In: SPIE Proceedings vol. 6492: Human Vision and Electronic Imaging XII, (B.E. Rogowitz, T.N. Pappas, S.J. Daly, Eds.), Jan 28-Feb 1, 2007, San Jose, California

Bilinear Models of Natural Images

Bruno A. Olshausen[a], Charles Cadieu[b], Jack Culpepper[c], and David K. Warland[d]

[a,b]Helen Wills Neuroscience Institute, [a]School of Optometry, and [c]Dept. of Computer Science, UC Berkeley, Berkeley, California 94720, USA

[d]Dept. of Neurobiology, Physiology and Behavior, UC Davis, Davis, California 95616, USA

ABSTRACT

Previous work on unsupervised learning has shown that it is possible to learn Gabor-like feature representations, similar to those employed in the primary visual cortex, from the statistics of natural images. However, such representations are still not readily suited for object recognition or other high-level visual tasks because they can change drastically as the image changes to due object motion, variations in viewpoint, lighting, and other factors. In this paper, we describe how bilinear image models can be used to learn independent representations of the invariances, and their transformations, in natural image sequences. These models provide the foundation for learning higher-order feature representations that could serve as models of higher stages of processing in the cortex, in addition to having practical merit for computer vision tasks.

Keywords: Natural images, sparse coding, bilinear models, invariance, motion

1. INTRODUCTION

The problem of feature extraction—i.e., parsing an image into a set of local descriptors which reflect its structure—is central to both neurophysiological investigations of vision and computer vision. Neuroscientists have traditionally probed the response properties of visual neurons by asking what features of the visual scene they encode. Early studies began with spots of light, and later Hubel and Wiesel discovered orientation selectivity in neurons within the primary visual cortex (area V1) which shifted the emphasis toward shape and other local image properties. In the field of computer vision, the most successful methods for object recognition and tracking depend upon extracting key feature points from an image, which are then matched to an object (represented in terms of the same features).[1] For example, the popular method of SIFT[2] (scale-invariant feature transform) utilizes a bank of multiscale, oriented gradient filters to find keypoints which are candidates for matching to an object.

Despite the initial successes of the feature-based approach in both realms, investigators in neuroscience and computer vision are increasingly faced with the question of how to choose the features to be extracted to begin with. In both realms this process has mostly been guided by good intuitions and guesswork (e.g., Hubel and Wiesel's discovery of orientation selectivity was more accidental than the purposeful test of an hypothesis). Beyond V1, though, there is very little agreement and few concrete ideas about what features are being represented. And although SIFT features appear to be robust to changes in viewpoint and other variations, it begs the question of whether there is a more principled set of features or method for extracting them that would exhibit even greater robustness.

In recent years, a growing community of researchers in both biological and machine vision has been addressing the question of what features to represent by asking, what is the structure of natural images? This problem may be approached within the principled framework of density estimation, or maximum likelihood, in which one attempts to derive, via unsupervised learning, a model that best captures the statistical structure of natural scenes. Using this approach, for example, it has been possible to account for the feature selective properties of neurons in primary visual cortex in terms a sparse coding strategy adapted to the statistics of natural images.[3–5] Our goal in this paper is to build upon this work in order to learn higher-order representations that could serve as models of higher stages of processing in the cortex, in addition to having practical merit for computer vision tasks.

Send correspondence to B.A.O.: baolshausen@berkeley.edu

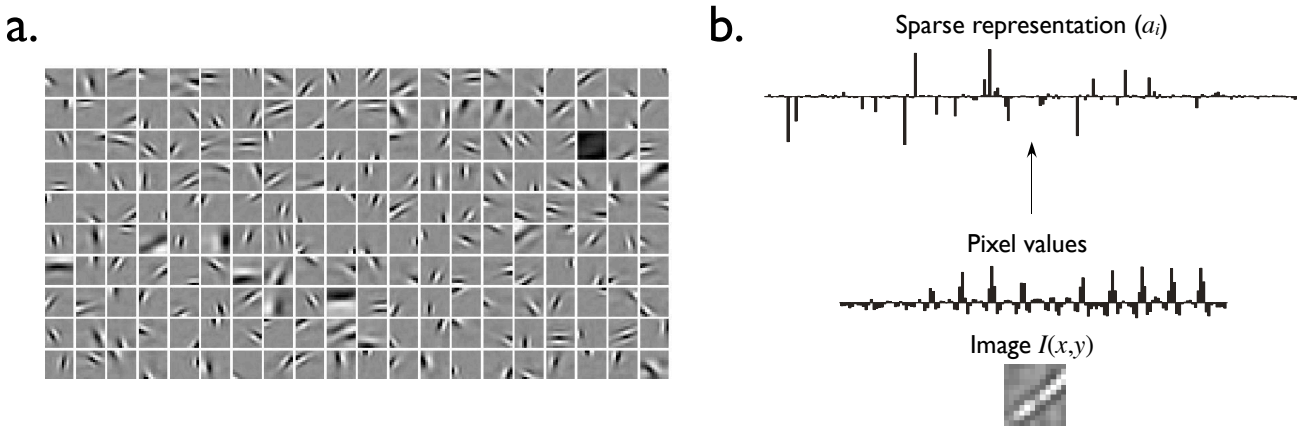

Figure 1. a. Features learned from sparse coding of natural images. Each patch shows a different basis function, ϕ_i, and together the entire set of functions can be used to describe a 12×12 pixel image patch. b. An example image patch (bottom) and its representation in terms of the coefficients, a_i, shown as a bar chart (top).

The previous work on sparse coding models utilized a linear generative model of the form:

$$I(\mathbf{x}) = \sum_i a_i \, \phi_i(\mathbf{x}) + \nu(\mathbf{x}) \tag{1}$$

The image, $I(\mathbf{x})$, is represented in terms of a set of basis functions, or feature descriptors, ϕ_i, and the presence or absence of these features (or degree to which they are present in the image) is indicated by the coefficients, a_i. The term ν is taken to be Gaussian noise and is small relative to the first term. The basis functions, ϕ_i, are adapted to the statistics of natural images so as to minimize the number of non-zero coefficients needed to represent an image (on average), thus forming a *sparse code*. The basis functions that emerge from this optimization are localized, oriented, and bandpass, in a manner that both qualitatively and quantitatively resembles the spatial receptive field properties of V1 simple cells[3] (see figure 1).

Although the features learned from sparse coding reflect intrinsic structural properties of natural images (i.e., edges), the resulting code is still not readily suited for object recognition or other high-level visual tasks because the coefficients, a_i, can change drastically as the image changes to due object motion, variations in viewpoint, lighting, and other factors. What is really desired, then, is to represent *invariant* properties of the environment (i.e., structural properties of surfaces and objects) independently from the factors causing the change (i.e., motion, changes in viewpoint, etc.).

Most attempts at achieving invariant representations utilize successive stages of feature extraction and pooling.[6–8] A major drawback of these models, however, is that they attempt to represent *only* the invariant part without modeling the transformations that created the change in the image. Many aspects of perception, such as scene segmentation and figure-ground assignment, rely on high-level knowledge to resolve ambiguity at lower levels of representation, and if higher cortical areas are to send meaningful expectations to lower areas then they need to know *where* to send it. A high-level representation that achieves invariance by throwing away information about transformations will not be able to do this.

In order to preserve information about both the invariances and transformations that occur in images, a number of theorists have been exploring a more powerful class of models, called *bilinear models*, for image representation. These models are called 'bilinear' because the hidden variables combine via pairwise multiplication, thus they are linear in one set of hidden variables when the others are held constant. In previous work, we and others have described how neural architectures with bilinear forms could be used to remap visual information into an object-based reference frame at higher levels of representation.[9–11] At each stage of processing, information

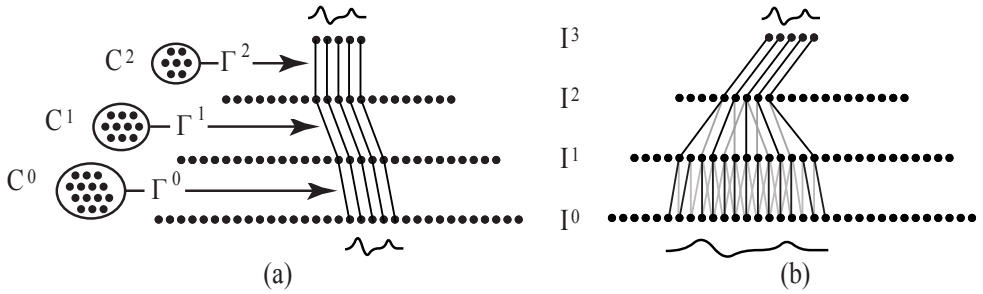

Figure 2. Hierarchical remapping circuit composed of three bilinear stages. The units at each level l, I_i^l, compute their values from a weighted sum of units in the level below, $I_i^l = \sum_j W_{ij}^l \, I_j^{l-1}$. The weights in turn are dynamically modulated by the control neurons, C, via $W_{ij}^l = \sum_k \Gamma_{ijk}^l \, C_k^l$. Shown in a. and b. are two different settings of the weights which allow different instantiations of the same pattern to be remapped into a common pattern at higher levels.

from one layer ($\mathbf{I}^0$) is dynamically gated by a set of control neurons ($\mathbf{c}$) as it goes to the next layer ($\mathbf{I}^1$):

$$I_i^1 = \sum_j \sum_k \Gamma_{ijk} \, c_k \, I_j^0 \tag{2}$$

where the term Γ_{ijk} is a three-way weight that determines how much the k-th control neuron modulates the connection from unit j in the input layer to unit i in the output layer. When these remapping circuits are combined into a hierarchical network composed of multiple stages, it is possible to implement remappings covering a large range of positions and scales with a manageable number of three-way connections,[10] as shown in figure 2.

While it is relatively straightforward to set the three-way weights Γ_{ijk} so as to achieve a certain class of remappings, we aim to *learn* these parameters by observing the class of transformations that objects actually undergo in time-varying natural images. In addition, we would like to incorporate the features learned by sparse coding into the representation at each level, rather that simply remapping pixels.

Recently, Grimes and Rao,[12] building upon earlier work by Tenenbaum & Freeman,[13] proposed a method for learning both the features as well as their remappings via sparse coding. They proposed a bilinear generative model in which the image is described in terms of a superposition of basis functions $B_{ik}(\mathbf{x})$ with two sets of coefficient multipliers, a_i and c_k:

$$I(\mathbf{x}) = \sum_i \sum_k a_i \, c_k \, B_{ik}(\mathbf{x}) + \nu(\mathbf{x}) \tag{3}$$

The a_i's play the same role as before—i.e., they indicate the presence or absence of features in the image. The c_k's now represent the local transformation. During learning, sparseness is imposed upon both the a_i's and c_k's, and the a_i's are clamped as a local scene fragment is translated by a small number of pixels over the image array, $I(\mathbf{x})$, thus forcing the c_k's to represent the transformation. After training on many such image sequences extracted from natural scenes, the basis functions $B_{ik}(\mathbf{x})$ converge to a set of localized, oriented shape features, indexed by i (similar to those above), and they take on different positions, indexed by k. Somewhat similar approaches have been employed by Frey & Jojic[14] and Vasilescu & Terzopoulos,[15] although in both cases strong assumptions are made about the class of transformations to be implemented.

While this model allows for an invariant representation in the a_i's, with respect to local transformations, the training scheme was still somewhat supervised in that a teacher signal was used to specify when the a_i's are to be unclamped and recomputed due to a scene change vs. being clamped during the translation of the image. In addition, the model was trained on image patch sequences obtained by scanning a window over static natural scenes, moving in discrete, integer steps. Not suprisingly then, the B_{ik} simply learn shifted versions of the same set of features. As mentioned earlier, what we really desire is to learn the transformations that actually occur in time-varying natural images, and we would like the model to discover what these are in a fully unsupervised manner.

In this paper, we explore two different types of bilinear models for learning separate representations of the invariances and their transformations in time-varying natural images. The first, based on remapping, has the same bilinear form of equation 3 but makes explicit the manner in which invariant and variant components are being modeled, which also suggests how the model can be made more efficient. This leads to the development of the second model, which is based on interpolation among the basis function coefficients via phase shifting, utilizing fewer multiplicative couplings. This model also points the way toward learning higher-order feature representations which could provide rich descriptions of the invariant and variant components in natural images.

2. BILINEAR MODELS

2.1. Remapping

We first consider the problem of modeling the small transformations that occur from one frame to the next in natural image sequences. Let us assume that each frame of the image sequence may be described as a remapping of the previous frame via

$$I(\mathbf{x}, t+1) = \sum_{\mathbf{x}'} T(\mathbf{x}, \mathbf{x}', t)\, I(\mathbf{x}', t) + \nu(\mathbf{x}, t) \tag{4}$$

where ν is included to account for residual structure not well described by remapping. The map, T, is modeled using a basis function decomposition:

$$T(\mathbf{x}, \mathbf{x}', t) = \sum_{k} c_k(t)\, \psi_k(\mathbf{x}, \mathbf{x}') \tag{5}$$

The problem of modeling transformations in natural image sequences thus amounts to one of finding a good set of basis functions $\{\psi_k\}$ for generating the appropriate remappings $T(\mathbf{x}, \mathbf{x}')$. By "good" we mean intuitively that the ψ_k should be well-matched to describe the transformations that typically occur. That is, only a small number of non-zero c_k should be needed to describe any given transformation. In the same way that we learned a sparse code of image content, then, we seek to learn a sparse representation of the space of *transformations*.

Sparseness is enforced on the coefficients, c_k, by imposing a cost function on their activity. The basis functions $\psi_k(\mathbf{x}, \mathbf{x}')$ are then adapted to an image sequence by forcing each frame transition to be described using the fewest non-zero coefficients c_k. This is accomplished by the following optimization procedure:

$$\{\hat{\psi}_k\} = \arg\min_{\{\psi_k\}} \left\langle \min_{\mathbf{c}} \left[\sum_{\mathbf{x},t} [I(\mathbf{x}, t+1) - \sum_{\mathbf{x}'} \sum_{k} c_k(t)\, \psi_k(\mathbf{x}, \mathbf{x}')\, I(\mathbf{x}', t)]^2 + \sum_{k,t} S(c_k(t)) \right] \right\rangle$$

where $\langle\, \rangle$ denotes 'average over many image sequences.' The sparseness penalty S is of the form $S(x) = \log(1+x^2)$. The minimization with respect ot the basis functions ψ_k and coefficients c_k may be accomplished via gradient descent methods similar to those described previously.[3]

Now, if we represent the image at each time, t, using the basis function decomposition of the sparse coding model:

$$I(\mathbf{x}, t) = \sum_{i} a_i(t)\, \phi_i(\mathbf{x})$$

we obtain from equations 4 and 5:

$$\begin{aligned} I(\mathbf{x}, t+1) &= \sum_{\mathbf{x}'} \sum_{k} c_k(t)\, \psi_k(\mathbf{x}, \mathbf{x}') \sum_{i} a_i(t)\, \phi_i(\mathbf{x}') && (6) \\ &= \sum_{i} \sum_{k} a_i\, c_k\, B_{ik}(\mathbf{x}) && (7) \end{aligned}$$

where $B_{ik}(\mathbf{x}) = \sum_{\mathbf{x}'} \psi_k(\mathbf{x}, \mathbf{x}')\, \phi_i(\mathbf{x}')$. Thus, it can be seen that the remapping model is equivalent to the bilinear model of Grimes & Rao, described above (eq. 3), when the image being remapped is represented by the coefficients of a basis function expansion. The difference here is that we are learning the remapping components

ψ_k explicitly, whereas in Grimes & Rao's model they are learned implicitly, along with the invariant components ϕ_i, via the B_{ik}. Also, in lieu of using a teacher signal during learning we are imposing a form of "perceptual stability" on the a_i's by essentially clamping their values for each pair of adjacent frames in the image sequence, similar in spirit to 'slow feature analysis' methods.[16–19]

This analysis helps to clarify *how* the bilinear model factors an image into 'what' and 'where' components—i.e., in terms of transformation components $\psi_k(\mathbf{x}, \mathbf{x}')$ and shape components $\phi_i(\mathbf{x})$. At the same time, it shows that it may actually be disadvantageous to collapse the ϕ_i and ψ_k bases into a single function $B_{ik}(\mathbf{x})$, because in doing so you lose the ability to learn structure independently from the transformations. By keeping them separate, you can learn new shape features without having to relearn all the transformations for those features. While this may not be a concern at lower levels of representation since the shape features are likely to be rather generic, it could prove to be important at higher levels.

A potential problem with this approach is that the dimensionality of the transformation bases, ψ_k, can be extremely large. Even for an 8×8 pixel image patch, each ψ_k is of dimension 64^2, and if we wish to learn many such bases the number of parameters to learn, and thus local minima, could prove intractable. One way to reduce the complexity is to exploit the compact representation provided by the sparse coding model. That is, we can model the transformations in terms of the basis function coefficients, rather than directly on the image pixels, which we turn to next.

2.2. Phase-shifting

Looking back at figure 1a, one of the striking properties of the features learned by sparse coding is that they resemble Gabor functions—i.e., Gaussian modulated sines and cosines. An interesting property of Gabor functions, for our purposes, is that they allow for shift in the image domain to be modeled simply via interpolation among the coefficients, as shown in figure 3a. One way of understanding the solution discovered by sparse coding, then, is that the model is attempting to describe edges or other features that occur over a continuum of different positions, and since it is being forced to do so using a linear generative model it has essentially learned a good set of interpolating functions. Our task then is to make explicit the *transitions* among coefficient values that occur as the result of translation (or other transformations) in the image, rather than treating the coefficients as independent variables.

A natural way to model these transitions among the coefficients is via *phase-shifting*. Consider a complex basis function with real and imaginary parts as follows:

$$\phi_i(\mathbf{x}) = \phi_i^R(\mathbf{x}) + j\phi_i^I(\mathbf{x})$$

Multiplying $\phi_i(\mathbf{x})$ by a complex coefficient z_i and taking the real part of the product essentially interpolates between the real and imaginary parts according to the phase of z_i:

$$\Re\{z_i^* \, \phi_i(\mathbf{x})\} = \sigma_i \left[\cos \alpha_i \; \phi_i^R(\mathbf{x}) + \sin \alpha_i \; \phi_i^I(\mathbf{x})\right]$$

where $\Re\{\,\}$ denotes 'real part,' and σ_i and α_i are the amplitude and phase of z_i:

$$z_i = \sigma_i \, e^{j \, \alpha_i}$$

Thus, we have constructed a 'shiftable' feature descriptor in which the amplitude of the coefficient, σ_i, indicates its presence or absence and is invariant with respect to some local transformation, and the phase of the coefficient, α_i, indicates the transformation. Note that in general a phase shift in z_i need not correspond only to translation in the image domain. If the real and imaginary basis functions are Gaussian derivatives with orthogonal orientations, for example, then shifting the phase of z_i will rotate the combined function.[20]

Now we can construct a complete representation of the image using a full set of such complex basis functions as follows:

$$\begin{aligned} I(\mathbf{x}, t) &= \sum_i \Re\{z_i^* \, \phi_i(\mathbf{x})\} \\ &= \sum_i \sigma_i(t) \left[\cos \alpha_i(t) \; \phi_i^R(\mathbf{x}) + \sin \alpha_i(t) \; \phi_i^I(\mathbf{x})\right] \; + \; \nu(\mathbf{x}, t) \end{aligned} \tag{8}$$

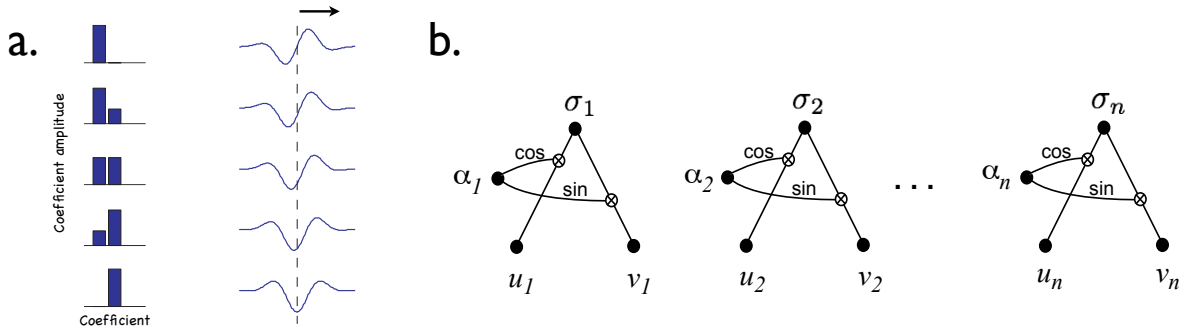

Figure 3. a. Phase-shifting via interpolation in the coefficients. At left is shown a bar chart of the coefficients corresponding to two different basis functions (in this case, Gabor functions in quadrature phase). At right is shown the combined function that is formed by adding the basis functions weighted by the amounts shown at left. b. Complex basis function model. The real and imaginary coefficients for each basis function, u_i, v_i, are controlled by amplitude and phase variables, σ_i, α_i. The σ_i's are invariant to local transformations, and the α_i's represent those transformations.

This model is illustrated graphically in figure 3b. Each complex basis function has two parameters that describe how it used. The amplitude, σ, describes the locally invariant part, while the phase, α, represents the local transformation.

Note that we can also recast the model in the same form as the original sparse coding model (eq. 1) as follows:

$$I(\mathbf{x},t) = \sum_i u_i(t)\,\phi_i^R(\mathbf{x}) + v_i(t)\phi_i^I(\mathbf{x}) \;+\; \nu(\mathbf{x},t)$$

where

$$\begin{aligned} u_i(t) &= \sigma_i(t)\,\cos\alpha_i(t) \\ v_i(t) &= \sigma_i(t)\,\sin\alpha_i(t) \end{aligned}$$

This allows us to see that the model is essentially bilinear in the amplitudes σ_i and the cosine or sine of the phase α_i. However, the model here is much more constrained and there are no longer any three-way weights, thus reducing the number of parameters that need to be learned.

The model is adapted to time-varying natural image sequences by imposing both sparseness and *slowness* on the amplitudes, $\sigma_i(t)$, in order to encourage the model to learn invariant features in the images. This is accomplished by the following optimization procedure:

$$\{\hat{\phi}_i\} = \arg\min_{\{\phi_i\}} \left\langle \min_{\sigma,\alpha} \left[\sum_{\mathbf{x},t} \left[I(\mathbf{x},t) - \sum_i \sigma_i(t)\,[\cos\,\alpha_i(t)\,\phi_i^R(\mathbf{x}) + \sin\,\alpha_i(t)\,\phi_i^I(\mathbf{x})] \right]^2 + \sum_{i,t} S(\sigma_i(t)) + |\dot{\sigma}_i(t)|^2 \right] \right\rangle$$

where the sparseness penalty S is the same as before. Note that there is no penalty on the phases variables, which allows them to spin as needed in order to best match structure in the image.

3. RESULTS

3.1. Remapping

Figure 4 shows a sample of some of the bases $\psi_k(\mathbf{x},\mathbf{x}')$ learned by training the remapping model on image sequences extracted from a natural movie (in this case, a nature documentary). Interestingly, one of them learns the identity mapping, which simply maps each pixel into itself, while the others learn to compute directional

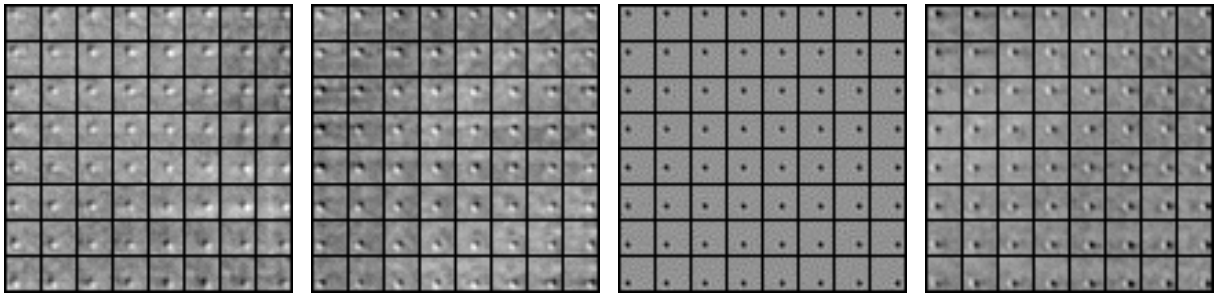

Figure 4. Four of the remapping bases, $\Psi_k(\mathbf{x}, \mathbf{x}')$, learned from the transformations contained in natural image sequences. Each basis function shows how a pixel in one frame is mapped into the next frame—i.e., each patch within a basis function corresponds to a pixel within the originating frame (ordered according to its position in the frame), and the values within the patch denote how it is weighted into the next frame. For example, the identity function (third from left) simply maps each pixel into itself.

gradients with different orientations. This solution to can be understood as a first-order approximation to the Lie group operator for translation.[21] That is, one can approximate a translated image as

$$I(\mathbf{x} + \Delta\mathbf{x}) \cong I(\mathbf{x}) + \Delta\mathbf{x} \cdot \nabla_{\mathbf{x}} I(\mathbf{x})$$

Thus, the model has essentially learned the basis functions needed to translate an image patch by adding a copy of the image patch to its derivative along a certain direction ($\Delta\mathbf{x}$).

3.2. Phase-shifting

Figure 5a shows two examples of complex basis functions learned as a result of adapting the model to natural image sequences. The basis functions take on a similar form as before (localized, oriented, bandpass), except now they come in pairs that appear roughly in quadrature. When added together weighted by the cosine and sine of the phase, α_i, they combine to form a set of shiftable basis functions. One can see the range of variation expressed by each function by holding the amplitude of its coefficient fixed and spinning the phase from 0 to 2π (see `http://redwood.berkeley.edu/bruno/complexbfs`). As can be seen from the joint histograms of real and imaginary coefficients in figure 5b, the phases have a uniform distribution over the interval $0 : 2\pi$, indicating that each complex basis function is being utilized in all of its shifts.

Figure 6 shows the result of coding a time-varying natural image sequence using the complex basis function model. Note that the local invariances are now made explicit via the complex amplitudes, $\sigma_i(t)$, which change relatively slowly over time, typically sustaining their value over 5-10 frames. By comparison, the real and imaginary coefficients, $u_i(t)$, $v_i(t)$, tend to undulate with each frame. In addition, motion is explicitly represented as a linear ramp in phase during the periods when the corresponding amplitude is significant. Importantly, the joint distribution of the amplitudes and phases of neighboring complex basis functions exhibits strong dependencies (fig. 5c), suggesting that another layer of sparse coding could learn higher-order features based on this structure.

4. DISCUSSION

We have shown in this work how bilinear models may be used to learn independent representations of 'what' (invariances) and 'where' (transformations) components of time-varying natural images. In constrast to previous models that focus either on forming invariant representations of objects[6–8] or representations of motion and optic flow,[22–24] our approach combines both of these into the same model. Moreover, computing the transformations is necessary for extracting the invariances, and vice-versa. Together, both the invariances and their transformations provide a complete description of the content of time-varying images.

Interestingly, the complex basis function model bears a strong resemblance to models of 'complex cells' in primary visual cortex. However the models are actually very different in terms of how they achieve invariance. In

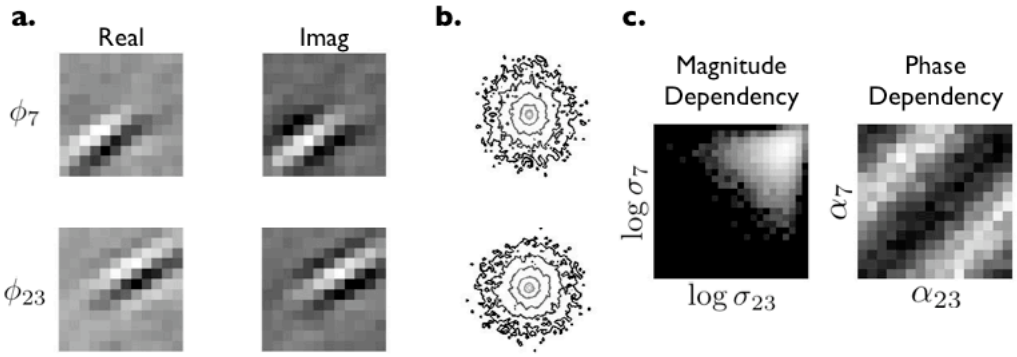

Figure 5. a. Each row shows a different complex basis function pair learned in the shiftable basis function model. The real part of each function is shown at left, and the corresponding imaginary part is shown at right. b. Joint histogram of the real and imaginary coefficients, u_i, v_i for each complex pair. c. Joint histograms of log-amplitude (left) and phase (right) for the pair of complex basis functions shown at left.

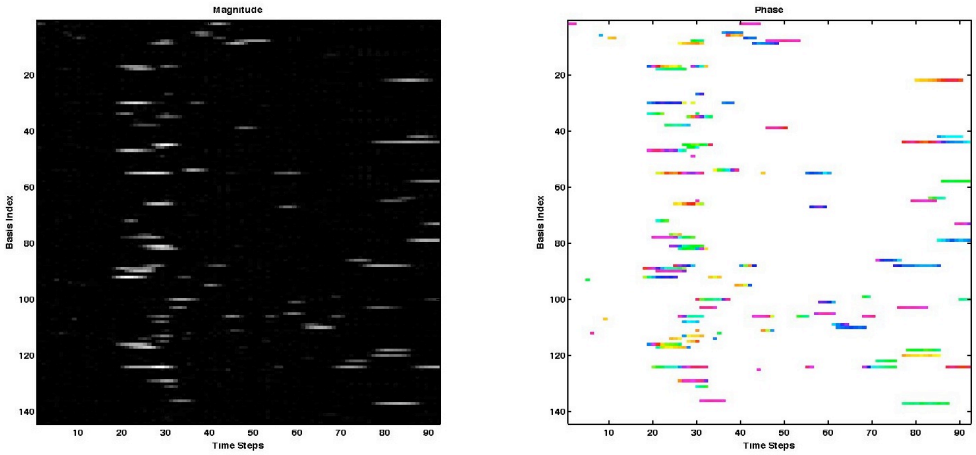

Figure 6. Coding of a 100-frame image sequence in terms of magnitude (left) and phase (right). Vertical axis is coefficient index and horizontal axis is time. Phase is displayed for only those points in time when the corresponding amplitude is significant.

the standard "energy model" of complex cells,[22] invariance is achieved by simply pooling over a set of simple-cell subunits. By contrast, in the complex basis function model, invariance is achieved by dynamically changing the linkages with the simple-cell subunits (real and imaginary components, u_i, v_i) via the phase variables, α_i (fig. 3b). The phase variables contain information about relative spatial relationships, which is important to preserve and represent explicitly since it contains useful, structural information about the content of images. For example, the state of the art method for iris recognition encodes the phase (and discards the amplitudes) of complex Gabor wavelet filters to form the feature vector used for pattern matching.[25]

The complex basis function model is also similar to the linear subspace model of Hyvarinen & Hoyer[26] when the subspace size is two. Again, however, the major difference here is that we are explicitly representing and exploiting the phase variables in order to provide a sparse, locally invariant representation of image content.

The manner in which both the remapping model and complex basis function model are trained has much in common (and in fact was inspired by) slow feature analysis methods.[16–19] The idea behind these models is to learn about the invariances in natural image sequences by imposing "perceptual stability" on the representation, since objects and other invariant properties of the visual environment tend to change slowly over time (by definition), in contrast to the pixels which typically change rapidly. However, a major drawback of most of these models is that they attempt to represent *only* the invariant part, without modeling the transformations that created the changes in the image. As emphasized earlier, the philosophy driving our approach is that the information about the transformations is equally important and should be explicitly represented along with the invariant part so that higher levels can generate specific predictions about content at lower levels of representation.

As it currently stands, the complex basis function model is somewhat contrived in that it relies upon grouping pairs of basis functions together in order to model the transitions among their coefficients. However, this particular model is intended merely as a starting point, to demonstrate what can be gained in a generative model that utilizes multiplicative interactions among hidden variables. Generalizing this model beyond simple pairs, so as to learn the groupings from the statistics of natural images, is the subject of current research. We believe this approach holds great promise for learning higher-order feature representations within a hierarchical architecture that mirrors the 'what' and 'where' streams of visual cortex.

ACKNOWLEDGMENTS

We thank Kilian Koepsell for useful discussions during the preparation of this manuscript. Work supported by NGA grant MCA 015894-UCB and NSF grant IIS-06-25223 to B.A.O.

REFERENCES

1. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, p. 1615, 2005.
2. D. G. Lowe, "Distinctive image features from scale-invariant keypoint," *International Journal of Computer Vision* **60**(2), pp. 91–110, 2004.
3. B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature* **381**(381), pp. 607–609, 1996.
4. A. J. Bell and T. J. Sejnowski, "The independent components of natural images are edge filters," *Vision Research* **37**, pp. 3327–3338, 1997.
5. 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.Lond. B* **265**, pp. 359–366, 1998.
6. K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics* **36**, pp. 193–202, 1980.
7. G. Wallis and E. T. Rolls, "Invariant face and object recognition in the visual system," *Prog Neurobiol.* **51**(2), pp. 167–94, 1997.
8. M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nat Neurosci.* **2**(11), pp. 1019–25, 1999.
9. G. E. Hinton, "A parallel computation that assigns canonical object-based frames of reference," in *International Joint Conference on Artificial Intelligence*, pp. 683–685, 1981.

10. B. A. Olshausen, C. H. Anderson, and D. C. Van Essen, "A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information," *The Journal of Neuroscience* **13**(11), pp. 4700–4719, 1993.
11. D. W. Arathorn, *Map-Seeking Circuits in Visual Cognition: A Computational Mechanism for Biological and Machine Vision*, Stanford University Press, 2002.
12. D. Grimes and R. Rao, "Bilinear sparse coding for invariant vision," *Neural Computation* **17**(1), pp. 47–73, 2005.
13. W. T. Freeman and J. B. Tenenbaum, "Learning bilinear models for two-factor problems in vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 1997.
14. B. J. Frey and N. Jojic, "Estimating mixture models of images and inferring spatial transformations using the em algorithm," in *CVPR proceedings*, pp. 1416–1422, 1999.
15. M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces.," in *Proc. European Conf. on Computer Vision, ECCV*, 2002.
16. P. Foldiak, "Learning invariance from transformation sequences," *Neural Computation* **3**(2), pp. 194–200, 1991.
17. L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Computation* **14**(4), pp. 715–770, 2002.
18. W. Einhauser, C. Kayser, and K. P. Kording, "Learning multiple feature representations from natural image sequences," in *International Conference on Artificial Neural Networks (ICANN)*, pp. 21–26, August 2002.
19. A. Hyvarinen, J. Hurri, and J. Vayrynen, "Bubbles: A unifying framework for low-level statistical properties of natural image sequences," *Journal of the Optical Society of America* **20**(7), pp. 1237–1252, 2003.
20. D. K. Hammond and E. P. Simoncelli, "Nonlinear image representation via local multiscale orientation," *Courant Institute Tech Report* **TR2005-875**, 2005.
21. R. P. N. Rao and D. L. Ruderman, "Learning lie groups for invariant visual perception," in *Advances in Neural Information Processing Systems (NIPS 1999)*, 1999.
22. E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America, A* **2**(2), pp. 284–299, 1985.
23. E. P. Simoncelli and D. J. Heeger, "A model of neuronal responses in visual area mt," *Vision Research* **38**(5), pp. 743–761, 1998.
24. J. A. Perrone, "A visual motion sensor based on the properties of v1 and mt neurons," *Vision Res* **44**, pp. 1733–55, 2004.
25. J. Daugman, "How iris recognition works," *IEEE transactions on circuits and systems for video technology* **14**(1), pp. 21–30, 2004.
26. A. Hyvarinen and P. O. Hoyer, "Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation* **12**(7), pp. 1705–1720, 2000.