Sparse coding in practice

Chakra Chennubhotla & Allan Jepson
Department of Computer Science
University of Toronto, 6 King's College Road
Toronto, ON M5S 3H5, Canada
Email: {chakra,jepson}@cs.toronto.edu

Abstract

The goal in sparse coding is to seek a linear basis representation where each image is represented by a small number of active coefficients. The learning algorithm involves adapting a basis vector set while imposing a *low-entropy*, or sparse, prior on the output coefficients. Sparse coding applied on natural images has been shown to extract wavelet-like structure [9, 4]. However, our experience in using sparse coding for extracting multi-scale structure in object-specific ensembles, such as face images or images of a gesturing hand, has been negative. In this paper we highlight three points about the reliability of sparse coding for extracting the desired structure: (1) using an *overcomplete* representation (2) projecting data into a low-dimensional subspace before attempting to resolve the sparse structure and (3) applying sparsity constraint on the basis elements, as opposed to the output coefficients.

1 Introduction

What is the structure one can expect to find in an ensemble of face images? Assume the dataset has been collected keeping the viewpoint and illumination conditions roughly identical and with the subjects all looking at the camera from a fixed distance. Under these conditions the ensemble captures variations in the appearance of the whole (overall face) and the parts (forehead, nose, mouth etc) of the object class (faces). We expect these variations to result in groups of pixels that are potentially disconnected yet exhibit statistical coherencies over multiple scales. In this paper, we investigate whether *sparse coding* can extract such object-specific multi-scale structure.

The goal in sparse coding is to seek a linear basis representation where each image is represented by a very *few* active coefficients. The learning algorithm involves adapting a basis vector set while imposing a *low-entropy*, or sparse, prior on the output coefficients. The intuition for sparse priors comes from analyzing the statistics of natural scenes. The evidence for sparse structure can be seen by filtering natural images with wavelets. The histograms of the resulting coefficients are sharply peaked at zero and have extended tails. When compared to a Gaussian distribution of the same variance, the wavelet coefficient histograms have a lower entropy. The hope in using a sparse prior for the coefficients, while adapting the basis vector set, is that the basis functions will acquire the shape and form of the underlying structure.

Sparse coding applied on natural images has been shown to extract wavelet-like structure [9, 4]. The basis vectors appear self-similar, localized, oriented and bandpass. However, our experience with sparse coding, when it was used to extract multi-scale structure in object-specific ensembles such as face images or images of a gesturing hand, has been negative. Using simple yet interesting models of structure often found in real images, we highlight three points about the reliability of sparse coding for extracting the desired structure:

- **Overcomplete Representation**: A representation is said to be overcomplete if the number of basis vectors, and hence the output coefficients, exceeds the input dimensionality. We demonstrate that there are simple ensembles for which the adapted basis vectors capture the intrinsic structure *only* when the representation is overcomplete.

- **Subspace Projection**: Similarly we exhibit image ensembles where large number of sparse components exist in a low-dimensional subspace. We find that a sparse coding algorithm fails to simultaneously compute the linear subspace and resolve the structure within, unless the data is first projected into the subspace.

- **Sparsity constraint on the basis matrix**: In a typical sparse coding formulation, a low-entropy (or sparse) prior is applied on the output coefficients so as to represent each image with a small number of active coefficients. In this paper we show that to reliably extract object-specific multi-scale structure, it is useful to have the sparsity constraint applied on the basis elements instead.

The rest of the paper is organized as follows. In section 2 we provide a brief introduction to sparse coding and its relation to the independent component analysis (ICA). In sections 3 and 4 we discuss the issues of overcomplete representation and subspace projection. In section 5 we analyze the results obtained from applying sparse coding to an object-specific ensemble. In section 6 we argue for the application of sparsity constraint on the elements of the basis matrix. This lead us to a new framework, Sparse PCA, for extracting multi-scale structure in object-specific ensembles[1]. We provide a brief summary of this algorithm and show results in section 7.

2 Sparse coding

In the sparse coding framework, each image is represented by a linear superposition of basis vectors plus noise:

$$\vec{t} = B\vec{c} + \vec{\epsilon}, \tag{1}$$

where $\vec{t}$ is n-element input image, B is a $n \times m$ matrix whose columns are basis vectors, $\vec{c}$ is a m-element coefficient vector, and $\vec{\epsilon}$ represents n-element noise vector sampled from a Normal distribution. When the number of basis vectors m is greater than the dimensionality of the inputs n (i.e. $m > n$) then the basis matrix B is considered to be overcomplete. For a given image ensemble $\{t_i\}_{i=1..k}$ a basis matrix yielding a sparse representation is learned by *minimizing* the following cost function [9, 4]:

$$E(\lambda) = \sum_i E_i(\lambda) = \sum_i \left[||\vec{t}_i - B\vec{c}_i||_2^2 + \lambda \sum_j \Omega(c_{ij}) \right]. \tag{2}$$

The cost function $E_i(\lambda)$ for each image $\vec{t}_i$ contains two terms: a reconstruction error term given by $||\vec{t}_i - B\vec{c}_i||_2^2$ and a regularization term given by $\lambda \sum_j \Omega(c_{ij})$. The reconstruction error term forces B to span the input space. The regularization term is set up to favor solutions in which only *few* elements c_{ij} of the coefficient vector c_i respond to each image t_i. This is achieved by imposing a low-entropy (or sparse) prior on the coefficient vector $\vec{c}$. The prior on the coefficients is assumed to be factorial, that is, $p(\vec{c}) = \prod_j p(c_j)$ and $\Omega(c_j) = \log(p(c_j))$. One example of a low-entropy prior is the double-sided Laplacian: $p(x) \propto \exp(-\theta|x|)$. The regularization parameter λ controls the amount of contribution from the prior to the cost function. It is useful to note that λ is really a combination of parameters that are needed to specify both the noise and coefficient prior distributions.

The overall cost function $E(\lambda)$ is minimized in two stages. In the inner loop for each image $\vec{t}_i$, holding the basis matrix B fixed, $E_i(\lambda)$ is minimized with respect to the coefficient vector $\vec{c}_i$. In the outer loop, after coding several training images, the accumulated $E(\lambda)$ is minimized with respect to B. The regularization parameter λ need not be held fixed. Our implementation follows the improved version of the learning algorithm presented in [7]. Sparse coding on natural images [9, 4] generates wavelet-like basis vectors: self-similar, localized, oriented and bandpass. This result was seen to be invariant to the exact functional form of the low-entropy prior.

[1] A longer version of the Sparse PCA algorithm appears in ICCV'01 [3]. We are including a brief summary here for the sake of completeness.

2.1 Independent Component Analysis

Sparse coding is closely related to Independent Component Analysis (ICA) ([2]). In ICA the idea is to learn a filter matrix D that can generate statistically independent coefficients. In other words a coefficient vector $\vec{s}$ for each image $\vec{t}$ is obtained by: $\vec{s} = D\vec{t}$. The image ensemble is assumed to be noise-free. If the basis matrix is invertible, then it is related to the filter matrix D by: $D = B^{-1}$. ICA, unlike sparse coding, emphasizes independence over sparsity in the output coefficients. However, *a transformation that minimizes the entropy of the individual outputs also promotes their statistical independence*. Hence, we abuse the terminology slightly in this paper by denoting the structure that can be captured by a low-entropy prior as being, not just sparse, but also *independent*. We work with image ensembles where the sparse structure has a clear interpretation.

In the next several sections, we explore the process of sparse coding. We highlight three structural conditions necessary for sparse coding to succeed. Before initiating the learning procedure, several decisions have to be made: choosing a functional form for the low-entropy prior, fixing an appropriate value for λ or deciding to learn it from the data and guessing a lower bound on number of basis vectors needed to extract independent components in the images. In each of the datasets used in this paper, we make particular choices for what the independent components are. Our synthetic ensembles are idealized yet interesting models of structure often found in the images of the real world.

3 Overcomplete Representation

Imagine placing a 2-pixel wide window at random locations in a natural image and recording the intensities. We assume that natural images are typically made up of spatially coherent objects (or blobs). The window, because it is placed at random, will either fall on the same side of object or straddle an edge that separates objects. Hence, the ensemble will have two kinds of images: (a) images with correlated pixel intensities, caused by the window falling on a spatially coherent object (b) images with a potential edge between the two pixels, caused by independently varying pixel intensities sampled from two different objects.

As shown in Fig. 1, each 2-pixel image can be seen as a point in a 2-dimensional space. We can model one such image cloud by the following expression:

$$\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \cdot r \\ c_2 \cdot r \\ c_3 \cdot (1-r) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix}, \tag{3}$$

where $\vec{t} = [t_1, t_2]^T$ is a 2-pixel image generated by linearly combining the basis vectors in $B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ using the coefficient vector $\vec{c} = [c_1, c_2, c_3]^T$ sampled from a factorial low-entropy prior and mixed with additive noise vector $\vec{\epsilon} = [\epsilon_1, \epsilon_2]^T$ obtained from a Normal distribution. The low-entropy prior on the coefficient vector $\vec{c}$ is appropriate to use, particularly if the ensemble were to be generated by sampling a wavelet-filtered natural image. Here we are modeling images to be real-valued functions of continuous variables.

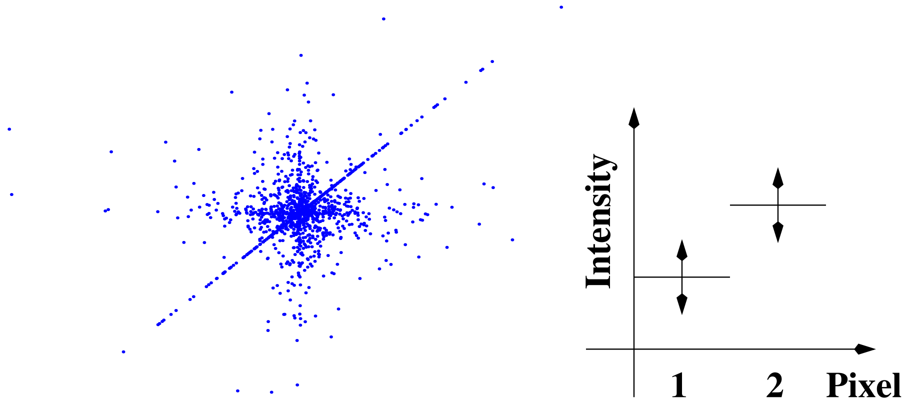


Figure 1: (Left) A 2-pixel image cloud is generated by sampling intensities from a low-entropy (or sparse) distribution (Right). The horizontal and vertical arms of the distribution have images each with an *edge* between the two pixels, caused by independent sampling of the pixel intensities. In comparison, the images lying on the diagonal were forced to have correlated pixel values. Thus, the 2-pixel distribution comprises of images with at most *one* edge but *three* independent components.

The image cloud shown in Fig. 1 exhibits 3 arms: one along the diagonal and two aligned with the horizontal/vertical axes. The arms are caused by the binary random variable r. Following Eq. 3, when r is 0 intensities at the two pixels are correlated and the resulting images are distributed along the diagonal of the point cloud. When $r = 1$, it signifies the presence of an *edge* in the image because the pixel intensities now vary independently, thus giving rise to the axis-aligned arms of the image cloud. This observation about the 2-pixel ensemble is likely to hold regardless of the shape of the window. In particular, the window can be designed to measure intensities in either one of the directions: horizontal, vertical or diagonal.

Each independent component thus corresponds to a unique direction in the image cloud. To capture structure in the 2-pixel ensemble, the sparse coding procedure requires 3 basis vectors. In Fig. 2 we show the structure extracted by sparse coding when initiated with 2 and 3 basis vectors. The $\times$'s in each sub-plot indicate the trajectories of the basis vectors over iterations, the final values being shown with a solid line. The estimated independent components that result from initializing 2 basis vectors appear *edge*-like. Clearly, the edge directions carry information on more then one independent component. For the 2 component case the basis vectors converge to the directions: $[1, 1]^T$ and $[1, -1]^T$. In this case, when sparse coding cannot achieve independence, it seemed to have decorrelated the input ensemble with orthogonal basis vectors. The independent components that result from initiating 3 basis vectors converge to the structure intrinsic to the ensemble.

We can extend the above argument for images with N pixels having at most 1 edge at a random pixel location. For simplicity, assume the signal before and after the the edge each requires C basis vectors and hence C coefficients for a complete description. As there are $N-1$ possible locations where an edge can appear, the total number of independent components F in the N-pixel ensemble is given by: $F = (N-1)(2C) + C = C(2N-1)$. Note this count includes the one component needed to describe the *absence* of an edge in a given image. For example, in the 2-pixel ensemble that we just considered, the signal before and after the edge

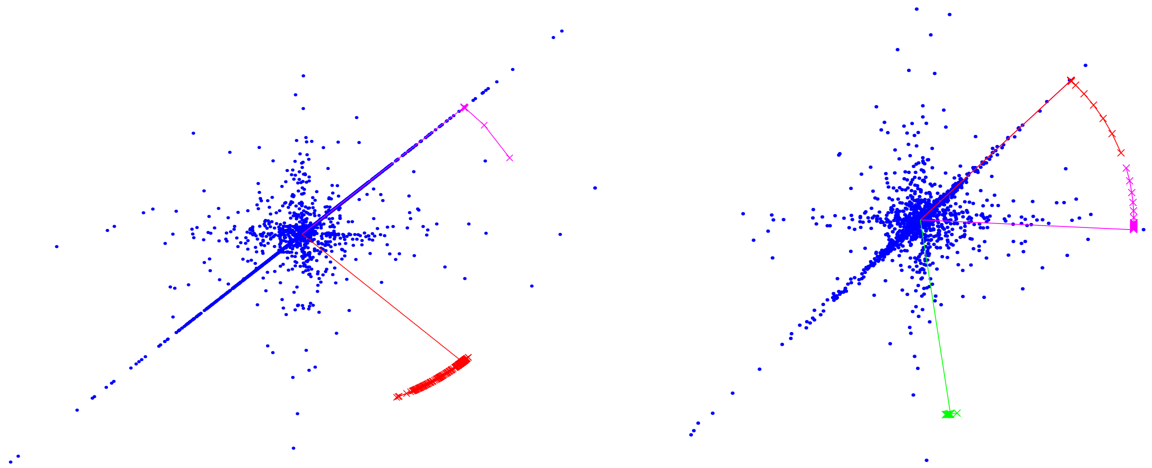


Figure 2: *Are edge-directions the independent components of images?* Independent components learned for the ensemble of 2-pixel images (Fig. 1) when initialized with 2 (Left) and 3 (Right) basis vectors. The crosses indicate the basis vector updates of the sparse coding algorithm, while the solid lines indicate the converged directions. Note *edge-like* directions that result from using 2 basis vectors summarize information on more than 1 independent component.

each needed 1 basis vector and hence 1 coefficient for a complete description. Hence the total number of independent components F comes to $(2 - 1)(2 \times 1) + 1 = 3$.

We generated a synthetic image ensemble using an expression similar to Eq. 3 with $N = 5$ and $C = 1$. The generative process is very simple. First, decide if the new image should should have an edge. If so, pick a pixel position and draw two coefficient samples from a low-entropy distribution. Depending on the choice made on having an edge, use the coefficients to describe the signal before and after the edge. The total number of independent components in this ensemble will be 9. In Figures 3 and 4 we show the structure extracted by sparse coding when initiated with 9 and 5 basis vectors respectively. It is clear that the independent components learned while using the lower dimensional basis appear edge-like. The effect was duplicated for other basis dimensions so long as they were smaller than 9. In the next Section, we discuss the case where the number of basis vectors initialized is more than required.

To summarize, it is possible to have independent components in images that are *not* edges and where the total number of independent components can far exceed the input dimensionality [2]. Each independent component introduces a novel direction in the image cloud created by the ensemble. If the sparse coding algorithm is initiated with fewer components than the ensemble requires, it converges on a state in which some of the learned directions are edge-like.

4 Subspace projection before sparse coding

Here we show that, even with the correct number of basis vectors, there is no guarantee that sparse coding will capture the structure intrinsic to the ensemble. To understand why, consider images generated by three interacting objects, named the background, middle and foreground objects. In our simple ensemble, these objects appear at fixed locations in the image, but they

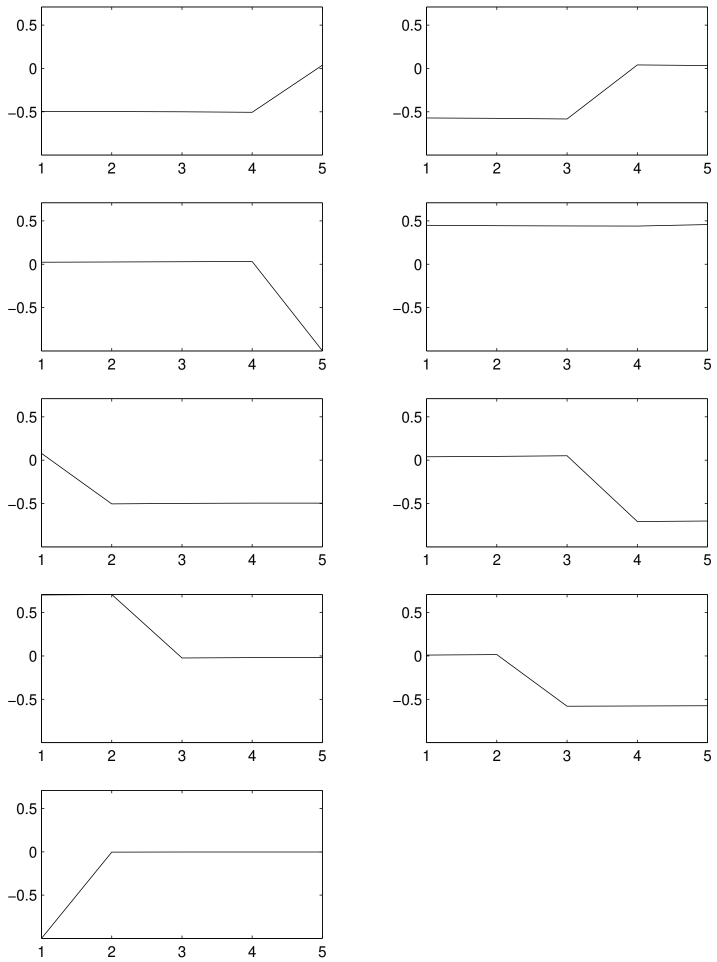


Figure 3: Structure found by sparse coding for a 5-pixel image ensemble with at most 1 edge in each image. Each sub-figure corresponds to an independent component or equivalently a basis vector. The learning algorithm was initiated with the *correct* number of basis vectors, that is 9. The *non-zero* portion of each vector highlights pixels that vary together, but independently of the other pixels, over the ensemble. For example, the basis vector appearing in (row 4,column 2) highlights pixels 3, 4 and 5 as varying independently. When combined with the vector shown in (row 4,column 1), it is possible to describe all the images in the ensemble that have an edge between pixels 2 and 3. Note the basis vector in (row 2, column 2) describes images that do *not* have an edge.

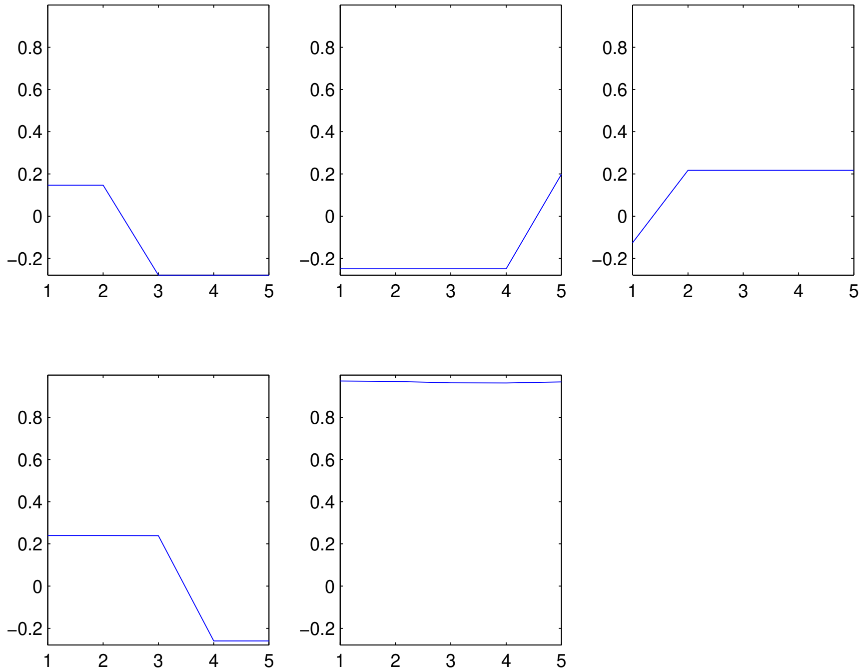


Figure 4: Structure found by sparse coding for a 5-pixel image ensemble with at most 1 edge in each image. The learning algorithm has been initiated with 5 basis vectors to capture information on 9 independent components. Each sub-figure corresponds to a basis vector. Most of the basis vectors appear *edge*-like because they seem to carry information on more than 1 independently varying component. It is illustrative to compare these results with ones shown in Fig. 3.

potentially occlude one another. The shape of each object is the same but their sizes differ. The imaging model and the prototypical images are shown in Fig. 5. The background object appears in all images but the objects in the middle and the foreground appear with certain probability. Also, when the objects in the middle and the foreground appear together, the foreground object always occludes the one in the middle. As shown in Fig. 5 the intensity of each object is sampled from a low-entropy distribution.

As defined earlier, independent components are groups of pixels that each vary in a statistically coherent manner. For example the occlusion of the object in the background by the object in the middle creates 2 independently varying pixel groups. A simple count of all the uniquely varying pixel groups, over all possible object interactions, yields a total of 7 independent components. The independent components can be seen as waveforms in Fig. 9. There are 7 components as opposed to 8 because the foreground object is never occluded by the object in the middle. It is beyond the scope of our linear model to collapse information contained in the 7 independent components as really coming from 3 interacting objects.

For the occluding object ensemble, the number of independent components (7) is clearly less than the dimensionality of each input image (20). In theory the sparse coding algorithm should learn the independent components in a low-dimensional subspace while setting the extra basis vectors to zero. The resulting basis vectors are shown in Fig. 6. Only the 4 *non-zero* basis

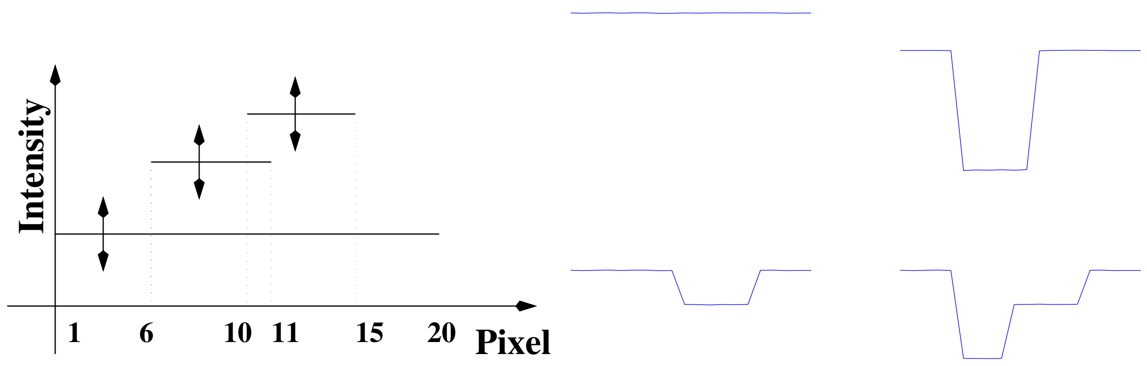


Figure 5: (Left) A generative model of 3 interacting objects and sample images from the resulting ensemble (Right). Each image is 20 pixels long. The background object appears in every image and is 20 pixels long, while the rest two appear each with a certain probability. The spatial extents and the pixel locations where these three objects appear is marked in the plot on the Left. The sample images shown on the Right can be interpreted as (going clockwise from top left): (a) image with just the background object, (b) image of the background object occluded by an object in the middle extending from pixel locations 6 to 11, (c) image of the background object occluded by two objects, one in the middle extending from pixel locations 6 to 11 (but visible only until pixel 10) and the other in the foreground extending from pixel locations 10 to 15, (d) image of the background object occluded by an object in the foreground extending from pixel locations 10 to 15. Note the intensities with which each of these objects appear in a given image is a random number sampled from a low-entropy distribution. Each pixel value here could correspond to the brightness of the object. The ensemble in turn is generated by three planar objects in the world that may or may not be present, and appear at various brightness values.

vectors are shown and they clearly indicate the low-dimensionality of the image ensemble [2]. However the structure has not been fully resolved, except for the background object (row 2, column 1).

We hypothesize that perhaps it is difficult for the sparse coding algorithm to simultaneously compute the linear subspace and resolve the independent components within. During optimization, perturbations of the basis matrix B will cause the representation to move away from the spanning space. Optimization steps mainly cause the updates to move the basis matrix back to the spanning space, instead of modifying the representation so it can come closer to capturing structure *within* the spanning space. Assuming this is the case, we split the problem into two sub-problems, namely the restriction to the subspace and the representation within the subspace.

The restriction to the subspace is achieved by principal component analysis (PCA). Images are projected to a low-dimensional subspace spanned by an *orthogonal* set of principal component vectors: $\vec{q} = U^T\vec{t}$, where $\vec{t}$ is the input image, U is a basis matrix with a small number of principal components, U^T is the PCA matrix transposed and $\vec{q}$ is the resulting low-dimensional coefficient vector. We then search for the sparse structure matrix B' in the projected space:

[2] We actually ran the ICA algorithm on these ensembles, as it is convenient to use whenever the required basis/filter matrix has a square form [2]. As mentioned in Sec 2.1, ICA learns the filter matrix D first. We then compute its (pseudo-)inverse to obtain the basis matrix B.

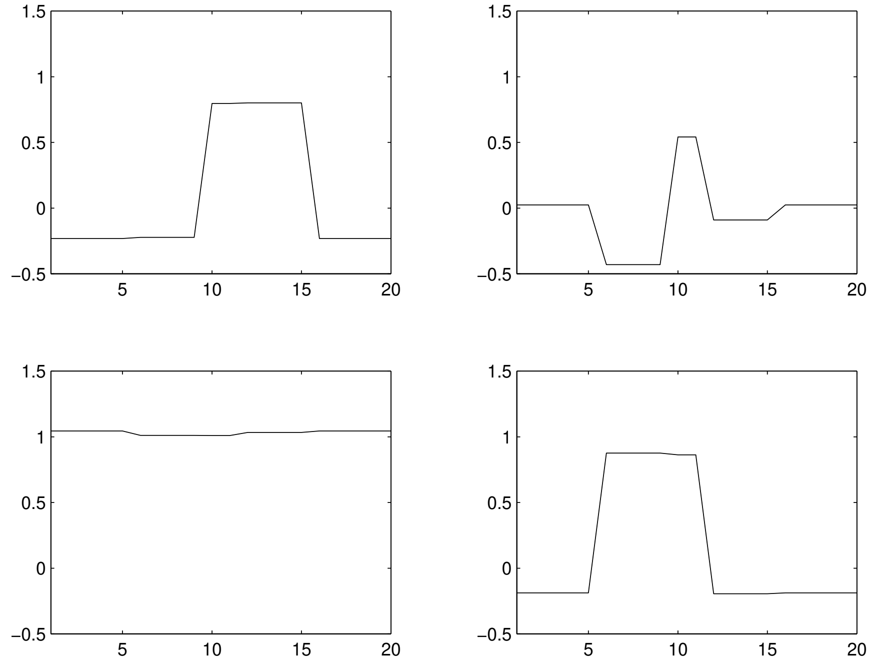


Figure 6: Structure found by sparse coding for the interacting object ensemble. Only the four non-zero basis vectors are shown. The objects have not been resolved except for the background, seen in (row 2, column 1). The rest of the basis vectors appear, once again, edge-like.

$\vec{q} = B'\vec{c}$, where $\vec{c}$ has a low-entropy prior and B' resolves the sparse structure in the set of low-dimensional data points $\vec{q}$. The resulting independent component matrix B can be derived as:

$$\begin{aligned} \vec{q} &= B'\vec{c}, \\ &= U^T\vec{t}, \\ U^T\vec{t} &= B'\vec{c}, \\ \vec{t} &= UB'\vec{c}, \\ &= B\vec{c}. \end{aligned}$$

In Fig. 7 we show the principal components obtained by performing PCA on the occluding object ensemble. It is clear once again that the data lies in a 4-dimensional subspace, beyond which the basis vectors appear to reflect the sampling noise in the ensemble. We make up a rank-4 PCA subspace matrix U using the first 4 basis vectors shown in Fig. 7. We project the ensemble onto U to obtain the coefficient vectors $\vec{q}$. In Fig. 8, we analyze the scatter plot of PCA coefficient 1 with coefficients 2, 3 and 4, to see if the sparse structure leaves any trace. Interestingly, the long tails or arms of the low-entropy distributions appear (just as in Fig. 1).

We now provide the 4 leading principal component coefficients of each image as input to the sparse coding algorithm. The input images are 4 dimensional but the basis vector matrix B' required to extract the sparse structure should be at least: 4×7. We initialized B' to be 4×8, thereby allowing one extra component to be used if needed. Sparse coding learns basis vectors minimizing Eq. 2. In Fig. 9 we plot the resulting independent component matrix B, given by

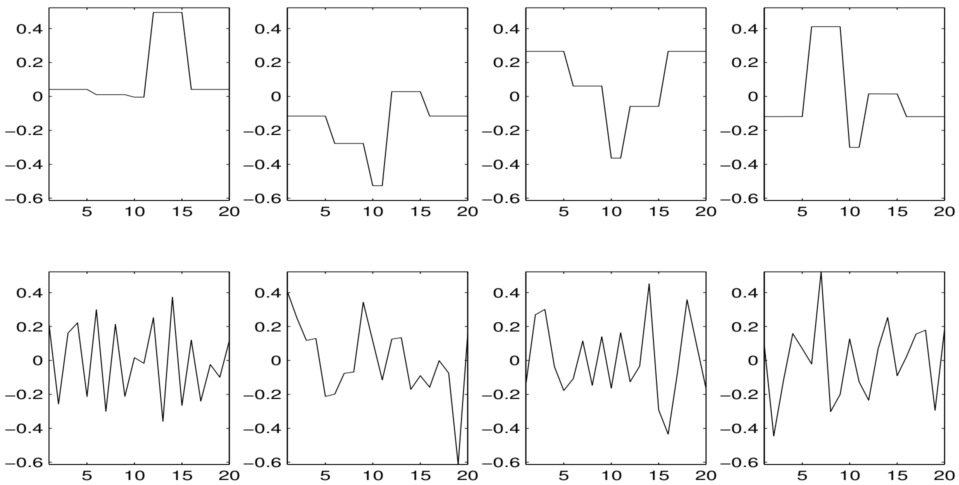


Figure 7: The first few principal components for the interacting object ensemble. They are displayed (from top left, going clockwise) in the decreasing order of the total variance captured by each basis vector. PCA identifies the principal subspace but fails to extract the independent components. Only the top 4 basis vectors have a significant variance, and these are used to project the input ensemble.

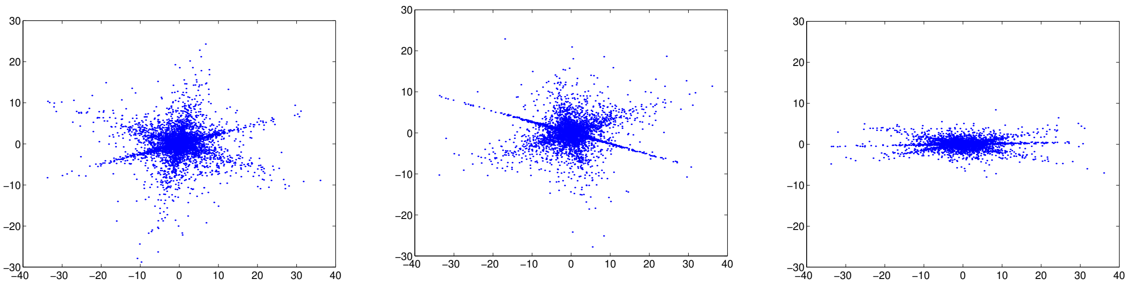


Figure 8: Scatter plot of PCA coefficients: $1-2$ (Left), $1-3$ (Middle) and $1-4$ (Right) for the interacting object image ensemble. The horizontal axis, in all the sub-plots, corresponds to the values of the first principal coefficient. Observe the long-tails typical of a low-entropy distribution.

UB'. The basis vectors successfully capture the 7 independent components. The additional basis vector is set to zero.

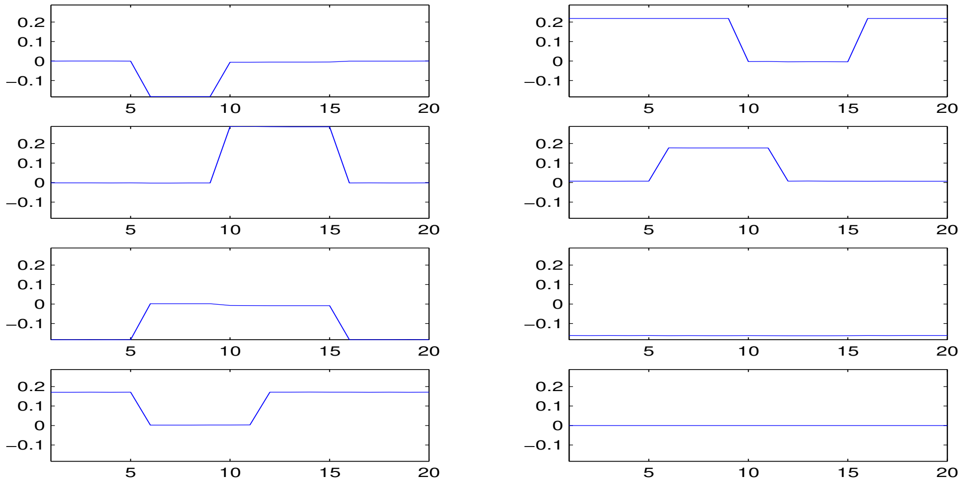


Figure 9: Independent components learned by sparse coding, after projecting input data first into a PCA subspace. The *non-zero* portion of each vector highlights pixels that vary together, but independently of the other pixels, over the ensemble. For example, the basis vectors appearing in (row 1,column 2) and in (row 2, column 1) explain all the images where the background object is occluded by the foreground. Likewise, basis vectors appearing in (rows $\{1,2,3\}$, column 1) describe images where all the objects appear together. It is beyond the scope of our linear model to collapse information contained in the 7 independent components as really coming from 3 interacting objects.

To summarize, we have shown that an arbitrarily large number of independent components can exist in a low-dimensional subspace. We found that sparse coding succeeds in extracting structure, only when the data is first projected into the subspace. Although we chose PCA, we suspect that any non-orthogonal basis which span the subspace would also be suitable for the preprocessing step.

5 Object-specific structure

What is the structure one can expect to find in an object-specific ensemble? We show that object-specific ensembles exhibit structure in a low-dimensional subspace in a sparse, scale-dependent form. *In particular, there is large-scale structure correlated across the entire image and fine-scale structure that is spatially localized.* Although we study a database of gesturing hand images (Fig. 10), the following argument holds true for other object-specific ensembles as well. Once the structure is made explicit, it will be easier to conclude whether sparse coding was successful in extracting object-specific structure.

We base our argument on the results shown by Penev and Atick [11]. We begin with a linear low-dimensional representation for the ensemble, which can be readily obtained by PCA. The principal components characterize the second-order statistical variation in the ensemble. The associated variance spectrum, for a majority of object-specific ensembles, shows a steady

Figure 10: Images of a gesturing hand. Each sub-image is of size :50×52. There are about 6 unique gestures.

drop. The images in the ensemble can be approximated using a small number of principal components.

We assume each input image $\vec{t}$ that is n-pixels long is approximated by m leading principal components to give:

$$\hat{\vec{t}} = U_{1:m} U_{1:m}^T \vec{t}, \tag{4}$$

where U is the principal component matrix with M columns *spanning* the image space, $U_{1:m}$ is a sub-matrix with m leading principal components and $m << n$ the size of the input image. Notice that $U_{1:m} U_{1:m}^T$ is the *projection* operator for the low-dimensional subspace approximating the image ensemble,

To understand structure, we ask what it means to reconstruct the intensity at a pixel position r in $\hat{\vec{t}}$? It involves the r^{th} row of the projection operator: $U_{1:m} U_{1:m}^T$. We rearrange the data in the r^{th} row of the matrix $U_{1:m} U_{1:m}^T$, so it can be seen as an image of the same size as the input. In Fig. 11, we show how the r^{th} row of $U_{1:m} U_{1:m}^T$ varies with m. Observe that with increasing m, the appearance of the r^{th} row changes from being global to very local. We interpret the projection operator in the following way:

- $U_{1:m} U_{1:m}^T$ captures correlations, not in the raw image space, but in the *whitened* space. For each image $\vec{t}$ there is a whitened image $\vec{j}$ given by:

$$\begin{aligned} \vec{j} &= U_{1:m} \Sigma^{-1/2} U_{1:m}^T \vec{t}, & (5) \\ &= U_{1:m} \Sigma^{-1/2} \vec{q}, & (6) \\ &= U_{1:m} \vec{d}, & (7) \end{aligned}$$

  where $\vec{q} = U_{1:m}^T \vec{t}$ is the projection of input image $\vec{t}$ onto the PCA subspace, Σ is a diagonal matrix providing the variance in each element of the coefficient vector $\vec{q}$ across

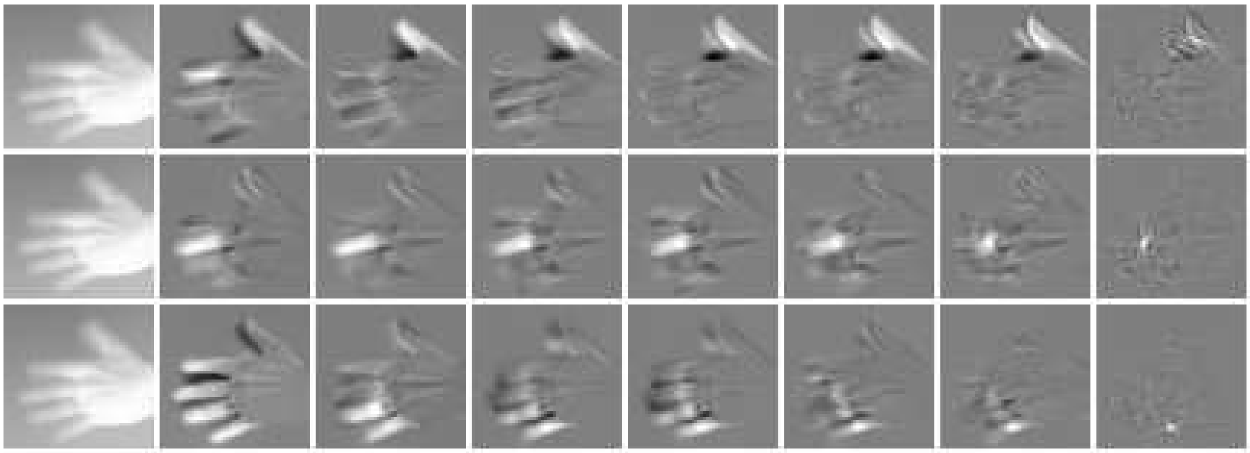


Figure 11: Structure in the hand image ensemble at 3 pixel locations centered each on the thumb (Top row), middle (Center row) and the last finger (Bottom row), corresponding to three different rows of the projection matrix: $U_{1:m}U_{1:m}^T$. Each column of the sub-figures corresponds to a value of m in $\{1, 4, 8, 12, 16, 20, 40, 156\}$, where $m = 1$ is the left most column. Sub-images are rescaled to have zero value correspond to a gray level of 127. Observe that fine scale structure caused by the moving fingers is spatially localized.

the ensemble, and $\vec{d}$ is a rescaling of $\vec{q}$ such that it can be treated as a sample from a zero-mean unit-variance Gaussian distribution (assuming the dataset is zero mean). The image $\vec{j}$ is considered whitened because the covariance matrix of $\vec{d}$ is an Identity. Hence, the covariance of the whitened image ensemble is given by:

$$< \vec{j}\vec{j}^T > \; = \; U_{1:m} < \vec{d}\vec{d}^T > U_{1:m}^T, \quad (8)$$

$$= \; U_{1:m}U_{1:m}^T. \quad (9)$$

- *The correlation in the space of whitened images* $< \vec{j}\vec{j}^T >$ *provides a measure of how appropriate it was to use* m *global models in representing the image ensemble.* If the correlations are large, global models such as PCA are appropriate. Otherwise, at least some spatially local models (and perhaps some global models) would be more appropriate. This is clearly the case for $m = 12$ and higher in Fig. 11.

The ensemble is not only low dimensional but the structure which fills the low-dimensional space has a sparse form. In Fig. 12 we show how object-specific structure, measured as correlations in the whitened image space, varies across pixels. The correlation map at each pixel takes a shape characteristic of a portion of the object (i.e. the hand) and varies smoothly between adjacent pixels. For pixels in large coherent regions, such as the back of the hand, the extent of the correlation is large. At other points the correlation maps appear sparse by being dominant only in local regions. PCA ignores this sparse form, coalescing local structure to form global basis vectors.

The fact that structure (or correlation) between any two pixels is related at multiple scales becomes evident if we regenerate the projection matrix the following way. Suppose the principal components are grouped into consecutive bands of roughly equal power. Power here is

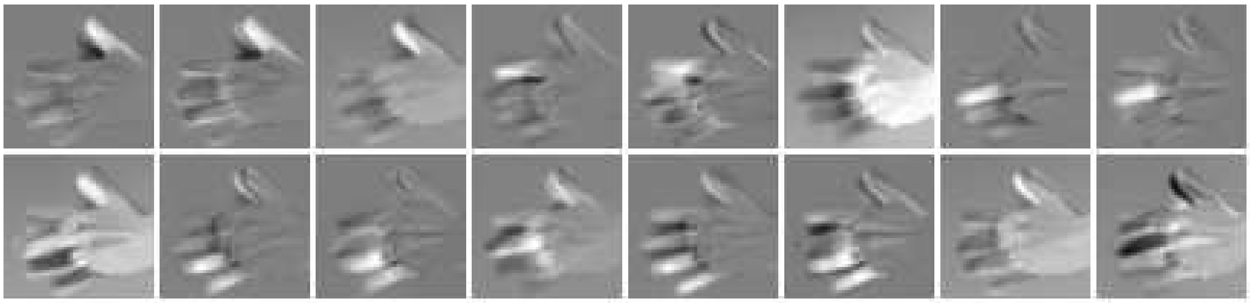


Figure 12: A sample of rows of the projection matrix $U_{1:m}U_{1:m}^T$ with $m = 10$ displayed as structure. Depending on the row selected, sub-images have shapes characterizing parts of the hand. Sub-images are rescaled to have zero value correspond to a gray level of 127.

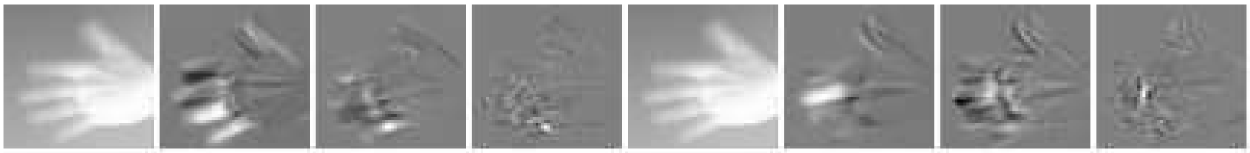


Figure 13: Multi-scale structure in the hand image ensemble measured at pixels located on the last (first 4 columns) and the middle finger (last 4 columns). Sub-images are organized by grouping sets of principal components: $[1]$, $[2-5]$, $[6-24]$, $[25-99]$. Each banded projection operator denotes a different scale for correlation.

measured as the total amount of input variance captured. Let $U_{s_l:s_h}$ be one such matrix in the s^{th} band having principal components numbered consecutively from l to h. It is clear from Fig. 13 that each banded projection operator corresponds to a different *scale* of correlation.

To summarize, we have shown that in object-specific ensembles there is a large-scale structure correlated across the entire image and fine-scale structure that is localized spatially. As we demonstrate in the next section, placing a sparse prior on the basis elements, instead of output coefficients, is a powerful way to predispose a learning mechanism to converge to this naturally-occurring, sparse, multi-scale structure. Moreover, current ICA and sparse coding algorithms do not extract this structure, as shown in Section 7.

6 Sparse PCA

In this section we provide a brief summary of a new framework, Sparse Principal Component Analysis (S-PCA), that we have proposed in [3]. The review is being included here for the sake of continuity and to allow comparison of results with the original sparse coding formulation.

S-PCA learns an orthonormal basis by simply *rotating* basis vectors that span the principal subspace. Rotation achieves sparsity in the basis vectors at the cost of introducing correlations in the output coefficients. If the input ensemble is a multi-dimensional Gaussian with widely separated variance distribution, then S-PCA returns a redundancy minimizing solution, namely the basis vectors of PCA. On the other hand, if the input ensemble is devoid of any structure (i.e. i.i.d. pixel intensities), then S-PCA returns a maximally sparse representation, with each basis vector representing the brightness at a single pixel.

The idea behind S-PCA is to retain the PCA directions when there is correlational structure in the data set, and otherwise rotate them to be as sparse as possible. We propose a cost function $C(\lambda) = C_1 + \lambda C_2$, where C_1 is a function of the variances of the data projected onto the individual basis vectors, and C_2 is a function of the elements of the basis vectors themselves.

Let $U = \{\vec{u}_1 \, \vec{u}_2 \, \cdots \, \vec{u}_m\}$ be a basis matrix spanning a m-dimensional principal subspace, with $\vec{u_k} = (u_{k,1}, \ldots, u_{k,n})^T$. Let σ_k^2 be the variances of the data projected on the direction $\vec{u}_k$. Set $\vec{\delta} = (\delta_1, \ldots, \delta_m)^T$ to be the vector of relative variances for each of the basis functions, that is, $\delta_k = \sigma_k^2 / \sum_{p=1}^{m} \sigma_p^2$. Then the first term of the cost function, namely $C_1(\vec{\delta})$, is defined to be $C_1(\vec{\delta}) = \sum_{k=1}^{m} -\delta_k \log(\delta_k)$. It can be shown that C_1 minimized only when the basis vectors are PCA directions [6].

The second term of the cost function is defined as: $C_2(U) = \sum_{i=1}^{m} \sum_{j=1}^{n} -u_{i,j}^2 \log(u_{i,j}^2)$. Notice that this is just the sum of the entropies of the distributions defined by the square of the elements for each basis vector $\vec{u}_m$ (recall the basis vectors have unit norm). If the elements of the basis vector have a Gaussian-like distribution, as in PCA, entropy is high and so is the cost function C_2. If the basis vectors from an Identity matrix, entropy is low. Thus, $C_2(U)$ can be seen as promoting sparsity.

In our experience, the exact form for C_1 and C_2 was not so important, so long as C_1 is designed to retain the PCA directions while C_2 promotes sparsity. The λ parameter in the cost function provides the relative importance of the sparsity term. In particular we chose the λ parameter to make the contributions from C_1 and C_2 have the same scale: $\lambda = (n * \log(m))^{-1}$.

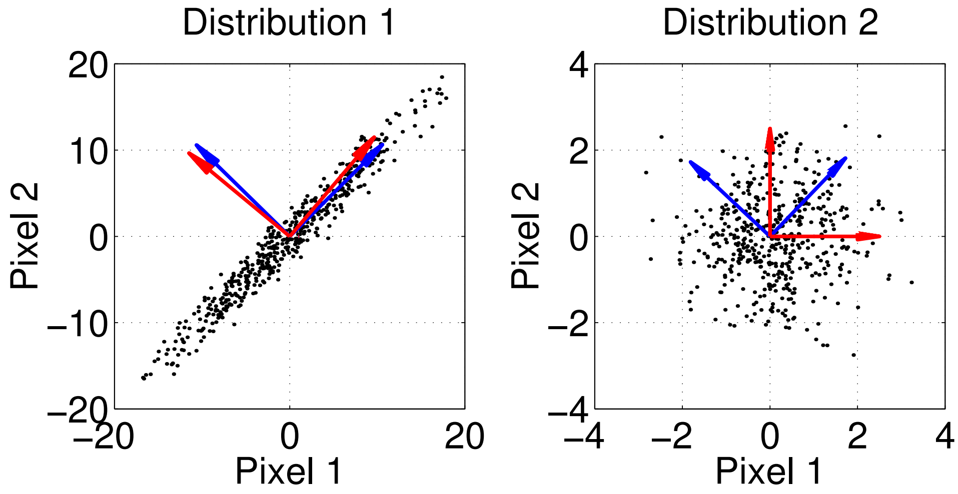
The learning algorithm of S-PCA is very simple. The basis vectors are initialized to be the principal components. The dimensionality of this principal subspace is chosen before hand. Every pair of these basis vectors defines a hyper-plane, and we successively select suitable rotations within these hyper-planes to minimize $C(\lambda)$. We sweep through every possible basis vector pair doing these rotations, and these sweeps are repeated until the change in $C(\lambda)$ is below a threshold. The product of the pairwise rotations provides a composite rotation matrix which, when applied to the PCA vectors generates the S-PCA basis. In particular, the S-PCA and PCA bases are orthonormal representations for the same subspace and are related to each by other by this rotation matrix.

The PCA basis is used as the starting point since it identifies the principal subspace best suited to recovering correlational structure. The job of the S-PCA algorithm is then simply to resolve the range of the spatial correlations. Note that the S-PCA basis is always a rotation of the original basis, so some care should be taken in choosing this starting basis. In cases for which we want a complete representation, we have found that the trivial basis (i.e. provided by the columns of the identity matrix) can also be used as a starting point for the S-PCA algorithm.

In [3] we discuss in detail the properties of this framework. Fig. 14 provides a useful summary of S-PCA.

7 Results

We report results on two different ensembles: an ensemble of gesture images and a database of face images. We show results from applying PCA, sparse coding and S-PCA.



(a) (b)

Figure 14: What is PCA good for? Distributions 1 and 2 (black dots) are 2-pixel image ensembles sampled from multi-dimensional Gaussian priors. (a) Distribution 1 has a dominant orientation indicated by the PCA basis (blue). (b) Distribution 2 has no orientational structure, and the PCA basis (blue) reflects sampling noise. In both cases the preferred S-PCA basis vectors are obtained by rotating PCA directions. In (a) the rotation is minimal, while in (b) the rotation maximizes sparsity in the basis vector description by aligning them with the pixel basis.

7.1 Images of a gesturing hand

The dataset has images of a hand undergoing various gestures (in all about 6 distinct gestures). We show a sample of images in the Fig. 10. Each image is of size: 50×52. The database has 156 images. The total number of principal components would be same as the rank of the image dataset, which is 156.

In Fig. 15, we show the results of PCA. The first few basis vectors appear relatively smooth and hence they represent global correlations. As the index k for the principal component increases, the basis vectors represent fine scale structure but remain global in space. This clearly is not representative of the object-specific structure that was just highlighted.

We now show that it is difficult to extract object-specific structure by imposing a low-entropy prior on the coefficients. In Figures 16 & 17 we show results of sparse coding on the hand image dataset. The ensemble has a low-dimensional description, so we assume independent components, if any, will reside within. To facilitate sparse coding, we first projected the data into space spanned by the first 10 principal components. It is not clear how many independent components one should search for, so we show the results from extracting 11 and 16 independent components. The overall appearance of the basis vectors is global and edge-like (alternating dark and bright regions). They cannot be simply interpreted in terms of object parts, i.e. fingers of a hand. It is also *not* clear if a low-entropy prior is appropriate on the coefficients. In Fig. 18 we show scatter plots of the coefficients of PCA for the hand images. The plots do not have *arms* the kind seen in Fig. 8 to support a low-entropy prior. We also applied the ICA algorithm but the results are not very different from the ones shown for sparse coding.

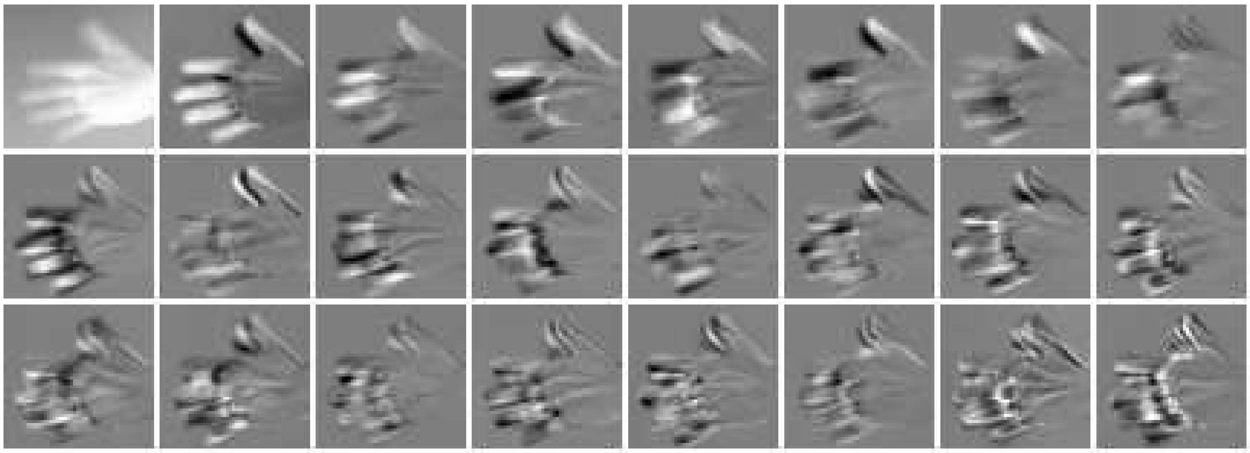


Figure 15: PCA results on the hand images. Note fine-scale information is spread across the entire image.

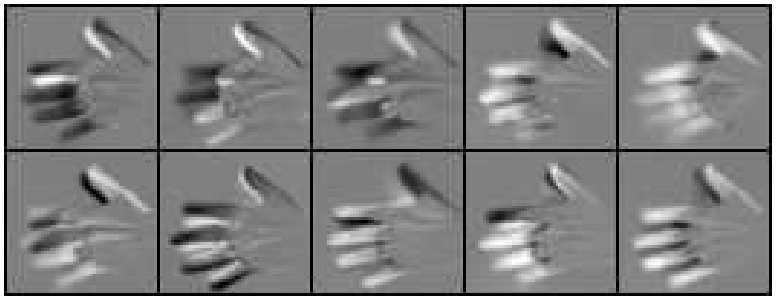


Figure 16: Sparse coding results after projecting the data into a 10-dim subspace and then searching for 11 components. This includes the mean component of the ensemble which is not shown in this figure.

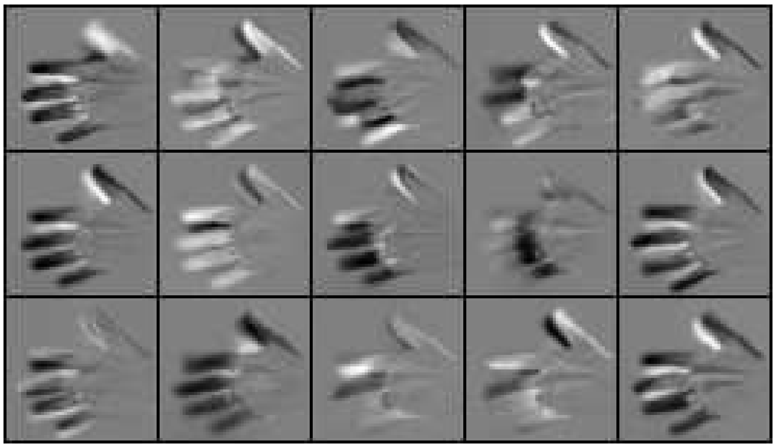


Figure 17: Sparse coding results after projecting the data into a 10-dim subspace and then searching for 16 components. This includes the mean component of the ensemble which is not shown in this figure.

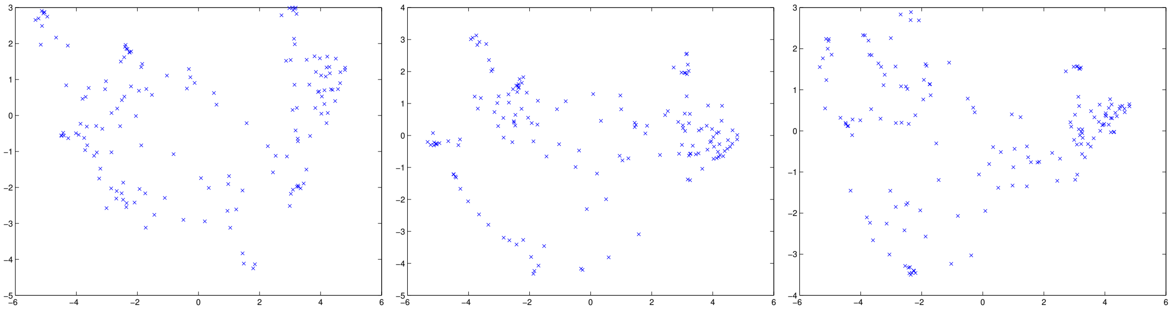


Figure 18: Scatter plot of PCA coefficient 1 (horizontal axis) with coefficients 2, 3 and 4, for the hand image ensemble.

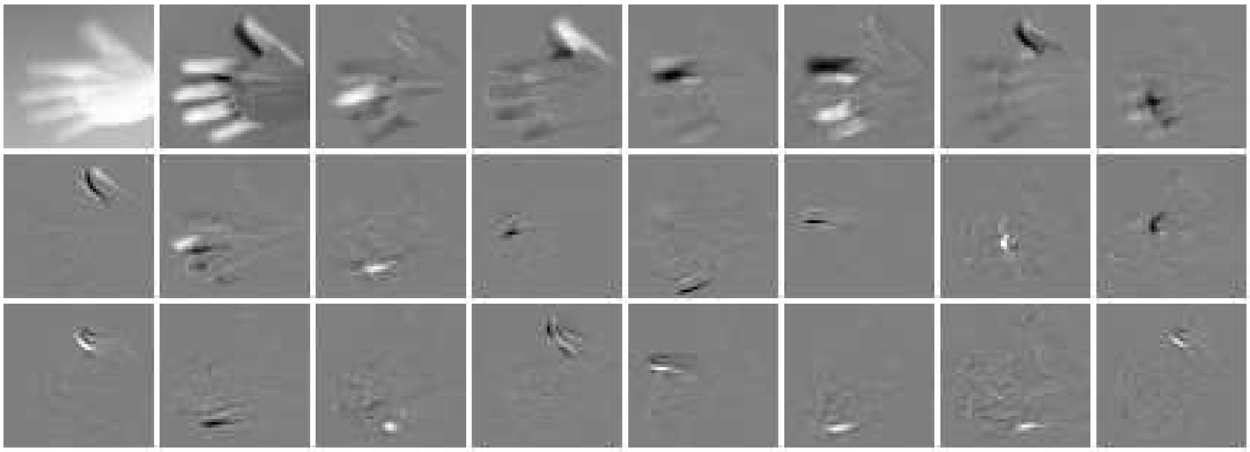


Figure 19: Sparse-PCA results on the hand images. Only the first 24 basis vectors are displayed here. Unlike PCA, basis vectors representing fine-scale information are spatially localized.

However, S-PCA, which promotes sparsity in the basis matrix, results in basis vectors that are far more representative of the object-specific structure. As shown in Fig. 19 basis vectors representing fine-scale information are spatially localized.

7.2 Face images

We used the publicly available ORL database [10]. These are face images, frontal-looking but only roughly aligned. Each image is of size: 112×92. The database has 400 images.

The PCA results again show lack of spatial localization of fine-scale structure (Fig. 20). We are not reporting sparse coding results here. Instead we refer the reader to Marian Bartlett's thesis [1], where she applied ICA to a face database. The filter matrix obtained from ICA has some resemblence to facial features, but the resulting basis matrix appears "holistic". In particular, the scale-separation of structure is not evident. In comparison, we show results from S-PCA and the resulting basis vectors appear to respond to object-specific structure (Fig. 21).

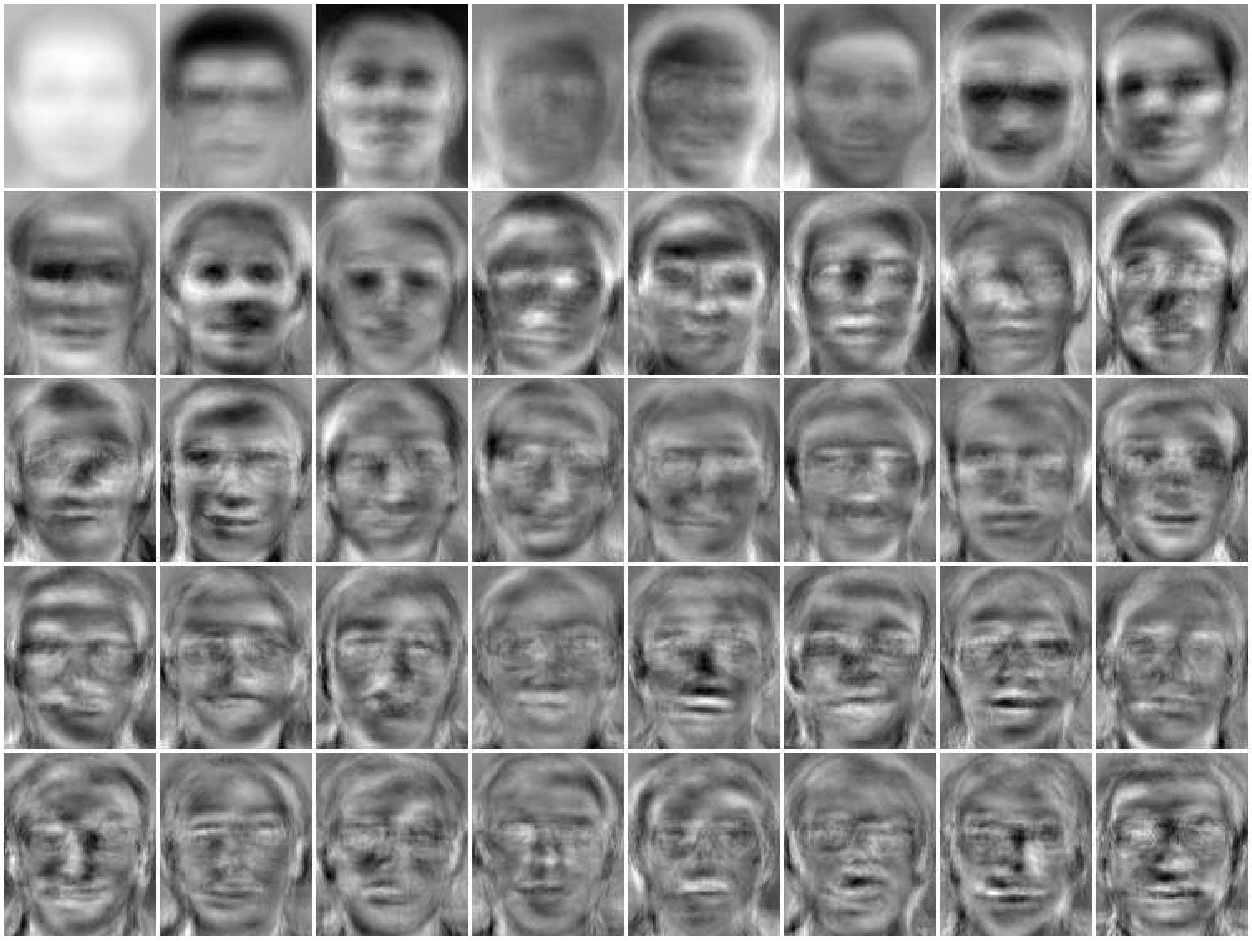


Figure 20: PCA basis vectors on face images. The first 40 principal components are displayed.

8 Related Work

The references in this section are by no means exhaustive. We highlight material that is most relevant to our work.

Recently, Ruderman, Huang, Lee and Mumford articulated occlusion models that can generate scale-invariant statistics [12, 5, 8]. They proposed generative models which simulate statistics measured in natural scenes. In comparison, our interest is in the inference problem: to determine what the objects are. Of course, we cannot approach this problem with out having proper generative models. This lead to us to design simple models that can highlight issues with a particular inference algorithm: sparse coding. Also, our interest is in object-specific ensembles, such as face images. As we have shown, these images are not likely to exhibit scale-invariant statistics but they clearly have structure at multiple scales, which is of considerable interest.

The two issues: overcomplete representation and subspace projection before sparse coding, have been argued before as desirable for various reasons. An overcomplete representation, besides promoting sparsity, was shown to provide smooth and interpolated output for small changes in the input [9]. Subspace projection using PCA has been found to be useful in practice

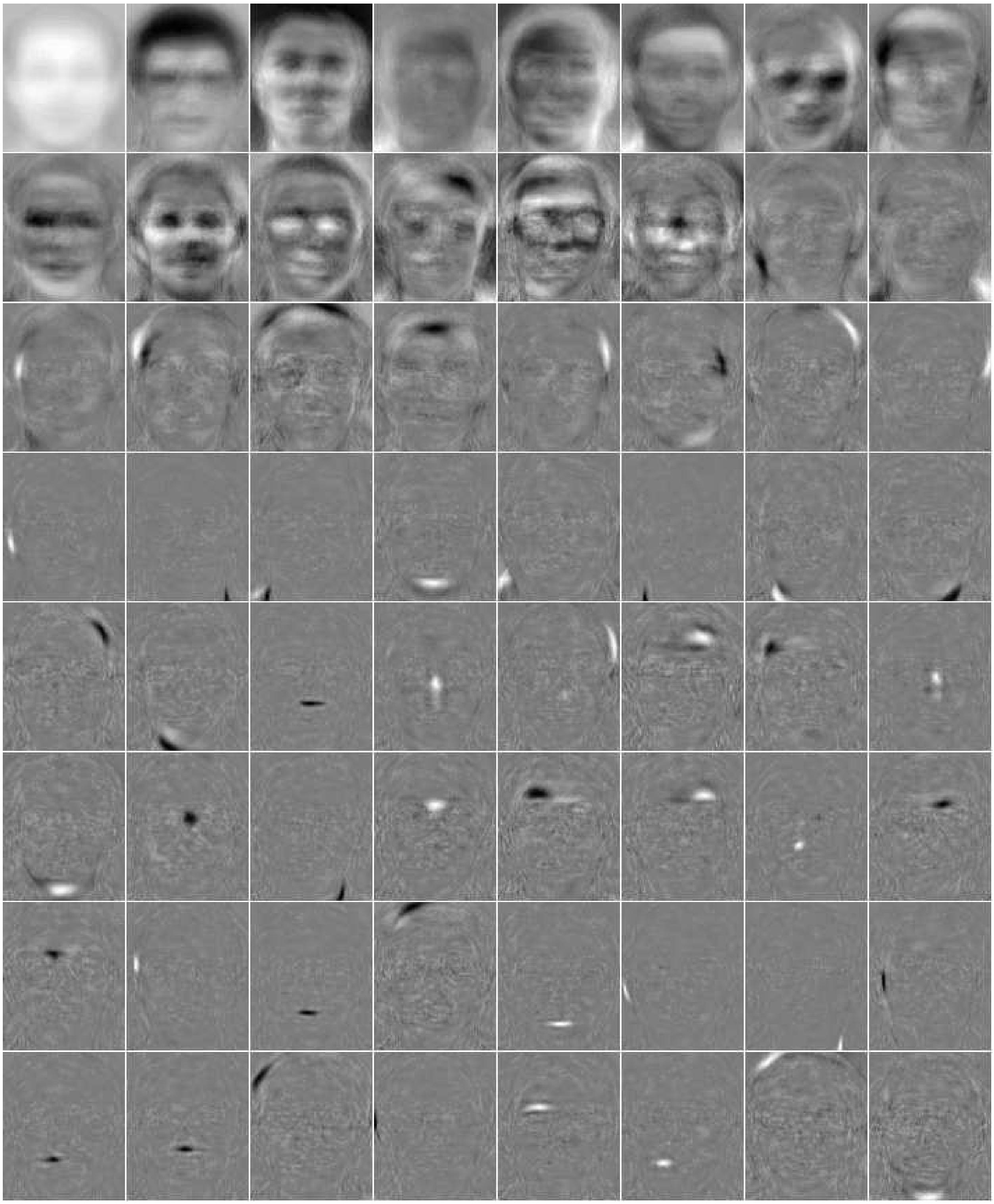


Figure 21: Sparse-PCA results on the face images. Only the first 64 basis vectors are displayed here. Unlike PCA, basis vectors representing fine-scale information are spatially localized.

for resolving independent components in fMRI/EEG/MEG data. In this paper, we have shown the critical role played by these two issues in extracting object-specific structure, using simple models for the real world images.

The structure argument we make for object-specific ensembles is based on the work by Penev and Atick on Local Feature Analysis [11]. We extend the argument by forming banded projection operators (also suggested by Penev in his thesis [11]), downplay the role of eigenvalues and associate structural information to correlations appearing in the whitened image space. Penev and Atick chose the columns of the projection matrix as a representation for the data. Instead, we learn an orthogonal basis representation, that captures multi-scale information in the input data, by simply promoting sparsity in a basis matrix [3].

9 Conclusions

In trying to infer object-specific structure by sparse coding, we found three conditions necessary for the sparse coding algorithm to succeed. First, we showed ensembles where overcomplete representation is critical. Otherwise, sparse coding extracts edge directions that potentially summarize more than one independent component. Second, sparse coding may fail if it is not initialized in the low-dimensional subspace where the actual structure exists. Third, we showed how object-specific ensembles exhibit structure in a low-dimensional subspace in a sparse, scale-dependent form. In particular, fine-scale structure typically requires small spatial support, while large-scale structure is correlated (globally) across the image. The extraction of this form of structure appears to require a sparsity constraint on the basis matrix, as opposed to the coefficients.

Acknowledgements: We thank Michael Black for the hand image sequence.

References

[1] Bartlett, M. Face image analysis by unsupervised learning and redundancy reduction Doctoral Dissertation, University of California, San Diego, 1998

[2] Bell, A. J. and T. J. Sejnowski. The Independent Components of natural scenes are edge filters. *Vision Research*, 37:3327-3338. 1997.

[3] Chakra Chennubhotla and Allan D. Jepson. Sparse PCA: Extracting multi-scale structure from data *International Conference on Computer Vision*, Vancouver, July 2001.

[4] Harpur, G. F. and R. W. Prager Development of Low-entropy coding in a recurrent framework. *Network: Computation in Neural Systems* , 7(2):277-284, 1996.

[5] Huang, J. and D. Mumford Statistics of Natural Images and Models *CVPR* , 541-547, 1999.

[6] Devijver, P. A and J. Kittler. Pattern Recognition: A Statistical Approach. Prentice Hall, 1982.

[7] Lewicki, M. S. and T. J. Sejnowski. Learning Overcomplete Representations. *Neural Computation*, 2000.

[8] Lee, A. and D. Mumford Scale-Invariant Random-Collage Model for Natural Images *Workshop on Statistical and Computational Theories of Vision*, Fort Collins, CO 1999

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

[10] ORL Database, AT&T Laboratories Cambridge ftp://ftp.uk.research.att.com:pub/data/att_faces.tar.Z

[11] Penev, P. S. and J. J. Atick. Local Feature Analysis. *Network: Computation in Neural Systems*, 7(3), 477-500, 1996.

[12] Ruderman, D. Origins of Scaling in Natural Images *Vision Research*, 3385-3395, Vol 37, No 23, 1997.