Multi-Scale Hybrid Linear Models for Lossy Image Representation

Wei Hong, John Wright, Kun Huang, *Member, IEEE*, Yi Ma, *Senior Member, IEEE*

***Abstract*— In this paper, we introduce a simple and efficient representation for natural images. We view an image (in either the spatial domain or the wavelet domain) as a collection of vectors in a high-dimensional space. We then fit a piece-wise linear model (i.e. a union of affine subspaces) to the vectors at each down-sampling scale. We call this a *multi-scale hybrid linear model* for the image. The model can be effectively estimated via a new algebraic method known as generalized principal component analysis (GPCA). The hybrid and hierarchical structure of this model allows us to effectively extract and exploit multi-modal correlations among the imagery data at different scales. It conceptually and computationally remedies limitations of many existing image representation methods that are based on either a fixed linear transformation (e.g. DCT, wavelets), or an adaptive uni-modal linear transformation (e.g. PCA), or a multi-modal model that uses only cluster means (e.g. VQ). We will justify both quantitatively and experimentally why and how such a simple multi-scale hybrid model is able to reduce simultaneously the model complexity and computational cost. Despite a small overhead of the model, our careful and extensive experimental results show that this new model gives more compact representations for a wide variety of natural images under a wide range of signal-to-noise ratio than many existing methods, including wavelets. We also briefly address how the same (hybrid linear) modelling paradigm can be extended to be potentially useful for other applications, such as image segmentation.**

***Index Terms*— image representation, hybrid linear model, generalized principal component analysis, wavelets.**

I. Introduction

RESEARCHERS in image processing and computer vision have long sought efficient and sparse representations of images. Except for a few image representations such as fractal-based approaches [1], most existing sparse image representations use an effective linear transformation so that the (transformed) image energy will be concentrated in the coefficients of a small set of bases of the transformation. Computing such a representation is typically the first step of subsequent lossy coding/decoding of the image. The result can also be potentially useful for other purposes such as image segmentation, classification, and recognition.

Wei Hong, John Wright, and Yi Ma are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Their work is supported by the grants NSF CAREER IIS-0347456, NSF CRS-EHS-0509151, NSF CCF-TF-0514955, and ONR YIP-N000140510633.

Kun Huang is with the Department of Biomedical Informatics, Ohio State University.

State of the Knowledge. Roughly speaking, most of the popular methods for obtaining a sparse image representation can be classified into two categories.

Methods of the first category seek to transform all images using a *pre-fixed* linear transformation. Each image is then represented as a superposition of a set of basis functions (specified by the transformation). These methods mainly evolved from the classical Fourier Transform. Its variation, the Discrete Cosine Transform (DCT), serves as the core of the JPEG standard [2]. Due to the Gibbs' phenomenon, DCT is poor at approximating discontinuities or impulses in the imagery signal. Wavelets [3]–[7] have been developed to remedy this problem and have been shown to be optimal for representing 1-D signals with discontinuities.[1] JPEG-2000 adopted wavelets as its standard. However, because wavelet transforms only deal with 1-D discontinuities, they are not well-suited for representing 2-D singularities along edges or contours. Anisotropic bases such as wedgelets [9], curvelets [10] and countourlets [11] have been proposed explicitly to capture different 2-D discontinuities. These x-lets have been shown to be (approximately) optimal for representing objects with singularities along C^2-smooth contours. However, natural images, especially images that have complex textures and patterns, do not consist solely of discontinuities along C^2-smooth edges. This is probably the reason why these edge-based methods do not seem to outperform separable wavelets on complex images [12].

Methods of the second category aim to identify the optimal (or approximately optimal) representation that is *adaptive* to the specific statistics or structures of each image.[2] The correlation across different regions and different color channels of a image can be captured by the adaptive basis. The Karhunen-Loève transform (KLT) or principal component analysis (PCA) [13] identifies the optimal principal subspace based on the statistical correlation of the imagery data and represents the image as a superposition of the basis of the subspace. In theory, PCA provides the optimal linear sparse representation for imagery data that satisfy a uni-modal

[1]Here, "optimality" means that the transformation achieves the optimal asymptotic rate for approximating the class of piece-wise smooth functions [8]. The optimality of x-lets is also used in the same sense.

[2]Here, unlike in the case of prefixed transformations, "optimality" means the representation obtained is the optimal one within the class of models considered, in the sense that it minimizes certain discrepancy between the model and the data.

distribution. However, natural images typically exhibit multi-modal statistics as they usually contain many heterogeneous regions with significantly different geometric structures or statistical characteristics (e.g. Figure 3). Heterogeneous data can be better-represented using a mixture of parametric models, one for each homogeneous subset. Bases for each model are adaptive to the particular homogeneous subset. Such a mixture of models is often referred to as a *hybrid model*. Vector quantization (VQ) [14] is a special hybrid model which assumes the imagery data are clustered around many different centers. From the dimension reduction point of view, VQ represents the imagery data with many 0-dimensional (affine) subspaces. This model typically leads to an excessive number of clusters or subspaces.[3] Bandelets [12] are another hybrid model, which partitions an image into squares according to the geometric flows. Within each square, the image is represented by warped wavelet bases, oriented along the direction of the geometric flows. But for images with complex textures which do not contain obvious oriented geometrical structures, segmenting and estimating geometric flows will be costly and inaccurate. The primal sketch model [15] is another hybrid model which represents the high entropy parts of images with multiple Markov random fields [16]–[18] and the low entropy parts with sketches. The result is also a "sparse" representation of the image as superposition of the random fields and sketches. However, the primary goal of primal sketch is not to authentically represent and approximate the original image. It is meant to capture the (stochastic) generative model that produces the image (as random samples). Therefore, this type of models are more suited for image parsing, recognition, and synthesis than compression. In addition, finding the sketches and estimating the parameters of the random fields are computationally expensive and therefore less appealing for developing efficient image representation and compression schemes.

Motivations of This Paper. In this paper, we generalize the techniques of PCA and VQ and propose to represent an image by a collection of (affine) subspaces, one subspace for a different segment (region) of the image. The dimension and basis of each subspace will be chosen adaptively according to the variability and correlation of the data in the corresponding image segment. We call such a representation as a *hybrid linear model* for the image. Conceptually, the hybrid linear model is related to a mixture of Gaussian distributions (PCAs), which has been previously explored for image representation and segmentation [19]–[22]. However, the hybrid linear model differs from the mixed Gaussian distributions in several important aspects. The hybrid linear model does not explicitly assume any statistical distribution for the data (such as Gaussians or exponentials). It allows the subspaces to have different dimensions (for image regions of different textures), that do not need to be known *a priori*. Unlike a mixture of Gaussians whose estimation often requires iterative nonlinear optimization schemes such as Expectation and Maximization [23], there exist non-iterative and efficient (linear) algebraic methods for *simultaneously* estimating the subspaces and segmenting the data. Generalized principal component analysis (GPCA) [24], [25] is one such method. This paper is the first to apply such useful formulation and methods to sparse image representations. Our method is also different from the conventional sparse representation methods (e.g., that based on L^1 minimization [26], [27]) in at least two different aspects: First, the bases of the subspaces are no longer given or known; Second, the assignment of respective subspace bases to the imagery data is found simultaneously with the subspaces, rather than obtained subsequently via L^1 minimization. The hybrid linear model has been shown to strike a good balance between simplicity and expressiveness for representing natural images [28]. This paper aims to provide even more convincing results.

The conventional transformations such as DCT and x-lets apply primarily to scalar functions (i.e. grayscale images). For vector-valued functions (e.g., color images, hyper-spectral images, diffusion tensor images), normally one needs to apply the transformations to each scalar channel separately. In the literature, many effective methods have been proposed to harness (statistical or geometric) correlation among different color channels for the purpose of image representation (see [29] and references therein). The method introduced in this paper provides a natural framework for representing and approximating high-dimensional vector-valued functions. The results of this paper will demonstrate that one often can obtain equally compact representations for images by harnessing primarily the correlation among the multiple color channels and different regions, instead of the spatial smoothness of each color channel as a function.

Another important characteristic of natural images is that they are comprised of structures at many different (spatial or frequency) scales. Many existing frequency-domain techniques harness this characteristic [30]. For instance, wavelets, curvelets, and fractals have all demonstrated effectiveness in decomposing the original imagery signal into multiple scales (or subbands). As the result of such a *multi-scale* decomposition, the structures of the image at different scales (e.g., low v.s. high frequency/entropy) become better exposed and hence can be more compactly represented. The availability of multi-scale structures also significantly reduces the size and dimension of the problem and hence reduces the overall computational complexity. Therefore, in this paper, we propose a new approach to represent imagery data by combining the hybrid paradigm and the multi-scale paradigm. The result is a *multi-scale hybrid linear model* which is based on an extremely simple concept: Given an image, at

[3]Be aware that compared to methods in the first category, representations in the second category typically need additional memory to store the information about the resulting model itself, e.g., the basis of the subspace in PCA, the cluster means in VQ.

each scale level of its down-sample pyramid, fit the (residual) image or the wavelet coefficients by a (multiple-subspace) hybrid linear model. Compared to the single-scale hybrid linear model, the multi-scale scheme can reduce not only the size of the resulting representation but also the overall computational cost. Surprisingly, as we will demonstrate, such a simple scheme is able to generate representations for imagery data that are more compact, even with the overhead needed to store the model, than most state-of-the-art representations, including DCT, PCA, and wavelets.

Organization of This Paper. This paper aims to provide a comprehensive introduction and development of the multi-scale hybrid linear model for (lossy) image representation. This paper focuses on algorithms and implementations, but mathematical proofs and details of GPCA can be found in [24], [25]. Section II gives a brief review of the basic notions and techniques of using a subspace, called a linear model, to represent images. It makes this paper more self-contained for readers who might not be so familiar with subspace methods. Section III formally introduces the notion of hybrid linear model for image representation purposes. It develops necessary concepts and technical components for estimating from an given image a hybrid linear model that consists of multiple subspaces of possible different dimensions. It also discusses the complexity of a hybrid linear model and what is the final result of the image represented by such a model. Section IV shows how to implement the hybrid linear models in a multiple-scale fashion in the spatial domain (i.e., in the original image coordinates); while Section V extends the techniques to the multiple-scale wavelet domain. In both cases, we provide careful and extensive experimental results in comparison with other popular image representation methods (e.g., DCT, PCA, wavelets).

What We Do Not Do. In this paper, we are only interested in studying and comparing the efficiency of different transformations in approximating the image. The results are all compared immediately after such (linear or hybrid-linear) transformations *without* any further quantization or entropy coding. In order for the proposed hybrid linear models to be truly useful for image compression, one must investigate if there also exist efficient methods for quantizing and coding such models. That is however beyond the scope of this paper and we leave it for future research. However, as we will contend later that the merit of the proposed image representation scheme is not limited only to image representation and compression. Potentially, it can also be very useful for segmentation of image and other types of data, as Sections VI suggest.

II. Linear Models

A. Imagery Data Vectors in Spatial Domain

An image $\boldsymbol{I}$ with width W, height H, and c color channels can be viewed as an array of total $W \times H \times c$ numbers. Thus, the image $\boldsymbol{I}$ resides in a very high-dimensional space $\mathbb{R}^{W\times H\times c}$. We may first reduce the dimension by dividing the image into a set of (non-overlapping) b by b blocks.[4] Each b by b block can then be stacked into a vector $\boldsymbol{x} \in \mathbb{R}^K$, where $K = b^2c$ is the dimension of the ambient space. For example, if $c = 3$ and $b = 2$, then $K = 12$ (see Figure 1). In this way, the image $\boldsymbol{I}$ is converted to a set of vectors $\{\boldsymbol{x}_i \in \mathbb{R}^K\}_{i=1}^M$, where $M = WH/b^2$ is the total number of vectors. The so-defined vectors will be referred to as the imagery vectors "in the spatial domain," to be distinguished from the imagery vectors defined "in the wavelets domain" that we will introduce later in Section V.

Borrowing ideas from existing unsupervised learning paradigms, it is tempting to assume the imagery data vectors $\{\boldsymbol{x}_i\}$ are random samples from a (non-singular) probability distribution or noisy samples from a smooth manifold. As the distribution or manifold can be very complicated, a common approach is to infer a best approximation within a simpler class of models for the distributions or manifolds. The "optimal" model is then the one that minimizes certain measure of discrepancy to the true model. Different choices of model classes and discrepancy measures have led to many different learning algorithms developed in machine learning, pattern recognition, computer vision, and image processing. For image compression, the most commonly adopted discrepancy measure is the Mean Square Error (MSE) between the original image $\boldsymbol{I}$ and approximated image $\hat{\boldsymbol{I}}$,

$$\epsilon_I^2 = \frac{1}{WHc}\|\hat{\boldsymbol{I}} - \boldsymbol{I}\|^2. \tag{1}$$

Since we will be approximating the (block) vectors $\{\boldsymbol{x}_i\}$ rather than the image pixels, in the following derivation, it is more convenient for us to define the Mean Square Error (MSE) *per vector* which is different from ϵ_I^2 by a scale,

$$\begin{aligned}\epsilon^2 &= \frac{1}{M}\sum_{i=1}^{M}\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|^2 = \frac{b^2}{WH}\sum_{i=1}^{M}\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|^2 \\ &= \frac{b^2}{WH}\|\hat{\boldsymbol{I}} - \boldsymbol{I}\|^2 = (b^2c)\epsilon_I^2 = K\epsilon_I^2.\end{aligned} \tag{2}$$

The Peak Signal to Noise Ratio (PSNR) of the approximated image will be used in experiment sections and it is defined to be,[5]

$$\text{PSNR} \doteq -10\log\epsilon_I^2 = -10\log\frac{\epsilon^2}{b^2c}. \tag{3}$$

B. Estimation of A Linear Model

If we assume that the vectors $\boldsymbol{x}$ are drawn from an anisotropic Gaussian distribution or a linear subspace as illustrated in Figure 1, the optimal model subject to a given PSNR can be inferred by Principal Component Analysis

[4] Conventionally one chooses b to be a common divisor of W and H.

[5] The peak value of the imagery data is normalized to 1.

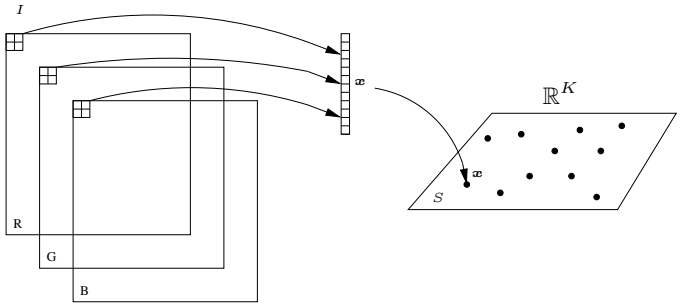

Fig. 1. In a linear model, the imagery data vectors $\{\boldsymbol{x}_i\}$ reside in an (affine) subspace.

(PCA) [31]–[33] or equivalently the Karhunen-Loève Transform (KLT) [13]. We review below briefly a solution to PCA based on singular value decomposition (SVD), also known as the Erkart and Young decomposition [34], [35].

Typically, the principal subspace is an affine subspace which does not necessarily pass through the origin, we can move the affine subspace into a linear subspace by subtracting the mean of the samples from each sample vector. Let $\bar{\boldsymbol{x}} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{x}_i$ be the mean of the imagery data vectors, and $\boldsymbol{X} = [\boldsymbol{x}_1 - \bar{\boldsymbol{x}}, \boldsymbol{x}_2 - \bar{\boldsymbol{x}}, \ldots, \boldsymbol{x}_M - \bar{\boldsymbol{x}}] = U\Sigma V^T$ be the SVD of the mean-subtracted data matrix $\boldsymbol{X}$. Then all the vectors $\boldsymbol{x}_i$ can be represented as a linear superposition: $\boldsymbol{x}_i = \bar{\boldsymbol{x}} + \sum_{j=1}^{K} \alpha_i^j \phi_j, i = 1, \ldots, M$, where $\{\phi_j\}_{j=1}^{K}$ are just the columns of the matrix U.

The matrix $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K)$ contains the ordered singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_K$. It is well known that the optimal linear representation of $\boldsymbol{x}_i$ subject to the MSE ϵ^2 is obtained by keeping the first k (principal) components

$$\hat{\boldsymbol{x}}_i \doteq \bar{\boldsymbol{x}} + \sum_{j=1}^{k} \alpha_i^j \phi_j, \quad i = 1, \ldots, M, \tag{4}$$

where k is chosen to be

$$k = \min(n), \quad \text{s.t.} \quad \frac{1}{M}\sum_{i=n+1}^{K} \sigma_i^2 \leq \epsilon^2. \tag{5}$$

Symbolically, the process is represented by the following diagram:

$$\{\boldsymbol{x}_i\} \subset \mathbb{R}^K \xrightarrow{\text{PCA}} \{\boldsymbol{x}_i'\} \subset \mathbb{R}^{K'}.$$

C. Model Complexity of A Linear Model

The model complexity of a linear model, denoted as Ω, is the total number of coefficients needed for representing the model $\{\alpha_i^j, \phi_j, \bar{\boldsymbol{x}}\}$ and subsequently a lossy approximation $\hat{\boldsymbol{I}}$ of the image $\boldsymbol{I}$. It is given by

$$\Omega(M, k) \doteq Mk + k(K - k + 1), \tag{6}$$

where the first term is the number of coefficients $\{\alpha_i^j\}$ to represent $\{\hat{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}\}_{i=1}^{M}$ with respect to the basis $\Phi = \{\phi_j\}_{j=1}^{k}$ and the second term is the number of Grassmannian coordinates[6] needed for representing the basis Φ and the mean vector $\bar{\boldsymbol{x}}$. The second term is often called *overhead*.[7] Notice that the original set of vectors $\{\boldsymbol{x}_i\}$ contain MK coordinate entries. If $\Omega \ll MK$, the new representation, although lossy, is much more compact. The search for such a compact representation is at the heart of any (lossy) image compression method. When the image $\boldsymbol{I}$ is large and the block size b is small, M will be much larger than K so that the overhead will be much smaller than the first term. However, in order to compare fairly with other methods, in the subsequent discussions and experiments, we always count the total number of coefficients needed for the representation, including the overhead.

III. Hybrid Linear Models

In this section we introduce and examine the hybrid linear model. The relationship between hybrid linear models across different scale levels will be discussed in Section IV.

A. Hybrid Linear Models

The linear model is very efficient when the target manifold or distribution function is unimodal. However, if the image $\boldsymbol{I}$ contains several heterogeneous regions $\{\boldsymbol{I}_n\}_{n=1}^{N}$, the data vectors $\boldsymbol{x}_i$ can be viewed as samples from a collection of (affine) subspaces of possibly different dimensions or from a mixture of multiple (Gaussian) distributions as shown in Figure 2. Figure 3 shows the first three principal components

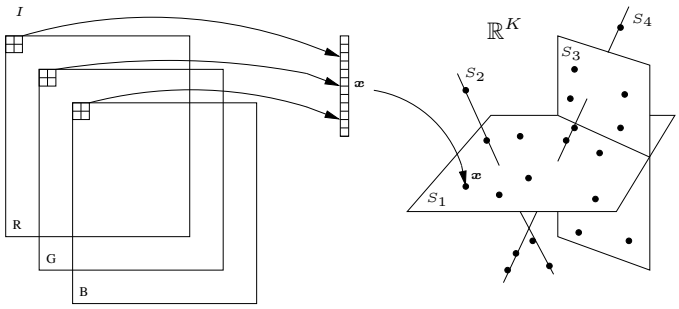

Fig. 2. In hybrid linear models, the imagery data vectors $\{\boldsymbol{x}_i\}$ reside in multiple (affine) subspaces which may have different dimensions.

of the data vectors $\boldsymbol{x}_i$ (as dots in $\mathbb{R}^3$) of the baboon image. Note the clear multi-modal characteristic in the distribution of the data.

[6]Notice that to represent a k dimensional subspace in a K dimensional space, we only need to specify a basis of k linearly independent vectors for the subspace. We may stack these vectors as rows of a $k \times K$ matrix. Any nonsingular linear transformation of these vectors span the same subspace. Thus, without loss of generality, we may assume that the matrix is of the normal form $[I_{k\times k}, G]$ where G is a $k \times (K - k)$ matrix consisting of the so-called Grassmannian coordinates.

[7]Notice that if one uses a pre-chosen basis such as discrete Fourier transform, discrete cosine transform (JPEG), and wavelets (JPEG-2000), there is no such overhead.

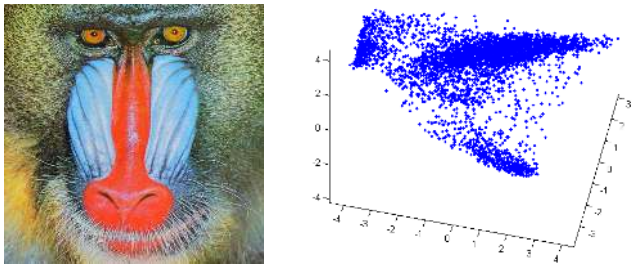

Fig. 3. Left: The baboon image. Right: The coordinates of each dot are the first three principal components of the vectors $\boldsymbol{x}_i$. There is a clear multi-modal structure in the data.

Thus, it is probably more reasonable to assume that a natural image $\boldsymbol{I}$ can be segmented into N disjoint regions $\boldsymbol{I} = \cup_{n=1}^{N} \boldsymbol{I}_n$ with $\boldsymbol{I}_n \cap \boldsymbol{I}_{n'} = \emptyset$ for $n \neq n'$. In each region $\boldsymbol{I}_n$, a linear model (4) is approximately valid for the subset of vectors $\{\boldsymbol{x}_{n,i}\}_{i=1}^{M_n}$ in $\boldsymbol{I}_n$:

$$\hat{\boldsymbol{x}}_{n,i} = \bar{\boldsymbol{x}}_n + \sum_{j=1}^{k_n} \alpha_i^j \phi_{n,j}, \quad i = 1, \ldots, M_n. \tag{7}$$

As in the linear model, the dimension k_n of each subspace should be determined by a common desired MSE ϵ^2 as in equation (5). We call the result $\{\phi_{n,j}, \hat{\boldsymbol{x}}_{n,i}\}$ a *hybrid linear model* for the image. In essence, the hybrid linear model assumes that the imagery data $\{\boldsymbol{x}_i\}$ belong to a collection of subspaces $\{S_n\}_{n=1}^{N}$:

$$\boldsymbol{I}_n = \{\boldsymbol{x}_{n,i}\} \subset S_n \quad \subset \mathbb{R}^K, \quad \forall n = 1, \ldots, N. \tag{8}$$

A collection of subspaces $Z \doteq \cup_{n=1}^{N} S_n$ is formally known in algebraic geometry as a *subspace arrangement*.

We emphasize that here the subspaces S_n are allowed to intersect with each other and can have different dimensions (as different textures in the image can have different complexities). Thus, a subspace arrangement is *not* a smooth manifold or a mixture of (non-degenerate) Gaussians. Strictly speaking, a subspace arrangement is an algebraic set.[8] In this paper, we contend that subspace arrangements are a more flexible, relevant, and efficient class of models for representing images than nonlinear smooth manifolds or mixtures of Gaussian distributions.

Subspace arrangements constitute of a very special but important class of algebraic sets that have been studied in mathematics for centuries (see [36], [37] and references therein). The importance as well as the difficulty of studying subspace arrangements can hardly be exaggerated. Different aspects of their properties have been and are still being investigated and exploited in many mathematical fields, including algebraic geometry & topology, combinatorics and complexity theory, and graph and lattice theory, etc. Interested readers may see [37] for a general review. Although the results about subspace arrangements are extremely rich and deep, only a few special classes of subspace arrangements have been fully characterized.

B. Identifying A Hybrid Linear Model

Here we are interested in only how to estimate such an arrangement of subspaces from the un-segmented imagery vectors $\{\boldsymbol{x}_i\}$. Notice that the subspaces S_n do not necessarily pass through the origin. Such a subspace is called an affine subspace. Technically affine subspaces are slightly less convenient to deal with than linear subspaces that pass through the origin. Notice that all vectors in an affine subspace of dimension k in $\mathbb{R}^K$ span a linear subspace of dimension $k+1$ in $\mathbb{R}^{K+1}$ by identifying a point in $\mathbb{R}^K$ as a point in $\mathbb{R}^{K+1}$ via the so-called homogeneous coordinates,[9]

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \end{bmatrix} \in \mathbb{R}^K \longrightarrow \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_K \\ 1 \end{bmatrix} \in \mathbb{R}^{K+1}.$$

Using the homogeneous coordinates, identifying an arrangement of affine spaces in $\mathbb{R}^K$ is converted to identifying an arrangement of linear subspaces, also known as a central arrangement, in $\mathbb{R}^{K+1}$. However, identifying multiple linear subspaces is still a very difficult problem in its full generality. If the segmentation of the vectors $\{\boldsymbol{x}_i\}$ were known, the optimal subspace for each subset could be easily found using PCA; and conversely, if the subspaces were known, the vectors $\{\boldsymbol{x}_i\}$ (and hence the image) could be easily segmented into their closest subspaces. It seems that this is a "chicken-and-egg" problem if we do not know either the segmentation or the subspaces. Traditionally, this kind of problem is approached with one of many clustering methods developed in statistics or machine learning (e.g. Expectation Maximization [23], [38], K-means [39] or its variation K-subspaces [40]). Some of the methods have been applied to problems in image processing [19], [20]. However, these techniques are iterative and incremental in nature and therefore prone to converge to local minima if the initialization is far off. Moreover, these methods cannot be extended so easily to hybrid models whose components may have *different and unknown* dimensions. In [24], [25], it is shown that a more pertinent method for estimating a hybrid linear model is the so-called Generalized Principal Component Analysis (GPCA). This method does not require prior knowledge of the number and dimension of the subspaces and can simultaneously estimate multiple subspaces and segment the data into them, well suited for our purpose here.

A general solution to GPCA can be found in [25]. In this paper, since the true nature of natural images is unknown and

[8] An algebraic set is defined as a set of points that are the zeros of set of polynomial equations.

[9] This process is formally known in algebraic geometry as "projectivitization."

we only adopt an arrangement of subspaces to approximate the imagery data, we need to solve a special case of GPCA that can identify a hybrid linear structure for an image subject to a given MSE threshold ϵ^2. That is, unlike the algorithm given in [25], the special algorithm needs to automatically determine the smallest number of subspaces of lowest possible dimensions for a given image subject to the MSE threshold. We describe in the next few sections the development of such a customized GPCA algorithm for the image representation problem. But for the sake of completeness, we first give a brief introduction to the basic ideas of GPCA via a simplified case that will also be used later in the customized GPCA algorithm (see Algorithm 2) as well as in the extended multi-scale version (see Algorithm 3 and 4).

For simplicity, we will assume for now that N, the number of subspaces, is known. We will discuss how it can be determined later in the Section III-E. We also assume that all the subspaces are hyperplanes in the ambient space $\mathbb{R}^{K+1}$. That is, all subspaces have the same dimension $k_1 = \cdots = k_n = k = (K+1) - 1 = K$. Such a model may initially over-fit the data as the dimensions of some of the subspaces could be strictly smaller than K. Nevertheless, the grouping of the data points will in general be correct because it is a zero-measure event that any over-estimated hyperplane simultaneously contains more than one subspace. Once the grouping is obtained, we can easily determine the true dimension of each subspace by applying PCA to data points that belong to each group.

We start by noticing that every K-dimensional hyperplane $S \subset \mathbb{R}^{K+1}$ can be defined in terms of a nonzero *normal* vector $\boldsymbol{b} \in \mathbb{R}^{K+1}$ as follows:

$$S \doteq \big\{\boldsymbol{x} : \boldsymbol{b}^T\boldsymbol{x} \doteq b_1x_1 + b_2x_2 + \cdots + b_{K+1}x_{K+1} = 0\big\}. \quad (9)$$

Therefore, a point $\boldsymbol{x} \in \mathbb{R}^{K+1}$ lying on one of the hyperplanes $S_n, n = 1, \ldots, N$ must satisfy the formula:[10]

$$(\boldsymbol{b}_1^T\boldsymbol{x} = 0) \ \vee \ (\boldsymbol{b}_2^T\boldsymbol{x} = 0) \ \vee \cdots \vee \ (\boldsymbol{b}_n^T\boldsymbol{x} = 0). \quad (10)$$

In other words, the product of the n linear terms must be zero regardless which subspace $\boldsymbol{x}$ is in, i.e.,

$$p_N(\boldsymbol{x}) \doteq (\boldsymbol{b}_1^T\boldsymbol{x})(\boldsymbol{b}_2^T\boldsymbol{x})\cdots(\boldsymbol{b}_N^T\boldsymbol{x}) = 0. \quad (11)$$

The basic idea of GPCA is to find the polynomial(s) p_N and then retrieve information about the individual subspaces (or hyperplanes) from p_N.

Note that p_N is a homogeneous polynomial of degree n in $\boldsymbol{x}$ with real coefficients:

$$p_N(\boldsymbol{x}) = \sum c_{N_1,\ldots,N_{K+1}} x_1^{N_1}\cdots x_{K+1}^{N_{K+1}} = \nu_N(\boldsymbol{x})^T\boldsymbol{c}_N,$$

where $c_{N_1,N_2,\ldots,N_{K+1}} \in \mathbb{R}$ represents the coefficient of monomial $x_1^{N_1}x_2^{N_2}\cdots x_{K+1}^{N_{K+1}}$, $\boldsymbol{c}_N$ is the vector of all coefficients, and $\nu_N(\boldsymbol{x})$ is the stack of all possible monomials. ν_N is actually the Veronese map of degree N which is defined as follows.

Definition (Veronese Map). Given N and D, the *Veronese map* of degree N, $\nu_N : \mathbb{R}^D \to \mathbb{R}^{G_N(D)}$, is defined as: $\nu_N : [x_1, \ldots, x_D]^T \mapsto [\ldots, \boldsymbol{x}^{\boldsymbol{N}}, \ldots]^T$, where $\boldsymbol{x}^{\boldsymbol{N}}$ is a monomial of the form $x_1^{N_1}x_2^{N_2}\cdots x_D^{N_D}$ with $\boldsymbol{N} = (N_1, N_2, \ldots, N_D)$. The monomials $\boldsymbol{x}^{\boldsymbol{N}}$ in $\nu_N(\boldsymbol{x})$ are ordered in the degree-lexicographic order.

The number of linearly independent monomials is $G_N \doteq \binom{D+N-1}{D}$, hence $\boldsymbol{c}_N$ and $\nu_N(\boldsymbol{x})$ are vectors in $\mathbb{R}^{G_N}$.

Since the given set of samples $\{\boldsymbol{x}_i\}_{i=1}^M$ all satisfy the equation (11), we obtain the following system of linear equations on the vector of coefficients $\boldsymbol{c}_N$:

$$\boldsymbol{V}_N\,\boldsymbol{c}_N \doteq \begin{bmatrix} \nu_N(\boldsymbol{x}_1)^T \\ \nu_N(\boldsymbol{x}_2)^T \\ \vdots \\ \nu_N(\boldsymbol{x}_M)^T \end{bmatrix} \boldsymbol{c}_N = 0 \quad \in \mathbb{R}^M. \quad (12)$$

Because the number of hyperplanes N was known, we could immediately recover $\boldsymbol{c}_N$ as the eigenvector of $\boldsymbol{V}_N^T\boldsymbol{V}_N$ associated with its smallest eigenvalue.

Now let us consider the derivative of $p_N(\boldsymbol{x})$ evaluated at each $\boldsymbol{x}$. Suppose $\boldsymbol{x} \in S_j$, we have:

$$\nabla p_N(\boldsymbol{x}) = \frac{\partial p_N(\boldsymbol{x})}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}}\prod_{n=1}^{N}(\boldsymbol{b}_n^T\boldsymbol{x})$$
$$= \sum_{n=1}^{N}(\boldsymbol{b}_n)\prod_{\ell\neq n}(\boldsymbol{b}_\ell^T\boldsymbol{x}) = \boldsymbol{b}_j\prod_{\ell\neq j}(\boldsymbol{b}_\ell^T\boldsymbol{x}), \quad (13)$$

because $\prod_{\ell\neq n}(\boldsymbol{b}_\ell^T\boldsymbol{x}) = 0$ for $j \neq n$.[11] Thus, one can obtain a normal vector for each subspace,

$$\boldsymbol{b}_n = \frac{\nabla p_N(\boldsymbol{x}_j)}{|\nabla p_N(\boldsymbol{x}_j)|}, \quad \forall\ \boldsymbol{x}_j \in S_n. \quad (14)$$

After obtaining the normals of all the subspaces, all the data vectors $\{\boldsymbol{x}_i\}$ can be segmented into the subspaces to which they are the closest. Subsequently, for each group of data vectors, a linear model is identified using the method described in Section II-B. The correct dimension of each subspace is determined according to equation (5).[12]

We summarize in pseudocode the GPCA algorithm that we are using in this paper as Algorithm 1.[13]

Example 1 (Simulation of One Plane and Two Lines). Figure 4 shows an example of 5,000 points sampled from one plane and two lines in $\mathbb{R}^3$: 3,000 points from the plane

[10] Since the subspaces S_n are all different from each other, the normal vectors $\{\boldsymbol{b}_n\}_{n=1}^N$ are pairwise linearly independent.

[11] For noisy data, it is not exactly 0 but can be ignored because it is insignificant relative to the term with $j = n$.

[12] In identifying the linear model for each group, we do not need to use homogeneous coordinate anymore because the mean of the vectors is subtracted first in PCA.

[13] Be aware that the derivation and summary of the GPCA algorithm given in this paper is a gross simplification of the more complete version. The interested reader may refer to [24] for a more rigorous derivation and proofs.

Algorithm 1 (Subspace Segmentation – GPCA).

1: **function** $\{\phi_{n,j}, \boldsymbol{x}_{n,i}\} = \text{SubspaceSegmentation}(\{\boldsymbol{x}_i\}, N)$
2: Construct $\boldsymbol{V}_N$ from $\{\boldsymbol{x}_i\}$ in homogeneous coordinates;
3: Solve for $\boldsymbol{c}_N$ from $\boldsymbol{V}_N \boldsymbol{c}_N = 0$;
4: Set $p_N(\boldsymbol{x}) = \boldsymbol{c}_N^T \nu_N(\boldsymbol{x})$;
5: Determine the normal vectors of each subspace S_n as $\boldsymbol{b}_n = \frac{\nabla p_N(\boldsymbol{x}_j)}{|\nabla p_N(\boldsymbol{x}_j)|}, \quad \forall\, \boldsymbol{x}_j \in S_n$;
6: Assign $\boldsymbol{x}_i$ to the group n which minimizes $\|\boldsymbol{b}_n^T \boldsymbol{x}_i\|^2$.
7: Apply PCA to each group $\{\boldsymbol{x}_{n,i}\}$ and obtain the dimension and a basis $\{\phi_{n,j}\}$ for each subspace S_n.

and 1,000 points from each of the line. Each sample point is corrupted with 5% independent Gaussian noise. The results of Algorithm 1 applied to the data set are shown in Figure 4. The sample points are segmented correctly according to the three subspaces.

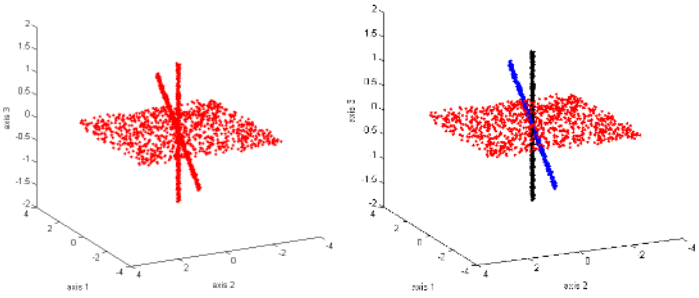

Fig. 4. An example of the subspace segmentation. Left: The sample points. Right: The sample points are segmented into 3 groups which are denoted by different colors.

As the above example shows, although Algorithm 1 is purely algebraic, each of its steps involves only numerically stable procedures and hence the algorithm is able to tolerate moderate amount of noise. In general, large noise and outliers pose significant challenges to GPCA. However, in our context, since we are seeking an approximation of the data set subject to a hard MSE threshold, we cannot simply throw away any subset of data as outliers and instead we need to either increase the number of subspaces or their dimensions in order to reduce the resulting fitting error. How to do so properly will be addressed in Section C-E.

C. Model Complexity of A Hybrid Linear Model

The model complexity, i.e., the total number of coefficients needed for representing the hybrid linear model $\{\phi_{n,j}, \hat{\boldsymbol{x}}_{n,i}\}$ is[14]

$$\begin{aligned} \Omega &= \Omega(M_1, k_1) + \cdots + \Omega(M_N, k_N) \\ &= \sum_{n=1}^{N} \big(M_n k_n + k_n(K - k_n + 1)\big). \end{aligned} \tag{15}$$

Notice that Ω is similar to the effective dimension (ED) of the hybrid linear representation defined in [41]. Thus, finding a representation that minimizes Ω is the same as minimizing the effective dimension of the imagery data set.[15]

Instead, if we model the union of all the vectors $\cup_{n=1}^{N} \{\boldsymbol{x}_{n,i}\}_{i=1}^{M_n}$ with a single subspace (subject to the same MSE), the dimension of the subspace in general needs to be $k = \min\{k_1 + \cdots + k_N, K\}$. It is easy to verify from the definition (6) that under reasonable conditions (e.g., N is bounded from being too large), we have

$$\Omega(M, k) \;>\; \Omega(M_1, k_1) + \cdots + \Omega(M_N, k_N). \tag{16}$$

Thus, if a hybrid linear model can be identified for an image, the resulting representation will in general be much more compressed than that with a single linear or affine subspace. This will also be verified by experiments on real images in subsequent sections.

D. Dimension Reduction via Projection.

The application of hybrid linear model to image representation has previously been explored in [28]. However, it misses two critical factors that prevent the final performance from being comparable to other competitive methods such as wavelets. The first factor is how to further reduce the negative effect of overhead by incorporating a pre-projection of the data; the second factor is to how to implement the hybrid linear model in a multi-scale fashion. We will discuss the first factor in the remainder of this section and leave the issues with multi-scale implementation to the next section.

In the complexity of the hybrid linear model (15), the first term is always smaller than that of the linear model (6) because $k_n \leq k$ for all n and $\sum_{n=1}^{N} M_n = M$. The second overhead term however can be larger than in that of the linear model (6) because the bases of multiple subspaces now must be stored. We here propose a method to further reduce the overhead by separating the estimation of the hybrid model into two steps.

In the first step, we may project the data vectors $\{\boldsymbol{x}_i\}$ onto a lower-dimensional subspace (e.g., via PCA) so as to reduce the dimension of the ambient space from K to K', with K' chosen to achieve an MSE $\frac{1}{2}\epsilon^2$. The data vectors in the lower ambient space $\mathbb{R}^{K'}$ are denoted as $\{\boldsymbol{x}_i'\}$. In the second step, we identify a hybrid linear model for $\{\boldsymbol{x}_i'\}$ within

[14] We also needs a very small number of binary bits to store the membership of the vectors. But those extra bits are insignificant comparing to Ω and often can be ignored.

[15] In fact, the minimal Ω is closely related to the Kolmogorov complexity or to the minimum description length (MDL) of the imagery data [41].

the lower-dimension ambient space $\mathbb{R}^{K'}$. In each subspace, we determine the dimension k_n subject to the MSE $\frac{1}{2}\epsilon^2$. Symbolically, the process is represented by the following diagram:

$$\{\boldsymbol{x}_i\} \subset \mathbb{R}^K \xrightarrow{\text{PCA}} \{\boldsymbol{x}'_i\} \subset \mathbb{R}^{K'} \xrightarrow{\text{GPCA}} \{\boldsymbol{x}'_i\} \subset \cup_{n=1}^N S'_n,$$

where $K' < K$. The two steps combined achieve an overall MSE ϵ^2, but they can actually reduce the total model complexity to

$$\Omega = \sum_{n=1}^{N} \big(M_n k_n + k_n(K - k_n + 1)\big) + K(K'+1). \quad (17)$$

This Ω will be smaller than the Ω in equation (15) because K' is smaller than K. The reduction of the ambient space will also make the identification of the hybrid linear model (say by GPCA) much more efficient.

E. Determining the Number of Subspaces

If the number of subspaces, N, is given, algorithms like GPCA or EM can always find a segmentation. The basis $\{\phi_{n,j}\}$ and dimension k_n of each subspace are determined by the desired MSE ϵ^2. As N increases, the dimension of the subspaces may decrease, but the overhead required to store the bases increases. The optimal N^* is then the one that minimizes Ω. From our experiments, we found that N^* is typically in the range from 2 to 6, especially in a multi-scale implementation that we will introduce next. Therefore, N^* can be easily identified by applying the algorithm repeatedly to different N's.

F. Algorithm for Estimating A Hybrid Linear Model

Algorithm 2 below describes in pseudocode how to estimate the hybrid linear model of an image $\boldsymbol{I}$, in which the *SubspaceSegmentation* function is implemented in this paper using the GPCA Algorithm 1. But it can also be implemented using some variations of EM or other subspace segmentation methods.[16]

Example 2 (A Hybrid Linear Model for the Gray-Scale Barbara Image). Figure 5 and Figure 6 show intuitively a hybrid linear model identified for the 8×8 blocks of the standard 512×512 gray-scale Barbara image. The total number of blocks is $N = 4,096$. The GPCA subspace segmentation algorithm identifies three subspaces for these blocks (for a given error tolerance), as shown in Figure 5. Figure 6 displays the three sets of bases for the three subspaces identified, respectively. It is worth noting that these bases are very consistent with the textures of the image blocks in the respective groups.

[16]However, by the time this paper is written, we are not aware of the existence of any such variation that provides a principled solution to the subspace segmentation problem while using as little prior knowledge on the subspaces as GPCA.

Algorithm 2 (Hybrid Linear Model Estimation).

function $\hat{\boldsymbol{I}}$ = HybridLinearModel($\boldsymbol{I}, \epsilon^2$)
$\{\boldsymbol{x}_i\}$ = StackImageIntoVectors($\boldsymbol{I}$);
$\{\boldsymbol{x}'_i\}, \{\phi_j\}, \{\alpha_i^j\}$ = PCA($\{\boldsymbol{x}_i - \bar{\boldsymbol{x}}\}, \frac{1}{2}\epsilon^2$);
for each possible N **do**
  $\{\boldsymbol{x}'_{n,i}\}$ =SubspaceSegmentation($\{\boldsymbol{x}'_i\}, N$);
  $\{\hat{\boldsymbol{x}'}_{n,i}\}, \{\phi_{n,j}\}, \{\alpha_{n,i}^j\}$ = PCA($\{\boldsymbol{x}'_{n,i} - \bar{\boldsymbol{x}}'_n\}, \frac{1}{2}\epsilon^2$);
  compute Ω_N;
end for
$\Omega_{opt} = \min(\Omega_N)$;
$\hat{\boldsymbol{I}}$ = UnstackVectorsIntoImage($\{\hat{\boldsymbol{x}'}_{n,i}\}$ with Ω_{opt});
output $\{\alpha_i^j\}, \{\phi_j\}, \bar{\boldsymbol{x}}, \{\alpha_{n,i}^j\}, \{\phi_{n,j}\}, \{\bar{\boldsymbol{x}}'_n\}$ with Ω_{opt};
return $\hat{\boldsymbol{I}}$.

Fig. 5. The segmentation of the 4,096 image blocks from the Barbara image. The image (left) is segmented into three groups (right three). Roughly speaking, the first subspace contains mostly image blocks with homogeneous textures; the second and third subspaces contain blocks with textures of different spatial orientations and frequencies.

IV. Multi-Scale Hybrid Linear Models in Spatial Domain

There are at least several reasons why the above hybrid linear model needs further improvement. Firstly, the hybrid linear model treats low frequency/entropy regions of the image in the same way as the high frequency/entropy regions, which is inefficient. Secondly, by treating all blocks the same, the hybrid linear model fails to exploit stronger correlations that typically exist among adjacent image blocks.[17] Finally, estimating the hybrid linear model is computationally expensive when the image is large. For example, we use 2 by 2 blocks, a 512 by 512 image will have $M = 65,536$ data vectors. Estimating a hybrid linear model for such a huge number of vectors is impossible for a PC. In this section, we introduce a multi-scale hybrid linear representation which is able to resolve the above issues.

[17]For instance, if we take all the b by b blocks and scramble them arbitrarily, the scrambled image would be fit equally well by the same hybrid linear model for the original image.

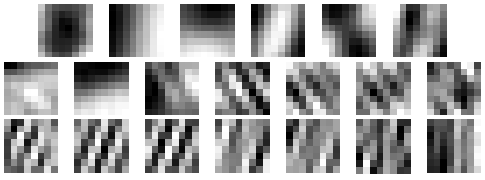

Fig. 6. The three sets of bases for the three subspaces (of blocks) shown in Figure 5, respectively. Each row is for one subspace and the number of base vectors (blocks) is the dimension of the subspace.

A. Laplacian Pyramid of Hybrid Linear Models

The basic ideas of multi-scale representations such as the Laplacian pyramid [30] have been exploited for image compression for decades (e.g., wavelets, subband coding). A multi-scale method will give a more compact representation because it encodes low frequency/entropy parts and high frequency/entropy parts separately. The low frequency/entropy parts are invariant after low-pass filtering and down-sampling, and can therefore be extracted from the much smaller down-sampled image. Only the high frequency/entropy parts need to be represented at a level of higher resolution. Furthermore, the stronger correlations among adjacent image blocks will be captured in the down-sampled images because every four images blocks are merged into one block in the down-sampled image. At each level, the number of imagery data vectors is one fourth of that at one level above. Thus, the computational cost can also be reduced.

We now introduce a multi-scale implementation of the hybrid linear model. We use the subscript l to indicate the level in the pyramid of down-sampled images.[18] The finest level (the original image) is indicated by $l = 0$. The larger is l, the coarser is the down-sampled image. We denote the highest level to be $l = L$.

Pyramid of Down-Sampled Images. First, the level-l image $\boldsymbol{I}_l$ passes a low-pass filter F_1 (averaging or Gaussian filter, etc) and is down-sampled by 2 to get a coarser version image $\boldsymbol{I}_{l+1}$:

$$\boldsymbol{I}_{l+1} \doteq \mathrm{F}_1(\boldsymbol{I}_l) \downarrow 2, \quad l = 0, \ldots, L-1. \tag{18}$$

The coarsest level-L image $\boldsymbol{I}_L$ is approximated by $\hat{\boldsymbol{I}}_L$ using a hybrid linear model with the MSE ϵ_L^2. The number of coefficients needed for the approximation is Ω_L.

Pyramid of Residual Images. At all other level-l, $l = 0, \ldots, L-1$, we do *not* need to approximate the down-sampled image $\boldsymbol{I}_l$ because it has been roughly approximated by the image at level-$(l+1)$ upsampled by 2. We only need to approximate the residual of this level, denoted as $\boldsymbol{I}'_l$:

$$\boldsymbol{I}'_l \doteq \boldsymbol{I}_l - \mathrm{F}_2(\hat{\boldsymbol{I}}_{l+1}) \uparrow 2, \quad l = 0, \ldots, L-1, \tag{19}$$

where the F_2 is an interpolation filter. Each of these residual images $\boldsymbol{I}'_l$, $l = 0, \ldots, L-1$ is approximated by $\hat{\boldsymbol{I}}'_l$ using a hybrid linear model with the MSE ϵ_l^2. The number of coefficients needed for the approximation is Ω_l, for each $l = 0, \ldots, L-1$.

Pyramid of Approximated Images. The approximated image at the level-l is denoted as $\hat{\boldsymbol{I}}_l$:

$$\hat{\boldsymbol{I}}_l \doteq \hat{\boldsymbol{I}}'_l + \mathrm{F}_2(\hat{\boldsymbol{I}}_{l+1}) \uparrow 2, \quad l = 0, \ldots, L-1. \tag{20}$$

The Figure 7 shows the structure of a three-level ($L = 2$) approximation of the image $\boldsymbol{I}$. Only the hybrid linear models for $\hat{\boldsymbol{I}}_2$, $\hat{\boldsymbol{I}}'_1$, and $\hat{\boldsymbol{I}}'_0$, which are approximation for $\boldsymbol{I}_2$, $\boldsymbol{I}'_1$, and $\boldsymbol{I}'_0$ respectively, are needed for the final representation of the image. Figure 8 shows the $\boldsymbol{I}_2$, $\boldsymbol{I}'_1$, and $\boldsymbol{I}'_0$ for the baboon image.

[18] This is not to be confused with the subscript n used to indicate different segments of an image.

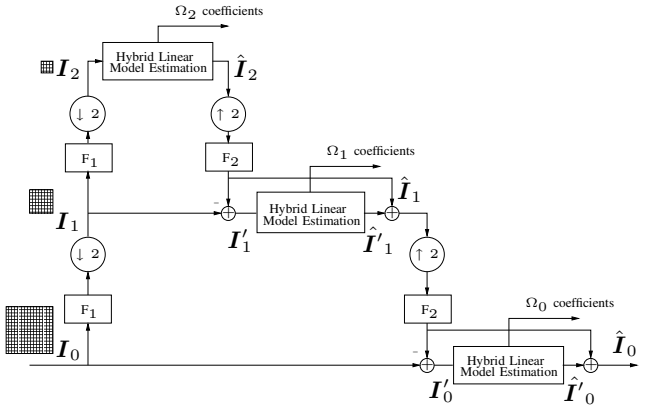

Fig. 7. Laplacian pyramid of the multi-scale hybrid linear model.

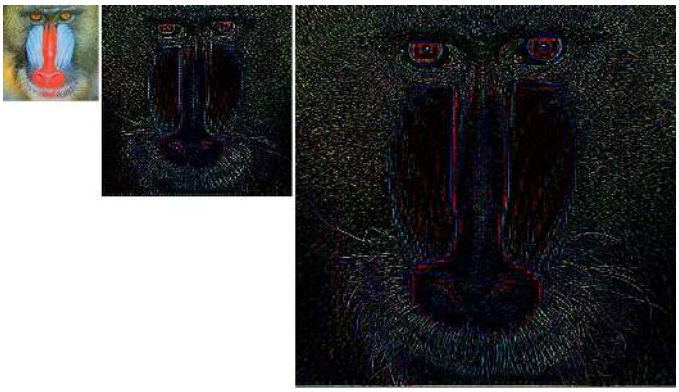

Fig. 8. Multi-scale representation of the Baboon image. Left: The coarsest level image $\boldsymbol{I}_2$. Middle: The residual image $\boldsymbol{I}'_1$. Right: The residual image $\boldsymbol{I}'_0$. The data at each level are modeled as the hybrid linear models. The contrast of the middle and right images has been adjusted so that they are visible.

The total number of coefficients needed for the representation will be

$$\Omega = \sum_{l=0}^{L} \Omega_l. \tag{21}$$

B. Constraints at Different Levels

MSE Threshold at Different Scale Levels. The MSE thresholds at different levels should be different but related because the up-sampling by 2 will enlarge 1 pixel at level-$(l+1)$ into 4 pixels at level-l. If the MSE of the level-$(l+1)$ is ϵ_{l+1}^2, the MSE of the level-l after the up-sampling will become $4\epsilon_{l+1}^2$. So the MSE thresholds of level-$(l+1)$ and level-l are related as

$$\epsilon_{l+1}^2 = \frac{1}{4}\epsilon_l^2, \quad l = 0, \ldots, L-1. \tag{22}$$

Usually, the user will only give the desired MSE for the approximation of original image which is ϵ^2. So we have

$$\epsilon_l^2 = \frac{1}{4^l}\epsilon^2, \quad l = 0, \ldots, L. \tag{23}$$

Vector Energy Constraint at Each Level. At each level-l, $l = 0, \ldots, L-1$, not all the vectors of the residual need to be approximated. We only need to approximate the (block) vectors $\{\boldsymbol{x}_i\}$ of the residual image $\boldsymbol{I}_l'$ that satisfy the following constraint:

$$\|\boldsymbol{x}_i\|^2 > \epsilon_l^2. \tag{24}$$

In practice, the energy of most of the residual vectors is close to zero. Only a small portion of the vectors at each level-l need to be modeled (e.g. Figure 9). This property

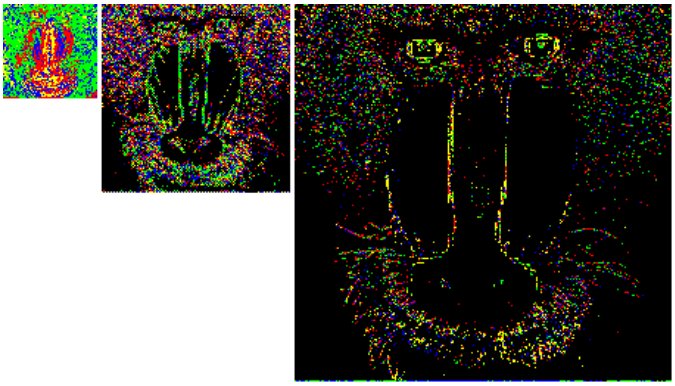

Fig. 9. The segmentation of (residual) vectors at the three levels—different subspaces are denoted by different colors. The black regions correspond to data vectors whose energy is below the MSE threshold ϵ_l^2 in equation (24).

of the multi-scale scheme not only significantly reduces the overall representation complexity Ω but also reduces the overall computational cost as the number of data vectors processed at each level is much less than those of the original image. In addition, for a single hybrid linear model, when the image size increases, the computational cost will increase in proportion to the square of the image size. In the multi-scale model, if the image size increases, we can correspondingly increase the number of levels and the complexity increases only linearly in proportion to the image size.

C. Algorithm of Estimating A Multi-scale Hybrid Linear Model

The overall process of estimating the multi-scale hybrid linear model can be written as the recursive pseudocode in Algorithm 3.

D. Experiments and Comparisons

Comparison of Different Lossy Representations. The first experiment is conducted on two standard images commonly used to compare image compression schemes: the 480×320 hill image and the 512×512 baboon image shown in Figure 10. We choose these two images because they are representative of two different types of images. The hill image contains large low frequency/entropy regions and the baboon image contains mostly high frenquency/entropy regions (see Section IV-E for more images). The size of the blocks b is chosen to be 2 and the level of the pyramid is 3 – we will test the effect of changing these parameters in subsequent experiments. In Figure 11, the results of our method are compared with several other commonly used image representations including DCT, PCA/KLT, single-scale hybrid linear model and Level-3 biorthogonal 4.4 wavelets (JPEG 2000). The x-axis of the figures is the ratio of coefficients (including the overhead) kept for the representation, which is defined as,

$$\eta = \frac{\Omega}{WHc}. \tag{25}$$

The y-axis is the PSNR of the approximated image defined in equation (3). The multi-scale hybrid linear model achieves the best PSNR among all the methods for both images. Figure 12 shows the two recovered images using the same amount of coefficients for the hybrid linear model and the wavelets. Notice that in the area around the whiskers of the baboon, the hybrid linear model preserves the detail of the textures better than the wavelets. However, the multi-scale hybrid

Algorithm 3 (Multi-Scale Hybrid Linear Model: Spatial Domain).

function $\hat{\boldsymbol{I}} = \text{MultiscaleModel}(\boldsymbol{I}, level, \epsilon^2)$
if $level <$ **MAXLEVEL then**
  $\boldsymbol{I}_{down} = \text{Downsample}(\text{F}_1(\boldsymbol{I}))$;
  $\hat{\boldsymbol{I}}_{nextlevel} = \text{MultiscaleModel}(\boldsymbol{I}_{down}, level+1, \frac{1}{4}\epsilon^2)$;
end if
if $level =$ **MAXLEVEL then**
  $\boldsymbol{I}' = \boldsymbol{I}$;
else
  $\boldsymbol{I}_{up} = \text{F}_2(\text{Upsample}(\hat{\boldsymbol{I}}_{nextlevel}))$;
  $\boldsymbol{I}' = \boldsymbol{I} - \boldsymbol{I}_{up}$;
end if
$\hat{\boldsymbol{I}}' = \text{HybridLinearModel}(\boldsymbol{I}', \epsilon^2)$;
return $\boldsymbol{I}_{up} + \hat{\boldsymbol{I}}'$.

Fig. 10. Testing images: the hill image (480×320) and the baboon image (512×512).

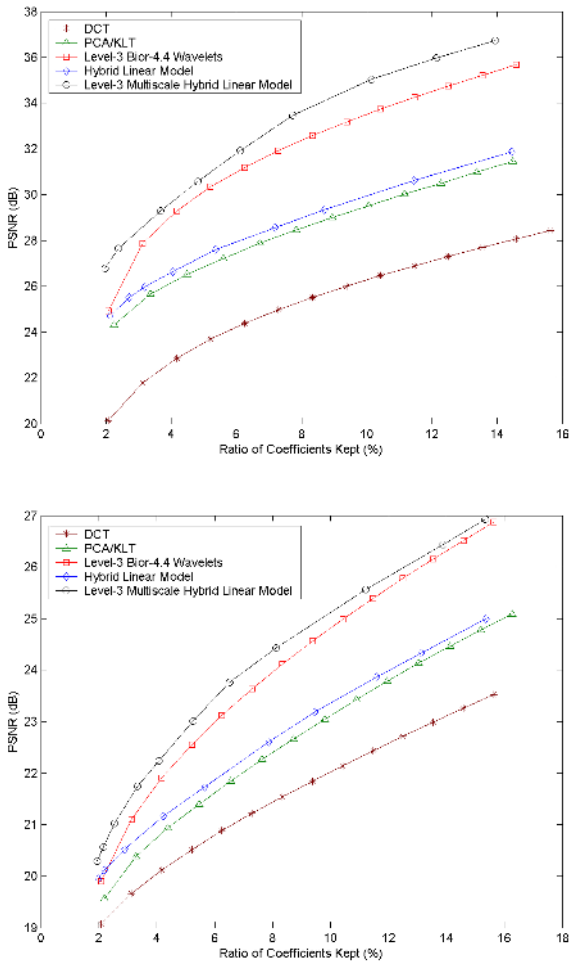

Fig. 11. Top: Comparison of several image representations for the hill image. Bottom: Comparison for the baboon image. The multi-scale hybrid linear model achieves the best PSNR among all the methods for both images.

linear model produces a slight block effect in the smooth regions.

Effect of the Number of Scale Levels. The second experiment shown in Figure 13 compares the multi-scale hybrid linear representation with wavelets for different number of levels. It is conducted on the hill and baboon image with 2 by 2 blocks. The performance increases while the number of levels is increased from 3 to 4. But if we keep increasing the number of levels to 5, the level-5 curves of both wavelets and our method (which are not shown in the figures) coincide with the level-4 curves. The performance cannot improve any more because the down-sampled images in the fifth level are so small that they cannot be further compressed. Only when the image is large, can we use more levels of down-sampling to achieve a more compressed representation.

Effect of the Window Size of the Blocks. The third experiment shown in Figure 14 compares the multi-scale

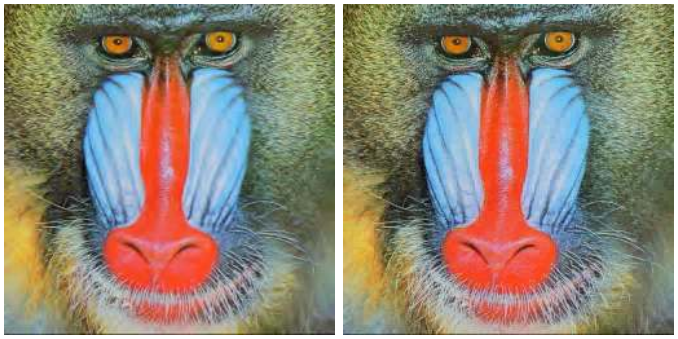

Fig. 12. Left: The baboon image recovered from level-3 biorthogonal 4.4 wavelets using 7.5% coefficients of the original image (PSNR=23.73). Right: The baboon image recovered from the level-3 multi-scale hybrid linear model in spatial domain using the same amount of coefficients. (PSNR=24.18).

hybrid linear models with different block sizes from 2×2 to 16×16. The dimension of the ambient space of the data vectors $\boldsymbol{x}$ ranges from 12 to 192 accordingly. The testing image is the baboon image and the number of down-sampling levels is 3. For large blocks, the number of data vectors is small but the dimension of the subspaces is large. So the overhead would be large and seriously degrade the performance. Also the block effect will be more obvious when the block size is large. This experiment shows that 2 is the optimal block size, which also happens to be compatible with the simplest down-sampling scheme.

E. Limitations

We have tested the multi-scale hybrid linear model on a wide range of images, with some representative ones shown in Figure 15.

From our experiments and experience, we observe that the multi-scale hybrid linear model is more suitable than wavelets for representing images with multiple high frequency/entropy regions, such as those with sharp 2-D edges and rich of textures. Wavelets are prone to blur sharp 2-D edges but better at representing low frequency/entropy regions. This probably explains why our model performs slightly worse than wavelets for the Lena and the monarch – the backgrounds of those two images are out of focus so that they do not contain much high frequency/entropy content.

Another limitation of our model is that it does not perform well on grayscale images. For a gray level image, the dimension K of a 2 by 2 block is only 4. Such a low dimension is not adequate for any further dimension reduction. If we use a larger block size, the block effect will also degrade the performance.

Unlike pre-fixed transformations such as wavelets, our method involves identifying the subspaces and their bases. Computationally, it is more costly. With unoptimized MATLAB codes, the overall model estimation takes 30 seconds to 3 minutes on a Pentium 4 1.8GHz PC depending on the

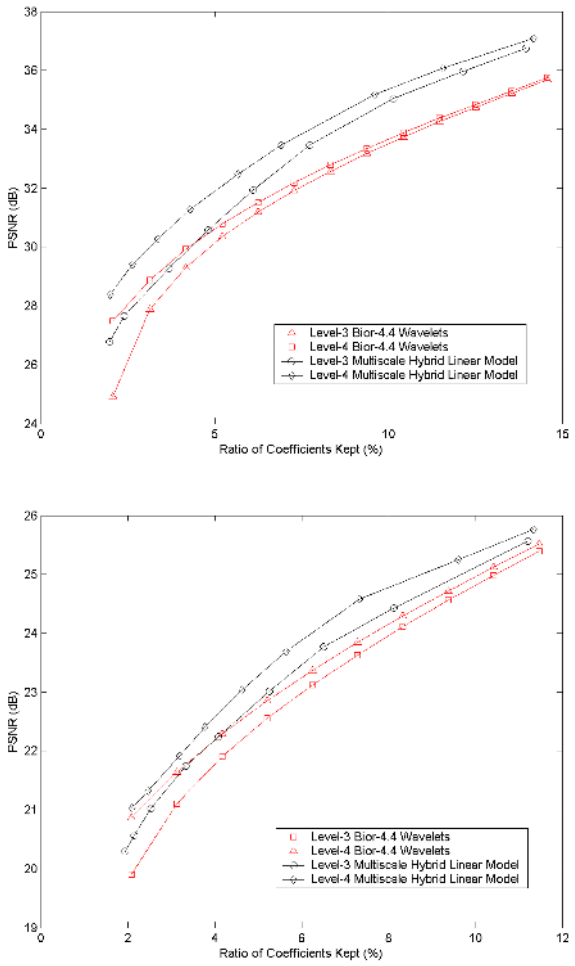

Fig. 13. Top: Comparison of the multi-scale hybrid linear model with wavelets for level-3 and level-4 for the hill image. Bottom: The same comparison for the baboon image. The performance increases while the number of levels increases from 3 to 4.

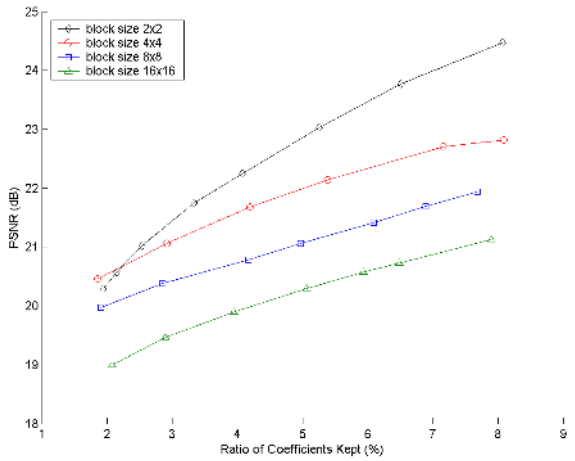

Fig. 14. Comparison of the multi-scale hybrid linear model with different block sizes: 16, 8, 4, 2. The performance increases while the size of blocks decreases.

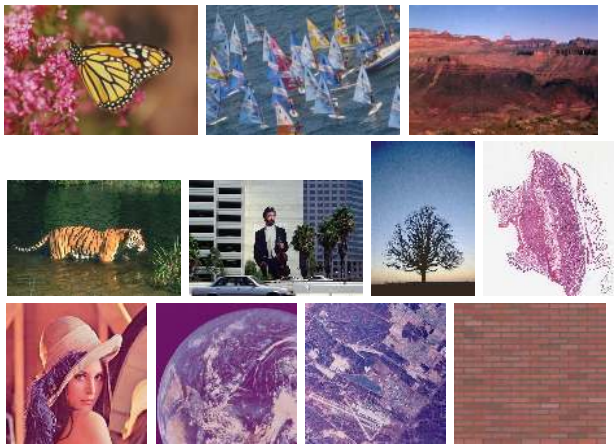

Fig. 15. A few standard testing images we used. From the top-left to the bottom-right: monarch (768×512), sail (768×512), canyon (752×512), tiger (480×320), street (480×320), tree (512×768), tissue (microscopic) (1408×1664), Lena (512×512), earth (satellite) (512×512), urban (aerial) (512×512), bricks (696×648). The multi-scale hybrid linear model outperforms wavelets except for the Lena and monarch images.

image size and the desired PSNR. The smaller the PSNR, the shorter the running time because the number of blocks needed to be coded in higher levels will be less.

V. Multi-scale Hybrid Linear Model in Wavelet Domain

From the discussion in the previous section, we have noticed that wavelets can achieve a better representation for smooth regions and avoid the block artifacts. Therefore, in this section, we will combine the hybrid linear model with the wavelet approach to build multi-scale hybrid linear models in the wavelet domain.

A. Imagery Data Vectors in Wavelet Domain

In the wavelet domain, an image is decomposed into an octave tree of subbands by certain separable wavelets. At each level, the LH, HL, HH subbands contain the information about high frequency edges and the LL subband is further decomposed into subbands at the next level. Figure 16 shows the octave tree structure of a level-2 wavelet decomposition. As shown in the Figure 17, the vectors $\{\boldsymbol{x}_i \in \mathbb{R}^K\}_{i=1}^M$ are constructed by stacking the corresponding wavelet coefficients in the LH, HL, HH subbands. The dimension of the vectors is $K = 3c$ because there are c color channels. One of the reasons for this choice of vectors is because for edges along the same direction, these coefficients are linearly related and reside in a lower dimensional subspace. To see this, let us first assume that the color along an edge is constant. If the edge is along the horizontal, vertical or

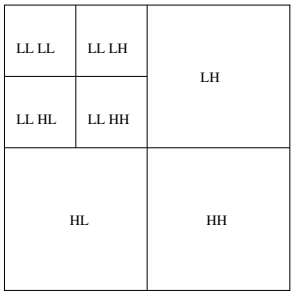

Fig. 16. The subbands of a level-2 wavelet decomposition.

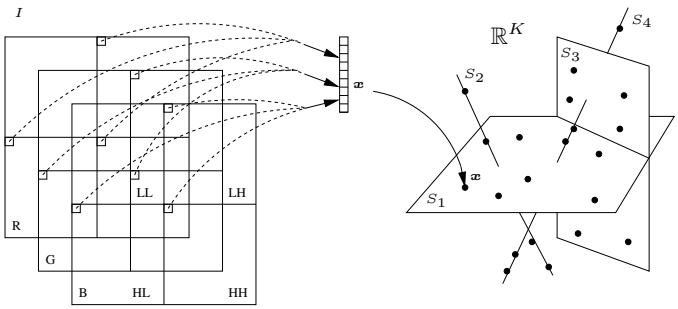

Fig. 17. The construction of imagery data vectors in the wavelet domain. These data vectors are assumed to reside in multiple (affine) subspaces which may have different dimensions.

diagonal direction, there will be an edge in the coefficients in the LH, HL, or HH subband, respectively. The other two subbands will be zero. So the dimension of the imagery data vectors associated with such an edge will be 1. If the edge is not exactly in one of these three directions, there will be an edge in the coefficients of all the three subbands. For example, if the direction of the edge is between the horizontal and diagonal, the amplitude of the coefficients in the LH and HH subbands will be large. The coefficients in the HL subband will be insignificant relative to the coefficients in the other two subbands. So the dimension of the data vectors associated with this edge is approximately 2 (subject to a small error ϵ^2). If the color along an edge is changing, the dimension the subspace will be higher but generally lower than the ordinal dimension $3c$. Notice that the above scheme is only one of many possible ways in which one may construct the imagery data vector in the wavelet domain. For instance, one may construct the vector using coefficients across different scales. It remains an open question whether such new constructions may lead to even more efficient representations than the one presented in this paper. We will leave this for future research.

B. Estimating A Multi-scale Hybrid Linear Model in Wavelet Domain

In the wavelet domain, there is no need to build a down-sampling pyramid. The multi-level wavelet decomposition already gives a multi-scale structure in the wavelet domain. For example, Figure 18 shows the octave three structure of a

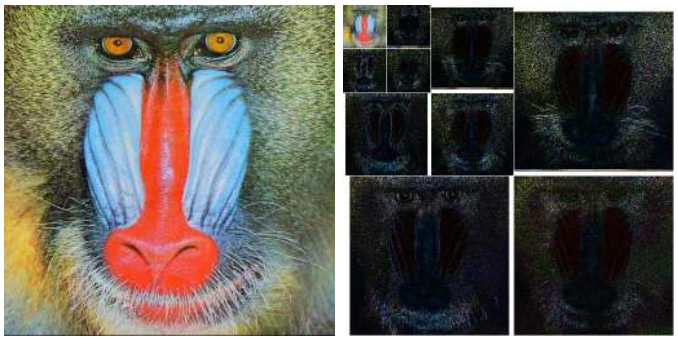

Fig. 18. The subbands of level-3 bior-4.4 wavelet decomposition of the baboon image.

level-3 bior-4.4 wavelet decomposition of the baboon image. At each level, we may construct the imagery data vectors in the wavelet domain according to the previous section. A hybrid linear model will be identified for the so-obtained vectors at each level. Figure 19 shows the segmentation results using the hybrid linear model at three scale levels for the baboon image.

Vector Energy Constraint at Each Level. In the nonlinear wavelet approximation, the coefficients which are below an error threshold will be ignored. Similarly in our model, not all the vectors of the imagery data vectors need to be modeled and approximated. We only need to approximate the (coefficient) vectors $\{\boldsymbol{x}_i\}$ that satisfy the following constraint:

$$\|\boldsymbol{x}_i\|^2 > \epsilon^2. \tag{26}$$

Notice that here we do not need to scale the error tolerance at different levels because the wavelet basis is orthonormal by construction. In practice, the energy of most of the vectors is close to zero. Only a small portion of the vectors at each level need to be modeled (e.g. Figure 19).

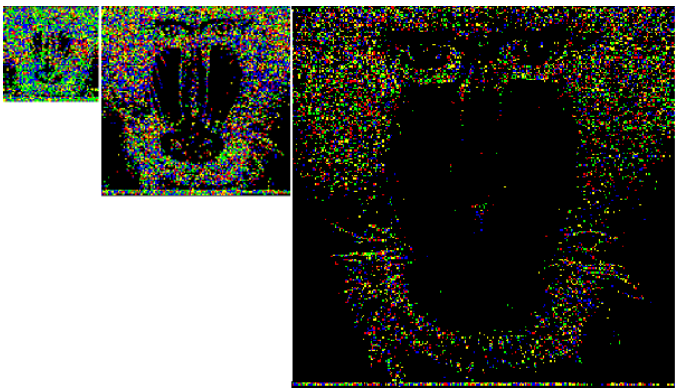

Fig. 19. The segmentation of data vectors constructed from the three subbands at each level—different subspaces are denoted by different colors. The black regions correspond to data vectors whose energy is below the MSE threshold ϵ^2 in equation (26).

C. Algorithm for Identifying A Multi-scale Hybrid Linear Model in Wavelet Domain

The overall process of estimating the multi-scale hybrid linear model in the wavelet domain can be summarized as the pseudocode in Algorithm 4.

Algorithm 4 (Multi-Scale Hybrid Linear Model: Wavelet Domain).

function $\hat{\boldsymbol{I}} = \text{MultiscaleModel}(\boldsymbol{I}, level, \epsilon^2)$
$\tilde{\boldsymbol{I}} = \text{WaveletTransform}(\boldsymbol{I}, level)$;
for each $level$ **do**
  $\hat{\tilde{\boldsymbol{I}}}_{level} = \text{HybridLinearModel}(\tilde{\boldsymbol{I}}_{level}, \epsilon^2)$;
end for
$\hat{\boldsymbol{I}} = \text{InverseWaveletTransform}(\hat{\tilde{\boldsymbol{I}}}, level)$;
return $\hat{\boldsymbol{I}}$.

D. Comparison with Other Lossy Representations

In this section, in order to obtain a fair comparison, the experimental setting is the same as that of the spatial domain in the previous section. The experiment is conducted on the same two standard images – the 480×320 hill image and the 512×512 baboon image shown in Figure 10.

The number of levels of the model is also chosen to be 3. In Figure 20, the results are compared with several other commonly used image representations including DCT, PCA/KLT, single-scale hybrid linear model and Level-3 biorthogonal 4.4 wavelets (JPEG 2000) as well as the multi-scale hybrid linear model in the spatial domain. The multi-scale hybrid linear model in the wavelet domain achieves better PSNR than that in the spatial domain. Figure 21 shows the three recovered images using the same amount of coefficients for wavelets, the hybrid linear model in the spatial domain, and that in the wavelet domain, respectively. Figure 22 shows the visual comparison with the enlarged bottom-right corners of the images in Figure 21.

Notice that in the area around the baboon's whiskers, the wavelets blur both the whiskers and the subtle details in the background. The multi-scale hybrid linear model (in the spatial domain) preserves the sharp edges around the whiskers but generates slight block artifacts in the relatively smooth background area. The multi-scale hybrid linear model in the wavelet domain successfully eliminates the block artifacts, keeps the sharp edges around the whiskers, and preserves more details than the wavelets in the background. Among the three methods, the multi-scale hybrid linear model in the wavelet domain achieves not only the highest PSNR, but also produces the best visual effect.

As we know from the previous section, the multi-scale hybrid linear model in the spatial domain performs slightly worse than the wavelets for the Lena and monarch images (Figure 15). Nevertheless, in the wavelet domain, the multi-scale hybrid linear model can generate very competitive results, as shown in Figure 23. The multi-scale hybrid linear

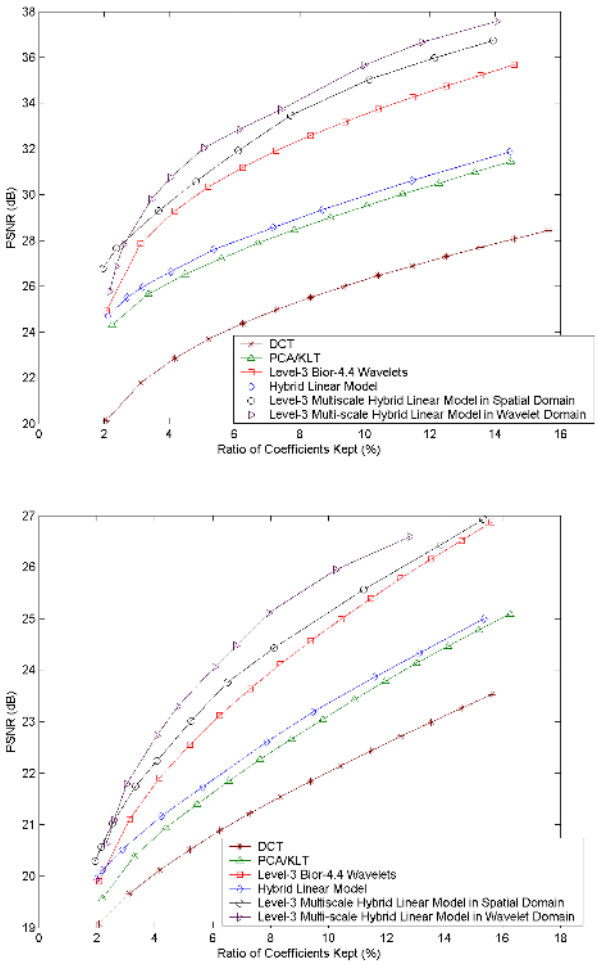

Fig. 20. Top: Comparison of several image representations for the hill image. Bottom: Comparison for the baboon image. The multi-scale hybrid linear model in the wavelet domain achieves better PSNR than that in the spatial domain.

model in the wavelet domain achieves better PSNR than the wavelets for the monarch image. For the Lena image, the comparison is mixed and merits further investigation.

Figure 24 demonstrates the average performance curves of the multi-scale hybrid linear models for all the images in Figure 15 both in spatial and wavelet domain. **The error bars on each average curve show the standard deviations of the performance curves for all the images. The lengths of the bars are scaled down by 0.1 to fit into the figure.** The hybrid linear models in spatial and wavelet domain both outperform wavelets. One interesting observation is that the hybrid linear model in spatial domain even outperforms the one in wavelet domain when the ratio of coefficients kept is high. One reason is that the block artifacts are largely reduced when the number of coefficients are large enough. Another reason is that the dimension of the ambient space in wavelet domain is lower than the one in spatial domain. So the room

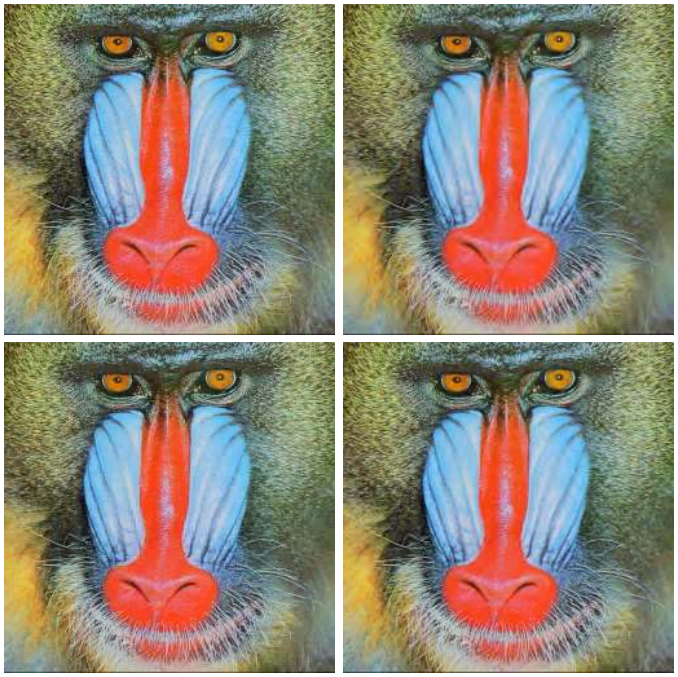

Fig. 21. Visual comparison of three representations for the baboon image approximated with 7.5% coefficients. Top-left: The original image. Top-right: The level-3 biorthogonal 4.4 wavelets (PSNR=23.73). Bottom-left: The level-3 multi-scale hybrid linear model in the spatial domain (PSNR=24.18). Bottom-right: The level-3 multi-scale hybrid linear model in the wavelet domain (PSNR=24.88).

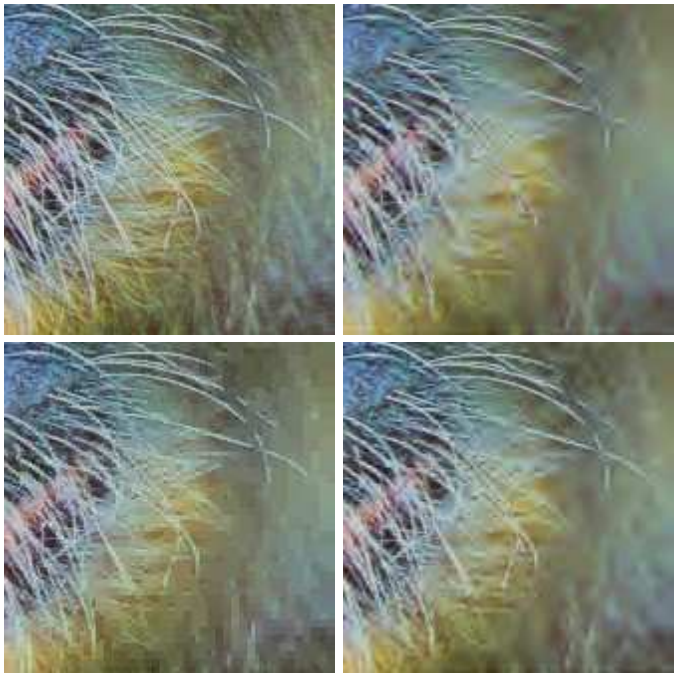

Fig. 22. Enlarged bottom-right corner of the images in Figure 21. Top-left: The original image. Top-right: The level-3 biorthogonal 4.4 wavelets. Bottom-left: The level-3 multi-scale hybrid linear model in the spatial domain. Bottom-right: the level-3 multi-scale hybrid linear model in the wavelet domain.

for dimension reduction will be very small in wavelet domain if the the ratio of coefficients kept is high. Hence, for a particular image, whether spatial or wavelet domain will offer better representation depends on the statistics of this image and the required PSNR.

E. Limitations

The above hybrid linear model (in the wavelet domain) does not produce so competitive results for gray-scale images as the dimension of the vector is merely 3 and there is little room for further reduction. For gray-scale images, one may have to choose a slightly larger window in the wavelet domain or to construct the vector using wavelet coefficients across different scales. A thorough investigation of all the possible cases is beyond the scope of this paper. The purpose of this paper is to demonstrate (using arguably the simplest cases) the vast potential of a new spectrum of image representations suggested by combining subspace methods with conventional image representation/approximation schemes. The quest for the more efficient and more compact representations for natural images without doubt will continue as our understanding of different classes of mathematical models improves while the nature of natural images remains largely a mystery.

VI. Image Segmentation

As we now know, the hybrid linear model uses different subspaces to model different regions of an image. It essentially gives a rough "segmentation" of the image. Pixels with a similar color or texture profile are likely grouped into the same subspace. It is well-known that image segmentation is an important problem in computer vision and image processing as it is the first step towards many important high-level tasks such as image understanding and object recognition. Many methods have been proposed in the literature for segmenting images based on different criteria or technical tools. The concepts and techncial tools (e.g., the GPCA Algorithm 1) introduced in this paper can potentially be used for image segmentation.

A. Hybrid Linear Models for Image Segmentation

Notice that for image representation, we normally divide the image $\boldsymbol{I}$ into *non-overlapping* blocks in the spatial domain (see the Section II-A). The hybrid linear model fit to the block vectors $\{\boldsymbol{x}_i\}$ essentially gives some kind of a segmentation of the image – pixels that belong to blocks in the same subspace are grouped into one segment. However, such a segmentation has a few undesirable features. If we choose a very large block size, then there will be severe "block effect" in the resulting segmentation, as all b^2 pixels in a block are assigned into the same segment (see Figure 5).

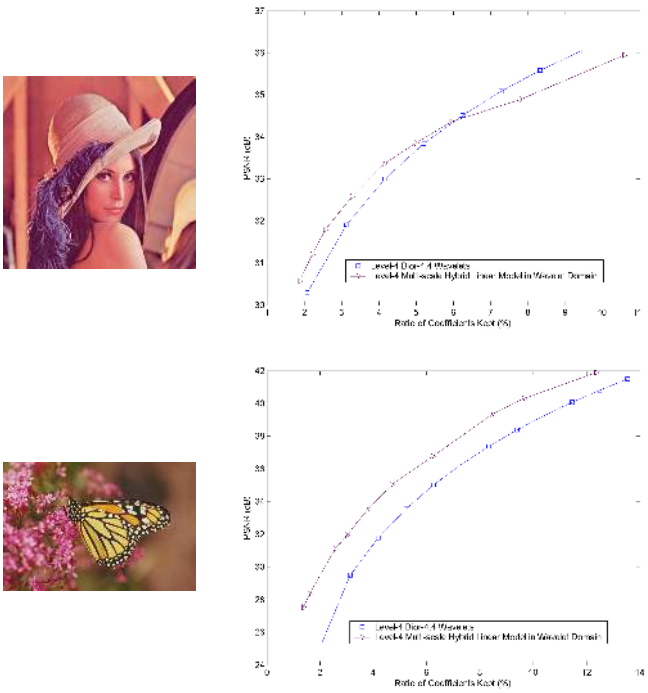

Fig. 23. Top: Comparison of multi-scale hybrid linear model in the wavelet domain with wavelets for the Lena image. Bottom: Comparison of multi-scale hybrid linear model in the wavelet domain with wavelets for the Monarch image. The multi-scale hybrid linear model in the wavelet domain achieves better PSNR than wavelets for a wide range of PSNR for these two images.

If we choose a smaller block size to reduce the block effect, then the block might not contain sufficient neighboring pixels that allow us to reliably extract the local texture.[19] Thus, the resulting segmentation will be determined primarily by the color of the pixels (in each small block) but not the texture.

One way to resolve the above problems is to choose a block of a reasonable size around *each* pixel and view the block as a (vector-valued) "label" or "feature" attached to the pixel. In many existing image segmentation methods, the feature (vector) is chosen instead to be the outputs of the block passing through a (pre-fixed) bank of filters (e.g., the Gabor filters). That is, the feature is the block transformed by a set of pre-fixed linear transformations. Subsequently, the image is segmented by grouping pixels that have "similar" features.

From the lessons that we have learned from image representation in the previous sections, we observe that the hybrid linear model may be adopted to facilitate this approach in several ways. First, we can directly fit a hybrid linear model to the un-transformed and un-processed block vectors, without having to choose beforehand which filter bank to

[19]Notice that a smaller block size is ok for compression as long as it can reduce the overhead and subsequently improve the overall compression ratio.

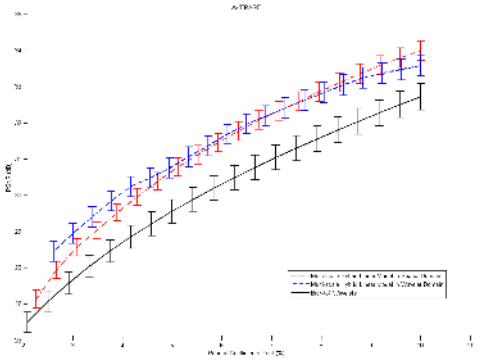

Fig. 24. The average performance curves of the multi-scale hybrid linear models for all the images in Figure 15 both in spatial and wavelet domain. **The error bars on each average curve show the standard deviations (scaled by 0.1) of the performance curves for all the images.**

use. The hybrid linear model essentially chooses the linear transformations (or filters) adaptively for different images and different image segments. Second, once the hybrid linear model is identified, there is no further need of introducing a similarity measure for the features. Feature vectors (and hence pixels) that belong to the same subspace are naturally grouped into the same image segment.

B. Dimension and Size Reduction

The problem of identifying a hybrid linear model for image segmentation is mathematically equivalent to that for image representation. However, two things have changed from image representation and make image segmentation a computationally much more challenging problem. First, the number of feature vectors (or blocks) is now always the same as the number of pixels: $M = WH$, which is larger than that ($M = WH/b^2$) in the case of image representation. For a typical 512×512 image, we have $M = 262,144$. Second, the block size b now can be much larger that in the case of image representation. Thus, the dimension of the block vector $K = b^2c$ is much higher. For instance, if we choose $b = 10$ and $c = 3$, then $K = 300$. It is impossible to implement the GPCA algorithm on a regular PC for $262,144$ vectors in $\mathbb{R}^{300}$, even if we are looking for up to only four or five subspaces.[20]

Dimension Reduction via Projection. To reduce dimension of the data, we rely on the assumption (or belief) that "the feature vectors lie on very low-dimensional subspaces in the high-dimensional ambient space $\mathbb{R}^K$." Then based on our discussion in Section III-D, we can project the data into a lower-dimensional space while still being able to preserve the separation of the subspaces. Principal component analysis (PCA) can be recruited for this purpose as the energy of the

[20]The dimension of the Veronese embedding of degree 5 will be in the order of 10^{10}.

feature vectors is mostly preserved by their first few principal components. From our experience, in practice it typically suffices to keep only the first ten principal components. Symbolically, the process is represented by the following diagram:

$$\{\boldsymbol{x}_i\} \subset \mathbb{R}^K \xrightarrow{\text{PCA}} \{\boldsymbol{x}'_i\} \subset \mathbb{R}^{K'} \xrightarrow{\text{GPCA}} \{\boldsymbol{x}'_i\} \subset \cup_{n=1}^N S'_n,$$

where $K' \ll K$ and $i = 1, \ldots, M = WH$.

Data Size Reduction via Down-Sampling. Notice that the number of feature vectors $M = WH$ might be too large for all the data to be processed together since a regular PC has trouble in performing singular value decomposition (SVD) for tens of thousands of vectors.[21] Thus, we have to down sample the data set and identify a hybrid linear model for only a subset of the data. The exact down-sampling scheme can be determined by the user. One can use periodic down-sampling (e.g., every other pixel) or random down-sampling. From our experience, we found periodic down-sampling often gives visually better segmentation results. The size of the down-sampled subset can be determined by the memory and speed of the computer the user has. Once the hybrid linear model is obtained, we may assign the remaining vectors to their closest subspaces. Of course, in practice, one may run the process on multiple subsets of the data and choose the one which gives the smallest fitting error for all the data. This is very much in the same spirit as the random sample consensus (RANSAC) method. Symbolically, the process is represented by the following diagram:

$$\{\boldsymbol{x}_i\} \xrightarrow{\text{sample}} \{\boldsymbol{x}_{i'}\} \subset \mathbb{R}^K \xrightarrow{\text{PCA}} \{\boldsymbol{x}'_{i'}\} \subset \mathbb{R}^{K'}$$
$$\xrightarrow{\text{GPCA}} \{\boldsymbol{x}_{i'}\} \subset \cup_{n=1}^N S_n \xrightarrow{\min d(\boldsymbol{x}_i, S_j)} \{\boldsymbol{x}_i\} \subset \cup_{n=1}^N S_n,$$

where $\{\boldsymbol{x}_{i'}\}$ is a (down-sampled) subset of $\{\boldsymbol{x}_i\}$.

C. Experiments

Figure 25 shows the results of applying the above schemes to the segmentation of some images from the Berkeley image database. The dimension of the subspace (in homogeneous coordinates) associated with each segment is marked by the number to its right. A $20 \times 20 \times 3$ "feature" vector is associated with each pixel that corresponds to the color values in a 20×20 block. We first apply PCA to project all the feature vectors onto a 6-dimensional subspace. We then apply the GPCA algorithm to further identify subspace-structures of the features in this 6-dimensional space and to segment the pixels to each subspace. The algorithm happens to find three segments for all the images shown below. Different choices in the error tolerance, window size, and color space (HSV or RGB) may affect the segmentation results. Empirically, we find that HSV gives visually better segments for most images. Figure 26 shows additional results of some more challenging images from the Berkeley image database.

[21] With the increase of memory and speed of modern computers, we hope this step will soon become unnecessary.

Fig. 25. Image segmentation results obtained from the hybrid linear model. The dimension of the subspace (in homogeneous coordinates) associated with each segment is given by the number to its right.

VII. Conclusions and Open Issues

In this paper, we have introduced a simple but effective class of mathematical models for representing and approximating natural images, namely the multi-scale hybrid linear models. This class of models can be efficiently computed from the given imagery data. The resulting model obtained for an image can effectively capture the heterogeneous and hierarchical structures of the image as well as harness the correlation among multiple color channels. Experiments show that the performance of the proposed method is comparable to or even better than most image representation schemes (including wavelets). In addition, this new method actually complements the existing schemes and together they achieve even higher performance.

As the proposed method is adaptive in nature, it is computationally more costly than methods that are based on pre-chosen (linear) transformations. We will investigate in the future how to reduce the complexity and develop more efficient implementations. We would also like to investigate optimal quantization and entropy coding schemes for hybrid linear models so as to develop a full package for compressing natural images.

There might be more efficient schemes to combine the hybrid linear models with wavelets and/or other x-lets. The scheme introduced in Section V is only one of the most straightforward options (that demonstrates the potential). A more systematic study of the approximation theory of vector-valued functions may lead to a principled (and therefore optimal) solution for the combination.

The proposed method is not limited to only representing

Fig. 26. Additional results of some more challenging images from the Berkeley image database.

(color) images. It is a general method for modeling any high-dimensional data that have heterogeneous internal structures. In this paper, we have demonstrated how the method can be extended to image segmentation – an application that is closely related to the proposed image representation scheme. However, the segmentation algorithm in this paper is only based on a single scale. More investigation about the consensus of segmentation results across different scales will improve the segmentation performance. In addition, our ongoing work has also shown that one can apply the same method to a collection of images that belongs to certain category, rather than a single image, and learn a hybrid linear model for the entire image category. The so-obtained models can potentially capture the essential statistical characteristics of different image categories and hence be useful for purposes such as image classification and retrieval. In the future, we will also investigate the potential of hybrid linear models for modeling other imagery data (e.g., MRI, DTI, hyper-spectral, etc.), as well as other data types (e.g., audio, video, gene expression data, etc.).

ACKNOWLEDGMENT

The authors would like to thank Professor Rene Vidal of the Biomedical Engineering Department of Johns Hopkins University and Professor Robert Fossum of the Mathematics Department of the University of Illinois for the joint work on subspace arrangements. We also thank Professor Minh Do of the Electrical and Computer Engineering Department of the University of Illinois for inspiring discussions on how to apply hybrid linear models in the wavelet (or curvelet) domain.

REFERENCES

[1] Y. Fisher, *Fractal Image Compression: Theory and Application*. Springer-Verlag Telos, 1995.
[2] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM. Special issue on digital multimedia systems*, vol. 34, no. 4, pp. 30–44, 1991.
[3] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 719–746, 1992.
[4] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice-Hall, 1995.
[5] D. L. Donoho, M. Vetterli, R. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2435–2476, 1998.
[6] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. Academic Press, 1999.
[7] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445–3463, 1993.
[8] R. DeVore, "Nonlinear approximation," *Acta Numer.*, vol. 7, pp. 51–150, 1998.
[9] D. L. Donoho, "Wedgelets: nearly-minimax estimation of edges," *Ann. Statist.*, vol. 27, pp. 859–897, 1999.
[10] E. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with smooth singularities," *Technical Report, Stanford University*, 2002.
[11] M. N. Do and M. Vetterli, "Contourlets: A directional multiresolution image representation," in *IEEE International Conference on Image Processing*, 2002.
[12] E. LePennec and S. Mallat, "Sparse geometric image representation with bandelets," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 423–438, 2005.
[13] M. Effros and P. Chou., "Weighted universal transform coding: Universal image compression with the Karhunen-Loéve transform," in *IEEE International Conference on Image Processing*, vol. 2, 1995, pp. 61–64.
[14] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
[15] C. Guo, S. Zhu, and Y. Wu, "A mathematical theory of primal sketch and sketchability." in *IEEE International Conference on Computer Vision*, 2003.
[16] S. C. Zhu, Y. N. Wu, and D. Mumford, "FRAME: Filters, random field and maximum entropy: — towards a unified theory for texture modeling," *International Journal on Computer Vision*, vol. 27, no. 2, pp. 1–20, 1998.
[17] Y. N. Wu, S. C. Zhu, and X. W. Liu, "Equivalence of Julesz ensemble and FRAME models," *International Journal on Computer Vision*, vol. 38, no. 3, pp. 247–265, 2000.
[18] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden markov models," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1056–1068, 2001.
[19] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.
[20] T. Lei and K. Udupa, "Performance evaluation of finite normal mixture model-based image segmentation techniques," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1153–69, 2003.
[21] M. S. Lewicki, "Learning optimal codes for natural images and sounds," in *Proc. SPIE Vol. 4119, p. 185-199, Wavelet Applications in Signal and Image Processing VIII, Akram Aldroubi; Andrew F. Laine; Michael A. Unser; Eds.*, Dec. 2000, pp. 185–199.
[22] A. Hyvärinen and M. Inki, "Estimating overcomplete independent component bases for image windows," *Journal of Mathematical Imaging and Vision*, vol. 17, pp. 139–152, 2002.
[23] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm (with discussion)," *J. R. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.
[24] R. Vidal, Y. Ma, and J. Piazzi, "A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials," in *IEEE*

International Conference on Computer Vision and Pattern Recognition, vol. I, 2004, pp. 510–517.

[25] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, 2005.

[26] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[27] D. Donoho, "For most large underdetermined systems of linear equations, the minimal l^1-norm solution is also the sparsest solution," *Stanford Technical Report*, 2004.

[28] K. Huang, A. Y. Yang, and Y. Ma, "Sparse representation of images with hybrid linear models," in *IEEE International Conference on Image Processing*, 2004.

[29] P. Scheunders, "A multivalued image wavelet representation based on multiscale fundamental forms," *IEEE Transactions on Image Processing*, vol. 11, no. 5, pp. 568– 575, 2002.

[30] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.

[31] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh and Dublin Philosphical Magazine and Journal of Science*, vol. 2, pp. 559–572, 1901.

[32] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.

[33] I. Jollife, *Principal Component Analysis*, 2nd ed. Springer-Verlag, 2002.

[34] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218, 1936.

[35] L. Hubert, J. Meulman, and W. Heiser, "Two purposes for matrix factorization: A historical appraisal," *SIAM Review*, vol. 42, no. 1, pp. 68–82, 2000.

[36] A. Bjorner, I. Peeva, and J. Sidman, "Subspace arrangements defined by products of linear forms," *Journal of the London Math. Society*, Preprint 2003.

[37] A. Bjorner, "Subspace arrangements," *First European Congress of Mathematics, (Paris, 1992)*, vol. I, no. 119, pp. 321–370, 1994.

[38] R. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," *Learning in Graphical Models*, 1998.

[39] R. Jancey, "Multidimensional group analysis," *Austral. J. Botany*, vol. 14, pp. 127–130, 1966.

[40] J. Ho, M. H. Yang, J. Lim, K. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions." in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2003.

[41] K. Huang, Y. Ma, and R. Vidal, "Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.