544 Newton Court, Davis, California

We apply a Bayesian method for inferring an optimal basis to the problem of finding efficient image codes for natural scenes. The basis functions learned by the algorithm are oriented and localized in both space and frequency, bearing a resemblance to two-dimensional Gabor functions, and increasing the number of basis functions results in a greater sampling density in position, orientation, and scale. These properties also resemble the spatial receptive fields of neurons in the primary visual cortex of mammals, suggesting that the receptive-field structure of these neurons can be accounted for by a general efficient coding principle. The probabilistic framework provides a method for comparing the coding efficiency of different bases objectively by calculating their probability given the observed data or by measuring the entropy of the basis function coefficients. The learned bases are shown to have better coding efficiency than traditional Fourier and wavelet bases. This framework also provides a Bayesian solution to the problems of image denoising and filling in of missing pixels. We demonstrate that the results obtained by applying the learned bases to these problems are improved over those obtained with traditional techniques.   © 1999 Optical Society of America
[S0740-3232(99)03107-5]

OCIS codes: 000.5490, 100.2960, 100.3010.

1. INTRODUCTION

The problem of encoding sensory information efficiently is relevant both to the design of practical vision systems and to advancing our understanding of how biological nervous systems process information. Within the image-processing community, much work has been done on image codes that utilize a linear basis function expansion, and considerable effort has gone into choosing sets of basis functions that satisfy certain mathematical desiderata or that have desirable properties, such as ease of computability.

An approach that has been largely overlooked, however, is consideration of the efficiency of the image code as defined by Shannon's source coding theorem, i.e., how well the basis functions capture the data's probability density. Typically, bases that are chosen for their low-entropy coding properties, such as two-dimensional (2D) Gabor functions or wavelets,[1–5] are hand designed rather than adapted to the data, so as to optimize coding efficiency.

We show in this paper how the problem of image coding may be cast within a probabilistic modeling framework. Instead of making prior assumptions about the shape or form of the basis functions, we adapt the bases to the data (natural images), using an algorithm that maximizes the log-probability of the data under the model,[6] thereby optimizing coding efficiency in the sense of Shannon.

Shannon's source coding theorem states that the lower bound on code-word length is determined by the entropy of the data:

$$\mathcal{L} \geqslant H(p) = -\sum p(x)\log p(x). \qquad (1)$$

In many cases, $p(x)$, the true density of the data, is unknown and must be approximated by a density $q(x)$. In this case the lower bound on the expected code-word length becomes

$$\mathcal{L} = E[l(X)] \geqslant \sum_x p(x)\log\frac{1}{q(x)}, \qquad (2)$$

$$= \sum_x p(x)\log\frac{p(x)}{q(x)} + \sum_x p(x)\log\frac{1}{p(x)}, \quad (3)$$

$$= D_{KL}(p\|q) + H(p), \qquad (4)$$

where $D_{KL}(p\|q)$ is the Kullback–Leibler divergence between p and q. Thus, if the model density is equal to the true density, then $D_{KL}(p\|q) = 0$ and the expected code length is bounded by the entropy. Otherwise, there is a penalty of $D_{KL}(p\|q)$ in the lower bound of the average code length. Thus the better the model captures the underlying probability density, the lower the bound on average code-word length.

Among existing techniques for modeling data with a set of basis functions are principal-components analysis (PCA) and a recently developed generalization of PCA called independent-components analysis (ICA). PCA assumes that the data distribution has Gaussian structure and fits an appropriate orthogonal basis, while ICA generalizes PCA by allowing for non-Gaussian distributions and nonorthogonal bases.[7–9] Two limitations common to

both of these techniques are that they do not allow for noise to be modeled separately from the signal structure and that they do not allow for overcomplete codes in which there are more basis functions than input dimensions.

In this paper we draw on an approach that generalizes ICA in two ways that are relevant to learning efficient image codes.[6,10] The first generalization is that additive noise is explicitly included in the model. For image codes, this allows direct specification of the encoding precision and calculation of theoretical rate-distortion curves by use of Shannon's source coding theorem. Explicitly modeling additive noise also provides a Bayesian solution for the problem of image denoising and filling in of missing pixels. We demonstrate here that using these methods with the learned bases produces improved results in comparison with results obtained with traditional techniques.

The second generalization is that the number of basis functions can be greater than the dimensionality of the inputs. Both ICA and PCA are restricted to the case in which the set of basis functions forms a complete or critically sampled basis, i.e., the number of basis vectors is equal to the dimensionality of the input. Overcomplete bases have been advocated because they allow certain advantages in terms of interpolation,[11] in achieving a tight frame with nonorthogonal basis functions,[5] or in achieving sparsity in the representation.[10,12] One approach that has been proposed for adapting a basis is to select from an overcomplete "dictionary" of basis functions a subset that yields a low-entropy description[12–14] of a particular signal or a class of signals such as texture (see Ref. 15, for example). A drawback of this approach is that the basis functions are still prespecified and are often chosen for rather *ad hoc* or intuitive criteria (e.g., 2D Gabor functions appear suitable for capturing oriented structure in images). In this paper the question of whether overcomplete representations can better capture the structure of images is tested directly by evaluating the relative coding efficiency.

An intriguing aspect of codes adapted to natural images is their resemblance to the receptive fields of neurons in the primary visual cortex.[10,16] Previous attempts to account for the structure of V1 receptive fields in terms of quantitative principles have been based either on PCA[17,18] or on the fact that 2D Gabor functions provide an optimal trade-off in achieving localization in both the spatial-position and spatial-frequency domains.[1,5] However, in the former case the basis functions learn only from the pairwise statistics in the images and so do not become localized unless artificially constrained (see also Refs. 19 and 20); in the latter case it is unclear why joint localization in the space/spatial-frequency domains is desirable in the first place or that it is a principle that could be generalized to higher stages of processing and other modalities. The results obtained here, as well as similar results obtained with related methods,[16,21,22] suggest that the localized, oriented, and bandpass structure of V1 receptive fields can be accounted for in terms of a rather general coding principle, i.e., formation of a probabilistic model of images in terms of a superposition of sparse, statistically independent representational elements.

We begin by describing the linear, generative image model and its probabilistic interpretation, and we show how to adapt the basis functions to maximize the probability of the model. The algorithm is then applied to natural images, and the resulting basis functions are fitted with 2D Gabor functions and analyzed in terms of their tiling of the joint domain of position, orientation, and peak spatial-frequency tuning. These results are compared with data from the receptive-field properties of neurons in the primary visual cortex. To compare the learned basis with traditional bases, we use the probabilistic model to compute the relative coding efficiencies. Finally, we show how the generative image model can be applied to practical problems such as image denoising and filling in.

2. MODEL FOR IMAGES

The proposed probabilistic model for images is based on a linear, generative model. Each observed image, $\mathbf{x} \equiv x_1, \ldots, x_L$, is assumed to be composed of a linear superposition of basis functions plus additive noise:

$$\mathbf{x} = \mathbf{As} + \boldsymbol{\epsilon}, \qquad (5)$$

where $\mathbf{A}$ is an $L \times M$ matrix whose columns are the basis functions, $\mathbf{s}$ is an M-element vector of basis coefficients, and $\boldsymbol{\epsilon}$ is assumed to be Gaussian white noise. Any given image $\mathbf{x}$ thus has an internal representation in the model $\mathbf{s}$, which specifies which basis functions in $\mathbf{A}$ compose the image. The number of basis functions can be greater than the number of dimensions in the input, in which case the basis is overcomplete.

Our goals are twofold: (1) to find a good matrix $\mathbf{A}$ for coding natural images and (2) to infer for each image the proper state of the coefficients $\mathbf{s}$. The first problem is one of adaptation and is analogous to the process of learning (through either development or evolution) in the visual system, while the second problem is one of image representation and is most analogous to perception. We shall take up the latter problem first and then address the problem of adaptation.

A. Inferring an Image Representation
The problem of determining the coefficients $\mathbf{s}$ in Eq. (5), given only information about the image $\mathbf{x}$, is ill-posed for two reasons. The first is that the basis functions will generally not be linearly independent of each other (owing to overcompleteness), and thus there will be multiple states of $\mathbf{s}$ that can account for the same image. The second reason is that the noise $\boldsymbol{\epsilon}$ is unknown. Thus $\mathbf{s}$ must be inferred from $\mathbf{x}$. We do this by maximizing the conditional probability distribution of $\mathbf{s}$ given $\mathbf{x}$, $P(\mathbf{s}|\mathbf{x}, \mathbf{A})$, which can be expressed by means of Bayes's rule as

$$P(\mathbf{s}|\mathbf{x}, \mathbf{A}) \propto P(\mathbf{x}|\mathbf{A}, \mathbf{s})P(\mathbf{s}). \qquad (6)$$

The first term specifies the likelihood of the image under the model for a given state of the coefficients. Because the noise is assumed to be Gaussian, this likelihood is given by $P(\mathbf{x}|\mathbf{A}, \mathbf{s}) \propto \exp[-(\lambda/2)|\mathbf{x} - \mathbf{As}|^2]$, where $\lambda = 1/\sigma^2$ and σ is the standard deviation of the additive noise. The noise level determines the encoding precision.

The second term specifies the prior probability distribution over the basis coefficients. (We assume that this prior does not depend on the basis matrix $\mathbf{A}$ and thus $P(\mathbf{s}|\mathbf{A}) = P(\mathbf{s})$.) We choose this distribution to be factorial and Laplacian, $P(s_m) \propto \exp(-\theta_m|s_m|)$, which assumes that $\mathbf{A}$ decomposes the images into sparse, statistically independent components.[16] More will be said about this choice of prior below.

Maximizing the posterior distribution $P(\mathbf{s}|\mathbf{x}, \mathbf{A})$ thus presents us with the following problem:

$$\hat{\mathbf{s}} = \max_{\mathbf{s}} P(\mathbf{s}|\mathbf{x}, \mathbf{A}), \tag{7}$$

$$= \max_{\mathbf{s}}[\log P(\mathbf{x}|\mathbf{A}, \mathbf{s}) + \log P(\mathbf{s})], \tag{8}$$

$$= \min_{\mathbf{s}}\left(\frac{\lambda}{2}|\mathbf{x} - \mathbf{As}|^2 + \theta^T|\mathbf{s}|\right). \tag{9}$$

In other words, we need to find the state of the coefficients with minimum L_1 norm that also minimizes mean squared reconstruction error. This problem is formally equivalent to that of "basis pursuit de-noising" proposed by Chen et al.[12] In our case, however, the L_1 norm arises from the Laplacian prior. Under this prior (or other super-Gaussian priors), finding the most probable basis coefficients essentially selects out a complete basis and sets the coefficients for the remaining vectors to zero. Thus, even though the generative image model is linear, the most probable basis coefficients are a nonlinear function of the image.

Note that if $P(\mathbf{s})$ is chosen to be factorial and Gaussian, then $\hat{\mathbf{s}}$ is simply a linear function of $\mathbf{x}$, given by the pseudoinverse of $\mathbf{A}$ when $\boldsymbol{\epsilon} = 0$:

$$\hat{\mathbf{s}} = \mathbf{A}^+\mathbf{x}. \tag{10}$$

In the special case where $\mathbf{A}$ is orthogonal, then we have $\hat{\mathbf{s}} = \mathbf{A}^T\mathbf{x}$ (again, assuming zero noise). If we specify $\mathbf{A}$ to be Fourier basis and the variance on s_i corresponds to the power spectrum of the images $\mathbf{x}$ then the linear mapping from $\mathbf{x}$ to $\mathbf{s}$ is the well-known Wiener filter. Our approach, by contrast, is entirely based on the use of non-Gaussian priors, which lead to nonlinear image codes.

B. Adapting the Basis Vectors
Our goal in adapting the basis vectors is to obtain a good model of the distribution of natural images. The goodness of fit can be assessed by computing the average log-probability of images under the model

$$\mathcal{L} = \langle \log P(\mathbf{x}|\mathbf{A})\rangle, \tag{11}$$

where the distribution $P(\mathbf{x}|\mathbf{A})$ is obtained by marginalizing over the internal states $\mathbf{s}$:

$$P(\mathbf{x}|\mathbf{A}) = \int d\mathbf{s} P(\mathbf{x}|\mathbf{A}, \mathbf{s})P(\mathbf{s}). \tag{12}$$

To the extent that the model is accurate, $-\mathcal{L}$ provides a lower bound for the number of bits required to code the images, and the more accurate the model, the closer one can approach the true bound. Thus our goal amounts to finding the basis matrix $\mathbf{A}$ that maximizes the data's log-probability ($\mathcal{L}$) or, equivalently, minimizes the data's coding cost ($-\mathcal{L}$).

Equation (12) makes clear that the form of the model distribution depends not only on the choice of basis functions $\mathbf{A}$ but also on the choice of prior $P(\mathbf{s})$. If the noise model is Gaussian, then choosing $P(\mathbf{s})$ to be Gaussian will result in the entire model distribution $P(\mathbf{x}|\mathbf{A})$ also being Gaussian. As such, it will be able to describe only second-order statistical structure, as specified by the covariance matrix. Because it is well established that images are not well described by Gaussian distributions,[4,16,23,24] we are thus obligated to choose a non-Gaussian prior. The specific form we choose for the prior is to be sparse and factorial. By a sparse prior, we mean that the probability distribution of each coefficient's activity, $P(s_i)$, is highly peaked around zero and with heavy tails. Such a distribution reflects the notion that natural images should be described in terms of a small number of descriptive elements[4,10]; thus any given coefficient will rarely be active, and when it does become active, it takes on a value along a continuum. We choose here to represent such a distribution using a Laplacian, but other super-Gaussian shapes are also possible. The joint distribution of the coefficients is chosen to be factorial, $P(\mathbf{s}) = \Pi_i P(s_i)$, in line with Barlow's proposal that an efficient code should try to decompose the image in terms of statistically independent elements.[25,26]

Maximizing $\mathcal{L}$ with respect to $\mathbf{A}$ will thus find a set of basis functions that best account for the structure in images in terms of sparse, statistically independent elements. This can be accomplished in the most straightforward fashion by gradient ascent:

$$\Delta\mathbf{A} \propto \frac{\partial\mathcal{L}}{\partial\mathbf{A}} = \frac{\partial}{\partial\mathbf{A}}\langle\log P(\mathbf{x}|\mathbf{A})\rangle, \tag{13}$$

$$= \left\langle\frac{1}{P(\mathbf{x}|\mathbf{A})}\int\frac{\partial}{\partial\mathbf{A}}P(\mathbf{x}|\mathbf{s}, \mathbf{A})P(\mathbf{s})d\mathbf{s}\right\rangle, \tag{14}$$

$$= \left\langle\frac{1}{P(\mathbf{x}|\mathbf{A})}\int\lambda\mathbf{es}^T P(\mathbf{x}|\mathbf{s}, \mathbf{A})P(\mathbf{s})d\mathbf{s}\right\rangle, \tag{15}$$

$$= \left\langle\int\lambda\mathbf{es}^T P(\mathbf{s}|\mathbf{x}, \mathbf{A})d\mathbf{s}\right\rangle, \tag{16}$$

$$= \lambda\langle\langle\mathbf{es}^T\rangle_{P(\mathbf{s}|\mathbf{x}, \mathbf{A})}\rangle, \tag{17}$$

where $\mathbf{e} = \mathbf{x} - \mathbf{As}$. The practical problem that is presented by Eq. (17) is that of averaging over the internal states $\mathbf{s}$ for each image presentation. One possible avenue might be to use efficient methods for sampling from the posterior $P(\mathbf{s}|\mathbf{x}, \mathbf{A})$. This characterizes in part the approach taken by Olshausen and Field,[10] although that algorithm samples the posterior only at its maximum, ignoring the volume and thus requiring an additional adaptive step to scale the basis functions so that the coefficients have the same variance as that dictated by the prior. Here we explore an alternative route[6] based on approximating the posterior with the best-fitting Gaussian distribution, which allows the integral to be solved analytically. Such an approximation seems reasonable because, for the Laplacian prior, there is in most cases a single maximum in the posterior (in some pathological cases, the posterior will be a ridge), and so a Gaussian could be capable of capturing most of the volume under the posterior. An advantage of this technique is that it allows us to calculate explicitly an approximation to $\mathcal{L}$,

which allows for the objective comparison of different image models (i.e., different bases or priors).

The Gaussian approximation of the posterior leads to the following expression for $\mathcal{L}$:

$$\mathcal{L} \approx \text{const.} - \left\langle \frac{\lambda}{2}|\mathbf{x} - \mathbf{A\hat{s}}|^2 + \log P(\mathbf{\hat{s}}) - \frac{1}{2}\log \det \mathbf{H}\right\rangle,$$
(18)

where $\mathbf{H}$ is the Hessian of the log posterior at $\mathbf{\hat{s}}$, given by $\lambda \mathbf{A}^T \mathbf{A} - \nabla_s \nabla_s \log P(\mathbf{\hat{s}})$. We assume the image patches to be independent. Performing gradient ascent on this expression yields the following learning rule (see Appendix A for derivation):

$$\Delta \mathbf{A} \propto \lambda \langle \mathbf{es}^T - \mathbf{AH}^{-1}\rangle.$$
(19)

Note that the first term is precisely Olshausen and Field's[16] learning rule, while the second term results from approximating the volume under the posterior and thus does away with the need for the additional rescaling step that was used in Olshausen and Field's algorithm. Premultiplying this rule by $\mathbf{AA}^T$ yields the form given by Lewicki and Sejnowski[6]:

$$\Delta \mathbf{A} \propto -\mathbf{A}(\mathbf{zs}^T + \mathbf{A}^T \mathbf{AH}^{-1}),$$
(20)

where $\mathbf{z} = \mathrm{d} \log P(\mathbf{s})/\mathrm{d}\mathbf{s}$ (see Appendix A for details). This form of the learning rule is more stable, and it was used to learn the basis functions in the examples below.

3. LEARNING CODES FOR NATURAL SCENES

Here we learn complete and 2×-overcomplete representations of natural scenes with the data set used by Olshausen and Field[16] (whitened images of Alaska nature scenes).

A. Learning Procedure

The bases were initialized to random Gaussian blobs with positions that were evenly distributed over the input area. The initial bases were generated by first setting the basis elements to random values between $[-1,1]$ and then scaling these values by a 2D Gaussian envelope that had a standard deviation of 0.25 pixels in the complete case and 1 pixel in the 2×-overcomplete case. This ensured that the initial set of basis functions spanned the input space. Similar results were obtained with random initial bases, but convergence was slower.

To further speed convergence, we used the modifications of the basic gradient-ascent procedure described previously.[6] For each gradient [Eq. (20)] a step size was computed by $\delta_i = \epsilon_i/a_{\max}$, where $a_{\max}$ is the element of the basis matrix $\mathbf{A}$ with largest absolute value. The parameter ϵ was reduced from $0.02r$ to $0.001r$ over the first 1000 iterations and fixed at $0.001r$ for the remaining iterations. The parameter r is a measure of the data range and was set equal to be the standard deviation of the data. Exponential averaging ($\epsilon_i = 0.9\epsilon_{i-1} + 0.1\delta_i$) was used to ensure smoothness of the steps. Learning was stopped after 10 000 gradient steps, at which point both of the bases learned here were stable. The training data consisted of 12×12 image patches randomly sampled
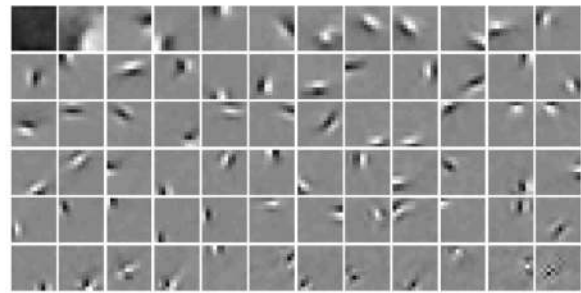
(a)
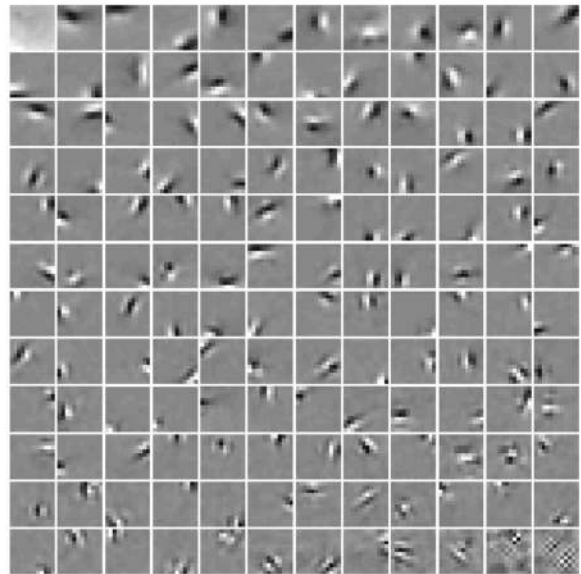
(b)

Fig. 1.   Results from training (a) complete and (b) 2×-overcomplete bases on natural scenes. The graphs plot and the odd-numbered basis functions in decreasing order of L^2 norm.

from the ten 512×512 images in the data set of Olshausen and Field.[16] The patches were repeatedly resampled throughout training to avoid reuse of any one set of patches. Training required approximately 12 h of computing time on a 200-Mhz processor.

The most probable basis function coefficients, $\mathbf{\hat{s}}$, were obtained with a modified conjugate-gradient routine.[27] The basic routine was modified to replace the line search with an approximate Newton step. This approach resulted in a substantial improvement in speed and produced much better solutions in a fixed amount of time than the standard routine. A convergence tolerance of 0.02 was used for the examples shown here. This was typically required between five and ten conjugate-gradient steps, each requiring computation of the posterior gradient. The noise level was set to $\lambda = 3000$, which corresponds to $\sim$7.8 bits of precision for the encoded image patches.

B. Results

A sample of the learned basis functions (the odd-numbered) is shown in Fig. 1, in decreasing order of L^2 norm. Nearly all the learned basis functions show a Gabor-like structure, as has been found previously.[16,21]

The basis functions largest in magnitude also have the lowest peak spatial-frequency tuning. Peak spatial-frequency tuning becomes progressively higher with decreasing magnitude. The checkerboardlike basis functions are smallest in magnitude and resemble those obtained from PCA which assumes that the data have Gaussian structure. These could reflect an attempt of the model to capture small-amplitude noise in the images.

The learned basis functions also resemble the spatial receptive-field profiles of simple cells found in the primary visual cortex of mammals, which numerous investigators have likened to Gabor functions.[1,5,28] One of the reasons that the Gabor basis has been advocated as a model of V1 image coding, as well as for efficient image coding in general, is that it possesses the attractive property of optimal localization in both the spatial-position and spatial-frequency domains.[1,5] It is thus quite interesting that, although the learned basis functions were completely unconstrained in terms of what form they take on within the 12×12 grid, the form that does emerge resembles a Gabor wavelet basis. However, besides the question of how well Gabor functions (or any functional form) fit individual receptive fields, there is the more difficult technical question of how a population of such func-

tions tile the joint space/spatial-frequency domain to form a complete basis for image representation. Fourier representations, for example, do this quite simply by tiling the spatial-frequency domain evenly in linear frequency. But the Gabor basis presents one with the choice of many parameters or degrees of freedom in deciding how to tile the space, e.g., achieving fine resolution in orientation versus coarse coverage in spatial position. The algorithm used here learns the basis functions and automatically chooses these parameters to tile the space so as to maximize the probability of the data under the model.

C. Analysis of Basis Function Tiling Properties

The basis functions learned by the algorithm can be compared with other attempts that have been made to tile a physiologically plausible Gabor basis "by hand."[2,3,5] To analyze the tiling properties of the learned basis functions, we fitted each basis function with a Gabor function of the form

$$g(x, y) = a \exp\left(-\frac{1}{2}\left\{\left[\frac{u(x, y)}{\sigma_u}\right]^2 + \left[\frac{v(x, y)}{\sigma_v}\right]^2\right\}\right)$$
$$\times \cos[2\pi f u(x, y) + \phi], \qquad (21)$$
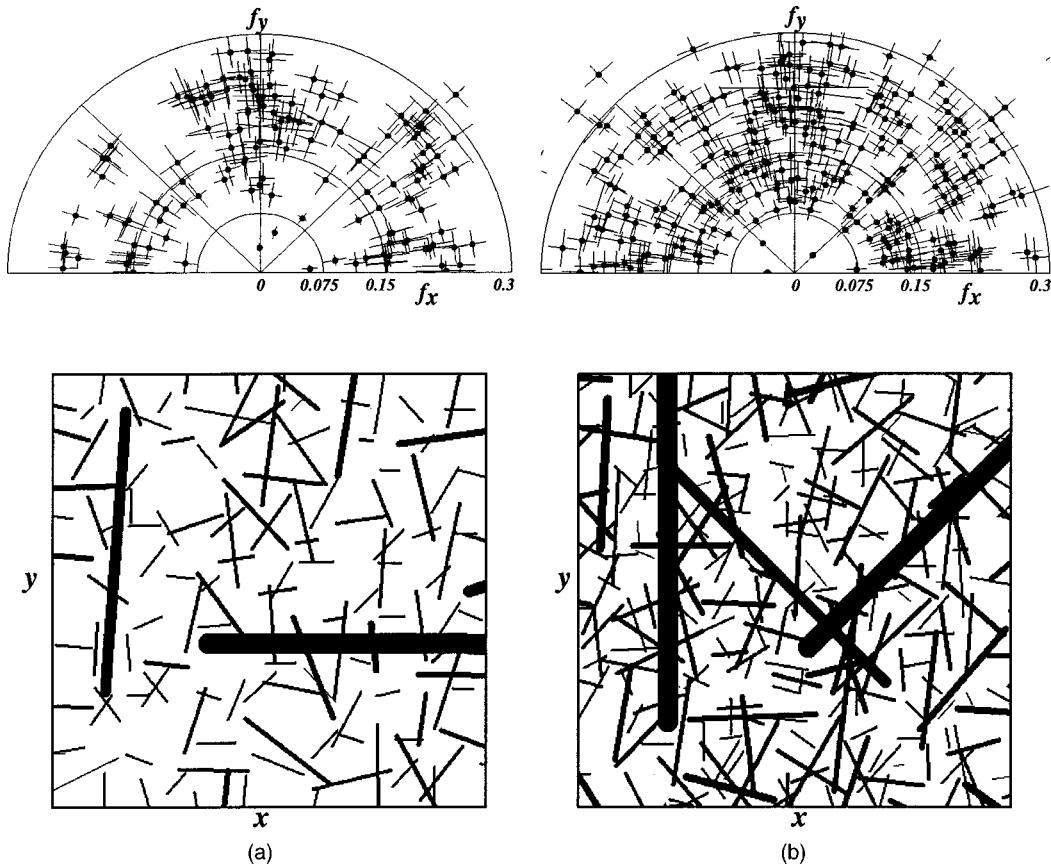

Fig. 2. Basis function characteristics for (a) the complete case and (b) the 2×-overcomplete case. Each basis function was fitted by a Gabor function to characterize its position, spatial-frequency selectivity, and orientation. At the top are polar plots of the peak spatial-frequency tuning and orientation selectivity. Each dot denotes the center spatial frequency and orientation of a fitted basis function. The cross hairs indicate the 1/4-bandwidth in spatial frequency and orientation. The plots at the bottom show the spatial layout of the same set of basis functions. Each bar denotes the center position and orientation of a fitted basis function within the 12×12 grid. The thickness and length of each line denotes its spatial-frequency band (lower spatial frequencies are represented with thicker lines, and vice versa). Increasing the degree of overcompleteness results in a denser tiling of the joint four-dimensional space of position, orientation, and spatial frequency.

$$u(x, y) = (x - x_0)\cos(\theta) + (y - y_0)\sin(\theta), \qquad (22)$$

$$v(x, y) = -(x - x_0)\sin(\theta) + (y - y_0)\cos(\theta). \qquad (23)$$

Note that Gabor functions, as defined by the joint minimization of uncertainty in spatial position and spatial frequency,[1] are complex valued. To obtain a physically meaningful expression, here we consider only its real part.

The parameters a, x_0, y_0, σ_u, σ_v, θ, f, ϕ were adjusted by conjugate-gradient descent to minimize the squared error between the learned basis function and the model Gabor $g(x, y)$. Because the error surface contains local minima, multiple initial conditions were used, and the parameters that formed the best fit were taken as the final solution. Note that the reference frame of the Gaussian envelope was locked to the orientation of the cosine grating. In addition, a soft constraint was placed on the size of the envelope so that its width ($\sim 2\sigma_u$) did not fall much below one-half wavelength of the carrier grating. This was necessary to discourage pathological solutions that combined a very small envelope with a very-low-frequency grating, which often happened in cases in which there was only one positive and one negative lobe in the learned basis function.

The Gabor functions parameterized in this way fitted the learned basis functions quite well, and the mean squared error for this example was 8% of the variance of the basis functions. Figure 2 shows the result of this analysis. One trend that appears immediately obvious is that the preferred orientation tends to align vertically and horizontally, but we suspect that this is an artifact of our having used a rectangular sampling grid to digitize the images rather than a reflection of an intrinsic property of the images themselves.

The Gabor characterization of the learned basis functions shows some differences when they are compared with physiologically determined receptive fields. The average bandwidths of the learned basis functions are 1.8 ± 0.2 and 1.7 ± 0.2 octaves for the 1× and 2× basis sets, respectively, whereas physiologically determined receptive fields tend to lie in the range of 1 to 1.5 octaves. The average aspect ratios of the basis functions were 1.32 ± 0.5 and 1.22 ± 0.3 (1× and 2× basis sets), compared with the 2:1 aspect ratios that tend to be more typical of the physiology. Thus the basis functions were somewhat more broadband and less selective in orientation than those found physiologically. However, it should be remembered that the basis functions of the model are not generally equivalent to receptive fields because of the nonlinear mapping from the image space **x** to the coefficient space **s** [see Eq. (9)]. To ascertain the receptive fields of the model, they would have to be mapped with spots and gratings, and previous experience with this[10] has shown that the nonlinearity tends to make units more selective than one would predict from a simple linear input–output relationship. Thus it is possible that this process could sharpen the tuning properties and bring them closer to the receptive fields determined physiologically.

Lee[5] has designed a physiologically plausible Gabor basis by constraining the aspect ratio of the functions to be 2:1 and the bandwidth to be between 1 and 1.5 octaves.

To form a complete code with such a basis, the higher-spatial-frequency functions must tile space more densely than the low-spatial-frequency functions. If the separation between each spatial-frequency band is one octave, then the sampling density increases by a factor of 4 for each octave. The learned bases also show a similar increase in sampling density, as shown in Fig. 3. The number of basis functions lying in the three spatial-frequency bands 0–0.075, 0.075–0.15, and 0.15–0.3 cycles/pixel are 4:23:117 for the 1× basis set and 4:78:206 for the 2× basis set. Thus there is an approximate 5-fold increase for each octave in the 1× basis, while the 2× basis shows a 2.6-fold increase for the middle-to-high-transition and a 19-fold increase for the low-to-middle transition. This general trend toward higher sampling density at higher spatial frequencies found in both Lee's basis and the bases learned here is at odds with the observed distribution of peak spatial-frequency tuning in V1 cortical neurons. The vast majority of recorded cells appear to reside in the mid- to low-spatial-frequency range when scaled to the retinal sampling lattice.[29,30] One possible explanation for this discrepancy is that the highest-spatial-frequency cells were greatly undersampled in these ex-
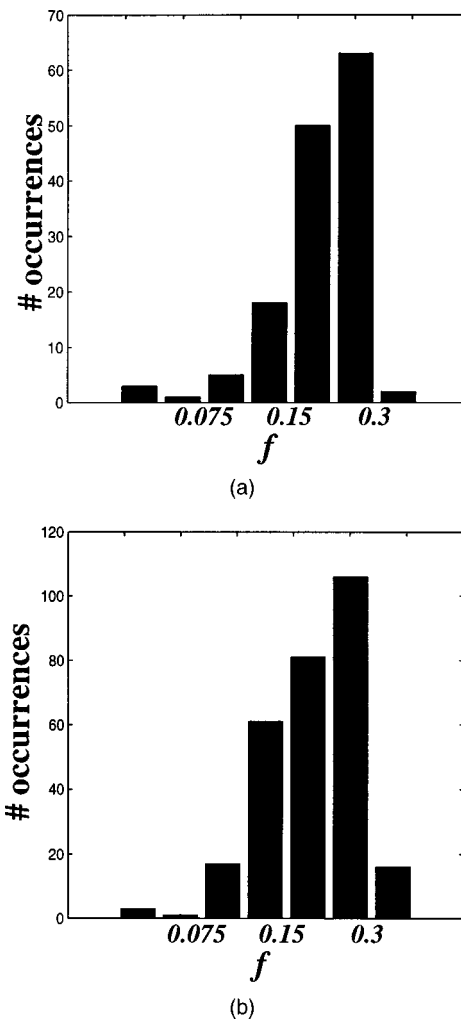

(a)

(b)

Fig. 3. Histogram of peak spatial-frequency bandwidths (in cycles per pixel) for (a) complete and (b) 2×-overcomplete learned basis functions.

periments, since these cells will be relatively difficult to isolate in comparison with the low-spatial-frequency cells. Alternatively, it has been suggested that this discrepancy might be resolved in part by considering the time domain of the visual signal.[31]   When basis functions are learned for natural movies, the distribution of spatial frequency is more spread out, presumably because of the trade-off between tiling velocity and spatial frequency.

The basis function characteristics found here are generally consistent with previous results obtained with related methods.[10,21,22]   The differences are that Olshausen and Field[10] show a somewhat more multimodal distribution of spatial-frequency tuning clustered at either low, medium, or high frequencies, while the bases of Bell and Sejnowski,[21] as well as those of van Hateren and van der Schaaf,[22] appear more highly skewed toward the highest spatial frequencies.   There are a number of differences between our approach and these previous approaches that could account for the broader distribution of spatial-frequency tuning found here.   One is the level of noise assumed.   Olshausen and Field assumed a relatively high noise level compared with that assumed here ($\lambda \approx 100$ versus $\lambda = 3000$), whereas the methods of Bell and Sejnowski and of van Hateren and van der Schaaf assume zero noise.   The other difference lies in the choice of prior on the coefficients.   Olshausen and Field used a generalized Cauchy prior, whereas Bell and Sejnowski use the prior $P(s_m) \propto \mathrm{sech}(s_m)$ (corresponding to the hyperbolic tangent output nonlinearity), which is less peaked at zero (approximately Gaussian for s_m between -1 and 1, or approximately 50% of the total probability) and as a result is less sparse than the Laplacian that we employ.   The method of van Hateren and van der Schaaf simply seeks non-Gaussianity and is thus ambivalent as to the degree of sparseness.

4.   COMPARISON WITH TRADITIONAL BASES

One of the advantages of the probabilistic framework is that alternative bases can be compared objectively in terms of coding efficiency.   To estimate how well a particular basis represented a given set of data, we followed two methods described in detail in Ref. 6.

The first method is to use Shannon's theorem directly to obtain a lower bound on the number of bits required to encode the pattern, which is

$$\text{number of bits} \geq -\log_2 P(\mathbf{x}|\mathbf{A}) - L \log_2(\sigma_x), \quad (24)$$

where L is the dimensionality of the input pattern $\mathbf{x}$ and σ_x is the standard deviation of the additive noise.   This defines the precision of the encoding and essentially discretizes the continuous distribution $P(\mathbf{x}|\mathbf{A})$ into hyperbins of size σ_x.   The higher the noise level, the coarser the encoding.   As the noise level goes to zero, the code words become longer.   Shannon's coding theorem states that this expression will give a lower bound on coding length if the model is correct, but if the assumptions of the model are wrong, e.g., if $P(\mathbf{s})$ does not match the observed coefficient distribution, this measure will overesti-

mate the bound by an amount equal to the Kullback–Leibler divergence between the model density and the true density.

The second method is to calculate the entropy of the basis vector coefficients.   A single function, $f(s)$, is used to estimate the probability density for all the coefficients from the observed distributions on a training set.   The coding cost for a test data set is computed by estimating the entropy (in bits per pattern) of the fitted coefficients

$$\text{number of bits} \geq -\sum_i \frac{n_i}{N} \log_2 f[i], \quad (25)$$

where the notation $f[i]$ represents the fact that $f(s)$ is quantized to a precision needed to maintain an encoding noise level of σ_x and n_i is the number of counts observed in each bin of $f[\ ]$ for each of the coefficients fitted to a data set consisting of N patterns.   See Ref. 6 for details. This method has the advantage over the probability method that it is not a bound but a direct measure of coding cost, albeit with a simple coding scheme.   The entropy method has the drawback that it does not include the cost of misfitting the data, but, because it uses the actual distribution of the observed coefficients, it can yield a more accurate estimate of the coding efficiency if the data are well fitted.   Thus the estimate of coding efficiency based on entropy can be higher or lower than the estimate of the bound based on probability, depending on the extent to which the data can be encoded to maintain an error of σ_x and on the accuracy of the probabilistic model.

In the comparisons below, the learned basis functions were obtained with the same methods as described above but with a data set consisting of randomly sampled 8×8 image patches.   The basis coefficients were fitted to test data by the same procedure as above but with a tolerance of 10^{-4}.   The Laplacian-prior parameter θ_m was adapted to fit the density of s_m obtained from the images.   The noise level was set to be the same as that used during the learning, ~ 7.8 bits/pixel [$\sigma_x = 1/(3000)^{1/2}$]. The estimated coding efficiencies were calculated by using 8×8 image patches randomly sampled from a test data set.   The standard deviations of these estimates were calculated by using ten different test data sets of 100 image patches each.

A.   Comparison of Complete Bases ($A_{64 \times 64}$)
Figure 4 shows some of the bases used in the comparison with the learned basis: a Gabor wavelet basis fitted to the learned basis, a basis obtained with PCA, the standard Fourier basis, the Haar wavelet basis, a Daubechies wavelet basis [without wrap-around, $N = 2$ (Ref. 32)] basis, and a Gabor wavelet basis.   The Gabor wavelet basis was constructed with the methods of Lee,[5] in which the Gabor basis functions are constrained so that they better match the properties of V1 simple cells.   The Gabor basis was constructed by using the parameters $K = 3$, $N = 1$, $a_0 = 2$, $b_0 = 2$ (using Lee's[5] notation) with an aspect ratio of 2:1 and a bandwidth of 1.5.   Three levels generated 67 basis functions (only 64 are shown in the figure).   Not shown are the learned bases for 8×8 image patches

(similar to those shown in Fig. 1), a Daubechies wavelet basis with wrap-around on the 8×8 grid, and a pixel basis (**A** is the identity).

We also tested a basis learned by using the standard ICA learning rule. Because these are complete bases, the only difference between the ICA learning rule and the rule used here is the assumption of additive noise and the

choice of the prior. ICA assumes zero noise and a prior corresponding to $P(s_m) \propto \mathrm{sech}(s_m)$, which assumes that the distribution of **s** is less sparse than under the Laplacian.

Table 1 shows the estimated coding efficiencies in bits per pixel for the various bases. The results show that the learned basis is between 0.7 and 1.1 bits/pixel more effi-
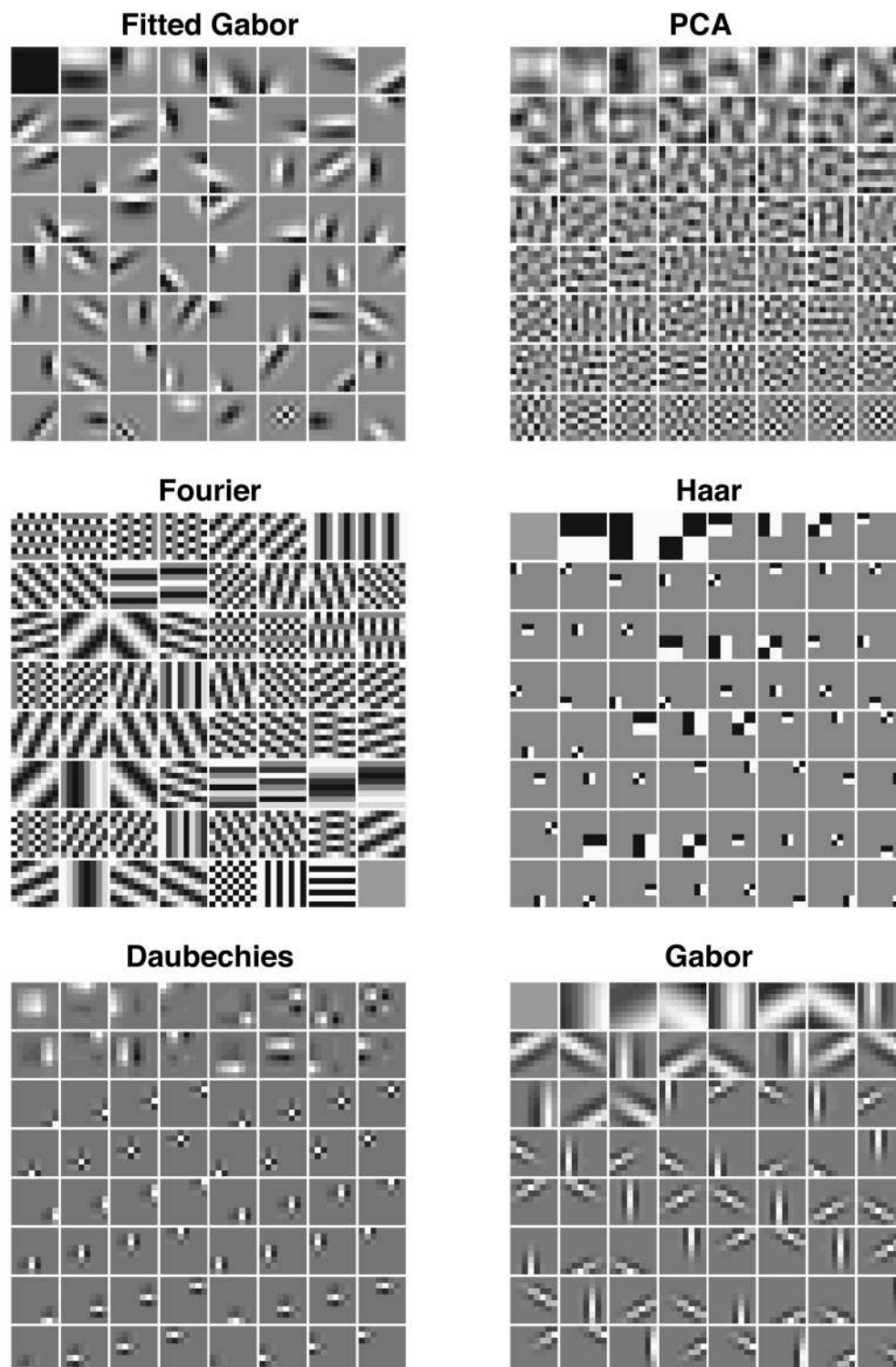

Fig. 4.   Some of the complete bases used for comparison with the learned basis.

Table 1. Bits per Pixel for Complete Bases on Natural Image Data Set

	Estimation Method	
Basis	$-\log_2 P(\mathbf{x}|\mathbf{A}) - L \log_2(\sigma_x)$	Entropy of $\mathbf{s}$
Learned	4.69 ± 0.05	4.11 ± 0.05
ICA	4.80 ± 0.06	4.71 ± 0.05
Gabor fit	5.86 ± 0.02	3.60 ± 0.04
PCA	5.43 ± 0.07	5.22 ± 0.06
Fourier	5.48 ± 0.08	5.34 ± 0.07
Haar	5.55 ± 0.07	5.46 ± 0.06
Daubechies (wrap)	5.60 ± 0.07	5.51 ± 0.07
Daubechies (nonwrap)	5.63 ± 0.07	5.49 ± 0.07
Gabor	7.28 ± 0.36	8.20 ± 0.10
Pixel	5.79 ± 0.08	5.77 ± 0.08
JPEG		4.36

cient than any of the nonadapted bases. Note that although the fitted Gabor basis achieves the best coding efficiency in terms of entropy, it is among the worst in terms of probability. This means that this set of basis functions does not efficiently span the space of natural images, i.e., achieving both a low-entropy-coefficient distribution and a misfit consistent with the assumed noise level. The misfit cost is not taken into account by the entropy estimate but is considered in the efficiency estimate based on probability.

One reason for why the Gabor bases do not represent the data efficiently is that in many cases, especially for images with significant high-spatial-frequency structure, very large coefficients are required to achieve low residual errors. The large difference in coding efficiency between the hand-generated and the fitted Gabor clearly shows that it is difficult to choose values for the large number of free parameters in a Gabor basis so that the space is optimally tiled. This problem is exacerbated by using a small sampling grid. The approach here (i.e., the learning algorithm) optimizes these parameters to maximize coding efficiency and solves the tiling problem by adapting the basis functions themselves, which are not necessarily constrained to be of Gabor form, to the data.

The coding efficiency is reflected in the distributions of the coefficients obtained when natural images are encoded. Figure 5 shows the histograms and kurtosis values for some of the bases used in the coding efficiency comparisons. The adapted bases, the $1\times$ and $2\times$ learned bases, the Gabor bases fitted to the $1\times$ learned basis, and the PCA basis, all show much larger kurtosis values than the Fourier, Haar, Daubechies, or Gabor basis. Note that all the histograms for the nonadapted bases are non-Gaussian, consistent with previous observations.[4,33] It is also evident from these figures that the Laplacian assumed by the model does not describe either of the distributions for the learned bases particularly well. It would be desirable to incorporate a prior that better describes the actual coefficient distribution, which would lead to better coding efficiency. The difficulties associated with this are discussed below.

The predicted compression rates for these bases might seem poor compared with standard compression algo-

rithms, but it should be kept in mind that these rates are for near-noiseless compression (7.8 bits of resolution/pixel) on black-and-white image patches that are already whitened. For comparison, the table also reports the compression rate achieved by using JPEG on the same data set. JPEG is designed as a lossy compression algorithm and contains a quality parameter that controls the trade-off between image quality and compressed size. To achieve a comparable encoding precision required a quality parameter between 96 and 97, which yielded a measured encoding precision of 7.62 and 7.90 bits/pixel, respectively, with a corresponding compression rate of 4.16 and 4.58 bits/pixel. Interpolating to get an encoding precision of 7.77 bits/pixel (the same precision used for the other estimates) gives an approximate compression rate of 4.36 bits/pixel. This efficiency is significantly better than that of the Fourier basis (on which JPEG is based), reflecting the fact that JPEG is able to reduce redundancy among the coefficients, whereas the model assumes that
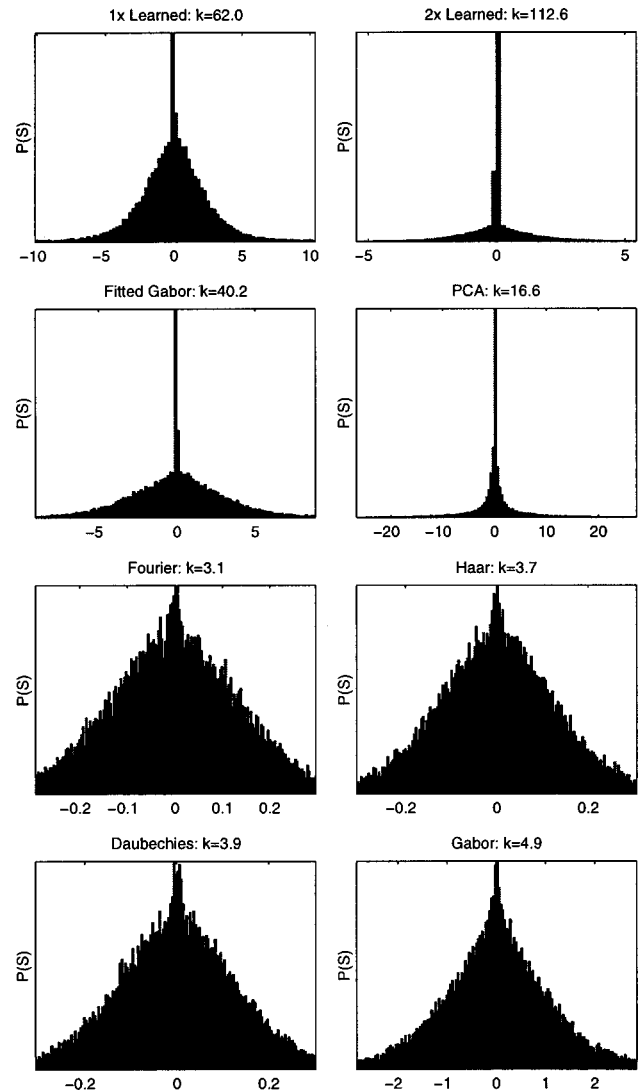

Fig. 5. Coefficient histograms for some of the bases used in the coding efficiency comparisons. Each histogram shows 97.5% of the coefficient range. The vertical axis is scaled so that each histogram peak falls within the plot. The sample kurtosis is shown for each histogram.

the coefficients are independent. The estimated coding efficiency of the learned basis functions (4.11, with the entropy estimate) shows a small but significant improvement, suggesting that incorporating learned basis functions into compression algorithms could yield improved compression rates. This comparison shows that there are at least two routes to an efficient code. Compression algorithms such as JPEG uses the mathematically convenient basis and compensate for the correlations in the coefficients. The approach presented here learns a set of basis functions such that the coefficients are maximally independent.

B. Checking the Coding Efficiency Estimates

To check the consistency of the coding efficiency estimates, we generated an artificial data set by synthesizing data from the pixel basis with a Laplacian distribution for the coefficients. This produced images that consisted of independent pixels, each with a Laplacian intensity distribution.

Table 2 shows the estimated coding efficiencies for the same set of bases. For this data set, the pixel basis achieves the best coding efficiency, and the learned and the Gabor-fit bases perform much worse than any of the fixed bases. This indicates that the coding efficiency estimates are consistent, because the true generating basis gives the best estimated coding efficiency. Also note that for the pixel basis, the two estimates of coding efficiency are nearly identical. This indicates that the approximation to $P(\mathbf{x}|\mathbf{A})$ is reasonably accurate, because in this case the assumptions of the model are correct.

C. Comparison of 2×-Overcomplete Bases ($\mathbf{A}_{64\times128}$)

Two bases were used for comparison with the learned 2×-overcomplete bases. A Fourier basis was generated by evenly sampling in frequency, orientation, and phase, and a Gabor basis was generated with the parameters $K = 3$, $N = 1$, $a_0 = 2$, $b_0 = 1.5$ (with Lee's[5] notation) with an aspect ratio of 2:1 and a bandwidth of 1.5. Four levels generated 118 basis functions. Table 3 shows the coding efficiency estimates in bits per pixel for these 2×-overcomplete bases, in comparison with the learned bases. Again, the learned basis achieves better coding efficiency than the hand-tiled, unadapted Fourier or Gabor basis on the same image data set. The relatively large values for the entropy of the Gabor basis suggest that there are strong correlations in the basis function coefficients.

One might expect that as more basis functions are added to the overcomplete representation, the coding efficiency should increase, because the basis functions can become more and more specific and obtain a better approximation of the underlying density. A comparison of Tables 1 and 3, however, shows that this is not the case. A likely reason for this is that the assumptions of the model are breaking down for higher degrees of overcompleteness. In particular, the present model assumes that the coefficients $\mathbf{s}$ are independent, an assumption that becomes increasingly inaccurate for higher degrees of overcompleteness. Models that can capture a greater variety

Table 2. Bits per Pixel for Complete Bases on Pixel Data Set

Basis	Estimation Method	
	$-\log_2 P(\mathbf{x}|\mathbf{A}) - L \log_2(\sigma_x)$	Entropy of $\mathbf{s}$
Natural image basis	7.45 ± 0.10	5.54 ± 0.02
Gabor fit	9.65 ± 0.29	4.88 ± 0.05
ICA	7.28 ± 0.09	6.12 ± 0.03
PCA	5.06 ± 0.01	4.99 ± 0.01
Fourier	5.06 ± 0.01	4.99 ± 0.03
Haar	5.03 ± 0.01	4.98 ± 0.01
Daubechies (wrap)	5.03 ± 0.01	4.97 ± 0.01
Daubechies (nonwrap)	5.02 ± 0.01	4.97 ± 0.01
Gabor	8.99 ± 0.26	8.63 ± 0.05
Pixel	4.89 ± 0.01	4.87 ± 0.01

Table 3. Bits per Pixel for 2×-Overcomplete Bases on Images Data Set

Basis	Estimation Method	
	$-\log_2 P(\mathbf{x}|\mathbf{A}) - L \log_2(\sigma_x)$	Entropy of $\mathbf{s}$
Learned	6.28 ± 0.04	6.81 ± 0.08
Gabor fit	6.46 ± 0.05	6.33 ± 0.07
Fourier	7.35 ± 0.17	7.89 ± 0.11
Gabor	7.16 ± 0.19	12.19 ± 0.16

of structure in the coefficients, such as with hierarchical priors, could achieve better representation and greater coding efficiency.

5. NOISE REMOVAL

To demonstrate the ability of the adapted bases to capture typical structure in the data, we applied the algorithm to the problem of noise removal in images. This task is well suited to the algorithm because Gaussian additive noise is incorporated into the specification of the image model. A set of bases that characterizes the probability distribution of the data well should have improved noise removal properties, because they will be better at inferring the most probable image in the face of uncertainty.

A 60×60 subimage was extracted from the training set, and Gaussian noise with variance of 0.05 was added to the image, which had a variance of 0.124 [signal-to-noise ratio (SNR) = 3.9 dB]. The image model was applied to nonoverlapping 12×12 blocks for different basis sets. On each image presentation the coefficients are fitted so as to maximize the probability of the image, with a Laplacian prior on the coefficients (i.e., the same as in learning):

$$\hat{\mathbf{s}} = \max_{\mathbf{s}} P(\mathbf{s}|\mathbf{x}, \mathbf{A}) \qquad (26)$$

$$= \min_{\mathbf{s}} \left[\frac{\lambda}{2} |\mathbf{x} - \mathbf{A}\mathbf{s}|^2 + \theta^T |\mathbf{s}| \right]. \qquad (27)$$

The denoised image $\hat{\mathbf{x}}$ is then computed as

$$\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{s}}. \qquad (28)$$

The resulting image reconstructions are shown in Fig. 6. The learned bases appear to do well at rejecting noise in the flat, uniform luminance regions of the image and also capture more details of the structure (e.g., the details in the lower left portion of the image are a bit sharper). Denoising with the Wiener filter, which uses the Fourier basis and a Gaussian prior as its image model, produces a perceptually different reconstruction, reflecting different assumptions about the underlying image structure. Quantitatively, there is only a slight improvement in using the learned basis functions over the Wiener filter (SNR = 9.5 versus 8.6 dB). It should be noted that because the images were preprocessed by low-pass filtering and whitening, the Wiener filter is using information that is mainly in the corners of the 2D frequency domain.

This method of denoising has many elements in common with the method of Bayesian wavelet "coring" developed by Simoncelli and Adelson.[34] Both utilize a transformation through a set of basis functions to reveal sparse, non-Gaussian histograms on the coefficients, and both utilize Bayesian inference based on these histograms to estimate the underlying signal. The main difference lies in the way that the optimal coefficient values are arrived at. The coefficient values in the approach presented here are computed iteratively by maximizing the posterior distribution over the coefficients, whereas Simoncelli and Adelson compute the coefficients by simply projecting the image onto the basis functions and then computing the mean of the posterior distribution given these coefficients (no iteration required). Both methods appear to yield sizable gains over Wiener filtering, but how they compare with each other on similar image ensembles at similar noise levels deserves further exploration.

The method described here is virtually identical to the "basis pursuit denoising" method.[35] The only difference here is that the learned bases (or "dictionaries," in their terms) have been adapted to the signal structure, according to the same probability model used for inferring the denoised image, rather than using a basis set that is specified *a priori*. Again, careful tests that compare the relative merits of these different techniques have yet to be done.

6. FILLING IN MISSING PIXELS

The same procedure that was used for denoising can also be used to fill in missing pixels, because missing information can be viewed as another form of noise. If the noise level of the missing pixels is set to infinity ($\lambda_m = 0$) in the likelihood function

$$P(\mathbf{x}|\mathbf{A}, \mathbf{s}) \propto \exp\left(-\sum_i \frac{\lambda_i}{2} |\mathbf{x} - \mathbf{As}|_i^2 \right), \qquad (29)$$

then the procedure for finding the most probable values of the coefficients,

$$\hat{\mathbf{s}} = \min_{\mathbf{s}}\left(\sum_i \frac{\lambda_i}{2} |\mathbf{x} - \mathbf{As}|_i^2 + \theta^T |\mathbf{s}| \right), \qquad (30)$$

will result in an image,

$$\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{s}}, \qquad (31)$$

that interpolates the values for these pixels. Figure 7 shows an example where the complete ($1\times$) learned bases were used to infer the image structure when 70% of the pixel values had been removed. Compared with spatial interpolation using, for example, cubic splines [Fig. 7(c)], the model reconstruction is able to fill in features such as lines or edges, whereas spline interpolation smoothes among the available pixels (compare, for example, the patches in the second row, first column, and also the patches in the third row, fourth column). The model reconstruction gives 5.3 dB versus 3.7 dB SNR for interpolation. Further experiments (not shown) indicate that reconstruction by the model performs increasingly better than spline interpolation (in terms of mean squared error) with an increasing amount of missing information.

Everson and Sirovich[36] describe a similar method for filling in missing pixels by using the Karhunen–Loève transform applied to a specific image class, i.e., faces. In terms of our framework, their procedure corresponds to using a Gaussian prior, and thus it captures only the second-order statistics in the data. The method described here, by contrast, utilizes the higher-order statistics because of the non-Gaussian prior. In the case of natural images, this approach allows the model to fill in missing data by using learned features like edges and lines.
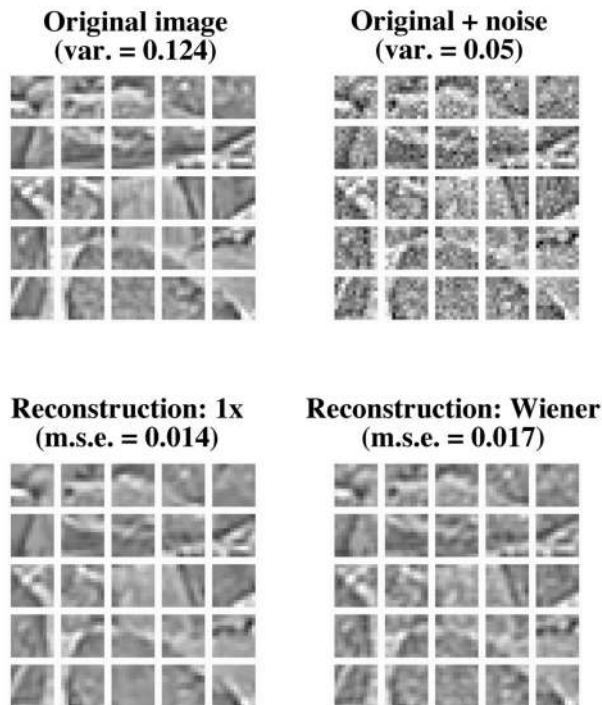

Original image (var. = 0.124)   **Original + noise (var. = 0.05)**

Reconstruction: 1x (m.s.e. = 0.014)   **Reconstruction: Wiener (m.s.e. = 0.017)**

Fig. 6. Demonstration of image denoising by use of the $1\times$ (complete) basis set. Each image is shown tiled into nonoverlapping 12×12 blocks, to which the image model was then applied. The results (lower left) show both a qualitative and a quantitative improvement over Wiener filtering (lower right).

original
(var. = 0.124)

71% pixels missing

model
reconstruction
(m.s.e. = 0.037)

interpolation
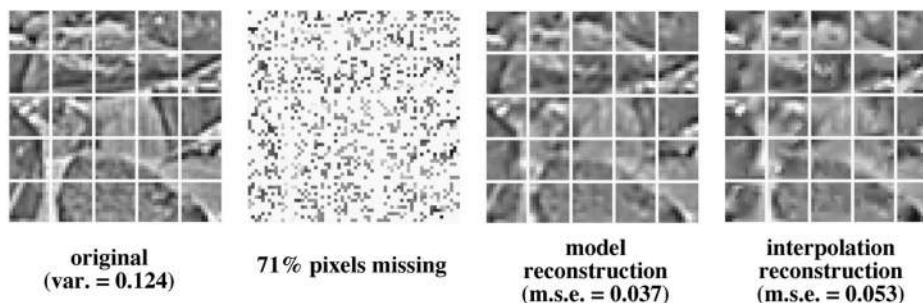reconstruction
(m.s.e. = 0.053)

Fig. 7.   Reconstruction of missing information.   From the original image, 71% of the pixels were removed and reconstructed with the methods described in the text.   The reconstruction by the model is superior to the reconstruction based on spline interpolation, because the model can fill in actual structure in the image, whereas spline interpolation only smoothes between available pixels.   This can be seen by comparing the image patches in row 2, column 1 and row 3, column 4.

7. DISCUSSION

We have shown in this paper how the framework of probabilistic inference can be employed both for learning efficient image codes [Eq. (20)] and for inferring the most probable representation for a given image [Eq. (9)].   We have demonstrated moderate success in learning bases that capture the underlying statistical structure of images and have demonstrated how these can be compared quantitatively with a number of standard image codes. In these comparisons, the learned bases show a 15–20% improvement over traditional bases and compare favorably with compression rates of highly optimized compression algorithms such as JPEG.

This framework also allowed us to test the viability of using overcomplete representations for the purposes of efficient coding.   Overcomplete codes have been shown to yield greater coding efficiency on some test data sets,[6] but for the natural image data used here, overcomplete codes did not yield an improvement in coding efficiency.   Although the entropy per coefficient was less in the overcomplete case than in the complete case, this reduction in entropy did not outweigh the cost of having to transmit twice as many coefficients.   This could be the result of both inaccuracies in the approximation used to estimate the data likelihood and shortcomings in the present image model.

An important concern in our present specification of the model is the accuracy of the coefficient prior, $P(\mathbf{s})$. Clearly, there is a need for this prior to be more flexible. For overcomplete codes, a Laplacian is not sparse enough to reflect the fact that, for each pattern, a subset of the coefficients will be zero.   Buccigrossi and Simoncelli[33] have observed that coefficients of wavelet representations of images are more sparse than predicted by the Laplace distribution and can be well modeled with a generalized Laplace distribution ($\log P(s) \propto -|\theta s|^p$).   Another possibility for improving the prior is to use a mixture distribution consisting of a delta function at zero and a second function describing the distribution of nonzero coefficients.   A more flexible approach proposed recently is to model $P(\mathbf{s})$ with Gaussian mixtures.[37,38]   Better approximation of the coefficient prior would allow the model to better capture the actual coefficient distribution, but two important problems must be addressed before the benefits of such a model can be realized.   The first is finding the most probable coefficient values.   For the cases of posi-

tive noise ($\epsilon > 0$) and overcomplete representations, computing the most probable coefficients is not straightforward, although recent research has made progress in finding the most probable coefficients for overcomplete representations with a generalized Laplacian prior.[39]   A second issue is how to evaluate or approximate the integral required to compute the data probability, $P(\mathbf{x}|\mathbf{A})$ [Eq. (12)].   This problem remains a challenge for general models.

In this report we approximate $P(\mathbf{x}|\mathbf{A})$ by approximating the coefficient posterior distribution $P(\mathbf{s}|\mathbf{x}, \mathbf{A})$ with a Gaussian.   One promising approach for more accurate estimates is to model $P(\mathbf{s})$ with a Gaussian mixture[37,38] and again use a Gaussian approximation at the (maximum) posterior mode.   A more general approach, suggested by Eq. (17), is to use Monte Carlo methods[40] to estimate $P(\mathbf{x}|\mathbf{A})$ by sampling the coefficient posterior.

It should be emphasized, however, that, despite these shortcomings of the particular models and approximations used, the probabilistic framework described here provides a new perspective on the utility of working with an adapted basis set, i.e., a better modeling of the underlying probability density.   In the case of nonzero additive noise or an overcomplete basis, the resulting image representation is not a simple linear transformation of the image but a result of a nonlinear inference process that finds the most probable explanation of the image.   A byproduct of the algorithm used to infer the representations (the encoding step) is that it naturally lends itself to denoising and filling in of missing data.   A further advantage of the probabilistic framework is that the assumptions about the form of the model as well as the noise are made explicit and can be tested objectively.

A second issue addressed in this article is the relevance of the learned codes to neurobiology.   Almost 40 years ago Barlow[25] proposed the principle of redundancy reduction for neural coding, i.e., that a population of neurons should form a factorial code in which the neural outputs for the particular data ensemble, such as natural scenes, are statistically independent.   The general framework applied here is one of density estimation, i.e., estimating $P(\mathbf{x}|\mathbf{A})$ [Eq. (12)], which minimizes the Kullback–Leibler divergence between the model density and the distribution of the data.   Under certain forms of the model, this is equivalent to the methods of redundancy reduction and maximizing the mutual information between the input

and the representation[41-43] and has been advocated by several researchers.[3,25,26,44-46] Because the model we have used here assumes statistical independence of the basis function coefficients, it provides a direct method for generating predictions from these principles about the structure of population codes.

The analyses of both complete and overcomplete bases adapted to natural images suggest that some of the properties of V1 receptive fields can be accounted for by Barlow's efficient coding principle. This result is consistent with previous observations,[10,16,21,22] and in fact the learning algorithm used by Olshausen and Field[10,16] and the ICA learning rule used by Bell and Sejnowski[21] can both be derived from this framework.[6] The notion of efficient coding can be defined only with respect to a model, and this represents perhaps the simplest non-Gaussian model that produces Gabor-like receptive fields. The class of models used here can capture only the most elementary structure in natural images. This perhaps is one reason that the overcomplete bases did not result in improved coding efficiency in spite of producing a more dense tiling of the Gabor spaces that is closer to that observed in the physiological population.

Whether these principles are actually used by the brain is an issue that can be addressed only by using these principles to make explicit predictions and contrasting these with physiology. The framework used here finds compact descriptions of arbitrary high-dimensional data spaces and has the potential for a wide variety of applications. It will be exciting to apply this framework to pattern domains where good codes remain largely unknown. We do not suggest that efficient coding is the only principle underlying cortical function. It remains to be seen to what extent these ideas, which show promise in accounting for elementary aspects of sensory coding, will apply to higher levels of cortical processing.

APPENDIX A

1. Derivation of the Learning Rule

A derivation of the learning rule has been presented previously by Lewicki and Sejnowski.[6] Here we present an alternate derivation that demonstrates more directly the connections to the previous learning rule of Olshausen and Field.[10] The log-probability of the data has the form

$$\log P(\mathbf{x}|\mathbf{A}) = \text{const.} + F(\mathbf{A}, \hat{\mathbf{s}}) + V(\hat{\mathbf{s}}), \tag{A1}$$

where

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} F(\mathbf{A}, \mathbf{s}), \tag{A2}$$

$$F(\mathbf{A}, \mathbf{s}) = -\frac{\lambda}{2}|\mathbf{x} - \mathbf{As}|^2 + \log P(\mathbf{s}), \tag{A3}$$

$$\propto \log P(\mathbf{s}|\mathbf{x}, \mathbf{A}), \tag{A4}$$

$$V(\mathbf{A}, \mathbf{s}) = -\frac{1}{2}\log|\det \mathbf{H}(\mathbf{s})|, \tag{A5}$$

$$H(\mathbf{s}) = -\nabla\nabla F(\mathbf{A}, \mathbf{s}). \tag{A6}$$

Throughout, ∇ denotes the gradient with respect to $\mathbf{s}$, and ∇_k denotes the kth component of that gradient. Some useful quantities are

$$\mathbf{e} = \mathbf{x} - \mathbf{As}, \tag{A7}$$

$$\mathbf{z} = \nabla \log P(\mathbf{s}), \tag{A8}$$

$$\nabla F = \lambda \mathbf{A}^T \mathbf{e} + \mathbf{z}, \tag{A9}$$

$$B(\mathbf{s}) = -\nabla\nabla \log P(\mathbf{s}), \tag{A10}$$

$$H(\mathbf{s}) = -\nabla\nabla F(\mathbf{s}) = \lambda \mathbf{A}^T\mathbf{A} + B(\mathbf{s}), \tag{A11}$$

$$y_k = 2\nabla_k V = (H^{-1})_{kk}\frac{\mathrm{d}\mathbf{B}_{kk}}{\mathrm{d}s_k}, \tag{A12}$$

$$\frac{\mathrm{d}\hat{s}_k}{\mathrm{d}A_{ij}} = \lambda[e_i \mathbf{H}_{kj}^{-1} - (\mathbf{AH}^{-1})_{ik}s_j]. \tag{A13}$$

The derivation for expression (A13) is given below.

The derivative of F with respect to A is

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{A}}F[\mathbf{A}, \hat{\mathbf{s}}(\mathbf{A})] = \frac{\partial F}{\partial \mathbf{A}} + \nabla F \cdot \frac{\partial \mathrm{d}\hat{\mathbf{s}}}{\partial \mathbf{A}}, \tag{A14}$$

and because $\nabla F = 0$ at $\hat{\mathbf{s}}$, we are left only with the first term, which yields

$$\frac{\mathrm{d}F}{\mathrm{d}\mathbf{A}} = \lambda \mathbf{e}\hat{\mathbf{s}}^T. \tag{A15}$$

The derivative of V with respect to $\mathbf{A}$ is

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{A}}V[\mathbf{A}, \hat{\mathbf{s}}(\mathbf{A})] = \frac{\partial V}{\partial \mathbf{A}} + \nabla V \cdot \frac{\partial \mathrm{d}\hat{\mathbf{s}}}{\partial \mathbf{A}}. \tag{A16}$$

The first term is

$$\frac{\partial V}{\partial \mathbf{A}} = -\lambda \mathbf{AH}^{-1}, \tag{A17}$$

and the second two terms are given above [Eqs. (A12) and (A13)], yielding

$$\frac{\mathrm{d}V}{\mathrm{d}\mathbf{A}} = \lambda\left[-\mathbf{AH}^{-1} + \frac{1}{2}(\mathbf{ey}^T\mathbf{H}^{-1} - \mathbf{AH}^{-1}\mathbf{y}\hat{\mathbf{s}}^T)\right]. \tag{A18}$$

Combining the derivatives of F [Eq. (A15)] and V [Eq. (A18)] gives us the total learning rule

$$\Delta\mathbf{A} \propto \lambda\left[\mathbf{e}\hat{\mathbf{s}}^T - \mathbf{AH}^{-1} + \frac{1}{2}(\mathbf{ey}^T\mathbf{H}^{-1} - \mathbf{AH}^{-1}\mathbf{y}\hat{\mathbf{s}}^T)\right]. \tag{A19}$$

We conjecture that the terms involving $\mathbf{y}$ can be ignored because they represent curvature components that are unrelated to the volume. In practice, omitting these terms yields more stable learning. The rule then reduces to

$$\Delta\mathbf{A} \propto \lambda(\mathbf{e}\mathbf{s}^T - \mathbf{AH}^{-1}). \tag{A20}$$

It might appear that the learning rule does not incorporate information about the gradient of the prior, but bear in mind that at the maximum of $F(\mathbf{s})$, $\nabla F = 0$ and so $\lambda \mathbf{A}^T\mathbf{e} = -\mathbf{z}$ [Eq. (A9)]. Thus if the posterior has been properly maximized, $\mathbf{e}$ reflects information about the prior.

This rule can be written in the form derived by Lewicki and Sejnowski[6] by premultiplying by $\mathbf{AA}^T$

$$\mathbf{A}\mathbf{A}^T\Delta\mathbf{A} \propto \lambda(\mathbf{A}\mathbf{A}^T\mathbf{e}\mathbf{s}^T - \mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{H}^{-1}], \qquad \text{(A21)}$$

$$= -\mathbf{A}(\mathbf{z}\mathbf{s}^T + \mathbf{A}^T\mathbf{A}\mathbf{H}^{-1}), \qquad \text{(A22)}$$

where the last step is obtained by using $\lambda\mathbf{A}^T\mathbf{e} = -\mathbf{z}$.

2. Derivation of $d\hat{s}_k/dA_{ij}$

The quantity $d\hat{s}_k/dA_{ij}$ can be computed by observing that at the maximum of $F(\mathbf{s})$, the gradient of F with respect to $\mathbf{s}$ must remain zero as we perturb A_{ij}. Thus if we perturb A_{ij} by a small amount, the resulting change in the coefficients, $d\hat{s}$, must be such that $\nabla F = 0$. The derivative of ∇F with respect to A_{ij} is

$$\frac{\partial\nabla_k F}{\partial A_{ij}} = \lambda(\delta_{kj}\mathbf{e}_i - A_{ik}\mathbf{s}_j), \qquad \text{(A23)}$$

where $\delta_{kj} = 1$ if $k = j$ and $\delta_{kj} = 0$ otherwise. The derivative of ∇F with respect to $\mathbf{s}$ is simply $-\mathbf{H}$ [Eq. (A6)]. For these two changes to cancel, we require that

$$\frac{\partial\nabla F}{\partial A_{ij}} - \mathbf{H}\frac{d\hat{\mathbf{s}}}{dA_{ij}} = 0. \qquad \text{(A24)}$$

Thus

$$\frac{d\hat{\mathbf{s}}}{dA_{ij}} = \mathbf{H}^{-1}\frac{\partial\nabla F}{\partial A_{ij}}. \qquad \text{(A25)}$$

Writing this out in terms of each component $d\hat{s}_k$ and substituting into Eq. (A23), we get

$$\frac{d\hat{s}_k}{dA_{ij}} = \lambda[e_i\mathbf{H}_{kj}^{-1} - (\mathbf{A}\mathbf{H}^{-1})_{ik}s_j]. \qquad \text{(A26)}$$

This relation assumes that changes in $\hat{s}_k$ are smooth with respect to changes in A_{ij}, which may not be true at a small number of critical points because the mapping from $\mathbf{x}$ to $\hat{\mathbf{s}}$ is nonlinear [Eq. (9)].

3. Approximating $\lambda\mathbf{A}^T\mathbf{A}\mathbf{H}^{-1}$

The expression for the term $\lambda\mathbf{A}^T\mathbf{A}\mathbf{H}^{-1}$ in expression (20) can be approximated with the identity matrix,[6] which works well in many cases but can break down under some circumstances. The following approximation works under a broader range of conditions. Letting $\mathbf{C} = \lambda\mathbf{A}^T\mathbf{A}$, we first apply a singular-value decomposition $\mathbf{C} = \mathbf{Q}\mathbf{V}\mathbf{Q}^T$:

$$\mathbf{H}^{-1} = (\mathbf{C} + \mathbf{B})^{-1}, \qquad \text{(A27)}$$

$$= (\mathbf{Q}\mathbf{V}\mathbf{Q}^T + \mathbf{B})^{-1}, \qquad \text{(A28)}$$

$$= \mathbf{Q}(\mathbf{V} + \mathbf{Q}^T\mathbf{B}\mathbf{Q})^{-1}\mathbf{Q}^T. \qquad \text{(A29)}$$

We can write

$$\mathbf{C}(\mathbf{C} + \mathbf{B})^{-1} = \mathbf{I} - \mathbf{B}(\mathbf{C} + \mathbf{B})^{-1}, \qquad \text{(A30)}$$

$$= \mathbf{I} - \mathbf{B}\mathbf{Q}(\mathbf{V} + \mathbf{Q}^T\mathbf{B}\mathbf{Q})^{-1}\mathbf{Q}^T, \qquad \text{(A31)}$$

$$\approx \mathbf{I} - \mathbf{B}\mathbf{Q}\,\text{diag}^{-1}(\mathbf{V} + \mathbf{Q}^T\mathbf{B}\mathbf{Q})\mathbf{Q}^T, \qquad \text{(A32)}$$

where $\text{diag}^{-1}[\,\cdot\,]$ represents a diagonal approximation to the inverse. For an ensemble of N patterns, this operation can be performed in $NO(M^2) + O(M^3)$ time if $\mathbf{Q}^T$ is factored out.

REFERENCES

1. J. G. Daugman, "Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters," J. Opt. Soc. Am. A **2**, 1160–1169 (1985).
2. J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image-analysis and compression," IEEE Trans. Acoust., Speech, Signal Process. **36**, 1169–1179 (1988).
3. J. G. Daugman, "Entropy reduction and decorrelation in visual coding by oriented neural receptive-fields," IEEE Trans. Biomed. Eng. **36**, 107–114 (1989).
4. D. J. Field, "What is the goal of sensory coding," Neural Comput. **6**, 559–601 (1994).
5. T. S. Lee, "Image representation using 2D Gabor wavelets," IEEE Trans. Pattern. Anal. Mach. Intell. **18**, 959–971 (1996).
6. M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," Neural Comput. (to be published).
7. C. Jutten and J. Herault, "Blind separation of sources. 1. An adaptive algorithm based on neuromimetic architecture," Signal Process. **24**, 1–10 (1991).
8. P. Comon, "Independent component analysis, a new concept," Signal Process. **36**, 287–314 (1994).
9. A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," Neural Comput. **7**, 1129–1159 (1995).
10. B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by V1?" Vision Res. **37**, 3311–3325 (1997).
11. E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multiscale transforms," IEEE Trans. Inf. Theory **38**, 587–607 (1992).
12. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," tech. rep. (Stanford University, Stanford, Calif., 1996).
13. R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," IEEE Trans. Inf. Theory **38**, 713–718 (1992).
14. S. G. Mallat and Z. F. Zhang, "Matching pursuits with time-frequency dictionaries," IEEE Trans. Signal Process. **41**, 3397–3415 (1993).
15. S. C. Zhu, Y. N. Wu, and D. Mumford, "Minimax entropy principle and its application to texture modeling," Neural Comput. **9**, 1627–1660 (1997).
16. B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive-field properties by learning a sparse code for natural images," Nature (London) **381**, 607–609 (1996).
17. P. J. B. Hancock, R. J. Baddeley, and L. S. Smith, "The principal components of natural images," Network Comput. Neural Syst. **3**, 61–70 (1992).

18. C. Fyfe and R. Baddeley, "Finding compact and sparse-distributed representations of visual images," Network Comput. Neural Syst. **6**, 333–344 (1995).

19. R. P. N. Rao and D. H. Ballard, "Dynamic-model of visual recognition predicts neural response properties in the visual-cortex," Neural Comput. **9**, 721–763 (1997).

20. R. P. N. Rao and D. H. Ballard, "Development of localized oriented receptive-fields by learning a translation-invariant code for natural images," Network Comput. Neural Syst. **9**, 219–234 (1998).

21. A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," Vision Res. **37**, 3327–3338 (1997).

22. J. H. van Hateren and A. van der Schaaf, "Independent component filters of natural images compared with simple cells in primary visual cortex," Proc. R. Soc. London, Ser. B **265**, 359–366 (1998).

23. C. Zetzsche, E. Barth, and B. Wegmann, "The importance of intrinsically two-dimensional image features in biological vision and picture coding," in *Digital Images and Human Vision*, A. B. Watson, ed. (MIT Press, Cambridge, Mass., 1993), pp. 109–138.

24. D. L. Ruderman, "The statistics of natural images," Network Comput. Neural Syst. **5**, 517–548 (1994).

25. H. B. Barlow, "Possible principles underlying the transformation of sensory messages," in *Sensory Communication*, W. A. Rosenbluth, ed. (MIT Press, Cambridge, Mass., 1961), pp. 217–234.

26. H. B. Barlow, "Unsupervised learning," Neural Comput. **1**, 295–311 (1989).

27. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipies in C: The Art of Scientific Programming*, 2nd ed. (Cambridge U. Press, Cambridge, England, 1992).

28. S. Marcelja, "Mathematical description of the responses of simple cortical cells," J. Opt. Soc. Am. **70**, 1297–1300 (1980).

29. R. L. De Valois, D. G. Albrecht, and L. G. Thorell, "Spatial frequency selectivity of cells in macaque visual cortex," Vision Res. **22**, 545–559 (1982).

30. A. J. Parker and M. J. Hawken, "Two-dimensional spatial structure of receptive fields in monkey striate cortex," J. Opt. Soc. Am. A **5**, 598–605 (1988).

31. J. H. van Hateren and D. L. Ruderman, "Independent component analysis of natural images sequences yield spatiotemporal filters similar to simple cells in primary visual cortex," Proc. R. Soc. London Ser. B **265**, 2315–2320 (1998).

32. I. Daubechies, "Orthonormal bases of compactly supported wavelets," Commun. Pure Appl. Math. **XLI**, 909–996 (1988).

33. R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," Tech. Rep. 414 (University of Pennsylvania, Philadelphia, Penn., May 1997).

34. E. P. Simoncelli and E. H. Adelson, "Noise removal via Bayesian wavelet coring," in *Proceedings of International Conference IEEE on Image Processing, III Lausanne, Switzerland* (Institute of Electrical and Electronics Engineers, New York, 1996), pp. 379–382.

35. S. Chen, "Basis pursuit," Ph.D. dissertation (Stanford University, Stanford, Calif., 1995). Available at http://www-stat.stanford.edu/reports/chen.s

36. R. Everson and L. Sirovich, "Karhunen–Loève procedure for gappy data," J. Opt. Soc. Am. A **12**, 1657–1664 (1995).

37. B. A. Pearlmutter and L. C. Parra, "Maximum likelihood blind source separation: a context-sensitive generalization of ICA," in *Advances in Neural and Information Processing Systems* M. C. Mozer, M. I. Jordan, and T. Petsche, eds. (Morgan Kaufmann, Los Altos, Calif., 1997), Vol. 9, pp. 613–619.

38. H. Attias, "Independent factor analysis," Neural Comput. **11**, 803–851 (1998).

39. B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," tech. rep. (Center for Information Engineering, University of California, San Diego, San Diego, Calif., 1997).

40. R. M. Neal, *Bayesian Learning for Neural Networks* (Springer-Verlag, New York, 1996).

41. J.-P. Nadal and N. Parga, "Nonlinear neurons in the low-noise limit: a factorial code maximizes information transfer," Network **5**, 565–581 (1994).

42. J.-P. Nadal and N. Parga, "Redundancy reduction and independent component analysis: conditions on cumulants and adaptive approaches," Network **5**, 565–581 (1994).

43. J-F. Cardoso, "Infomax and maximum likelihood for blind source separation," IEEE Signal Process. Lett. **4**, 109–111 (1997).

44. G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, eds. (MIT Press, Cambridge, Mass., 1986), Vol. 1, Chap. 7, pp. 282–317.

45. R. Linsker, "Self-organization in a perceptual network," Computer **21**, 105–117 (1988).

46. J. J. Atick, "Could information-theory provide an ecological theory of sensory processing," Network Comput. Neural Syst. **3**, 213–251 (1992).