1

A Method for Compact Image Representation using Sparse Matrix and Tensor Projections onto Exemplar Orthonormal Bases

Karthik S. Gurumoorthy, Ajit Rajwade, Arunava Banerjee and Anand Rangarajan

Abstract—**We present a new method for compact representation of large image datasets. Our method is based on treating small patches from a 2D image as matrices as opposed to the conventional vectorial representation, and encoding these patches as sparse projections onto a set of exemplar orthonormal bases, which are learned *a priori* from a training set. The end result is a low-error, highly compact image/patch representation that has significant theoretical merits and compares favorably with existing techniques (including JPEG) on experiments involving the compression of ORL and Yale face databases, as well as a database of miscellaneous natural images. In the context of learning multiple orthonormal bases, we show the easy tunability of our method to efficiently represent patches of different complexities. Furthermore, we show that our method is extensible in a theoretically sound manner to higher-order matrices ('tensors'). We demonstrate applications of this theory to compression of well-known color image datasets such as the GaTech and CMU-PIE face databases and show performance competitive with JPEG. Lastly, we also analyze the effect of image noise on the performance of our compression schemes.**

Index Terms—**compression, compact representation, sparse projections, singular value decomposition (SVD), higher-order singular value decomposition (HOSVD), greedy algorithm, tensor decompositions.**

I. INTRODUCTION

Most conventional techniques of image analysis treat images as elements of a vector space. Lately, there has been a steady growth of literature which regards images as matrices, e.g. [1], [2], [3], [4]. As compared to a vectorial method, the matrix-based representation helps to better exploit the spatial relationships between image pixels, as an image is essentially a two-dimensional function. In the image processing community, this notion of an image has been considered in [5], in the context of image coding by using singular value decomposition (SVD). Given any matrix, say $J \in R^{M_1 \times M_2}$, the SVD of the matrix expresses it as a product of a diagonal matrix $S \in R^{M_1 \times M_2}$ of singular values, with two orthonormal matrices $U \in R^{M_1 \times M_1}$ and $V \in R^{M_2 \times M_2}$. This decomposition is unique upto a sign factor on the vectors of U and V if the singular values are all distinct. For the specific case of natural images, it has been observed that the values along the diagonal of S are rapidly decreasing, which allows for the low-error reconstruction of the image by low-rank approximations. This property of SVD, coupled with the fact that it provides the best possible lower-rank approximation to a matrix, has motivated its application

to image compression in [5] and [6]. In [6], the SVD technique is further combined with vector quantization for compression applications.

To the best of our knowledge, the earliest technique on obtaining a *common* matrix-based representation for a set of images was developed by Rangarajan in [1]. In [1], a *single* pair of orthonormal bases (U, V) is learned from a set of some N images, and each image $I_j, 1 \leq j \leq N$ is represented by means of a projection S_j onto these bases, i.e. in the form $I_j = U S_j V^T$. Similar ideas are developed by Ye in [2] in terms of optimal lower-rank image approximations. In [3], Yang *et al.* develop a technique named 2D-PCA, which computes principal components of a column-column covariance matrix of a set of images. In [7], Ding *et al.* present a generalization of 2D-PCA, named as 2D-SVD, and some optimality properties of 2D-SVD are investigated. The work in [7] also unifies the approaches in [3] and [2], and provides a non-iterative approximate solution with bounded error to the problem of finding the single pair of common bases. In [4], He *et al.* develop a clustering application, in which a single set of orthonormal bases is learned in such a way that projections of a set of images on these bases are neighborhood-preserving.

It is quite intuitive to observe that, as compared to an entire image, a small image patch is a simpler, more local entity, and hence can be more accurately represented by means of a smaller number of bases. Following this intuition, we choose to regard an image as a set of matrices (one per image patch) instead of using a single matrix for the entire image as in [1], [2]. Furthermore, there usually exists a great deal of similarity between a large number of patches in one image or across several images of a similar kind. We exploit this fact to learn a small number of full-sized orthonormal bases (as opposed to a single set of low-rank bases learned in [1], [2] and [7], or a single set of low-rank projection vectors learned in [3]) to reconstruct a set of patches from a training set by means of *sparse* projections with least possible error. As we shall demonstrate later, this change of focus from learning lower-rank bases to learning full-rank bases but with sparse projection matrices, brings with itself several significant theoretical and practical advantages. This is because it is much easier to adjust the sparsity of a projection matrix in order to meet a certain reconstruction error threshold than to adjust for its rank [see Section II-A and Section III].

There exists a large body of work in the field of sparse image representation. Coding methods which express images as sparse combinations of bases using techniques such as the

The authors are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611, USA. e-mail: {ksg,avr,arunava,anand}@cise.ufl.edu.

discrete cosine transform (DCT) or Gabor-filter bases have been quite popular for a long period. The DCT is also an essential ingredient of the widely used JPEG image compression standard. These developments have also spurred interest in methods that *learn* a set of bases from a set of natural images as well as a *sparse* set of coefficients to represent images in terms of those bases. Pioneering work in this area has been done in [8], with a special emphasis on learning an over-complete set of bases and their sparse combinations. Some recent contributions in the field of over-complete representations include the work in [9] by Aharon *et al.*, which encodes image patches as a sparse linear combination of a set of dictionary vectors (learned from a training set). An important feature of all such learning-based approaches (as opposed to those that use a fixed set of bases such as DCT) is their tunability to datasets containing a *specific* type of images. In this paper, we develop such a learning technique, but with the key difference that our technique is matrix-based, unlike the aforementioned vector-based learning algorithms. The main advantage of our technique is as follows: we learn a small number of pairs of orthonormal bases to represent ensembles of image patches. Given any such pair, the computation of a sparse projection for any image patch (matrix) with least possible reconstruction error, can be accomplished by means of a very simple and optimal greedy algorithm. On the other hand, the computation of the optimal sparse linear combination of an over-complete set of basis vectors to represent another vector is a well-known NP-hard problem [10]. We demonstrate the applicability of our algorithm to the compression of databases of face images, with favorable results in comparison to existing approaches. Our main algorithm on ensembles of 2D images or image patches, was presented earlier by us in [11]. In this paper, we present more detailed comparisons, including with JPEG, and also study the effect of various parameters on our method, besides showing more detailed derivations of the theory.

In the current work, we bring out another significant extension of our previous algorithm - namely its elegant applicability to higher-order matrices (commonly and usually mistakenly termed tensors [1]), with sound theoretical foundations. In this paper, we represent patches from color images as tensors (third-order matrices). Next, we learn a small number of orthonormal matrices and represent these patches as sparse tensor projections onto these orthonormal matrices. The tensor-representation for image datasets, as such, is not new. Shashua and Levin [12] regard an ensemble of gray-scale (face) images, or a video sequence, as a single third-order tensor, and achieve compact representation by means of lower-rank approximations of the tensor. However, quite unlike the computation of the rank of a matrix, the computation of the rank of a tensor [2] is known to be an NP-complete problem [14]. Moreover, while the SVD is known to yield the best lower-rank approximation of a matrix in the 2D case,

its higher-dimensional analog (known as higher-order SVD or HOSVD) does not necessarily produce the best lower-rank approximation of a tensor (see the work of Lathauwer in [13] for an extensive development of the theory of HOSVD). Nevertheless, for the sake of simplicity, HOSVD has been used to produce lower-rank approximations of datasets (albeit a theoretically sub-optimal one [13]) which work well in the context of applications ranging from face recognition under change in pose, illumination and facial expression as in [15] by Vasilescu and Terzopoulos (though in that particular paper, each image is still represented as a vector) to dynamic texture synthesis from gray-scale and color videos in [16] by Costantini *et al*. In these applications, the authors demonstrate the efficacy of the multilinear representation in accounting for variability over several different modes (such as pose, illumination and expression in [15], or spatial, chromatic and temporal in [16]), as opposed to simple vector-based image representations.

An iterative scheme for a lower-rank tensor approximation is developed in [13], but the corresponding energy function is susceptible to the problem of local minima. In [12], two new lower-rank approximation schemes are designed: a closed-form solution under the assumption that the actual tensor rank equals the number of images (which usually may not be true in practice), or else an iterative approximation in other cases. Although the latter iterative algorithm is proved to be convergent, it is not guaranteed to yield a global minimum. Wang and Ahuja [17] also develop a new iterative rank-R approximation scheme using an alternating least-squares formulation, and also present another iterative method that is specially tuned for the case of third-order tensors. Very recently, Ding *et al.* [18] have derived upper and lower bounds for the error due to low-rank truncation of the core tensor obtained in HOSVD (which is a closed-form decomposition), and use this theory to find a *single* common triple of orthonormal bases to represent a database of color images (represented as 3D arrays) with minimal error in the L_2 sense. Nevertheless, there is still no method of directly obtaining the *optimal* lower-rank tensor approximation which is non-iterative in nature. Furthermore, the error bounds in [18] are applicable only when the entire set of images is coded using a single common triple of orthonormal bases. Likewise, the algorithms presented in [17] and [12] also seek to find a *single* common basis.

The algorithm we present here differs from the aforementioned ones in the following ways: (1) All the aforementioned methods learn a common basis, which may not be sufficient to account for the variability in the images. We do not learn a single common basis-tuple, but a *set* of some K orthonormal bases to represent some N patches in the database of images, with each image and each patch being represented as a higher-order matrix. Note that K is much less than N. (2) We do not seek to obtain lower-rank approximations to a tensor. Rather, we represent the tensor as a sparse projection onto a chosen tuple of orthonormal bases. This sparse projection, as we shall later on show, turns out to be optimal and can be obtained by a very simple greedy algorithm. We use our extension for the purpose of compression of a database of color images, with promising results. Note that this sparsity based approach

---

[1] A $N_1 \times N_2$ matrix is a tensor only when it is expressly characterized as a multilinear function from $V_1 \otimes V_2$ to R where V_1 and V_2 are N_1-dimensional and N_2-dimensional vector spaces respectively.

[2] The rank of a tensor is defined as the smallest number of rank-1 tensors whose linear combination gives back the tensor, with a tensor being of rank-1 if it is equal to the outer product of several vectors [13].

has advantages in terms of coding efficiency as compared to methods that look for lower-rank approximations, just as in the 2D case mentioned before.

Our paper is organized as follows. We describe the theory and the main algorithm for 2D datasets in Section II. Section III presents experimental results and comparisons with existing techniques. Section IV presents our extension to higher-order matrices, with experimental results and comparisons in Section V. We conclude in Section VI.

II. THEORY: 2D IMAGES

Consider a set of images, each of size $M_1 \times M_2$. We divide each image into non-overlapping patches of size $m_1 \times m_2, m_1 \ll M_1, m_2 \ll M_2$, and treat each patch as a separate matrix. Exploiting the similarity inherent in these patches, we effectively represent them by means of sparse projections onto (appropriately created) orthonormal bases, which we call as 'exemplar bases'. We learn these exemplars *a priori* from a set of training image patches. Before describing the learning procedure, we first explain the mathematical structure of the exemplars.

A. Exemplar Bases and Sparse Projections

Let $P \in R^{m_1 \times m_2}$ be an image patch. Using singular value decomposition (SVD), we can represent P as a combination of orthonormal bases $U \in R^{m_1 \times m_1}$ and $V \in R^{m_2 \times m_2}$ in the form $P = USV^T$, where $S \in R^{m_1 \times m_2}$ is a diagonal matrix of singular values. However P can also be represented as a combination of *any* set of orthonormal bases $\bar{U}$ and $\bar{V}$, different from those obtained from the SVD of P. In this case, we have $P = \bar{U}S\bar{V}^T$ where S turns out to be a *non-diagonal* matrix [3]. Contemporary SVD-based compression methods leverage the fact that the SVD provides the best *low-rank* approximation to a matrix [6], [19]. We choose to depart from this notion, and instead answer the following question: What *sparse* matrix $W \in R^{m_1 \times m_2}$ will reconstruct P from a pair of orthonormal bases $\bar{U}$ and $\bar{V}$ with the least error $\|P - \bar{U}W\bar{V}^T\|^2$? Sparsity is quantified by an upper bound T on the L_0 norm of W, i.e. on the number of non-zero elements in W (denoted as $\|W\|_0$)[4]. We prove that the *optimal W* with this sparsity constraint is obtained by nullifying the least (in absolute value) $m_1 m_2 - T$ elements of the estimated projection matrix $S = \bar{U}^T P\bar{V}$. Due to the ortho-normality of $\bar{U}$ and $\bar{V}$, *this simple greedy algorithm turns out to be optimal* as we prove below:.

Theorem 1: Given a pair of orthonormal bases $(\bar{U}, \bar{V})$, the optimal sparse projection matrix W with $\|W\|_0 = T$ is obtained by setting to zero $m_1 m_2 - T$ elements of the matrix $S = \bar{U}^T P\bar{V}$ having least absolute value.

Proof: We have $P = \bar{U}S\bar{V}^T$. The error in reconstructing a patch P using some other matrix W is $e = \|\bar{U}(S - W)\bar{V}^T\|^2 = \|S - W\|^2$ as $\bar{U}$ and $\bar{V}$ are orthonormal. Let

---

[3]The decomposition $P = \bar{U}S\bar{V}^T$ exists for any P even if $\bar{U}$ and $\bar{V}$ are not orthonormal. We still follow ortho-normality constraints to facilitate optimization and coding. See section II-C and III-B.

[4]See section III for the merits of our sparsity-based approach over the low-rank approach.

$I_1 = \{(i,j)|W_{ij} = 0\}$ and $I_2 = \{(i,j)|W_{ij} \neq 0\}$. Then $e = \sum_{(i,j) \in I_1} S_{ij}^2 + \sum_{(i,j) \in I_2} (S_{ij} - W_{ij})^2$. This error will be minimized when $S_{ij} = W_{ij}$ in all locations where $W_{ij} \neq 0$ and $W_{ij} = 0$ at those indices where the corresponding values in S are as small as possible. Thus if we want $\|W\|_0 = T$, then W is the matrix obtained by nullifying $m_1 m_2 - T$ entries from S that have the least absolute value and leaving the remaining elements intact.□

Hence, the problem of finding an optimal sparse projection of a matrix (image patch) onto a pair of orthonormal bases, is solvable in $O(m_1 m_2 (m_1 + m_2))$ time as it requires just two matrix multiplications ($S = U^T PV$). On the other hand, the problem of finding the optimal sparse linear combinations of an over-complete set of basis vectors in order to represent another vector (i.e. the vectorized form of an image patch) is a well-known NP-hard problem [10]. In actual applications [9], approximate solutions to these problems are sought, by means of pursuit algorithms such as OMP [20]. Unfortunately, the quality of the approximation in OMP is dependent on T, with an upper bound on the error that is directly proportional to $\sqrt{1 + 6T}$ [see [21], Theorem (C)] under certain conditions. Similar problems exist with other pursuit approximation algorithms such as Basis Pursuit (BP) as well [21]. For large values of T, there can be difficulties related to convergence, when pursuit algorithms are put inside an iterative optimization loop. Our technique avoids any such dependencies.

B. Learning the Bases

The essence of this paper lies in a learning method to produce K exemplar orthonormal bases $\{(U_a, V_a)\}, 1 \leq a \leq K$, to encode a training set of N image patches $P_i \in R^{m_1 \times m_2}$ ($1 \leq i \leq N$) with least possible error (in the sense of the L_2 norm of the difference between the original and reconstructed patches). Note that $K \ll N$. In addition, we impose a sparsity constraint that every S_{ia} (the matrix used to reconstruct P_i from (U_a, V_a)) has at most T non-zero elements. The main objective function to be minimized is:

$$E(\{U_a, V_a, S_{ia}, M_{ia}\}) = \sum_{i=1}^{N} \sum_{a=1}^{K} M_{ia} \|P_i - U_a S_{ia} V_a^T\|^2 \quad (1)$$

subject to the following constraints:

$$(1) U_a^T U_a = V_a^T V_a = I, \forall a. \quad (2) \|S_{ia}\|_0 \leq T, \forall (i,a).$$
$$(3) \sum_a M_{ia} = 1, \forall i \text{ and } M_{ia} \in \{0,1\}, \forall i, a. \quad (2)$$

Here M_{ia} is a binary matrix of size $N \times K$ which indicates whether the ith patch belongs to the space defined by (U_a, V_a). Notice that we are not trying to use a mixture of orthonormal bases for the projection, but just project onto a single suitable basis pair instead. The optimization of the above energy function is difficult, as M_{ia} is binary. A simple K-means will lead to local minima issues, and so we relax the binary membership constraint so that now $M_{ia} \in (0,1), \forall (i,a)$, subject to $\sum_{a=1}^{K} M_{ia} = 1, \forall i$. The naive mean field theory line of development culminating in [22], (with much of the theory developed already in [23]) leads to the following deterministic

annealing energy function:

$$E(\{U_a, V_a, S_{ia}, M_{ia}\}) = \sum_{i=1}^{N} \sum_{a=1}^{K} M_{ia} \|P_i - U_a S_{ia} V_a^T\|^2 +$$
$$\frac{1}{\beta} \sum_{ia} M_{ia} \log M_{ia} + \sum_{i} \mu_i (\sum_a M_{ia} - 1) +$$
$$\sum_a \text{trace}[\Lambda_{1a}(U_a^T U_a - I)] + \sum_a \text{trace}[\Lambda_{2a}(V_a^T V_a - I)]. \quad (3)$$

Note that in the above equation, $\{\mu_i\}$ are the Lagrange parameters, $\{\Lambda_{1a}, \Lambda_{2a}\}$ are symmetric Lagrange matrices, and β a temperature parameter.

We first initialize $\{U_a\}$ and $\{V_a\}$ to random orthonormal matrices $\forall a$, and $M_{ia} = \frac{1}{K}$, $\forall(i, a)$. As $\{U_a\}$ and $\{V_a\}$ are orthonormal, the projection matrix S_{ia} is computed by the following rule:

$$S_{ia} = U_a^T P_i V_a. \quad (4)$$

Note that this is the solution to the least squares energy function $\|P_i - U_a S_{ia} V_a^T\|^2$. Then $m_1 m_2 - T$ elements in S_{ia} with least absolute value are nullified to get the best sparse projection matrix. The updates to U_a and V_a are obtained as follows. Denoting the sum of the terms of the cost function in Eqn. (1) that are independent of U_a as C, we rewrite the cost function as follows:

$$E(\{U_a, V_a, S_{ia}, M_{ia}\}) = \sum_{i=1}^{N} \sum_{a=1}^{K} M_{ia} \|P_i - U_a S_{ia} V_a^T\|^2 +$$
$$\sum_a \text{trace}[\Lambda_{1a}(U_a^T U_a - I)] + C$$
$$= \sum_{i=1}^{N} \sum_{a=1}^{K} M_{ia}[\text{trace}(P_i - U_a S_{ia} V_a^T)^T (P_i - U_a S_{ia} V_a^T))] +$$
$$\sum_a \text{trace}[\Lambda_{1a}(U_a^T U_a - I)] + C$$
$$= \sum_{i=1}^{N} \sum_{a=1}^{K} M_{ia}[\text{trace}(P_i^T P_i) - 2\text{trace}(P_i^T U_a S_{ia} V_a^T)$$
$$+ \text{trace}(S_{ia}^T S_{ia})] + \sum_a \text{trace}(\Lambda_{1a}(U_a^T U_a - I)) + C. \quad (5)$$

Now setting derivatives w.r.t. U_a to zero, re-arranging the terms and eliminating the Lagrange matrices, we obtain the following update rule for U_a:

$$Z_{1a} = \sum_i M_{ia} P_i V_a S_{ia}^T; U_a = Z_{1a}(Z_{1a}^T Z_{1a})^{-\frac{1}{2}}. \quad (6)$$

An SVD of Z_{1a} will give us $Z_{1a} = \Gamma_{1a} \Psi \Upsilon_{1a}^T$ where Γ_{1a} and Υ_{1a} are orthonormal matrices and Ψ is a diagonal matrix. Using this, we can re-write U_a as follows:

$$U_a = (\Gamma_{1a} \Psi \Upsilon_{1a}^T)((\Gamma_{1a} \Psi \Upsilon_{1a}^T)^T \Gamma_{1a} \Psi \Upsilon_{1a}^T))^{-\frac{1}{2}} = \Gamma_{1a} \Upsilon_{1a}^T. \quad (7)$$

Notice that the steps above followed only owing to the fact that U_a and hence Γ_{1a} and Υ_{1a} were full-rank matrices. This gives us an update rule for U_a. A very similar update rule for

V_a follows along the same lines. The membership values are obtained by:

$$M_{ia} = \frac{e^{-\beta \|P_i - U_a S_{ia} V_a^T\|^2}}{\sum_{b=1}^{K} e^{-\beta \|P_i - U_b S_{ib} V_b^T\|^2}}. \quad (8)$$

The matrices $\{S_{ia}, U_a, V_a\}$ and M are then updated sequentially following one another for a fixed β value, until convergence. The value of β is then increased and the sequential updates are repeated. The entire process is repeated until an integrality condition is met.

Our algorithm could be modified to learn a set of orthonormal bases (single matrices and not a pair) for sparse representation of vectorized images, as well. Note that even then, our technique should not be confused with the 'mixtures of PCA' approach [24]. The emphasis of the latter is again on low-rank approximations and not on sparsity. Furthermore, in our algorithm we do not compute an actual probabilistic mixture (also see Sections II-C and III-E). However, we have not considered this vector-based variant in our experiments, because it does not exploit the fact that image patches are two-dimensional entities.

C. Application to Compact Image Representation

Our framework is geared towards compact but *low-error* patch reconstruction. We are not concerned with the *discriminating* assignment of a *specific kind* of patches to a *specific* exemplar, quite unlike in a clustering or classification application. In our method, after the optimization, each training patch P_i $(1 \leq i \leq N)$ gets represented as a projection onto one out of the K exemplar orthonormal bases, which produces the least reconstruction error, i.e. the kth exemplar is chosen if $\|P_i - U_k S_{ik} V_k^T\|^2 \leq \|P_i - U_a S_{ia} V_a^T\|^2, \forall a \in \{1, 2, ..., K\}, 1 \leq k \leq K$. For patch P_i, we denote the corresponding 'optimal' projection matrix as $S_i^\star = S_{ik}$, and the corresponding exemplar as $(U_i^\star, V_i^\star) = (U_k, V_k)$. Thus the entire training set is approximated by (1) the *common* set of basis-pairs $\{(U_a, V_a)\}, 1 \leq a \leq K$ $(K \ll N)$, and (2) the optimal sparse projection matrices $\{S_i^\star\}$ for each patch, with at most T non-zero elements each. The overall storage per image is thus greatly reduced (see also section III-B). Furthermore, these bases $\{(U_a, V_a)\}$ can now be used to encode patches from a new set of images that are somewhat similar to the ones existing in the training set. However, a practical application demands that the reconstruction meet a specific error threshold on unseen patches, and hence the L_0 norm of the projection matrix of the patch is adjusted dynamically in order to meet the error. Experimental results using such a scheme are described in the next section.

III. EXPERIMENTS: 2D IMAGES

In this section, we first describe the overall methodology for the training and testing phases of our algorithm based on our earlier work in [11]. As we are working on a compression application, we then describe the details of our image coding and quantization scheme. This is followed by a discussion of the comparisons between our technique and other competitive techniques (including JPEG), and an enumeration of the experimental results.

A. Training and Testing Phases

We tested our algorithm for compression of the entire ORL database [25] and the entire Yale database [26]. We divided the images in each database into patches of fixed size (12×12), and these sets of patches were segregated into training and test sets. The test set consisted of many more images than the training set, in the case of either database. For the purpose of training, we learned a total of $K = 50$ orthonormal bases using a fixed value of T to control the sparsity. For testing, we projected each patch P_i onto that exemplar $(U_i^\star, V_i^\star)$ which produced the *sparsest* projection matrix $S_i^\star$ that yielded an average per-pixel reconstruction error $\frac{\|P_i - U_i^\star S_i^\star V_i^{\star T}\|^2}{m_1 m_2}$ of no more than some chosen δ. Note that different test patches required different T values, depending upon their inherent 'complexity'. Hence, we varied the sparsity of the projection matrix (but keeping its size fixed to 12×12), by greedily nullifying the smallest elements in the matrix, without letting the reconstruction error go above δ. This gave us the flexibility to adjust to patches of different complexities, without altering the rank of the exemplar bases $(U_i^\star, V_i^\star)$. As any patch P_i is projected onto exemplar orthonormal bases that are different from those produced by its own SVD, the projection matrices turn out to be non-diagonal. Hence, there is no such thing as a hierarchy of 'singular values' as in ordinary SVD. As a result, we cannot resort to restricting the rank of the projection matrix (and thereby the rank of $(U_i^\star, V_i^\star)$) to adjust for patches of different complexity (unless we learn a separate set of K bases, each set for a different rank r where $1 < r < \min(m_1, m_2)$, which would make the training and even the image coding [see Section III-B] very cumbersome). *This highlights an advantage of our approach over that of algorithms that adjust the rank of the projection matrices.*

B. Details of Image Coding and Quantization

We obtain $S_i^\star$ by sparsifying $U_i^{\star T} P_i V_i^\star$. As $U_i^\star$ and $V_i^\star$ are orthonormal, we can show that the values in $S_i^\star$ will always lie in the range $[-m, m]$ for $m \times m$ patches, if the values of P_i lie in $[0, 1]$. This can be proved as follows: We know that $S_i^\star = U_i^{\star T} P_i V_i^\star$. Hence we write the element of $S_i^\star$ in the ath row and b^{th} column as follows:

$$S_{i\ ab}^\star = \sum_{k,l} U_{i\ ak}^{\star T} P_{ikl} V_{lb}^\star \leq \sum_{kl} \frac{1}{\sqrt{m}} P_{ikl} \frac{1}{\sqrt{m}} = m. \quad (9)$$

The first step follows because the maximum value of $S_{i\ ab}^\star$ is obtained when all values of $U_{i\ ak}^{\star T}$ are equal to $\sqrt{m}$. The second step follows because $S_{i\ ab}^\star$ will meet its upper bound when all the m^2 values in the patch are equal to 1. We can similarly prove that the lower bound on $S_{i\ ab}^\star$ is $-m$. As in our case, $m = 12$, the upper and lower bounds are $+12$ and -12 respectively.

In our experiment, we Huffman-encoded the integer parts of the values in the $\{S_i^\star\}$ matrices over the whole image (giving us an average of some Q_1 bits per entry) and quantized the fractional parts with Q_2 bits per entry. Thus, we needed to store the following information per test-patch to create the compressed image: (1) the index of the best exemplar, using a_1 bits, (2) the location and value of each non-zero element

in its $S_i^\star$ matrix, using a_2 bits per location and $Q_1 + Q_2$ bits for the value, and (3) the number of non-zero elements per patch encoded using a_3 bits. Hence the total number of bits per pixel for the whole image is given by:

$$RPP = \frac{N(a_1 + a_3) + T^{whole}(a_2 + Q_1 + Q_2)}{M_1 M_2}. \quad (10)$$

where $T^{whole} = \sum_{i=1}^N \|S_i^\star\|_0$. The values of a_1, a_2 and a_3 were obtained by Huffman encoding.

After performing quantization on the values of the coefficients in $S_i^\star$, we obtained new quantized projection matrices, denoted as $\hat{S}_i^\star$. Following this, the PSNR for each image was measured as $10 \log_{10} \frac{N m_1 m_2}{\sum_{i=1}^N \|P_i - U_i^\star \hat{S}_i^\star V_i^{\star T}\|^2}$, and then averaged over the entire test set. The average number of bits per pixel (RPP) was calculated as in Eqn. (10) for each image, and then averaged over the whole test set. We repeated this procedure for different δ values from 8×10^{-5} to 8×10^{-3} (range of image intensity values was $[0, 1]$) and plotted an ROC curve of average PSNR vs. average RPP.

C. Comparison with other techniques:

We pitted our method against four recent and competitive approaches: (1) KSVD, (2) overcomplete DCT, (3) SSVD and (4) the JPEG standard. (1) For the KSVD algorithm from [9], we used 441 unit norm dictionary vectors of size 144. In this method, each patch is vectorized and represented as a sparse linear combination of the dictionary vectors. During training, the value of T is kept fixed, and during testing it is dynamically adjusted depending upon the patch so as to meet the appropriate set of error thresholds represented by $\delta \in [8 \times 10^{-5}, 8 \times 10^{-3}]$. For the purpose of comparison between our method and KSVD, we used the exact same values of T and δ as in our algorithm, to facilitate a fair comparison. The method used to find the sparse projection onto the dictionary was the orthogonal matching pursuit (OMP) algorithm [21]. (2) We created the over-complete DCT dictionary with 441 unit norm vectors of size 144 by sampling cosine waves of various frequencies, again with the same T value as for KSVD, and using the OMP technique for sparse projection. (3) The SSVD method from [19] is not a learning based technique. It is based on the creation of sub-sampled versions of the image by traversing the image in a manner different from usual raster-scanning. (4) For the JPEG standard (for which we used the implementation provided in MATLAB®), we measured the RPP from the number of bytes for storage of the file.

For KSVD and over-complete DCT, there exist bounds on the values of the coefficients that are very similar to those for our technique, as presented in Eqn. (9). For both these methods, the RPP value was computed using the formula in [9], Eqn. (27), with the modification that the integer parts of the coefficients of the linear combination were Huffman-encoded and the fractional parts separately quantized (as it gave a better ROC curves for these methods). For the SSVD method, the RPP was calculated as in [19], section (5).
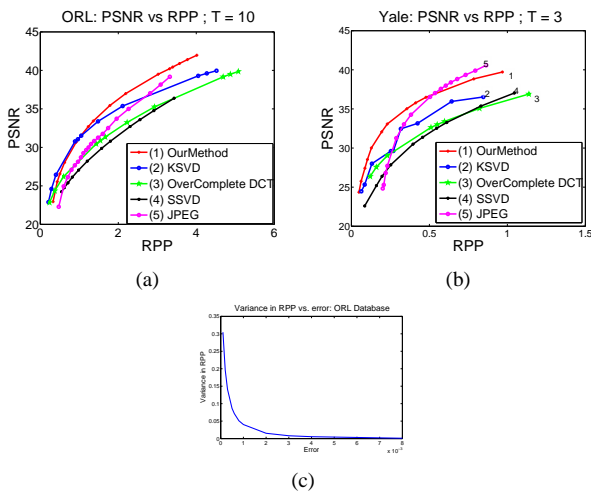
Fig. 1. ROC curves on (a) ORL and (b) Yale Databases. Legend- Red (1): Our Method, Blue (2): KSVD, Green (3): Over-complete DCT, Black (4): SSVD and (5): JPEG (Magenta). (c) Variance in RPP versus pre-quantization error for ORL. *These plots are viewed best in color.*
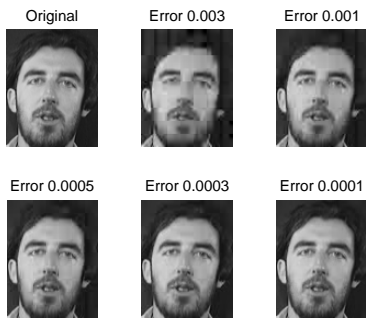

Fig. 2. Reconstructed versions of an image from the ORL database using 5 different error values. The original image is on the extreme left and top corner. Viewed better by ZOOMING into the pdf file.

D. Results

For the ORL database, we created a training set of patches of size 12×12 from images of 10 different people, with 10 images per person. Patches from images of the remaining 30 people (10 images per person) were treated as the test set. From the training set, a total of 50 pairs of orthonormal bases were learned using the algorithm described in Section II-B. The T value for sparsity of the projection matrices was set to 10 during training. As shown in Figure III-D(c), we also computed the variance in the RPP values for every different pre-quantization error value, for each image in the ORL database. Note that the variance in RPP decreases with increase in specified error. This is because at very low errors, different images require different number of coefficients in order to meet the error.

The same experiment was run on the Yale database with a value of $T = 3$ on patches of size 12×12. The training set consisted of a total of 64 images of one and the same person under different illumination conditions, whereas the testing set consisted of 65 images each of the remaining 38 people from the database (i.e. 2470 images in all), under different

Method	Dictionary Size (number of scalars)
Our Method	$50 \times 2 \times 12 \times 12 = 14400$
KSVD	$441 \times 12 \times 12 = 63504$
Overcomplete DCT	$441 \times 12 \times 12 = 63504$

TABLE I
COMPARISON OF DICTIONARY SIZE FOR VARIOUS METHODS FOR
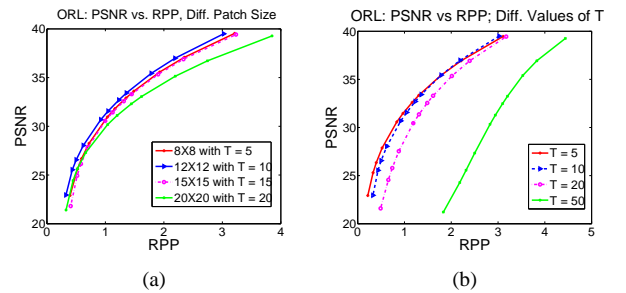EXPERIMENTS ON THE ORL AND YALE DATABASES.

Fig. 3. ROC curves for the ORL database using our technique with (a) different patch size and (b) different value of T for a fixed patch size of 12×12. *These plots are viewed best in color.*

illumination conditions. The ROC curves for our method were superior to those of other methods over a significant range of δ, for the ORL as well as the Yale database, as seen in Figures 1(a) and 1(b). Figure 2 shows reconstructed versions of an image from the ORL database using our method with different values of δ. For experiments on the ORL database, the number of bits used to code the fractional parts of the coefficients of the projection matrices [i.e. Q_2 in Eqn. (10)] was set to 5. For the Yale database, we often obtained pre-quantization errors significantly less than the chosen δ, and hence using a value of Q_2 less than 5 bits often did not raise the post-quantization error above δ. Keeping this in mind, we varied the value of Q_2 dynamically, for each pre-quantization error value. The same variation was applied to the KSVD and over-complete DCT techniques as well. For the experiments on these two databases, we also summarize the dictionary size of all the relevant methods being compared in Table I.

E. Effect of Different Parameters on the Performance of our Method:

The different parameters in our method include: (1) the size of the patches for training and testing, (2) the number of pairs of orthonormal bases, i.e. K, and (3) the sparsity of each patch during training, i.e. T. In the following, we describe the effect of varying these parameters: (1) *Patch Size:* If the size of the patch is too large, it becomes more and more unlikely that a fixed number of orthonormal matrices will serve as adequate bases for these patches in terms of a low-error sparse representation, owing to the curse of dimensionality. Furthermore, if the patch size is too large, any algorithm will lose out on the advantages of locality and simplicity. However, if the patch size is too small (say 2×2), there is very little a compression algorithm can do in terms of lowering the number of bits required to store these patches. We stress that the patch size as a free parameter is something common to *all* algorithms

to which we are comparing our technique (including JPEG). Also, the choice of this parameter is mainly empirical. We tested the performance of our algorithm with $K = 50$ pairs of orthonormal bases on the ORL database, using patches of size 8×8, 12×12, 15×15 and 20×20, with appropriately different values of T. As shown in Figure 3(a), the patch size of 12×12 using $T = 10$ yielded the best performance, though other patch sizes also performed quite well. (2) *The number of orthonormal bases K:* The choice of this number is not critical, and can be set to as high a value as desired without affecting the accuracy of the results. This parameter should not be confused with the number of mixture components in standard mixture-model based density estimation, because in our method each patch gets projected only onto a *single* set of orthonormal bases, i.e. we do not compute a combination of projections onto all the K different orthonormal basis pairs. The only down-side of a higher value of K is the added computational cost during training. Again note that this parameter will be part of any learning-based algorithm for compression that uses a set of bases to express image patches. (3) *The value of T during training:* The value of T is fixed only during training, and is varied for each patch during the testing phase so as to meet the required error threshold. A very high value of T during training can cause the orthonormal basis pairs to overfit to the training data (variance), whereas a very low value could cause a large bias error. This is an instance of the well-known bias-variance tradeoff common in machine learning algorithms. Our choice of T was empirical, though the value of $T = 10$ performed very well on nearly all the datasets we ran our experiments on. The effect of different values of T on the compression performance for the ORL database is shown in Figure 3(b). We again emphasize that the issue with the choice of an 'optimal' T is not an artifact of our algorithm *per se*. For instance, this issue of a choice of T will also be encountered in pursuit algorithms to find the approximately optimal linear combination of unit vectors from a dictionary. In the latter case, however, there will also be the problem of poorer and poorer approximation errors as T increases, under certain conditions on the dictionary of overcomplete basis vectors.

F. Performance on Random Collections of Images

We performed additional experiments to study the behavior of our algorithm when the training and test sets were very different from one another. To this end, we used the database of 155 natural images (in uncompressed format) from the Computer Vision Group at the University of Granada [27]. The database consists of images of faces, houses, natural scenery, inanimate objects and wildlife. We converted all the given images to grayscale. Our training set consisted of patches (of size 12×12) from 11 images. Patches from the remaining images were part of the test set. Though the images in the training and test sets were very different, our algorithm produced excellent results superior to JPEG upto an RPP value of 3.1 bits, as shown in Figure 4(a). Two training images and reconstructed versions of a test image for error values of 0.0002, 0.0006 and 0.002 respectively, are shown
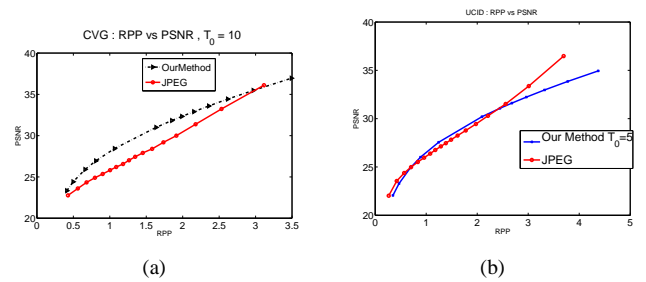

Fig. 4. PSNR vs. RPP curve for our method and JPEG on (a) the CVG Granada database and (b) the UCID database.
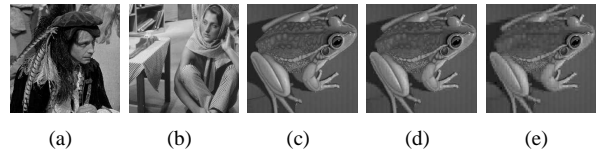

Fig. 5. (a) and (b): Training images from the CVG Granada database. (c) to (e): Reconstructed versions of a test image for error values of 0.0002, 0.0006 and 0.002 respectively. Viewed better by ZOOMING in the pdf file.

in Figure III-F. To test the effect of minute noise on the performance of our algorithm, a similar experiment was run on the UCID database [28] with a training set of 10 images and the remaining 234 test images from the database. The test images were significantly different from the ones used for training. The images were scaled down by a factor of two (i.e. to 320×240 instead of 640×480) for faster training and subjected to zero mean Gaussian noise of variance 0.001 (on a scale from 0 to 1). Our algorithm was yet again competitive with JPEG as shown in Figure 4(b).

G. Comparison with Wavelet Coding

In this section, we present experiments comparing the performance of our method with JPEG2000 and also with a wavelet-based compression method involving a very simple quantization scheme that employs Huffman encoding, similar to the quantization scheme we employed for our technique. We would like to emphasize that JPEG2000 is an industry standard that was developed by several researchers for over a decade. Its strongest point is its superlative quantization and encoding scheme, rather than the mere fact that it uses wavelet bases [29]. On the other hand, a wavelet-coding method employing a much simpler quantization scheme (similar to the one we used in our technique) performs quite poorly in practice (much worse than JPEG), with one of its major problems being that there are no bounds that can be imposed on the values of the wavelet coefficients in terms of the image intensity range or patch size. In our experiments, the typical range of coefficient values turned out to be around 10^4. As opposed to this, the values of the coefficients of the projected matrices in our technique are (provably) bounded between $-m$ and $+m$ for patch sizes of $m \times m$ (as shown in Section (3B)).

In our wavelet coding strategy, we performed a decomposition of upto four levels (level 1 giving us higher RPP and higher PSNR, and level 4 giving us low RPP and low PSNR). For any decomposition, the LL sub-band coefficients (also

called the approximation coefficients) of the final level were never thresholded. The integer parts of these coefficients were Huffman encoded (just as in our technique) and 4 bits were allocated for the fractional part. Having more bits for the fractional part did not improve the ROC curves in our experiments. The coefficients of the other sub-bands (HH, HL, LH) at every level were thresholded using the commonly employed Birge-Massart strategy (part of the compression implementation in the MATLAB wavelet toolbox). Different thresholds were used for each subband at each level. Again, the thresholded values were coded using Huffman encoding on the integer part followed by 4 bits for the floating point. The specific type of wavelet that we used was 'sym4' as it gave better performance than other wavelets that we tried (such as 'db7', biorthogonal wavelets etc).

We performed experiments comparing the compression performance of the following four techniques: our method, JPEG, JPEG2000 (the specific implementation being the commonly available Jasper package [30]), wavelet methods using 'sym4' followed by a simple quantizer (as decribed earlier). In Figure 6, we show results on ORL and Yale databases. For the ORL database, we also present results with db7 and biorthogonal wavelets. As these performed much worse than sym4, we do not show more results with them. As seen in Figure 6 (a) and (b), our method as well as JPEG (using DCT) outperformed the simple wavelet based techniques though not JPEG2000. Note that there are other reports in the literature [31] that indicate a superior performance of JPEG over wavelet-based methods that did not use sophisticated quantizers, especially for higher bit rates.

Thus we believe that our results will improve further if a more sophisticated quantizer/coder is used in conjunction with bases learned by our method. The improvement of the quantization scheme for our technique is well beyond the scope of the current paper. Nevertheless, we observed that the performance of our technique was competitive with JPEG2000 for low bit rates, when negligible amounts of noise (i.i.d. zero-mean Gaussian noise of variance 8×10^{-4} or 10^{-3}) were added to the images. It should be noted that the visual effect of such small amounts of noise is not obvious to the human eye. At lower bit rates, we observed that our method sometimes beat JPEG2000. This is despite the fact that we *trained our method on noiseless images, and tested on noisy images.* and the size of the test size was *several times larger than the size of the training set.* The results of our method, JPEG, JPEG2000 and wavelet coding with sym4 wavelets are shown in Figure 6(c) to (f), for two levels of noise on the ORL and Yale databases.

IV. THEORY: 3D (OR HIGHER-D) IMAGES

We now consider a set of images represented as third-order matrices, each of size $M_1 \times M_2 \times M_3$. We divide each image into non-overlapping patches of size $m_1 \times m_2 \times m_3, m_1 \ll M_1, m_2 \ll M_2, m_3 \ll M_3$, and treat each patch as a separate tensor. Just as before, we start by exploiting the similarity inherent in these patches, and represent them by means of sparse projections onto a triple of exemplar orthonormal bases learned from a set of training image patches. Note that our
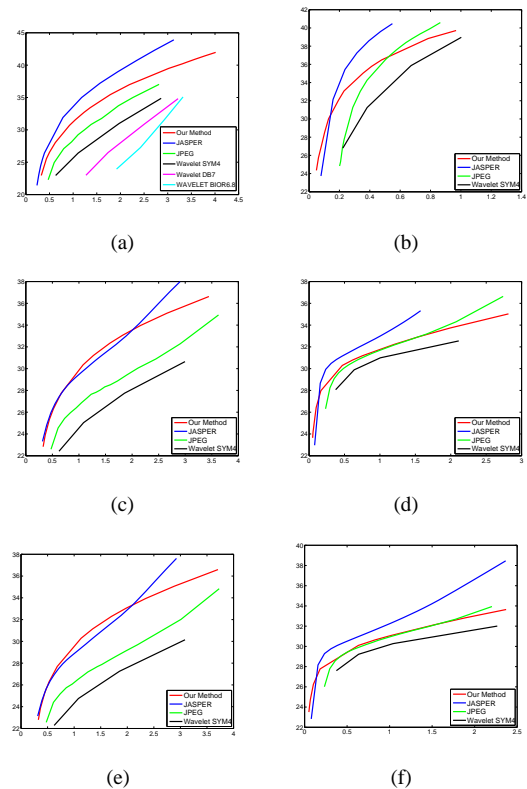

Fig. 6. PSNR vs. RPP curve for our method, JPEG, JPEG2000 (JASPER) and wavelet coding with 'sym4' wavelets on (a) ORL Database and (b) Yale Database (with no noise). (c) and (d): with iid Gaussian noise of mean zero and variance 8×10^{-4}, (e) and (f): with iid Gaussian noise of mean zero and variance 10^{-3}. Image are best viewed by ZOOMING into the pdf file.

extension to higher dimensions is non-trivial, especially given that a key property of the SVD in 2D (namely that SVD provides the optimal lower-rank reconstruction by simple nullification of the lowest singular values) does not extend into higher-dimensional analogs such as HOSVD [13]. In the following, we now describe the mathematical structure of the exemplars. Note that though the derivations presented in this paper are for 3D matrices, a very similar treatment is applicable to learning n-tuples of orthonormal bases for patches from n-D matrices (where $n \geq 4$).

A. Exemplar Bases and Sparse Projections

Let $P \in R^{m_1 \times m_2 \times m_3}$ be an image patch. Using HOSVD, we can represent P as a combination of orthonormal bases $U \in R^{m_1 \times m_1}$, $V \in R^{m_2 \times m_2}$ and $W \in R^{m_3 \times m_3}$ in the form $P = S \times_1 U \times_2 V \times_3 W$, where $S \in R^{m_1 \times m_2 \times m_3}$ is termed the core-tensor. The operators $\times_i$ refer to tensor-matrix multiplication over different axes. The core-tensor has special properties such as all-orthogonality and ordering. See [13] for more details. Now, P can also be represented as a combination of *any* set of orthonormal bases $\bar{U}$, $\bar{V}$ and $\bar{W}$, different from those obtained from the HOSVD of P. In this case, we have $P = S \times_1 \bar{U} \times_2 \bar{V} \times_3 \bar{W}$ where S is not guaranteed to be an all-orthogonal tensor, nor is it guaranteed to obey the ordering property.

As the HOSVD does not necessarily provide the best *low-*

rank approximation to a tensor [13], we choose to depart from this notion (instead of settling for a sub-optimal approximation), and instead answer the following question: What *sparse* tensor $Q \in R^{m_1 \times m_2 \times m_3}$ will reconstruct P from a triple of orthonormal bases $(\bar{U}, \bar{V}, \bar{W})$ with the least error $\|P - Q \times_1 \bar{U} \times_2 \bar{V} \times_3 \bar{W}\|^2$? Sparsity is again quantified by an upper bound T on the L_0 norm of Q (denoted as $\|Q\|_0$). We now prove that the *optimal* Q with this sparsity constraint is obtained by nullifying the least (in absolute value) $m_1 m_2 m_3 - T$ elements of the estimated projection tensor $S = P \times_1 \bar{U}^T \times_2 \bar{V}^T \times_3 \bar{W}^T$. Due to the ortho-normality of $\bar{U}, \bar{V}$ and $\bar{W}$, *this simple greedy algorithm turns out to be optimal* (see Theorem 2).

Theorem 2: Given a triple of orthonormal bases $(\bar{U}, \bar{V}, \bar{W})$, the optimal sparse projection tensor Q with $\|Q\|_0 = T$ is obtained by setting to zero $m_1 m_2 m_3 - T$ elements of the tensor $S = P \times_1 \bar{U}^T \times_2 \bar{V}^T \times_3 \bar{W}^T$ having least absolute value.

Proof: We have $P = S \times_1 \bar{U} \times_2 \bar{V} \times_3 \bar{W}$. The error in reconstructing a patch P using some other matrix Q is $e = \|(S - Q) \times_1 \bar{U} \times_2 \bar{V} \times_3 \bar{W}\|^2$. For any tensor X, we have $\|X\|^2 = \|X_{(n)}\|^2$ (i.e. the Frobenius norm of the tensor and its nth unfolding are the same [13]). Also, by the matrix representation of HOSVD, we have $X_{(n)} = \bar{U} \cdot S_{(n)} \cdot (\bar{V} \otimes \bar{W})^5$. Hence, it follows that $e = \|\bar{U} \cdot (S - Q)_{(1)} \cdot (\bar{V} \otimes \bar{W})^T\|^2$. This gives us $e = \|S - Q\|^2$. The last step follows because $\bar{U}, \bar{V}$ and $\bar{W}$, and hence $\bar{V} \otimes \bar{W}$ are orthonormal matrices. Let $I_1 = \{(i, j, k) | Q_{ijk} = 0\}$ and $I_2 = \{(i, j, k) | Q_{ijk} \neq 0\}$. Then $e = \sum_{(i,j,k) \in I_1} S_{ijk}^2 + \sum_{(i,j,k) \in I_2} (S_{ijk} - Q_{ijk})^2$. This error will be minimized when $S_{ijk} = Q_{ijk}$ in all locations where $Q_{ijk} \neq 0$ and $Q_{ijk} = 0$ at those indices where the corresponding values in S are as small as possible. Thus if we want $\|Q\|_0 = T$, then Q is the tensor obtained by nullifying $m_1 m_2 m_3 - T$ entries from S that have the least absolute value and leaving the remaining elements intact.$\square$

We wish to re-emphasize that a key feature of our approach is the fact that the same technique used for 2D images scales to higher dimensions. Tensor decompositions such as HOSVD do not share this feature, because the optimal low-rank reconstruction property for SVD does not extend to HOSVD. Furthermore, though the upper and lower error bounds for core-tensor truncation in HOSVD derived in [18] are interesting, they are applicable only when the entire set of images has a common basis (i.e. a common U, V and W matrix), which may not be sufficient to compactly account for the variability in real datasets.

B. Learning the Bases

We now describe a method to learn K exemplar orthonormal bases $\{(U_a, V_a, W_a)\}$, $1 \leq a \leq K$, to encode a training set of N image patches $P_i \in R^{m_1 \times m_2 \times m_3}$ ($1 \leq i \leq N$) with least possible error (in the L_2 norm sense). Note that $K \ll N$. In addition, we impose a sparsity constraint that every S_{ia} (the tensor used to reconstruct P_i from (U_a, V_a, W_a)) has at most T non-zero elements. The main objective function to be minimized is:

$$E(\{U_a, V_a, W_a, S_{ia}, M_{ia}\}) =$$
$$\sum_{i=1}^{N} \sum_{a=1}^{K} M_{ia} \|P_i - S_{ia} \times_1 U_a \times_2 V_a \times_3 W_a\|^2 \quad (11)$$

subject to the following constraints:

(1) $U_a^T U_a = V_a^T V_a = W_a^T W_a = I, \forall a$. (2) $\|S_{ia}\|_0 \leq T, \forall (i, a)$.
$$(3) \sum_a M_{ia} = 1, \forall i, \text{ and } M_{ia} \in \{0, 1\}, \forall i, a. \quad (12)$$

Here M_{ia} is a binary matrix of size $N \times K$ which indicates whether the ith patch belongs to the space defined by (U_a, V_a, W_a). Just as for the 2D case, we relax the binary membership constraint so that now $M_{ia} \in (0, 1), \forall (i, a)$, subject to $\sum_{a=1}^{K} M_{ia} = 1, \forall i$. Using Lagrange parameters $\{\mu_i\}$, symmetric Lagrange matrices $\{\Lambda_{1a}\}, \{\Lambda_{2a}\}$ and $\{\Lambda_{3a}\}$, and a temperature parameter β, we obtain the following deterministic annealing objective function:

$$E(\{U_a, V_a, W_a, S_{ia}, M_{ia}\}) =$$
$$\sum_{i=1}^{N} \sum_{a=1}^{K} M_{ia} \|P_i - S_{ia} \times_1 U_a \times_2 V_a \times_3 W_a\|^2 +$$
$$\frac{1}{\beta} \sum_{ia} M_{ia} \log M_{ia} + \sum_i \mu_i (\sum_a M_{ia} - 1) +$$
$$\sum_a \text{trace}(\Lambda_{1a}(U_a^T U_a - I)) +$$
$$\sum_a \text{trace}(\Lambda_{2a}(V_a^T V_a - I)) + \sum_a \text{trace}(\Lambda_{3a}(W_a^T W_a - I)). \quad (13)$$

We first initialize $\{U_a\}, \{V_a\}$ and $\{W_a\}$ to random orthonormal tensors $\forall a$, and $M_{ia} = \frac{1}{K}, \forall (i, a)$. Using the fact that $\{U_a\}, \{V_a\}$ and $\{W_a\}$ are orthonormal, the projection matrix S_{ia} is computed by the rule:

$$S_{ia} = P_i \times_1 U_a^T \times_2 V_a^T \times_3 W_a^T, \forall (i, a). \quad (14)$$

This is the minimum of the energy function $\|P_i - S_{ia} \times_1 U_a \times_2 V_a \times_3 W_a\|^2$. Then $m_1 m_2 m_3 - T$ elements in S_{ia} with least absolute value are nullified. Thereafter, U_a, V_a and W_a are updated as follows. Denote the sum of the terms in the previous energy function that are independent of U_a, as C, denoting the the nth unfolding of the tensor P_i as $P_{i(n)}$ (details in [13]). Using the fact that $\|X\|^2 = \|X_{(n)}\|^2$ for any tensor X, we write the energy function as:

$$E(\{U_a, V_a, W_a, S_{ia}, M_{ia}\}) =$$
$$\sum_{ia} M_{ia} \|P_i - (S_{ia} \times_1 U_a \times_2 V_a \times_3 W_a)\|^2 +$$
$$\sum_a \text{trace}(\Lambda_{1a}(U_a^T U_a - I)) + C \quad (15)$$

---

[5] Here $A \otimes B$ refers to the Kronecker product of matrices $A \in R^{E_1 \times E_2}$ and $B \in R^{F_1 \times F_2}$, which is given as $A \otimes B = (A_{e_1 e_2} B)_{1 \leq e_1 \leq E_1, 1 \leq e_2 \leq E_2}$.

This produces:

$$E(\{U_a, V_a, W_a, S_{ia}, M_{ia}\})$$
$$= \sum_{ia} M_{ia} \|P_{i(1)} - U_a \cdot S_{ia(1)} \cdot (V_a \otimes W_a)^T\|^2 +$$
$$\sum_a \text{trace}(\Lambda_{1a}(U_a^T U_a - I)) + C$$
$$= \sum_{ia} M_{ia}[\text{trace}(P_{i(1)}^T P_{i(1)}) -$$
$$2\text{trace}(P_{i(1)}^T U_a \cdot S_{ia(1)} \cdot (V_a \otimes W_a)^T) +$$
$$\text{trace}(S_{ia(1)}^T S_{ia(1)})] + \sum_a \text{trace}(\Lambda_{1a}(U_a^T U_a - I)) + C. \quad (16)$$

Setting the derivative of E w.r.t. U_a to zero, and eliminating Lagrange matrices, we have an update rule for U_a:

$$Z_{Ua} = \sum_i M_{ia} P_{i(1)}(V_a \otimes W_a) S_{ia(1)}^T;$$
$$U_a = Z_{Ua}(Z_{Ua}^T Z_{Ua})^{-\frac{1}{2}} = \Gamma_{1a} \Upsilon_{1a}^T. \quad (17)$$

Here Γ_{1a} and Υ_{1a} are orthonormal matrices obtained from the SVD of Z_{Ua}. V_a and W_a are updated similarly by second and third unfolding of the tensors respectively. The membership values are obtained by:

$$M_{ia} = \frac{e^{-\beta\|P_i - S_{ia} \times_1 U_a \times_2 V_a \times_3 W_a\|^2}}{\sum_{b=1}^K e^{-\beta\|P_i - S_{ib} \times_1 U_b \times_2 V_b \times_3 W_b\|^2}}. \quad (18)$$

The core tensors $\{S_{ia}\}$ and the matrices $\{U_a, V_a, W_a\}, M$ are then updated sequentially following one another for a fixed β value, until convergence. The value of β is then increased and the sequential updates are repeated. The entire process is repeated until an integrality condition is met.

C. Application to Compact Image Representation

Quite similar to the 2D case, after the optimization during the training phase, each training patch P_i ($1 \leq i \leq N$) gets represented as a projection onto one out of the K exemplar orthonormal bases, which produces the least reconstruction error, i.e. the kth exemplar is chosen if $\|P - S_{ik} \times_1 U_k \times_2 V_k \times_3 W_k\|^2 \leq \|P - S_{ia} \times_1 U_a \times_2 V_a \times_3 W_a\|^2, \forall a \in \{1, 2, ..., K\}, 1 \leq k \leq K$. For patch training P_i, we denote the corresponding 'optimal' projection tensor as $S_i^\star = S_{ik}$, and the corresponding exemplar as $(U_i^\star, V_i^\star, W_i^\star) = (U_k, V_k, W_k)$. Thus the entire set of patches can be closely approximated by (1) the *common* set of basis-pairs $\{(U_a, V_a, W_a)\}, 1 \leq a \leq K$ ($K \ll N$), and (2) the optimal sparse projection tensors $\{S_i^\star\}$ for each patch, with at most T non-zero elements each. The overall storage per image is thus greatly reduced. Furthermore, these bases $\{(U_a, V_a, W_a)\}$ can now be used to encode patches from a new set of images that are similar to the ones existing in the training set, though just as in the 2D case, the sparsity of the patch will be adjusted dynamically in order to meet a given error threshold. Experimental results for this are provided in the next section.

V. EXPERIMENTS: COLOR IMAGES

In this section, we describe the experiments we performed on color images represented in the RGB color scheme. Each image of size $M_1 \times M_2 \times 3$ was treated as a third-order matrix. Our compression algorithm was tested on the GaTech Face Database [32], which consists of RGB color images with 15 images each of 50 different people. The images in the database are already cropped to include just the face, but some of them contain small portions of a distinctly cluttered background. The average image size is 150×150 pixels, and all the images are in the JPEG format. For our experiments, we divided this database into a training set of one image each of 40 different people, and a test set of the remaining 14 images each of these 40 people, and all 15 images each of the remaining 10 people. Thus, the size of the test set was 710 images. The patch-size we chose for the training and test sets was $12 \times 12 \times 3$ and we experimented with $K = 100$ different orthonormal bases learned during training. The value of T during training was set to 10. At the time of testing, for our method, each patch was projected onto that triple of orthonormal bases ($U_i^\star, V_i^\star, W_i^\star$) which gave the sparsest projection tensor $S_i^\star$ such that the per-pixel reconstruction error $\frac{\|P_i - S_i \times_1 U_i^\star \times_2 V_i^\star \times_3 W_i^\star\|^2}{m_1 m_2}$ was no greater than a chosen δ. Note that, in calculating the per-pixel reconstruction error, we did not divide by the number of channels, i.e. 3, because at each pixel, there are three values defined. We experimented with different reconstruction error values δ ranging from 8×10^{-5} to 8×10^{-3}. Following the reconstruction, the PSNR for the entire image was measured, and averaged over the entire test test. The total number of bits per pixel, i.e. RPP, was also calculated for each image and averaged over the test set. The details of the training and testing methodology, and also the actual quantization and coding step are the same as presented previously in Section III-A and III-B.

A. Comparisons with Other Methods

The results obtained by our method were compared to those obtained by four techniques: (1) KSVD, (2) Our algorithm on 2D with separate channels and (3) the JPEG standard. (1) For KSVD, we used patches of size $12 \times 12 \times 3$ reshaped to give vectors of size 432, and used these to train a dictionary of 1340 vectors using a value of $T = 30$. (2) For our algorithm for 2D images from Section II with an independent (separate) encoding of each of the three channels. As an independent coding of the R, G and B slices would fail to account for the inherent correlation between the channels (and hence give inferior compression performance), we used principal components analysis (PCA) to find the three principal components of the R, G, B values of each pixel from the training set. The R, G, B pixel values from the test images were then projected onto these principal components to give a transformed image in which the values in each of the different channels are decorrelated. A similar approach has been taken earlier in [33] for compression of color images of faces using vector quantization, where the PCA method is empirically shown to produce channels that are even more decorrelated than those from the Y-Cb-Cr color model. The orthonormal bases were learned on each of the three (decorrelated) channels of the PCA image. This was followed by the quantization and coding step similar to that described in Section III-B. However

in the color image case, the Huffman encoding step for finding the optimal values of a_1, a_2 and a_3 in Eqn. (10) was performed using projection matrices from all three channels together. This was done to improve the coding efficiency. (3) For the JPEG standard (its MATLAB® implementation) we calculated RPP from the number of bytes of file storage on the disk. See Section V-B for more details.

We would like to mention here that we did not compare our technique with the face image compression technique presented in [33]. The latter requires prior demarcation of the salient feature regions, which are encoded with more bits. Our method does not require any such prior segmentation, as we manually tune the number of coefficients to meet the pre-specified reconstruction error.

B. Results

As can be seen from Figure 7(a), all three methods perform well, though the higher-order method produced the best results after a bit-rate of around 2 per pixel. For the GaTech database, we did not compare our algorithm directly with JPEG because the images in the database are already in the JPEG format. However, to facilitate comparison with JPEG, we used a subset of 54 images from the CMU-PIE database [34]. The CMU-PIE database contains images of several people against cluttered backgrounds with a large variation in pose, illumination, facial expression and occlusions created by spectacles. All the images are available in an uncompressed (.ppm) format, and their size is 631×467 pixels. We chose 54 images belonging to one and the same person, and used exactly one image for training, and all the remaining for testing. Experiments with our higher-order method, our method involving separate channels and also KSVD, revealed performance that is competitive with JPEG. For a bit rate of greater than 1.5 per pixel, our methods produced performance that was superior to that of JPEG in terms of the PSNR for a given RPP, as seen in Figure V-B. The parameters for this experiment were $K = 100$ and $T = 10$ for training. Sample reconstructions of an image from the CMU-PIE database using our higher-order method for different error values are shown in Figure V-B. This is quite interesting, since there is considerable variation between the training image and the test images, as is clear from Figure V-B. We would like to emphasize that the experiments were carried out on uncropped images of the full size with the complete cluttered background. Also, the dictionary sizes for the experiments on each of these databases are summarized in Table II. For color-image patches of size $m_1 \times m_2 \times 3$ using K sets of bases, our method in 3D requires a dictionary of size $3Km_1m_2$, whereas our method in 2D requires a dictionary of size $2Km_1m_2$ per channel, which is $6Km_1m_2$ in total.

C. Comparisons with JPEG on Noisy Datasets

As mentioned before, we did not directly compare our results to the JPEG technique for the GaTech Face Database, because the images in the database are already in the JPEG format. Instead, we added zero-mean Gaussian noise of variance 8×10^{-4} (on a color scale of $[0, 1]$) to the images of the GaTech database and converted them to a raw format.
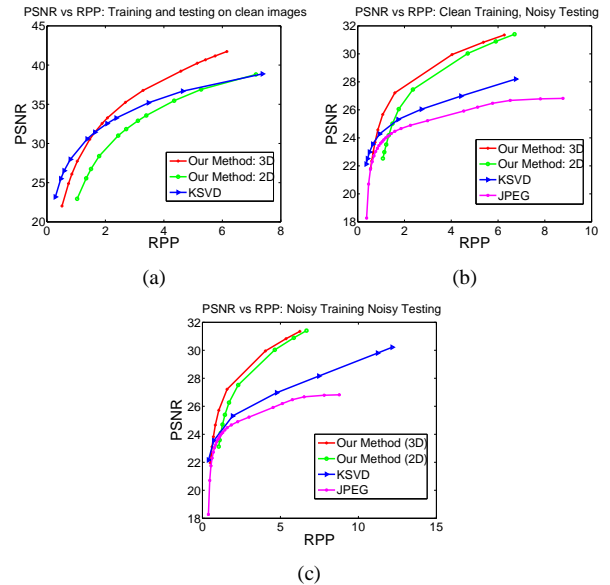

(a)

(b)

(c)

Fig. 7. ROC curves for the GaTech database: (a) when training and testing were done on the clean original images, (b) when training was done on clean images, but testing was done with additive zero mean Gaussian noise of variance 8×10^{-4} added to the test images, and (c) when training and testing were both done with zero mean Gaussian noise of variance 8×10^{-4} added to the respective images. The methods tested were our higher-order method, our method in 2D, KSVD and JPEG. Parameters for our methods: $K = 100$ and $T = 10$ during training. *These plots are viewed best in color*.
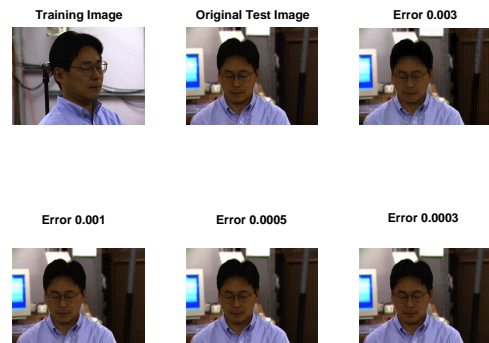

Fig. 8. Sample reconstructions of an image from the CMU-PIE database with different error values using our higher order method. The original image is on the top-left. *These images are viewed best in color and when zoomed in the pdf file.*
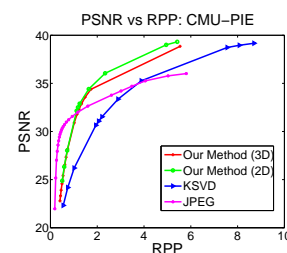

Fig. 9. ROC curves for the CMU-PIE database using our method in 3D, our method in 2D, KSVD and JPEG. Parameters for our methods: with $K = 100$ and $T = 10$ during training. *These plots are viewed best in color.*

Method	Dictionary Size (number of scalars)
Our Method (3D)	$100 \times 3 \times 12 \times 12 = 43200$
Our Method (2D)	$50 \times 2 \times 3 \times 12 \times 12 = 43200$
KSVD	$1340 \times 12 \times 12 \times 3 = 578880$

TABLE II
COMPARISON OF DICTIONARY SIZE FOR VARIOUS METHODS FOR
EXPERIMENTS ON THE GATECH AND CMU-PIE DATABASES.

Following this, we converted these raw images back to JPEG (using MATLAB®) and measured the RPP and PSNR. These figures were pitted against those obtained by our higher-order method, our method on separate channels, as well as KSVD, as shown in Figures 7(b) and 7(c). The performance of JPEG was distinctly inferior despite the fact that the noise added did not have a very high variance. The reason for this is that the algorithms used by JPEG cash in on the fact that while representing most natural images, the lower frequencies strongly dominate. This assumption can fall apart in case of sensor noise. As a result, the coefficients produced by the DCT step of JPEG on noisy images will possess prominently higher values, giving rise to higher bit rates for the same PSNR. For the purposes of comparison, we ran two experiments using our higher-order method, our separate channel method and KSVD as well. In the first experiment, noise was added only to the test set, though the orthonormal bases or the dictionary were learned on a clean training set (i.e. without noise being added to the training images). In the second experiment, noise was added to every image from the training set, and all methods were trained on these noisy images. The testing was performed on (noisy) images from the test set, and the ROC curves were plotted as usual. As can be seen from Figures 7(b) and 7(c), the PSNR values for JPEG begin to plateau off rather quickly. Note that in all these experiments, the PSNR is calculated from the squared difference between the reconstructed image and the *noisy* training image. This is because the 'original clean' image would be usually unavailable in a practical application that required compression of noisy data.

VI. CONCLUSION

We have presented a new technique for sparse representation of image patches, which is based on the notion of treating an image as a 2D entity. We go beyond the usual singular value decomposition of individual images or image-patches to learn a set of common, full-sized orthonormal bases for *sparse* representation, as opposed to low-rank representation. For the projection of a patch onto these bases, we have provided a very simple and provably optimal greedy algorithm, unlike the approximation algorithms that are part and parcel of projection pursuit techniques, and which may require restrictive assumptions on the nature or properties of the dictionary [21]. Based on the developed theory, we have demonstrated a successful application of our technique for the purpose of compression of two well-known face databases. Our compression method is able to handle the varying complexity of different patches by dynamically altering the number of coefficients (and hence the bit-rate for storage) in the projection matrix

of the patch. Furthermore, this paper also presents a clean and elegant extension to higher order matrices, for which we have presented applications to color-image compression. Unlike decompositions like HOSVD which do not retain the optimal low-rank reconstruction property of SVD, our method scales cleanly into higher dimensions. The experimental results show that our technique compares very favorably to other existing approaches from recent literature, including the JPEG standard. We have also empirically examined the performance of our algorithm on noisy color image datasets.

Directions for future work include: (1) investigation of alternative matrix representations tuned for specific applications (as opposed to using the default rows and columns of the image), (2) application of our technique for compression of gray-scale and color video (represented as 3D and 4D matrices respectively), or for denoising, and (3) a theoretical study of the method in the context of natural image statistics.

REFERENCES

[1] A. Rangarajan, "Learning matrix space image representations", in *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*. 2001, vol. 2134 of *LNCS*, pp. 153–168, Springer Verlag.
[2] J. Ye, "Generalized low rank approximations of matrices", *Machine Learning*, vol. 61, pp. 167–191, 2005.
[3] J. Yang, D. Zhang, A. F. Frangi, and J. y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, 2004.
[4] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis", in *Neural Information Processing Systems*, 2005, pp. 499–506.
[5] H. C. Andrews and C. L. Patterson, "Singular value decomposition (SVD) image coding", *IEEE Transactions on Communications*, pp. 425–432, 1976.
[6] J.-F. Yang and C.-L. Lu, "Combined techniques of singular value decomposition and vector quantization for image coding", *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1141–1146, 1995.
[7] C. Ding and J. Ye, "Two-dimensional singular value decomposition (2DSVD) for 2D maps and images", in *SIAM Int'l Conf. Data Mining*, 2005, pp. 32–43.
[8] M. Lewicki, T. Sejnowski, and H. Hughes, "Learning overcomplete representations", *Neural Computation*, vol. 12, pp. 337–365, 2000.
[9] M. Aharon, M. Elad, and A. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation", *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, November 2006.
[10] D. Mallat and M. Avellaneda, "Adaptive greedy approximations", *Journal of Constructive Approximations*, vol. 13, pp. 57–98, 1997.
[11] K. Gurumoorthy, A. Rajwade, A. Banerjee, and A. Rangarajan, "Beyond SVD - sparse projections onto exemplar orthonormal bases for compact image representation", in *International Conference on Pattern Recognition (ICPR)*, 2008.
[12] A. Shashua and A. Levin, "Linear image coding for regression and classification using the tensor-rank principle", in *CVPR(1)*, 2001, pp. 42–49.
[13] L. de Lathauwer, *Signal Processing Based on Multilinear Algebra*, PhD thesis, Katholieke Universiteit Leuven, Belgium, 1997.
[14] J. Håstad, "Tensor rank is NP-Complete", in *ICALP '89: Proceedings of the 16th International Colloquium on Automata, Languages and Programming*, 1989, pp. 451–460.
[15] M.A.O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces", in *International Conference on Pattern Recognition (ICPR)*, 2002, pp. 511–514.
[16] R. Costantini, L. Sbaiz, and S. Susstrunk, "Higher order SVD analysis for dynamic texture synthesis", *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 42–52, Jan. 2008.
[17] H. Wang and N. Ahuja, "Rank-R approximation of tensors: Using image-as-matrix representation", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, vol. 2, pp. 346–353.
[18] C. Ding, H. Huang, and D. Luo, "Tensor reduction error analysis - applications to video compression and classification", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[19] A. Ranade, S. Mahabalarao, and S. Kale, "A variation on SVD based image compression", *Image and Vision Computing*, vol. 25, no. 6, pp. 771–777, 2007.

[20] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", in *27th Asilomar Conference on Signals, Systems and Computation*, 1993, vol. 1, pp. 40–44.

[21] J. Tropp, "Greed is good: algorithmic results for sparse approximation", *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[22] A. Yuille and J. Kosowsky, "Statistical physics algorithms that converge", *Neural Computation*, vol. 6, no. 3, pp. 341–356, 1994.

[23] R. Hathaway, "Another interpretation of the EM algorithm for mixture distributions", *Statistics and Probability Letters*, vol. 4, pp. 53–56, 1986.

[24] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analysers", *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.

[25] "The ORL Database of Faces", http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[26] "The Yale Face Database", http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

[27] "The CVG Granada Database", http://decsai.ugr.es/cvg/dbimagenes/index.php.

[28] "The Uncompressed Colour Image Database", http://vision.cs.aston.ac.uk/datasets/UCID/ucid.html.

[29] M. Marcellin, M. Lepley, A. Bilgin, T. Flohr, T. Chinen, and J. Kasner, "An overview of quantization in jpeg 2000", *Signal Processing: Image Communication*, vol. 17, no. 1, pp. 73–84, 2000.

[30] M. Adams and R. Ward, "Jasper: A portable flexible open-source software toolkit for image coding/processing", in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004, pp. 241–244.

[31] M. Hilton, B. Jawerth, and A. Sengupta, "Compressing still and moving images with wavelets", *Multimedia Systems*, vol. 2, no. 5, pp. 218–227, 1994.

[32] "The Georgia Tech Face Database", http://www.anefian.com/face_reco.htm.

[33] J. Huang and Y. Wang, "Compression of color facial images using feature correction two-stage vector quantization", *IEEE Transactions on Image Processing*, vol. 8, pp. 102–109, 1999.

[34] "The CMU Pose, Illumination and Expression (PIE) Database", http://www.ri.cmu.edu/projects/project_418.html.