PERGAMON

Pattern Recognition 34 (2001) 1765–1784

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Image approximation and modeling via least statistically dependent bases

Naoki Saito*

Department of Mathematics, University of California, Davis, CA 95616, USA

Received 21 October 1999; accepted 29 June 2000

Abstract

Statistical independence is one of the most desirable properties of a coordinate system for representing and modeling images. In reality, however, truly independent coordinates may not exist for a given set of images, or it may be too difficult to compute them in practice. Therefore, we propose a new method to rapidly compute the *least statistically dependent basis* (LSDB) from a *basis dictionary* (e.g., the local cosine or wavelet packet dictionaries) containing a huge number of orthonormal (or biorthogonal) bases. Our new basis selection criterion is minimization of the mutual information of the distributions of the basis coefficients as a measure of statistical dependence, which in turn is equivalent to minimization of the sum of the differential entropy of each coordinate in the basis dictionary. In this sense, we can view this LSDB algorithm as the best-basis version of the Independent Component Analysis (ICA), which is increasingly gaining popularity. This criterion is different from that of the Joint Best Basis (JBB) proposed by Wickerhauser, which can be viewed as the best-basis version of the Karhunen–Loève basis (KLB). We demonstrate the usefulness of the LSDB for image approximation and modeling and compare its performance with that of KLB and JBB using a collection of real geophysical acoustic waveforms and an image database of human faces.

Keywords: Statistical independence; Karhunen–Loève expansion; Principal component analysis; Independent component analysis; Dimension reduction; Best basis; Wavelet packets; Local cosine transform; Image approximation; Image modeling

1. Introduction

Suppose we are given a set of similar images such as human faces (or a set of finger prints or a set of mammograms) and we want to *learn* the characteristics of those images, i.e., to represent or approximate them efficiently, analyze certain features, and build a stochastic model that can generate new images that are similar to those given images. What should we do, then? The best possible scenario would be to find a *statistically independent* coordinate system (basis) of that class of images. With this coordinate system we could achieve optimal compression of the images in that class by transmitting each coordinate (feature) separately using quantization scheme depending on the statistics of each coordinate. Moreover, a complete probabilistic description of an image class would be made possible by simply characterizing the probability distributions of each coordinate. We could *sample* or *simulate* as many new images from this stochastic model as we want so that we can examine variability of images in this class and how they look like. This would be a great tool for image diagnostics. In reality, however, it may not be possible to obtain truly independent coordinates because (1) the data may not be composed of truly independent features in the first place, and (2) even if the images consist of independent features, it may be too difficult to construct a feasible algorithm to extract such features because of the high dimensionality of the problem (imagine a large database consisting of

* Tel.: +1-530-754-2121; fax: +1-530-752-6635.
E-mail address: saito@math.ucdavis.edu (N. Saito).

0031-3203/01/$20.00
PII: S0031-3203(00)00116-3

512×512 pixel images). Therefore, it makes sense to devise an algorithm to rapidly compute a good coordinate system which is "closest" to the statistically independent one, and to examine how much we can achieve in approximation and stochastic modeling using such coordinates by assuming that they are truly independent.

In fact, the importance of the independent coordinates has long been recognized by several researchers in the various fields including statistics, signal and image processing, and pattern recognition. In the seminal paper by Watanabe [1] about the Karhunen–Loève (KL) expansion — also known as Principal Component Analysis (PCA) — and its application to pattern recognition, he argued the justification of the use of the KL coordinates for "feature compression" as follows:

> It would be desirable, from the viewpoint that information compression means elimination of redundancy, to use variables which are statistically independent, but in the absence of such variables, statistically uncorrelated variables may be the next best.

Then, he went on to show that the KL basis (KLB) is the *minimum entropy basis* among all the orthonormal bases in $\mathbb{R}^n$, where n is a number of pixels in images under consideration. This was a great achievement around 1965, and in fact, KLB was probably the best available feature extraction tool around that time. However, KLB-PCA only provides the *decorrelated* coordinates, and only takes care of the second-order statistics. Of course, if the underlying data obeys the multivariate Gaussian distribution, decorrelation implies independence. But in general, the natural images such as faces are far from Gaussian (see, e.g., [2]). Moreover, KLB-PCA has other drawbacks such as high computational cost and inaccuracy of sample estimate of covariance matrices which will be described in detail in Section 2.

More recently, the concept called Independent Component Analysis (ICA) has become popular, in particular, in the field of signal processing [3] and computational neuroscience [4]. The ICA incorporates higher-order statistics than KLB-PCA; it tries to obtain the statistically independent coordinate system more directly than KLB-PCA. It is very difficult, however, to compute it numerically, in particular for high-dimensional data, since they rely on the higher-order cumulants.

Thirty years since Watanabe's work has changed the landscape. We have now a *library* of local bases, which consists of various *dictionaries* of bases such as wavelet packet bases and local cosine bases, at our disposal as feature extraction tools. These are adaptable and flexible set of bases that can be tailored to one's needs very efficiently. They have been increasingly popular in various feature extraction business such as denoising [5–7], classification and regression [8–10]. The author and his colleagues, in particular, R.R. Coifman and M.V. Wickerhauser, have been advocating the use of the so-called "best basis paradigm" consisting of the following three steps: (1) Select a best possible basis from a dictionary or library of bases by optimizing a certain functional that quickly evaluates the efficacy of each basis in the dictionary/library for the problem at hand; (2) Discard the unimportant coordinates from the selected basis; and (3) Use the survived coordinates to solve the problem. Depending on the problem at hand, we need to use a different efficacy measure for the basis evaluation, and it is of critical importance to choose an appropriate measure.

Wickerhauser proposed the so-called "joint best basis" (JBB) with which he tried to alleviate some of the drawbacks of the KLB-PCA [11]. Independently from Watanabe, he proposed to find a basis from a dictionary that minimizes entropy of the energy distribution over its coordinates. Watanabe's argument is that a KLB is the best basis over all possible orthonormal bases of $\mathbb{R}^n$ with respect to the minimum entropy criterion whereas Wickerhauser's algorithm can quickly compute an approximate KLB that is the best basis over all bases selectable from the dictionary or library of orthonormal bases with the same criterion. Thus, the JBB corresponds to the KLB-PCA, but not to the ICA: it does not address the statistical independence of the coordinates explicitly.

In this paper, we propose yet another best basis aiming more directly to the statistical independence than KLB and JBB. Since there is no guarantee that the images under consideration consist of truly independent coordinates, a compromised but efficient strategy is to extract a basis whose coordinates are *least statistically dependent* from the dictionary or library of bases. We call this basis the *least statistically dependent basis* (LSDB).

This paper is organized as follows. In Section 2, we set up our notation and briefly review the KLB-PCA, ICA, and JBB using our notation. In Section 3, we consider a measure of statistical dependence of a given basis and propose the LSDB algorithm. In Section 4, we apply LSDB to an important problem of signal and image approximation and compare our method with KLB and JBB using the geophysical acoustic waveforms and the human face images. Then in Section 5, we consider how to build a stochastic models given a collection of similar signals or images. We propose a few simple models using the LSDB coordinates. We end this paper with discussion of the relation of the LSDB to the other methods and describe some of our ongoing and future work in Section 6.

2. Feature extraction and basis search

Let $\mathscr{X} \in \mathbb{R}^n$ be an input image space, i.e., a set of all images of a particular class under consideration, where

n is a number of pixels in each image. Suppose we are given N training (sample) images, $\mathscr{T} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathscr{X}$, and let us assume that these images are N independent realizations of a random vector $\mathbf{X} \in \mathscr{X}$ whose unknown probability density function (pdf) is $f_{\mathbf{X}}$. The ultimate characterization of a given image class entails estimating $f_{\mathbf{X}}$ from the training dataset $\mathscr{T}$. Estimating the empirical pdf from the available samples in $\mathscr{T}$, however, is very difficult because of the curse of dimensionality; we need a huge number of training samples to estimate $f_{\mathbf{X}}$ reliably, which we normally cannot access or handle. In our typical situation, we have $n \gg N$. Therefore, we need to reduce the dimensionality of the problem without losing important information for image approximation and modeling. As Scott mentions in his book [12, Chapter 7], this strategy is also supported by the empirical observation that multivariate data in $\mathbb{R}^n$ are almost never n-dimensional and there often exist lower-dimensional structures of data. That is, a class of images often has an *intrinsic dimension* $m < n$ (often $m \ll n$). Therefore, it would be much more efficient and effective to analyze the data in the smaller-dimensional subspace $\mathscr{F}$ of $\mathscr{X}$, if possible. We call $\mathscr{F}$ a *feature space*, and a map $\phi : \mathscr{X} \to \mathscr{F}$ a *feature extractor*. Then, the key is how to construct this "good" feature space $\mathscr{F}$ consisting of important features and to design the corresponding feature extractor ϕ.

Now, let us consider what are the "good" features for approximation and modeling of images. In this study, we define *image features* as the expansion coefficients (or their nonlinear functions) of an image relative to some basis. Let B be any basis spanning $\mathscr{X} \subset \mathbb{R}^n$. We also view B as a matrix whose columns are the basis vectors representing B, and assume $B \in \mathrm{GL}(n, \mathbb{R})$, a collection (in fact a group) of all invertible real-valued matrices of size $n \times n$. Let $\mathscr{C}(B \mid \mathscr{T})$ be a certain functional measuring the cost or inefficiency of the basis B for approximation and modeling of the image class given the training dataset $\mathscr{T}$. Then, we seek the best coordinates B_*:

$$B_* = \arg\min_{B \in \mathscr{L}} \mathscr{C}(B \mid \mathscr{T}),$$

where $\mathscr{L}$ is a set of all possible bases under consideration. Whether we constrain our search by restricting $\mathscr{L}$ or not makes a big difference as we will see soon. Now the feature extractor ϕ can be defined as the selection of m coordinates from the basis B_* potentially followed by some nonlinear mapping of them (e.g., computing energy).

2.1. Karhunen–Loève basis — principal component analysis

The Karhunen–Loève basis (KLB), also known as Principal Component Analysis (PCA), provides a decorrelated coordinate system. The KLB vectors are the eigenvectors of the covariance matrix of the process obeying $f_{\mathbf{X}}$. The KLB satisfies a number of optimality criteria, and in particular, it is *the minimum entropy basis* among all the orthonormal bases $\mathrm{O}(n)$, i.e., all the rotations of the coordinates in $\mathbb{R}^n$ [1]. Let B be any basis $B \in \mathrm{O}(n)$, and let $\mathbf{Y} = B^{\mathrm{T}}\mathbf{X}$ be the coordinates of the image $\mathbf{X}$ relative to the basis B. Entropy of the energy distribution over the coordinate axes can be considered as the inefficiency of that coordinate system since the entropy of the energy distribution measures the 'evenness' or 'flatness' of that distribution. Hence, in general, the larger the entropy, the less efficient for image approximation. Note that this entropy is different from the Shannon entropy of the process $\mathbf{X}$, which we discuss in the next section in details. Watanabe's viewpoint is to interpret the energy distribution over the coordinates (after normalization) as the discrete probability distribution. Let us now define the *entropy function* as

$$h(\gamma[B]) \triangleq -\sum_{i=1}^{n} \gamma_i[B] \log \gamma_i[B],$$

where $\gamma_i[B]$ is a normalized energy (or variance) of the ith coordinate of B, i.e., $\gamma_i[B] = E[Y_i^2]/\sum_{j=1}^n E[Y_j^2]$, or $\gamma_i[B] = \mathrm{Var}[Y_i]/\sum_{j=1}^n \mathrm{Var}[Y_j]$. In practice, we need to use the sample estimates $\hat{\gamma}_i[B]$ of $\gamma_i[B]$ using the training dataset $\mathscr{T}$. Then, the KLB is characterized by

$$B_{KLB} = \arg\min_{B \in \mathrm{O}(n)} h(\hat{\gamma}[B]). \tag{1}$$

On the other hand, the KLB has several drawbacks. First of all, criterion (1) does not measure the statistical independence of the coordinates. The KLB only takes care of the second-order statistics: it does just "decorrelation", and gives us only "the next best" coordinates as Watanabe put it. Therefore, the KLB provides a statistically independent coordinate system — which is the best thing one can hope for description and modeling — only for the multivariate Gaussian data since the decorrelation implies the independence for Gaussian data. The next serious problem is an inaccuracy of the sample estimate of the covariance matrix of the underlying process $f_{\mathbf{X}}$. In general, we do not know this matrix a priori, therefore, we need to estimate it using the available training samples. This inaccuracy is particularly severe for large n (dimension of the problem) with small N (the number of training samples). This entangles with the computational complexity as follows. Let $\bar{\mathbf{x}}$ be the sample mean of the training dataset $\mathscr{T}$, and let $X = (\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_N - \bar{\mathbf{x}}) \in \mathbb{R}^{n \times N}$. Then, the sample covariance matrix is $(1/N)XX^{\mathrm{T}}$. Suppose the singular value decomposition (SVD) of X is $X = U\Sigma V^{\mathrm{T}}$. (There is no need to perform full SVD in practice. This is just for the explanation of the KLB computation.) Note that the rank of X is $\min(n, N)$. Therefore, if $n < N$ (this is a classical situation in statistics where the dimensionality is small and the large number of samples are available), the KLB is $B_{KLB} = U \in \mathrm{O}(n)$ and its computational cost is

$O(n^3)$ for solving the eigenvalue problem, $XX^{\mathrm{T}}U = U\Sigma\Sigma^{\mathrm{T}}$. Now if $n > N$ (most of our problems of interest are under this category), the column vectors of XV are the first N eigenvectors of the sample covariance matrix $(1/N)XX^{\mathrm{T}}$ because $XX^{\mathrm{T}}XV = XV\Sigma^{\mathrm{T}}\Sigma$. We then need to solve the eigenvalue problem $X^{\mathrm{T}}XV = V\Sigma^{\mathrm{T}}\Sigma$ which is simply an $N \times N$ problem, i.e., requires $O(N^3)$ computation. In summary, the KLB computation costs $O(\min(n, N)^3)$. Note that having a small N is advantageous only for computational speed, not for the statistical accuracy. On the other hand, if N increases, then the computational cost increases cubically. This is a dilemma of the KLB computation.

2.2. Independent component analysis

To lift the PCA from its limitation to the second-order statistics, Comon [3] proposed the so-called Independent Component Analysis (ICA). Bell and Sejnowski discussed the closely related concept of "information maximization" and its neural network implementation [4].

Given a training dataset $\mathscr{T}$, the ICA tries to find an invertible linear transformation in $\mathrm{GL}(n, \mathbb{R})$ that minimizes the statistical dependence among its coordinates. In our notation, ICA can be written as

$$B_{ICA} = \arg \min_{B \in \mathrm{GL}(n,\mathbb{R})} \mathscr{C}_{ICA}(B|\mathscr{T}),$$

where $\mathscr{C}_{ICA}(B\,|\,\mathscr{T})$ measures the degree of statistical dependence of the coordinate system B using the training dataset $\mathscr{T}$. Let us now define differential entropy $H(\mathbf{X})$ of the process obeying $f_{\mathbf{X}}$.

$$H(\mathbf{X}) \triangleq -\int f_{\mathbf{X}}(\mathbf{x}) \log f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{2}$$

A convenient measure to quantify the statistical dependence among the components of $\mathbf{X}$ is the so-called *mutual information*:

$$I(\mathbf{X}) \triangleq \int f_{\mathbf{X}}(\mathbf{x}) \log \frac{f_{\mathbf{X}}(\mathbf{x})}{\prod_{i=1}^{n} f_{X_i}(x_i)}\, \mathrm{d}\mathbf{x}$$
$$= -H(\mathbf{X}) + \sum_{i=1}^{n} H(X_i),$$

which is simply *relative entropy* between $f_{\mathbf{X}}$ and the product of the marginal pdf's $\{f_{X_i}\}$. We note that $I(\mathbf{X}) = 0$ if and only if the components $X_1, \ldots, X_n$ are mutually independent. Now, we can write the inefficiency of the coordinate system $B \in \mathrm{GL}(n,\mathbb{R})$ as

$$\mathscr{C}_{ICA}(B\,|\,\mathscr{T}) = \hat{I}(\mathbf{Y}) = -\hat{H}(\mathbf{Y}) + \sum_{i=1}^{n} \hat{H}(Y_i), \tag{3}$$

where $\mathbf{Y} = B^{-1}\mathbf{X}$, and the $\hat{H}(\mathbf{Y})$ and $\{\hat{H}(Y_i)\}$ are the empirical estimates of the corresponding entropies using the training dataset $\mathscr{T}$. It is extremely difficult, however, to have a good estimate of $H(\mathbf{Y})$ via the empirical pdf $\hat{f}_{\mathbf{Y}}$ for large n, and even the case with $n > 3$ is difficult in practice. Therefore, Comon proposed to approximate Eq. (3) using the Edgeworth expansion of $f_{\mathbf{Y}}$ around the multivariate normal distribution with the same mean and variance as the original process, and this amounts to using the higher-order cumulants of $\mathbf{Y}$. This computational procedure is even more complicated and expensive than that of KLB; it costs $O(n^{2.5}N)$. Therefore, the direct application of the ICA of Comon is not feasible for the problems with very high dimension, $n \gg N$.

2.3. A dictionary and library of bases

Throughout this study we will use the *local basis library* as a basic tool to extract features since this library can resolve the problems of the PCA and the ICA. Below we will summarize the characteristics of this library. For the details, see [13–17]. This basis library consists of a collection of the *local basis dictionaries*, such as wavelet packets, local cosine/sine bases, local Fourier bases, and brushlets. Each dictionary consists of a redundant number (e.g., $n \log n$) of the basis vectors with the specific characters in scale, position, and frequency. These basis vectors are organized as a quadtree in a *hierarchical* manner ranging from very localized spikes to global oscillations with different frequencies (and orientations in the local Fourier and brushlet dictionaries). Expanding an image into such a dictionary is fast, $O(n[\log n]^p)$, where $p = 1$ for a wavelet packet dictionary and $p = 2$ for the local cosine/sine/Fourier and brushlet dictionaries. Thanks to this tree structure, each dictionary contains a huge number of possible bases (e.g., more than 2^n bases). Moreover, one can use the bottom-up procedure to efficiently search a good basis tailored to a specific application from such a huge number of possible bases by optimizing a certain criterion. This search algorithm, using the divide-and-conquer (i.e., split-and-merge) algorithm, is called the *best-basis* algorithm [13]. Therefore, this dictionary provides an adaptive, flexible, hierarchical, and computationally efficient set of features at our disposal. With a library of bases in our hands, our pattern descriptive power are enhanced, yet we can keep its computational complexity low. This strategy — viewing an image as a collection of more meaningful features rather than a collection of pixels — also appears to be employed in the primate vision-brain systems [18,19].

2.4. Joint best basis

Under the best-basis paradigm Wickerhauser [11] proposed (independent of Watanabe) a concept of a Joint Best Basis (JBB) that is the minimum entropy basis among all the bases in a dictionary of orthonormal bases. Wickerhauser's original motivation was to compute

the KLB vectors and coefficients approximately but efficiently. The JBB criterion is simply written as:

$$B_{JBB} = \arg\min_{B\in\mathscr{D}} h(\hat{\gamma}[B]).$$

A key difference from Eq. (1) is that B is searched within a specified dictionary of orthonormal bases $\mathscr{D}$ instead of all possible rotations $O(n)$. Therefore, its computational complexity is reduced to $O(n[\log n]^p)$, $p = 1, 2$. Recall that a dictionary $\mathscr{D}$ contains more than 2^n different orthonormal bases [15]. Moreover, since each feature is localized both in space and spatial frequency, the analysis and interpretation of the images become easier and more intuitive.

3. Least statistically dependent basis

Faced with the difficulty of ICA, it makes sense to find a basis from a dictionary or library of bases that minimizes the statistical dependency among its coordinates. To do this, let us consider a change of the basis of $\mathbf{X}$ in the definition of the differential entropy (2). We can easily get

$$\begin{aligned} H(\mathbf{Y}) &= H(B^{-1}\mathbf{X}) = H(\mathbf{X}) + \log|\det(B^{-1})| \\ &= H(\mathbf{X}) - \log|\det(B)|. \end{aligned}$$

Therefore, if B is a volume-preserving linear transformation, or more specifically, $B\in \mathrm{SL}(n,\mathbb{R})$, then the differential entropy is *invariant* under such a transformation:

$$H(\mathbf{Y}) = H(B^{-1}\mathbf{X}) = H(\mathbf{X}).$$

This invariance property is the key for our algorithm. The degree of the statistical dependence among the coordinates in a basis in $\mathrm{SL}(n,\mathbb{R})$ can be quantified by only considering the second term in Eq. (3), i.e., the sum of the differential entropy of the individual coordinates. Estimating $H(\mathbf{X})$ of high dimensional images is an extremely difficult task, but we do not need to estimate it as long as we compare the efficacy of the bases in $\mathrm{SL}(n,\mathbb{R})$. Our recent discussion with J.O. Strömberg clarified that $\mathrm{SL}(n,\mathbb{R})$ contains all the *biorthogonal* wavelet packet dictionaries if they are realized by the fast rotation algorithms described in Ref. [20, Chapter 2]. These biorthogonal dictionaries significantly increase our "vocabulary" for pattern description.

Now, we can state the selection criterion of our *Least Statistically Dependent Basis* (LSDB):

$$B_{LSDB} = \arg\min_{B\in\mathscr{D}} \sum_{i=1}^{n} \hat{H}(Y_i). \tag{4}$$

The LSDB is thus obtained by minimizing the sum of the coordinate-wise differential entropy among all possible (bi)orthogonal bases in a specified basis dictionary $\mathscr{D}$. We note that the basis search in Eq. (4) is fast since the sum of the coordinate-wise differential entropy is an additive measure. In practice, as Hall and Morton [21] suggests, the empirical estimate $\hat{H}(Y_i)$ of the entropy $H(Y_i)$ can be obtained by

$$\hat{H}(Y_i) = -\frac{1}{N}\sum_{k=1}^{N} \log \hat{f}_{Y_i}(y_{i,k}), \tag{5}$$

where $\hat{f}_{Y_i}$ is an empirical estimate of f_{Y_i} using the training dataset $\mathscr{T}$ by either histograms or kernels, and $y_{i,k}$ is the ith expansion coefficient (relative to B) of the training vector $\mathbf{x}_k$, $k = 1, \ldots, N$. Since the histogram computation is relatively cheap, i.e., $O(n)$, the computational complexity of the entire algorithm is dominated by the cost of expanding input images in a basis dictionary, i.e., $O(n[\log n]^p)$.

Remark 3.1. We can contrast our LSDB with KLB and JBB now. In the LSDB criterion (4), we have

$$\sum_{i=1}^{n} H(Y_i) = \sum_{i=1}^{n} E\left[\log\frac{1}{\hat{f}_{Y_i}}\right].$$

On the other hand, for KLB and JBB assuming that $\sum_{i=1}^{n} E[Y_i^2] = 1$, we have

$$\sum_{i=1}^{n} h(E[Y_i^2]) \geqslant \sum_{i=1}^{n} E[h(Y_i^2)] = \sum_{i=1}^{n} E\left[\log\frac{1}{Y_i^{2Y_i^2}}\right],$$

where we used Jensen's inequality. We can easily see that the criterion used in KLB and JBB is not suitable for measuring dependency among the coordinates in a basis. To illustrate this point, we conducted the following simple experiments. Let $\mathbf{X}$ be a two-dimensional random vector that is generated by two independent uniform random variable on $[-1,1]$ followed by 45° rotation, and we generated 1000 samples as displayed in Fig. 1(a). Then, we computed the KLB, JBB, and LSDB. Both the JBB and LSDB were obtained using the Haar-Walsh dictionary since we have only $n = 2$ in this example. The original points were projected onto these coordinates that are shown in Figs. 1(b)–(d).

The LSDB recovered the independent coordinates. The JBB selected the standard basis (i.e., no change). The KLB selected some rotated coordinate system, which is similar to the standard basis rather than the 45° rotation. Because the original distribution was not Gaussian, the KLB could not give us the independent coordinates. We tested the 15° and 30° rotations instead of 45° rotation so that the LSDB with Haar-Walsh dictionary cannot exactly capture the independent coordinates. In these cases, the LSDB selected the standard basis for 15° rotation, and 45° rotation for 30° rotation. The JBB still selected the standard basis, and the KLB selected the one similar to the standard basis, for both cases. Hence, the

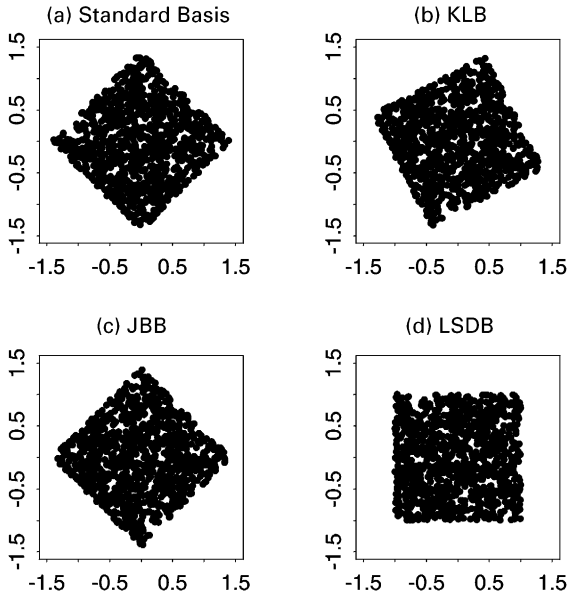

Fig. 1. Comparison of the KLB, JBB, and LSDB coordinates for the simple two-dimensional distribution. Points obeying a simple two-dimensional probability distribution are shown relative to (a) the standard basis; (b) the KLB; (c) the JBB; and (d) the LSDB.

LSDB provided the better (less dependent) coordinates than the KLB and JBB.

4. Signal and image approximation by LSDB

In this section we apply the LSDB to signal and image approximation and compression problems, and compare its performance with that of the KLB and the JBB. Since the redundancy is reduced *explicitly* using criteria (4), our strategy for approximation is simple: sort the LSDB coordinates in energy decreasing order, keep only the top m coordinates instead of n, and apply the inverse transform. For the KLB and JBB, we use the same strategy. We use geophysical acoustic waveforms and face images for our experiments.

4.1. Geophysical acoustic waveforms

For the detailed background of this dataset, see Ref. [10]. Here, we want to approximate/compress the acoustic waveforms (recorded in a borehole with 256 time samples per waveform) propagated through sandstone layers in the subsurface. We have 201 such "sand waveforms" as shown in Fig. 2. First we randomly split them into the training dataset consisting of 101 waveforms and the test dataset consisting of 100 waveforms.

First, we computed the mean signal of the training dataset and removed this mean waveform from both the

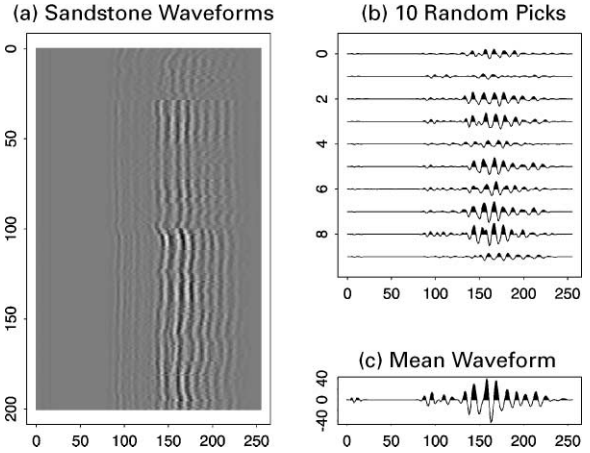

Fig. 2. The acoustic waveforms propagated through sandstone layers: (a) Original 201 waveforms displayed as gray scale images. The horizontal axis represents time samples (with sampling rate 10 μs). (b) Ten waveforms randomly selected from the 201 waveforms are displayed as wiggles (the positive parts are painted in black). (c) The mean waveform of the training dataset consisting of 101 randomly picked waveforms.

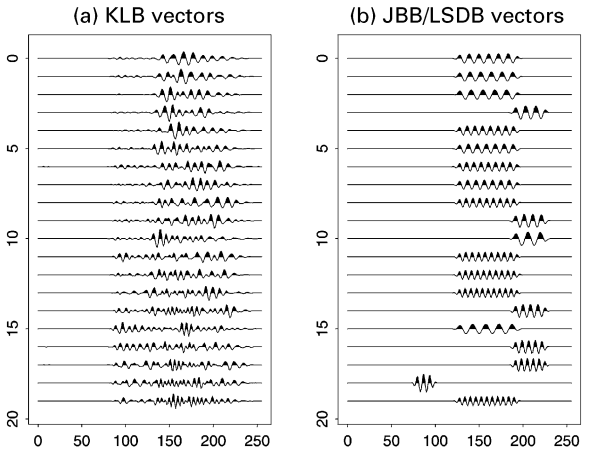

Fig. 3. (a) Top 20 KLB vectors. (b) Top 20 JBB/LSDB vectors. The basis vectors are sorted in the energy-decreasing order.

training and test datasets. Then, we computed the KLB, JBB, and LSDB of the training dataset. We used the local cosine dictionary for computing the JBB and LSDB since the local cosine dictionary allows us to segment time axis more easily than the wavelet packet dictionaries. It turned out that both the JBB and LSDB selected exactly the same basis. Top 20 most energetic basis vectors are shown in Fig. 3.

We then computed the relative ℓ^2 error of the approximation using these basis vectors as a function of the number of terms retained for approximating the original signals. Fig. 4 compares the performance of the KLB

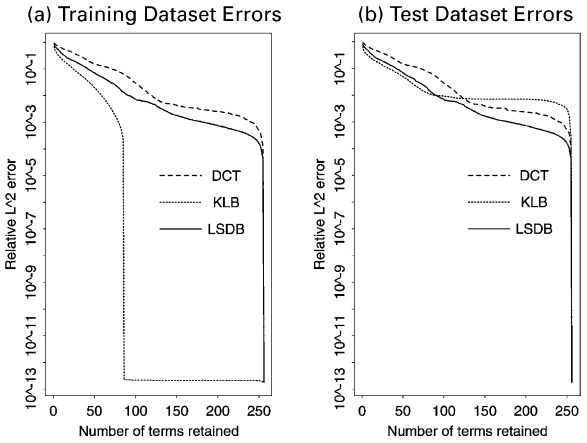

Fig. 4. Relative ℓ^2 approximation errors of the geophysical acoustic waveforms using DCT, KLB, LSDB plotted as functions of the number of terms used for approximation: (a) average errors over all the training signals; (b) average errors over all the test signals.

with that of the LSDB/JBB as well as DCT for the training and test datasets.

For the training dataset, the KLB approximation was perfect. In fact, the KLB approximation with 86 terms already reached the relative ℓ^2 error of 2.425×10^{-13} on average. The same KLB approximates the test dataset better than the LSDB only up to 89 terms. If we try to have more accuracy by increasing the number of terms, it got worse than the LSDB approximation. This implies that these geophysical acoustic waveforms do not obey the multivariate Gaussian distribution, and the sample mean and the covariance matrices computed from the training dataset were not enough to capture the statistics of the test dataset. On the other hand, the LSDB/JBB and DCT approximations are quite consistent for both the training and the test datasets. The locality of the basis functions of the LSDB/JBB in this case clearly gave a better performance than the DCT basis functions that are completely global in time.

4.2. "*Rogues' gallery*" *problem*

We now examine the approximation capability of LSDB for a set of face images, the so-called "Rogues' gallery" problem. This dataset consists of digitized pictures of faces of 143 people. These 143 people are a specific group of people; Caucasian students (and some faculty) at Brown University, without glasses, mustache, beard. The dataset was provided to us by L. Sirovich via M.V. Wickerhauser. For more detailed description of these images, see Ref. [22]. We note that horizontal dilation has been applied so that the pupils are placed on two fixed points if necessary. Fig. 5 displays some samples from this dataset as well as the "average" face, i.e., the mean of the 143 faces.

In the following experiments, we split the available 143 faces randomly into the training dataset $\mathscr{T}$ containing 72 face images and the test dataset containing 71 faces. We now examine how the KLB, JBB, and LSDB approximate the faces of the training and test datasets. We first removed the "average face" of the training dataset (which is quite similar to the average face of the all 143 faces displayed in Fig. 5) from each face in both the training and test datasets to make "caricatures", as Kirby and Sirovich put it [22]. All of our basis computations and processing are based on these caricatures.

For all the JBB and LSDB computations below, we use the multiple folding 2D local cosine dictionary (with DCT IV) [23] because their compression capability is superior to the fixed folding local cosines as Fang and Séré demonstrated in [23].

Fig. 6 compares the performance of the KLB, JBB, and LSDB using the top 72 terms.

Since the number of training images is 72, the KLB approximation here is simply a projection of a target image onto the 72 dimensional subspace spanned by the 72 "eigenfaces" (the computable KLB vectors). This original image in Fig. 6 belongs to the test dataset, not to the training dataset. If the target image were in the training dataset, then the KLB approximation would be perfect. However, because the target image is not in the training dataset, the approximation using these 72 KLB vectors is not impressive. In fact, it is not clear whether one can judge whether this approximation represents the same person as the original image. Using this standard KLB, we cannot do better than this. This essentially implies that the faces in the "Rogues' gallery" dataset do not obey the multivariate Gaussian distribution, and the mean and the covariance matrix computed from the training dataset did not capture the variability of the faces in the test dataset. Now, let us examine the JBB and LSDB approximations. Compared to the KLB, which has only 72 meaningful vectors in this case, we can compute a complete basis for both the JBB and the LSDB. Let us first note that the LSDB nicely split the faces into a set of meaningful regions. In particular, the regions around the eyes are split into a set of small segments, and most of the background regions are split into a larger segments. It is interesting to note that Kirby and Sirovich carefully cropped the oval-shaped portion of the faces containing the eyes, noses, and mouths and removed all the background and most of the hair portion for their approximations since "it significantly reduced the accuracy of the expression" [22]. We note that this natural splitting was done automatically in our case. On the other hand, JBB simply splits the images into four quadrants. The 72 term approximations by the JBB and LSDB shown in Fig. 6 are not necessarily better than the one by the KLB. However, they offer much more than

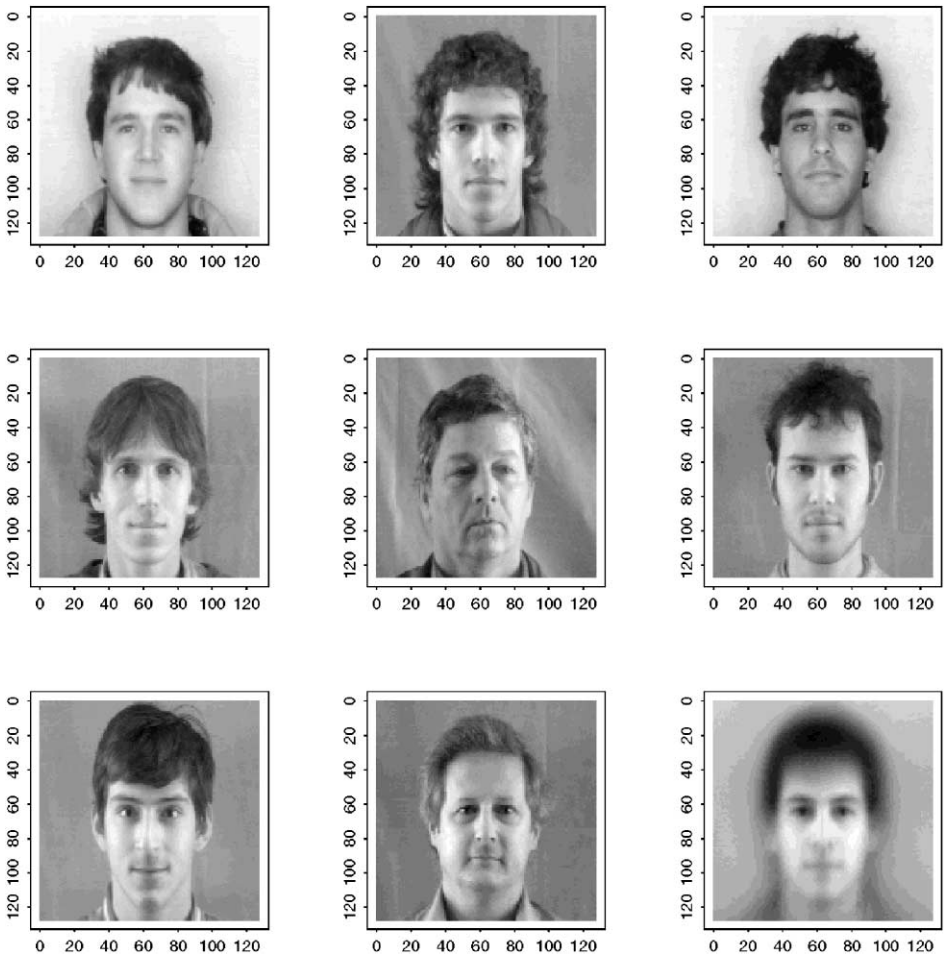

Fig. 5. Eight random samples from the "Rogues' gallery" dataset. The last (bottom right) figure shows the average face of the 143 faces.

the KLB. With the JBB and LSDB, we can use more terms to perform better approximation. With the most energetic 800 terms (i.e., about 6% of the total number of dimensions) instead of 72 terms, we can get the very good approximation as shown in Fig. 7.

In this figure, we compare the performance of the various adaptive and non-adaptive bases. As non-adaptive bases, we used the wavelet basis with the 12-tap Coiflet filter and the fixed folding local cosine transform (FLCT) by splitting the images homogeneously into a set of subimages of 8×8 pixels. The latter is very close to the block DCT algorithm used in the JPEG compression, although the FLCT has less edge effect than the block DCT. As adaptive bases, we used the JBB with multiple folding local cosine transform (MFLCT), and the LSDB with MFLCT. We observe that the LSDB approximation perceptually performs best, especially around important signatures such as the eyes, nose, and mouth.

Fig. 8 shows the top 5 most energetic basis vectors in each of the bases, i.e., KLB, C12-wavelet basis, FLCT 8×8, JBB, and LSDB. As one can see, the KLB vectors, of course, resemble actual faces. The basis vectors of the wavelet basis and JBB both show relatively global behavior. The FLCT 8×8 shows extremely local behavior: they are DC components at the particular blocks. The LSDB with MFLCT shows intermediate behavior featuring multiple folding, i.e., some "shadows" in the symmetric manner.

We computed the efficiency of the approximation of these bases in terms of average relative ℓ^2 error versus the number of coordinates retained. Fig. 9 compares their performance of the top 72 terms with that of the KLB. The KLB performed best for both the training and test datasets. As explained above, although the KLB worked perfectly for the training dataset, the other bases performed closely to the KLB for the test dataset. Moreover, the KLB approximation only allows us to use 72 terms in this case whereas the other bases allows us as many terms as we wish up to 128×128. As shown in Fig. 10, if we want to have better approximations, then we need to

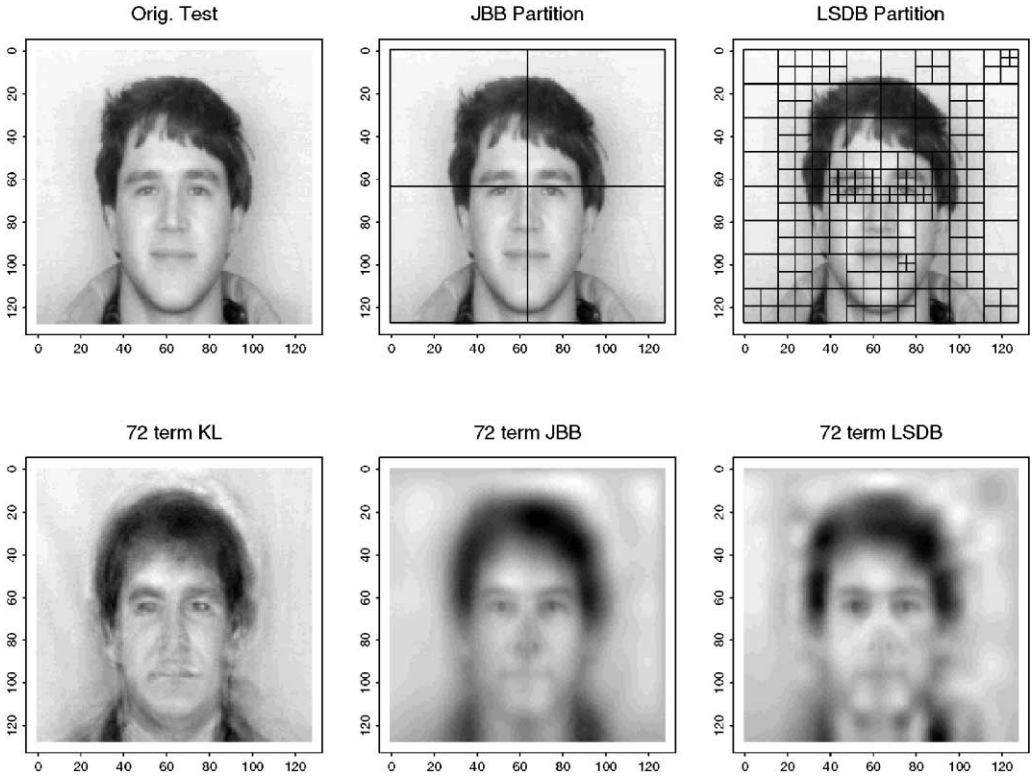

Fig. 6. Comparison of KLB, JBB, and LSDB using the top 72 terms. The original data was not in the training dataset.

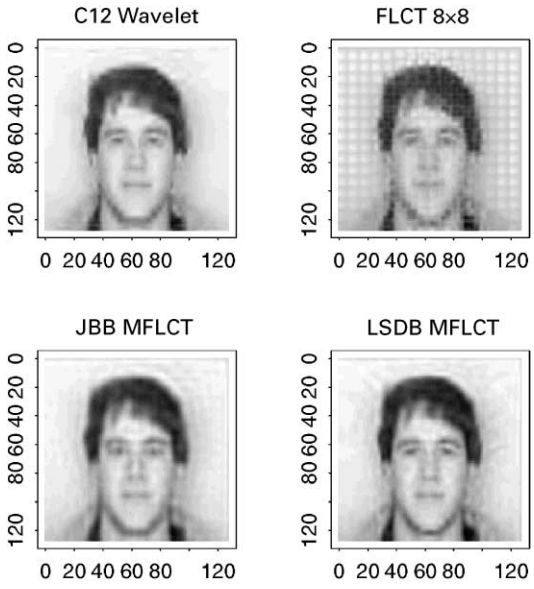

Fig. 7. Comparison of the approximations with 800 terms of various bases.

use the other bases with more terms. From this figure, we observe that the LSDB performs best if one keeps more than 315 terms. Another interesting observation is the stability of the adaptive bases. The behavior of the KLB was drastically different between the training and test datasets. On the other hand, the LSDB and the JBB behaved consistently, just like the other non-adaptive bases such as the wavelets and FLCT.

4.3. '*Second rotation*' *by KLB*

As discussed in Section 2 and demonstrated above with "Rogues' gallery" problem, we can compute only N KLB vectors if we have only N training images. We cannot go beyond this number and this can be a serious limitation for the KLB. We can compute more than N KLB vectors by the following idea. First, we compress the training images using some good basis, say LSDB. Then, using the top k coordinates of that basis, where $N < k < n$, we can compute the KLB on top of those k coordinates. In other words, we perform the *second rotation* of the coordinates (the basis used for compression does the first rotation). Of course, the rank of the covariance matrix is still N in this case, but nothing except the limitation on our computational resources prevents us from computing the k-dimensional KLB. For example, on a desktop PC or a workstation, one can easily use $k = 800$ to $k = 1000$. Fig. 11 demonstrates this idea.

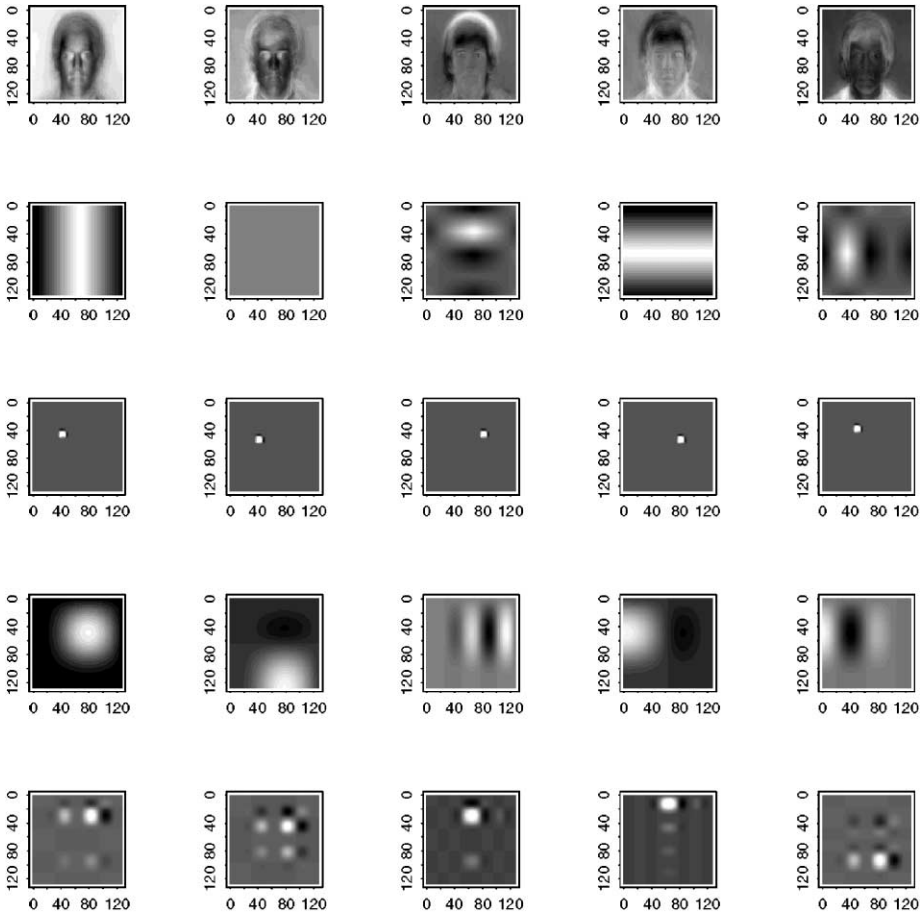

Fig. 8. Comparison of the five most energetic basis vectors in the five bases. First row: KLB, second row: C12 wavelet basis, third row: FLCT 8×8, fourth row: JBB, and the last row: LSDB.

Fig. 9. Relative ℓ^2 approximation errors of the "Rogues' gallery" dataset using various bases versus the number of terms used for approximation for the first 72 terms: (a) average errors over all the training images; (b) average errors over all the test images.

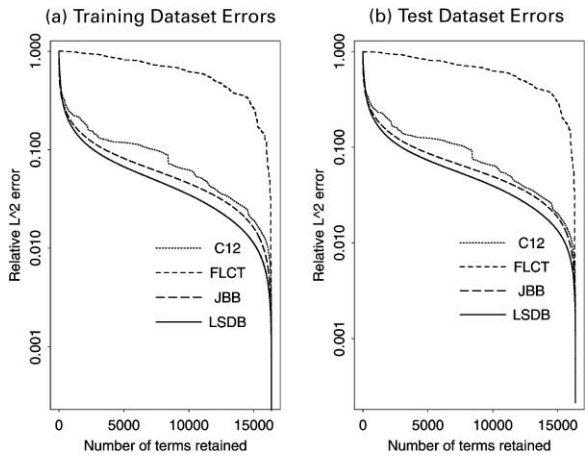

Fig. 10. Relative ℓ^2 approximation errors of the "Rogues' gallery" dataset using various bases versus the number of terms used for approximation: (a) average over all the training images; (b) average over all the test images.

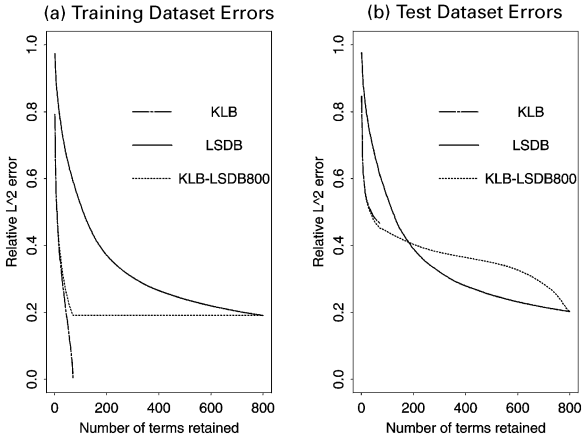

Fig. 11. Relative ℓ^2 approximation errors versus the number of terms used for approximation for the first 800 terms: (a) average over all the training images; (b) average over all the test images. Now we compare the performance of the KLB (only 72 terms available), LSDB, and the KLB computed on top of the 800 LSDB coordinates.

As one can see, the second rotation by KLB computed from the 800 LSDB coordinates (we abbreviate this as KLB-LSDB800) gives us the same performance as the KLB up to the first 60 terms or so. But this does not end with 72 terms. It continues until it reaches to $k = 800$, where the approximation error is exactly the same as the LSDB. For the training dataset, the KLB-LSDB800 reaches to the minimum with the 72 terms and does not improve anymore, as expected. For the test dataset, however, it performs better than the LSDB up to 182 terms; then the LSDB takes over. This indicates that the second rotation by the KLB may not always be advantages for image approximation. It turned out, however, that such rotations can be quite useful for image modeling, which will be discussed in the next section.

5. Stochastic model building using LSDB

Image modeling is an important application area where LSDB may contribute. As mentioned in Introduction, if we can successfully build a good stochastic model of a specific image class, then we can sample and simulate as many new images from the model as we wish. Such simulation may be particularly useful for image diagnostics. In this section, we propose two stochastic models of an image class using the LSDB coordinates.

5.1. Image models with LSDB as independent coordinates

We start with the simplest possible model. This model assumes that the LSDB coordinates of a given image class are truly statistically independent, i.e., a probabilistic description of that image class is a product of empirical marginal pdf's of the LSDB coordinates. Let $\mathbf{Y} = B_{LSDB}^{-1}\mathbf{X}$ be a random vector representing an input random vector $\mathbf{X}$ in the LSDB coordinates. (Note that if $B_{LSDB} \in \mathrm{O}(n)$, then $B_{LSDB}^{-1} = B_{LSDB}^{\mathrm{T}}$.) Now this simplest model can be written as

$$f_{\mathbf{Y}}(\mathbf{y}) = f_{\mathbf{Y}}(y_1, \ldots, y_n) \approx \prod_{i=1}^{n} \hat{f}_{Y_i}(y_i). \tag{6}$$

Therefore, this model can be described as

Image Model = Description of the LSDB
+ Statistics of each LSDB coordinate. (7)

Here, the description of the LSDB consists of the specification of the basis dictionary used and the specification of the LSDB vectors obtained via (4) in that dictionary. The statistics of each LSDB coordinate means either its empirical pdf (epdf) or empirical cumulative distribution function (ecdf). Sampling new images from this model is easy. We use the inversion method for each coordinate to sample a typical coefficient of that coordinate. Let $F_{N,i}(y)$ be an ecdf of the ith LSDB coordinate Y_i and let $F_i(y)$ be an interpolated version of $F_{N,i}$ so that the inverse exists. If $U \sim \mathrm{unif}(0,1)$, then $F_i^{-1}(U)$ obeys F_i, i.e., Y_i and $F_i^{-1}(U)$ share the same ecdf F_i since $\Pr\{F_i^{-1}(U) < y\} = \Pr\{U < F_i(y)\} = F_i(y)$. Once we sample all the coefficients of the LSDB coordinates to get $\mathbf{Y}_{new} = (F_1^{-1}(U_1), \ldots, F_n^{-1}(U_n))^{\mathrm{T}}$ where $U_1, \ldots, U_n$ are different realizations of the unif(0,1) distribution, then we can synthesize an typical image by the inverse transform $\mathbf{X}_{new} = B_{LSDB}\mathbf{Y}_{new}$. See Ref. [24] for simulation methods other than the inversion method.

If the images of the class contain noise and the noise model is known a priori, e.g., additive white Gaussian noise (WGN), then we can set up a better model including denoising as follows:

Image Model = Description of the LSDB
+ Statistics of the top m LSDB coordinates
+ Statistics of the remaining $(n - m)$
LSDB coordinates. (8)

Here, the m coordinates to be kept as a part of the signal (meaningful) component can be selected via a certain criterion such as the MDL criterion developed in Ref. [6]. The last $(n - m)$ terms correspond to noise. So if we do not want to include noise in the model, we can throw away this part.

5.1.1. Geophysical acoustic waveforms

Here, we want to model the sandstone waveforms used in the previous section. Fig. 12 shows 10 synthesized

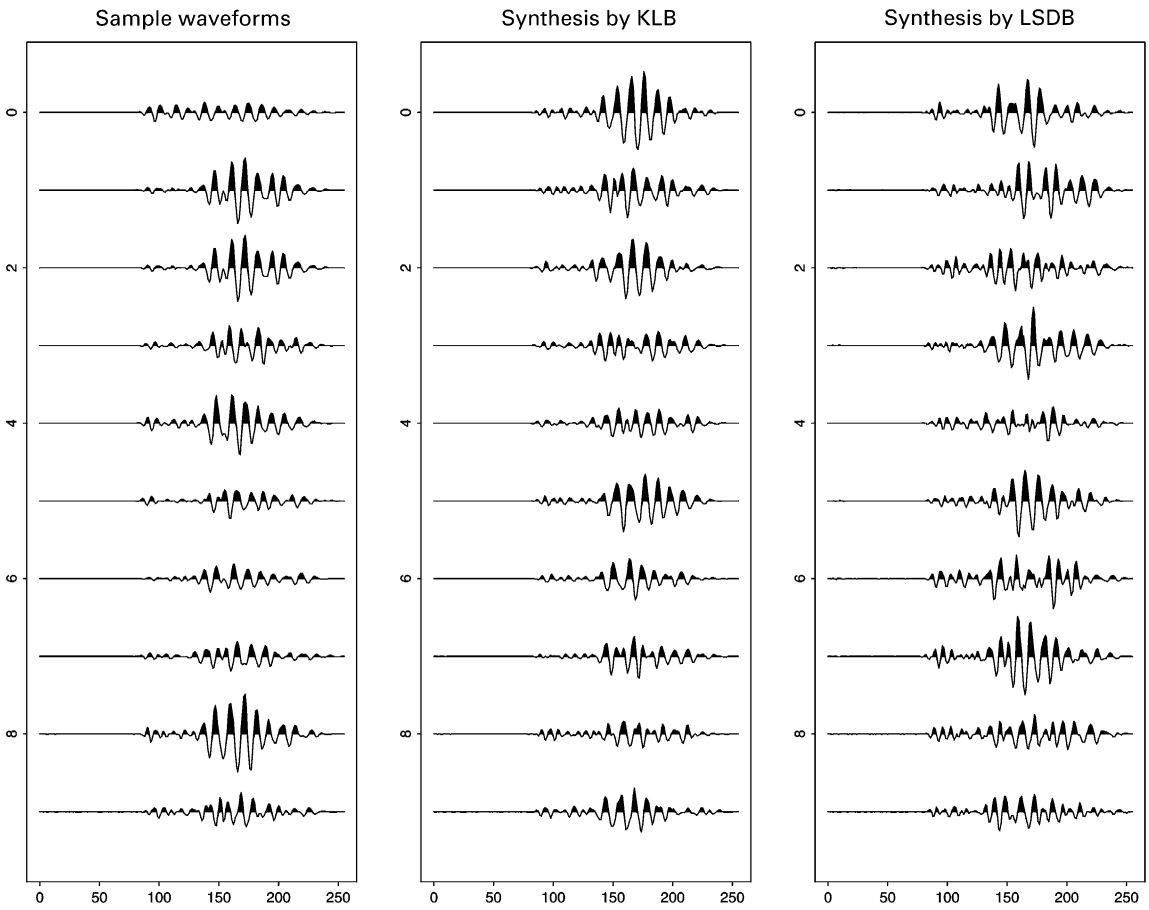

Fig. 12. (a) 10 example sand waveforms randomly selected from the training dataset; (b) 10 synthesized waveforms using the KLB; (c) 10 synthesized waveforms using the LSDB.

waveforms by assuming that the basis coordinates are all statistically independent, sampling each coefficient separately, and reconstructing them. For each case, we used all 256 coordinates.

As we can see from Fig. 12, the waveforms using the KLB and the LSDB both visually look similar to the original waveforms shown in Fig. 2(b). Although we cannot quantify how close to the true independence unless we can compute the entropy of the process $H(\mathbf{X})$ in Eq. (3), this experiment indicates that both the KLB and the LSDB coordinates are almost statistically independent for this dataset. On the other hand, considering the physics of the wave propagation, there should be some dependency between the so-called P wave components (compressional waves) around time samples 80 and the so-called S wave components (shear waves) around time samples 160, and this dependency is characteristic to the underlying media where the waves propagated [10]. Exploring such dependency is one of our future projects as will be discussed in Section 6.

5.1.2. "Rogues' gallery" problem

Similar to the examples of the geophysical acoustic waveforms, we compare the simple independence model (7) built on the pixel basis (i.e., the standard basis), the KLB, and the LSDB in Fig. 13.

These are the "new faces" generated by sampling from the models. The model assuming the independence of the pixel coordinates are clearly worst. They are simply the average face plus noise, and this validates the fact that the pixel coordinates are strongly statistically dependent. The independence model using the KLB, which we call the KLB-STD model for clarification, worked quite well although its coordinates are only guaranteed to be uncorrelated. The synthesized faces using the LSDB are "clouded" and do not look like representative faces of this class of images. This experiment indicates that the LSDB coordinates are not really mutually independent for this dataset. Synthesis using the JBB coordinates (not shown) does not work well either. This failure took us to the next level of the modeling.

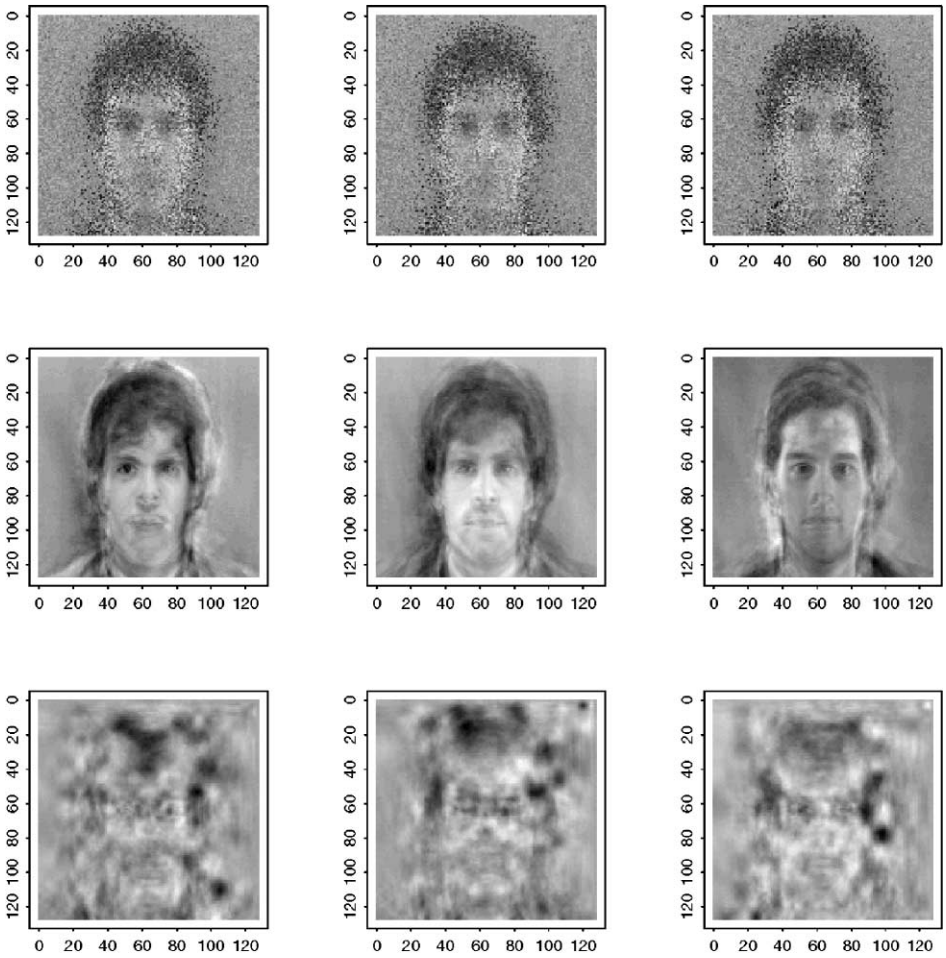

Fig. 13. Sampled images from the simple independence model (7) using the pixel basis (the first row), the KLB (the second row) and the LSDB (the last row).

5.2. "*Second rotation*" *by KLB*

Fig. 13 is somewhat discouraging for the image modeling using the LSDB. How can we improve the model using the LSDB? Can we do more with the LSDB than with KLB-STD? These questions, in fact, drove us to examine the "second rotation" by KLB, which was used for image approximation in Section 4.3: we form m-dimensional feature space $\mathscr{F}$ by selecting the top m LSDB coordinates, then rotate this feature space coordinates further to have decorrelated coordinates. Now we have the following image model:

Image Model = Description of the LSDB

+ Description of the KLB of

the top m LSDB coordinates

+ Statistics of these m KLB coordinates

+ Statistics of the $(n-m)$ LSDB coordinates. (9)

The last term is again optional. It may be better not to include this term for noisy image classes. This model can be quite powerful since these m coordinates are already statistically less dependent than the original coordinates and we can compute the m-dimensional KLB rather quickly if $m \ll n$. The assumption here is that the decorrelated KLB coordinates computed on top of the m LSDB coordinates are now statistically independent. Fig. 14 shows nine realizations from the KLB-LSDB800 model (9) with $m = 800$.

We can see some dramatic improvements over the last row of Fig. 13. Of course, these decorrelated coordinates may not necessarily be statistically independent, but they should be much less dependent than the LSDB

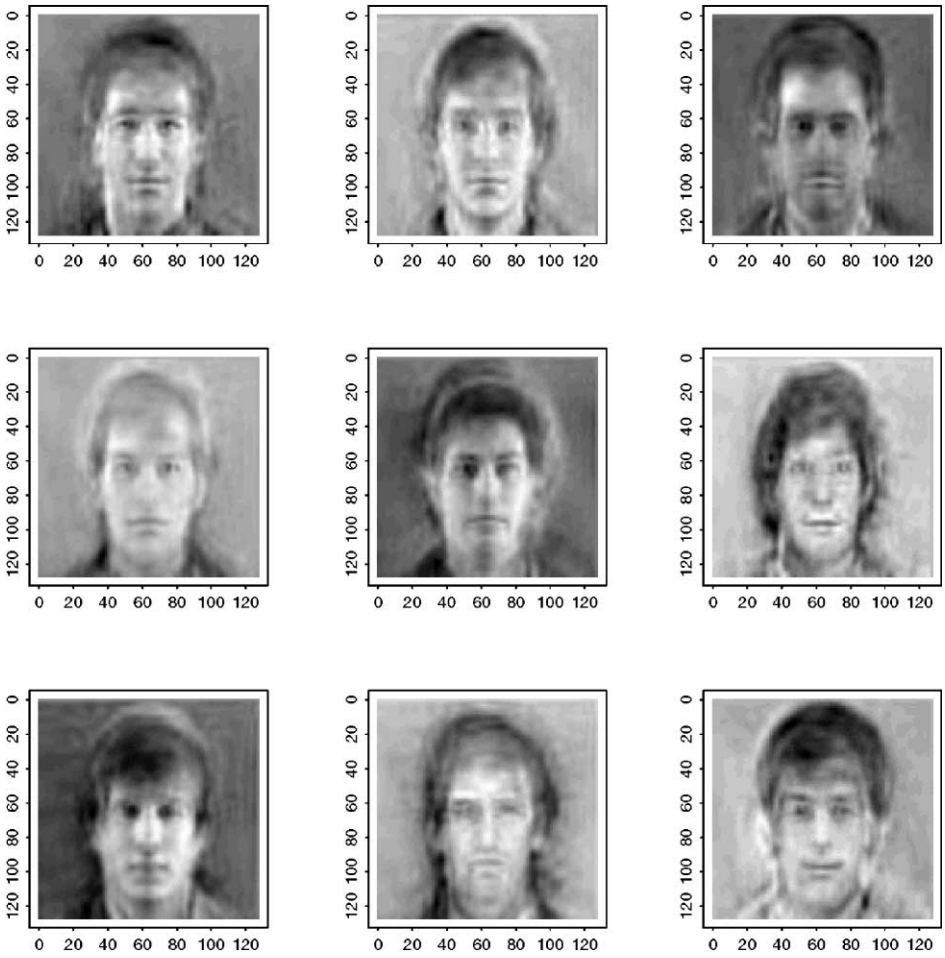

Fig. 14. Nine sampled faces from the KLB-LSDB800 model. Compare with Fig. 13.

coordinates. Fig. 14 testifies this. Do the underlying coordinates before the second rotation matter? The answer is yes. In order to see this, we computed the second rotation from the top 800 JBB (i.e., the KLB-JBB800 model), and follow the same sampling and reconstruction procedure to get the realizations shown in Fig. 15.

We can see that the realizations from the KLB-JBB800 model is blurred compared to those from the KLB-LSDB800 model shown in Fig 14. This difference is inherited from the approximation quality of the first stage (i.e. the top 800 LSDB versus the top 800 JBB).

Note here that out of 800 KLB-LSDB800 vectors, the first 72 vectors (when they are transformed back to the pixel basis) are essentially the same as the eigenfaces computed directly from the pixel basis representations. The question here is that what the other 728 basis vectors we computed are and whether these help simulation or not. To understand this effect, we synthesized the faces using only the last 728 terms of the KLB-LSDB800 model and obtained the new faces shown in Fig. 16.

As we can see here, these are better than the LSDB model shown in the last row of Fig. 13, and show the variations in facial expressions, which may have contributed to the good realization quality of the full KLB-LSDB800 model shown in Fig. 14.

6. Discussion

In this section, we discuss the relations of our proposed methods to the other methods and describe some of our ongoing and future projects.

6.1. Relation with local Karhunen–Loève bases

As we demonstrated in Section 5, the second rotation by the KLB computed from the selected LSDB coordinates was useful for stochastic image modeling. Over there, we simply selected the most energetic 800 LSDB coordinates from the total 16,834 LSDB coordinates. We

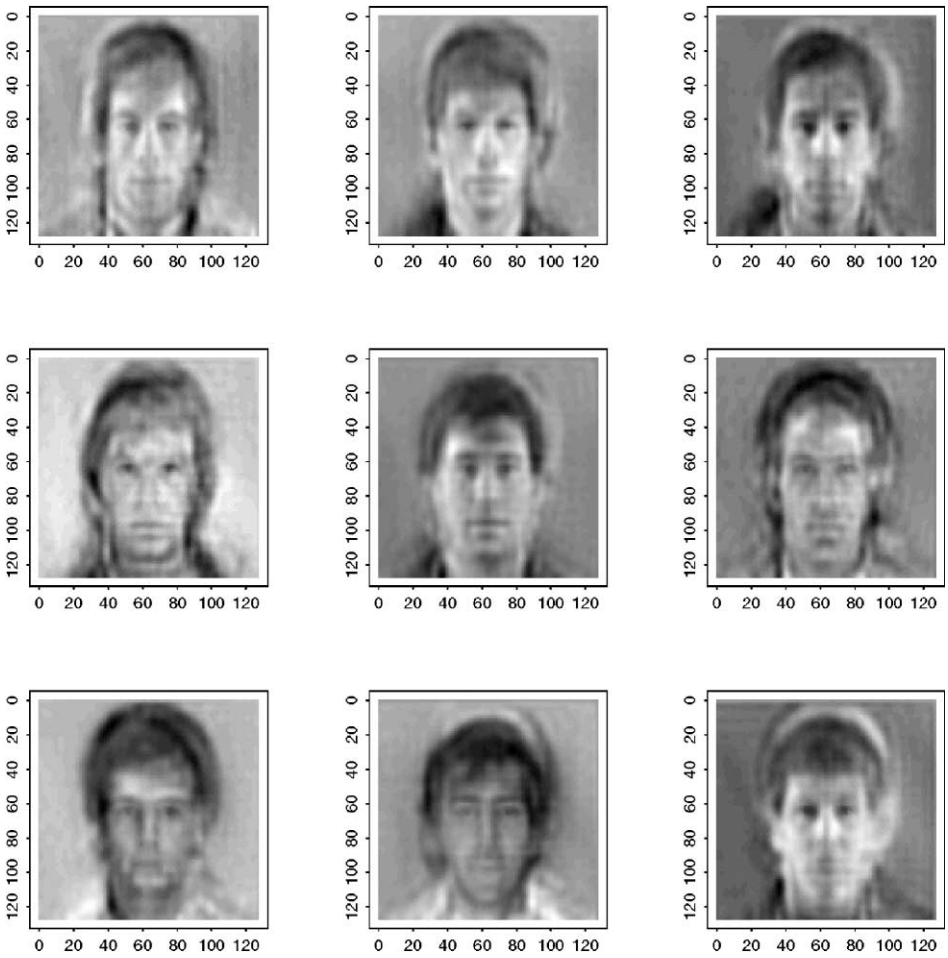

Fig. 15. Nine sampled faces from the KLB-JBB800 model. Compare with Fig. 14.

can select the LSDB coordinates in completely different manner. For example, we can select the LSDB coordinates only related to a specific region. Such selection is easy for our case because the coefficients expanded by the local cosine/Fourier dictionaries are nicely organized according to their spatial locations. (Note that it is also possible to select such coefficients in the wavelet packets/brushlets dictionaries with a little bit of extra effort of organizing their indices.) The selection of the LSDB coordinates in the local regions and the computations of the KLBs on those regions is closely related to the notion of the local Karhunen–Loève basis (LKLB) that Coifman and the author proposed in Ref. [25]. The idea of LKLB is to split the signals/images into tree-structured segments by the smooth orthogonal projections [26], compute the KLB locally within each segment, then invoke the best-basis algorithm to prune the tree and merge the segments. As a result, we can have a basis consisting of localized and spatially adapted versions of the KLBs. Computing the localized KLBs makes sense both computationally and statistically since splitting the images into segments provides better statistics. That is, the number of available samples N and the dimension of the problem for each segment get closer. For the smaller segments, N can be even larger than their dimensions. In Ref. [25], however, we had difficulty in deciding the basis selection (i.e., tree-pruning) criterion. We examined a few alternative criteria, but all of them were based on the eigenvalues of the autocorrelation or covariance matrices computed at those segments. But now, the LSDB offers a sound and justifiable (we are selecting a least statistically dependent basis) split of signals and images into segments, we can further compute all sorts of bases in each segment. In particular, we can compute a KLB in each segment. These operations are, in fact, a set of *local second rotations*. Compared to the original LKLB algorithm that uses the pixel coordinates to compute each local KLBs, this new version via the LSDB is much more computationally efficient since we can reduce the dimension of

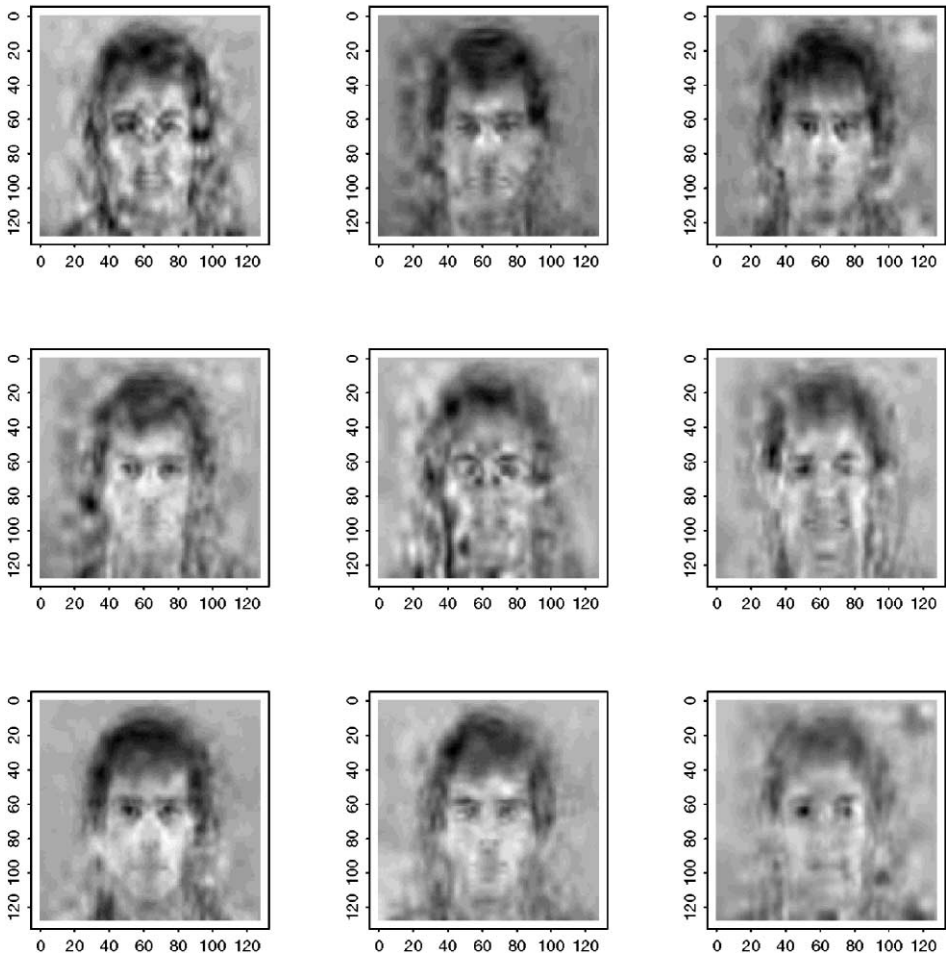

Fig. 16. Nine sampled faces using only the 73rd to 800th coordinates of the KLB-LSDB800 model. Compare with Fig. 14.

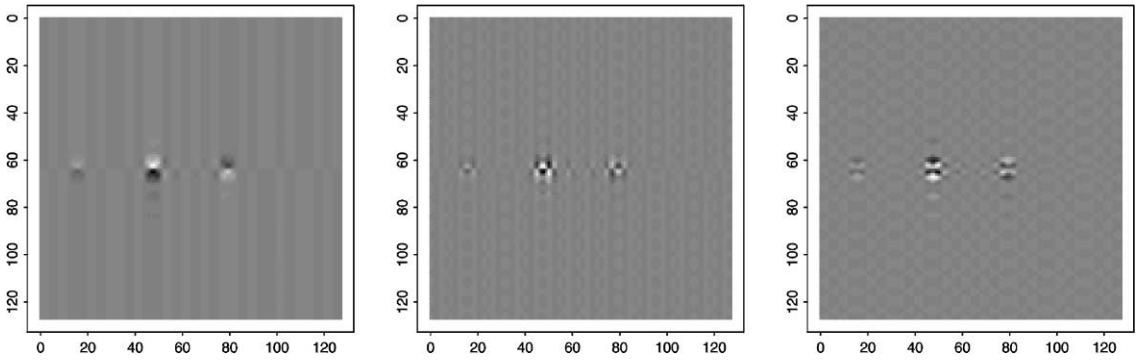

Fig. 17. Top 3 "eigen-eyes" computed from the LSDB coefficients belonging to a small region around right eyes of the "Rogues' gallery" dataset.

each segment using the LSDB coordinates prior to the KLB computation.

Fig. 17 shows one interesting example. This shows the top 3 KLB vectors computed from the LSDB coefficients belonging to a small segment around the right eye region of the "Rogues' gallery" dataset. This segment is one of the small segments (4×4 pixels) around the right eye shown in the LSDB partition pattern in Fig. 6. The first

"eigen-eye" checks the symmetry between the upper and lower part of the right eyes and left eyes. Recall that we used the 2D local cosine basis dictionary with multiple folding. That is why the eigen-eyes have activities outside of this right eye region. Then the subsequent eigen-eyes reveal more detailed structures of the eyes. For example, the second eigen-eye analyzes the pupils. One can view these as "tailor-made" local orthonormal basis vectors. If we operate these local second rotations in the frequency domain, this amounts to constructing "tailor-made" wavelets and wavelet packets whose details we will report at a later date.

6.2. Relation with other work

Moreau and Pesquet [27] independently and concurrently proposed a similar algorithm. The difference between theirs and ours lies in two aspects. One is the measure of the statistical independence, and the other is the motivation. First, their measure of statistical independence is a generalized version of the measures proposed by Comon [3], which is based on the higher-order cumulants, whereas ours is based on the empirical entropy estimation using the histogram or kernel pdf estimators. There are pros and cons on the estimation of entropy using the empirical pdf estimation using histograms or kernels. First, if we restrict the search of the best basis within the group $\mathrm{SL}(n,\mathbb{R})$, as we discussed in Section 3, the sum of the coordinate-wise entropy estimate using the pdf correctly and directly measures the statistical dependence of the data relative to that coordinate system. The estimation error solely comes from that of the pdf estimation from available samples. On the other hand, the convergence of the estimate to the true entropy is not necessarily fast, i.e., $o(1/\sqrt{N})$ as the number of available samples $N \to \infty$ as shown by Hall and Morton [21]. Furthermore, this convergence rate is only guaranteed for the low-dimensional cases ($n = 1$ for the histograms, and $n = 1, 2, 3$ for the special kernel-based estimators with heavy tails). The cumulant-based estimators can have faster convergence rate, i.e., $O(N^{-3/2})$ [28], and in principle, can handle any high-dimensional problems; however, the cumulant-based entropy estimation is always dependent on the number of terms in the Edgeworth expansions used for approximating the pdf, and for the higher-order terms and higher-dimensional problems, the mathematical expressions become extremely cumbersome as shown in [28]. For our LSDB algorithm, we do not need to evaluate the high dimensional pdf's; we only need to evaluate 1D coordinate-wise pdf's. From this point of view, the histogram-based pdf estimation should give very good estimates of the coordinate-wise entropy, and there is no need to use the cumulant-based estimators. In fact, the reason why people developed the cumulant-based estimators seems that they need to estimate the entropy for higher-dimensional distributions [29]. As for the difference in motivations, ours is to efficiently approximate and model a specific class of images whereas theirs is to separate mixed signal sources.

Also, related is the work of Buckheit and Donoho [30]. They proposed several measures to find an orthonormal basis from a dictionary whose coordinates are maximally non-Gaussian. Searching the maximally non-Gaussian basis is advocated by Diaconis and Freedman [31] who showed that most low-dimensional projections of high-dimensional datasets are approximately Gaussian. Therefore, the non-Gaussian projections reveal some intrinsic features of the datasets. This argument is also the basis of the projection pursuit [29,32]. Now, in order to evaluate a "non-Gaussianness" of a basis in the dictionary, they proposed to maximize either (1) the sum of the kurtosis of each coordinate; or (2) the sum of the certain statistical distances or information measures (such as the Anderson–Darling, Kolmogorov–Smirnov, Kullback–Leibler distances, or Fisher information) between the empirical distribution and the Gaussian distribution with the same mean and variance as that empirical distribution. If the Kullback–Leibler distance (i.e., relative entropy) is used for the measure of non-Gaussianness, the relationship between the LSDB and the maximally non-Gaussian basis (MNGB) of Buckheit and Donoho can be made precise. Let $D(f_{Y_i}\|\phi_{Y_i})$ be the Kullback–Leibler distance (i.e., relative entropy) between the pdf f_{Y_i} of the coordinate Y_i and the Gaussian distribution ϕ_{Y_i} whose mean and variance are the same as those of f_{Y_i}. Then,

$$\begin{aligned} D(f_{Y_i}\|\phi_{Y_i}) &= \int f_{Y_i}(y_i)\log\frac{f_{Y_i}(y_i)}{\phi_{Y_i}(y_i)}\,\mathrm{d}y_i \\ &= -H(Y_i) - \int f_{Y_i}(y_i)\log\phi_{Y_i}(y_i)\,\mathrm{d}y_i \\ &= -H(Y_i) - \int \phi_{Y_i}(y_i)\log\phi_{Y_i}(y_i)\,\mathrm{d}y_i \\ &= -H(Y_i) + \frac{1}{2}\log \mathrm{Var}[Y_i] + \frac{1}{2}\log(2\pi\mathrm{e}). \end{aligned}$$

The third equality comes from the fact that the f_{Y_i} and ϕ_{Y_i} share the same mean and variance. Therefore, the MNGB is obtained by the following criterion:

$$\begin{aligned} B_{MNGB} &= \arg\max_{B\in\mathscr{D}} \sum_{i=1}^{n} D(f_{Y_i}\|\phi_{Y_i}) \\ &= \arg\min_{B\in\mathscr{D}} \sum_{i=1}^{n} \left(H(Y_i) - \frac{1}{2}\log \mathrm{Var}[Y_i]\right) \end{aligned} \tag{10}$$

whereas the LSDB criterion (4) does not have the second term of $-\frac{1}{2}\log \mathrm{Var}[Y_i]$. This implies that the MNGB prefers the coordinates with small entropy and large variance, whereas the LSDB prefers the coordinates with small entropy only.

6.3. Further challenge

6.3.1. Nonlinear representations

All we have considered so far as a set of possible bases for image approximation and modeling are just a subset of invertible linear transformations $\mathrm{GL}(n,\mathbb{R})$, in fact, a subset of $\mathrm{O}(n)$ or $\mathrm{SL}(n,\mathbb{R})$, which can be searchable via the LSDB algorithm. Potentially, there may be a nonlinear transformation that outperforms any of these linear transformations in terms of image approximation and modeling. Along this line, we are currently investigating algorithms to find such nonlinear transformations [33]. One of the algorithms tries to find a nonlinear transformation that maps data to the new coordinates where the transformed data obey the standard multivariate Gaussian distribution, $N(\mathbf{0},\mathbf{I})$, which, of course, has independent coordinates. Although it is more computationally expensive than the LSDB algorithm, it may provide a coordinate system which is much closer to the statistical independence than the LSDB.

6.3.2. Image models with the LSDB with pairwise conditioning

As our experiments showed, the LSDB does not guarantee a truly independent coordinate system in general. So, we considered model (9) using the KLB of the top m LSDB coordinates as an attempt to make them more independent (in this case only decorrelation is achieved, of course). Alternatively, we can examine the dependency among the selected LSDB coordinates more explicitly. We cannot afford to examine and model the deep statistical dependence among the coordinates if we need the computational efficiency and algorithmic simplicity. The simplest dependency we can model is perhaps the pairwise dependency model that approximates the true pdf by a product of bivariate pdf's:

$$f_{\mathbf{Y}}(y_1,\ldots,y_n) \approx \prod_{i\neq j} \hat{f}_{Y_iY_j}(y_i,y_j). \tag{11}$$

Currently, we are working on algorithms to check this pairwise dependency among the LSDB coordinates and to sample the LSDB coefficients conditionally using 2D pdf estimation techniques such as ASH2D [12, Chapter 5]. We are also investigating the Markov chain model on the LSDB coordinates, which we hope to report at a later date.

7. Conclusion

We have presented a new criterion for the best-basis algorithm to find the least statistically dependent coordinate system from a given basis dictionary for a given collection of signals or images. This criterion is to minimize the mutual information of the coordinates, which is a measure of the statistical dependence among them. In this sense, this proposed algorithm can be viewed as the best-basis version of ICA. This basis (LSDB) can be computed rapidly, i.e., $O(n[\log n]^p)$, where n is the dimension of the problem, and $p=1$ for the wavelet packet dictionaries, and $p=2$ for the local cosine/Fourier/brushlet dictionaries. Using the geophysical acoustic waveforms and the "Rogues' gallery" dataset, we have demonstrated that LSDB performed best among a variety of bases including the KLB, JBB, DCT, and wavelets, for image approximation in terms of the average relative ℓ^2 error. We have also proposed simple stochastic models for a given class of signals or images based on the LSDB coordinates. The first model is to assume the statistical independence among the LSDB coordinates, which allows us to sample typical coefficients of each coordinate separately using the empirical distribution estimated from the available training coefficients of that coordinate, which in turn easily allows us to simulate new images at our disposal. This strategy worked well for the geophysical acoustic waveforms. Because the LSDB does not necessarily provide the truly statistically independent coordinates, this first model did not work well for the "Rogues' gallery" dataset. To deal with this problem, we have introduced the second model based on the "second rotation" by the KLB computed from the top m LSDB coordinates. This model gives us the decorrelated coordinates built on top of the localized least statistically dependent features. The simulation results using the second model suggest that this second rotation further reduced the statistical dependency among the coordinates. We believe that exploring the statistical dependency among the LSDB coordinates is likely to be a key for building a better stochastic image models, which we will tackle in our future project.

Acknowledgements

The earlier versions of this paper were presented at the SPIE conference on Wavelet Applications in Signal and Image Processing VI, San Diego, CA, July 1998, and the 32nd Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, November 1998.

The author would like to thank Professor L. Sirovich at Brown University and Professor M.V. Wickerhauser at Washington University at St. Louis for providing the digitized face images.

This research was partially supported by NSF DMS-9973032 and DMS-9978321.

References

[1] S. Watanabe, Karhunen-Loève expansion and factor analysis: theoretical remarks and applications, Transactions of

the Fourth Prague Conference on Information Theory, Statistic Decision Functions, Random Processes, Prague, Publishing House of the Czechoslovak Academy of Sciences, 1965, pp. 635–660.
[2] D.J. Field, What is the goal of sensory coding? Neural Comput. 6 (1994) 559–601.
[3] P. Comon, Independent component analysis, a new concept? Signal Process. 36 (1994) 287–314.
[4] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Comput. 7 (1995) 1129–1159.
[5] R.R. Coifman, D. Donoho, Translation-invariant de-noising, in: A. Antoniadis, G. Oppenheim (Eds.), Wavelets and Statistics, Lecture Notes in Statistics, Springer, Berlin, 1995, pp. 125–150.
[6] N. Saito, Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion, in: E. Foufoula-Georgiou, P. Kumar (Eds.), Wavelets in Geophysics, Academic Press, San Diego, CA, 1994, pp. 299–324 (Chapter XI).
[7] D.L. Donoho, I.M. Johnstone, Ideal denoising in an orthonormal basis chosen from a library of bases, C.R. Acad. Sci. Paris, Sér. I 319 (1994) 1317–1322.
[8] R.R. Coifman, N. Saito, Constructions of local orthonormal bases for classification and regression, C.R. Acad. Sci. Paris, Sér. I 319 (1994) 191–196.
[9] N. Saito, R.R. Coifman, Local discriminant bases and their applications, J. Math. Imaging Vision 5(4) (1995) 337–358, invited paper.
[10] N. Saito, R.R. Coifman, Extraction of geological information from acoustic well-logging waveforms using time-frequency wavelets, Geophysics 62 (6) (1997) 1921–1930.
[11] M.V. Wickerhauser, Fast approximate factor analysis, in: Curves and Surfaces in Computer Vision and Graphics II, October 1991, pp. 23–32; Proc. SPIE 1610.
[12] D.W. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization, Wiley, New York, 1992.
[13] R.R. Coifman, M.V. Wickerhauser, Entropy-based algorithms for best basis selection, IEEE Trans. Inform. Theory 38 (1992) 713–719.
[14] Y. Meyer, Wavelets: Algorithms and Applications, SIAM, Philadelphia, PA, 1993 (Translated and revised by R.D. Ryan).
[15] M.V. Wickerhauser, Adapted Wavelet Analysis from Theory to Software. A.K. Peters, Wellesley, MA, 1994 with diskette.
[16] N. Saito, Local feature extraction and its applications using a library of bases, Ph.D. Thesis, Department of Mathematics, Yale University, New Haven, CT 06520 USA, December 1994. Available via World Wide Web, http://www.math.yale.edu/pub/wavelets/papers/lfeulb.tar.gz.
[17] F.G. Meyer, R.R. Coifman, Brushlets: a tool for directional image analysis and image compression, Appl. Comput. Harmonic Anal. 4 (1997) 147–187.
[18] D.H. Hubel, Eye, Brain, and Vision, Scientific American Library, New York, 1995.
[19] I. Fujita, K. Tanaka, M. Ito, K. Cheng, Columns for visual features of objects in monkey inferotemporal cortex, Nature 360 (1992) 343–346.
[20] E. Fossgaard, Fast computational algorithms for the discrete wavelet transform and applications of localized orthonormal bases in signal classification, Master's Thesis, University of Tromsø, Norway, 1997.
[21] P. Hall, S.C. Morton, On the estimation of entropy, Ann. Inst. Stat. Math. 45 (1) (1993) 69–88.
[22] M. Kirby, L. Sirovich, Application of the Karhunen-Loève procedure for the characterization of human faces, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1) (1990) 103–108.
[23] X. Fang, E. Séré, Adapted multiple folding local trigonometric transforms and wavelet packets, Appl. Comput. Harmonic Anal. 1 (1994) 169–179.
[24] B.D. Ripley, Stochastic Simulation, Wiley, New York, 1987.
[25] R.R. Coifman, N. Saito, The local Karhunen-Loève bases, Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis, Paris, France, IEEE Press, New York, June 18–21, 1996, pp. 129–132.
[26] M.V. Wickerhauser, Smooth localized orthonormal bases, C.R. Acad. Sci. Paris Sér. I 316 (1993) 423–427.
[27] E. Moreau, J.-C. Pesquet, Independence/decorrelation measures with applications to optimized orthonormal representations, Proc. ICASSP-97.
[28] J.-J. Lin, N. Saito, R.A. Levine, Edgeworth expansions of the Kullback-Leibler information, Technical Report, Division of Statistics, University of California, Davis, 1999. (In preparation).
[29] M.C. Jones, R. Sibson, What is projection pursuit? (with discussion), J. Roy. Statist. Soc. A 150 (Part 1) (1987) 1–36.
[30] J.B. Buckheit, D.L. Donoho, Time-frequency tilings which best expose the non-Gaussian behavior of a stochastic process, Proceedings of the International Symposium on Time-Frequency and Time-Scale Analysis, Paris, France, IEEE Press, New York, June 18–21, 1996, pp. 1–4.
[31] P. Diaconis, D. Freedman, Asymptotics of graphical projection pursuit, Ann. Stat. 12 (1984) 793–815.
[32] P.J. Huber, Projection pursuit (with discussion), Ann. Stat. 13 (2) (1985) 435–525.
[33] J.-J. Lin, N. Saito, R.A. Levine, An iterative nonlinear Gaussianization algorithm for resampling dependent components, in: P. Pajunen, J. Karhunen (Eds.), Proceedings of the Second International Workshop on Independent Component Analysis and Blind Signal Separation, Helsinki, Finland, IEEE, New York, June 19–22, 2000, pp. 245–250.

About the Author—NAOKI SAITO received the B. Eng. and the M. Eng. degrees in mathematical engineering from the University of Tokyo, Japan, in 1982 and 1984, respectively, and the Ph.D. degree in applied mathematics from Yale University in 1994. In 1984, he joined Nippon Schlumberger K.K., Fuchinobe, Japan, and in 1986, he was transferred to Schlumberger-Doll Research, Ridgefield, CT, where he worked as a research scientist until 1997. In 1997, he joined Department of Mathematics, University of California, Davis, and currently he is an associate professor.

Dr. Saito received the Best Paper Award from SPIE for the wavelet applications in signal and image processing conference in 1994, and the Henri Doll Award, the highest honor in the technical papers presented at the annual symposium within the Schlumberger organization in 1997. In 1999, he was elevated to IEEE Senior Member. He also received an Office of Naval Research Young Investigator Award in 2000.

His research interests include: wavelet theory and its applications, feature extraction, pattern recognition and classification, statistical signal processing and analysis, human and machine vision, and geophysical inverse problems.

He is a member of IEEE, IMS, SEG, and SIAM.