# Linear Image Coding for Regression and Classification using the Tensor-rank Principle*

Amnon Shashua and Anat Levin
School of Computer Science and Engineering,
The Hebrew University,
Jerusalem 91904, Israel
e-mail: {shashua,alevin}@cs.huji.ac.il

## Abstract

*Given a collection of images (matrices) representing a "class" of objects we present a method for extracting the commonalities of the image space directly from the matrix representations (rather than from the vectorized representation which one would normally do in a PCA approach, for example). The general idea is to consider the collection of matrices as a tensor and to look for an approximation of its tensor-rank. The tensor-rank approximation is designed such that the SVD decomposition emerges in the special case where all the input matrices are the repeatition of a single matrix. We evaluate the coding technique both in terms of* regression, *i.e., the efficiency of the technique for functional approximation, and* classification. *We find that for regression the tensor-rank coding, as a dimensionality reduction technique, significantly outperforms other techniques like PCA. As for classification, the tensor-rank coding is at is best when the number of training examples is very small.*

## 1 Introduction

Given a collection of model images forming a "training" set of a class of objects we wish to find an image coding technique which captures the *spatial* and *temporal* (phase alignment) regularities shared by the training set of images. We will focus on those techniques in which the coding consists of a change of basis (linear coding). In the most general form, let $\phi_1(\boldsymbol{x}), ..., \phi_p(\boldsymbol{x})$ be the set of $p$ training images, representing an "object class" $\phi$, where $\boldsymbol{x}$ varies over the two-dimensional plane — for example, $\phi_i(\boldsymbol{x})$ is an $n \times m$ matrix. We seek a new set of images $\psi_1(\boldsymbol{x}), ..., \psi_r(\boldsymbol{x})$ which on one hand form a complete code, i.e., span the original training set of images as closely as possible, and while doing so capture the spatial and temporal regularities among the training set.

An image $\phi(\boldsymbol{x})$ of the class is represented in terms of the new basis $\phi(\boldsymbol{x}) = \sum_i y_i \psi_i(\boldsymbol{x})$ where the coefficients $y_i$ are dynamic variables that change from one image to the next, and are often referred to as the "feature" vector of the image $\phi(\boldsymbol{x})$. The measure of how well the image regularities are captured by the new basis is often represented in terms statistical properties on the variables $y_i$ — often the ultimate requirement being that the coefficient values be as statistically independent as possible over natural images.

One example of this approach is based on Principal Component Analysis (PCA), in which the goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated, $\sum \boldsymbol{y}\boldsymbol{y}^\top$ is a diagonal matrix. The popularity of PCA comes from its closed-form and efficient computation, the fact that it is well understood, and from its general applicability. For example, in computer vision applications it has been used for the representation and recognition of faces [21, 18, 4], recognition of 3D objects under varying pose [19], tracking of deformable objects [6] and for representations of 3D range data of heads [1].

Another example of this line of approach is minimum entropy coding [3] in which statistical dependence is reduced by lowering the individual entropies $H(y_i)$ or by enforcing a sparse structure of the feature vector, i.e., any image of the class can be represented in terms of a small number of basis images out of a large set [10, 20] and closely related to that is the work on Independent Component Analysis (ICA) by [8, 5].

The linear coding schemes mentioned above do not capture both the spatial and temporal (phase alignment) redundancy in a uniform manner. For example, when the training set is small, PCA captures mostly the temporal redundancy of each position across the images — where at the extreme case when the training set consists of a single image no cod-

*This work has been funded by the Israeli Science Ministry grant 1229. A.S. is on sabbatical at the Computer Science dept. at Stanford University and can be reached at shashua@cs.stanford.edu

ing is performed at all. The sparse coding of [20] on the other hand mostly captures the spatial redundancy as the resulting basis images represent the sparse components from which the training images were composed.

Moreover, most coding or classification schemes (such as SVM introduced by [22]) are designed for 1-dimensional signals and their adaptation to images requires an ad-hoc rasterization from 2D to 1D.

In this paper we introduce a different approach for capturing the temporal and spatial redundancy in a collection of training images and which is specifically designed for matrices (and could be easily generalized to higher dimensional signals as well). Rather than placing requirements on the statistical properties of the coefficients $y_i$ we look for the most *compact* representation of a collection of matrices as a linear super-position of rank-1 matrices (the "tensor rank" problem). This problem definition is a natural extension of the Singular Value Decomposition (SVD) of a single matrix to a collection of matrices. An SVD (of a single matrix) captures the spectral representation of the image (i.e., the spatial redundancy) therefore the "multi-image" rank-1 decomposition (as described later) would represent both the spatial and temporal spectra of the image set.

We introduce first a closed-form scheme for generating the rank-1 decomposition under special conditions, and then introduce a greedy algorithm for obtaining a multi-image rank-1 decomposition and prove its convergence properties both in the worst case and average case. We have experimented with various image sequences both for regression and classification. With regression we find a compression ratio of almost an order of magnitude better than other linear coding techniques such as PCA. As for classification, we demonstrate clear superiority over PCA for small sets of training images.

## 2 The Tensor-rank Problem

In the framework of algebraic complexity theory, the problem of finding optimal computations for bilinear forms, leads to the notion of tensorial rank: Let $X, Y, W$ be finite dimensional vector spaces and $t \in X \otimes Y \otimes W$. The task consists of finding a decomposition of the trivalent tensor $t$ into triads (rank-1 tensors):

$$t = \sum_{s=1}^{r} u_s \otimes v_s \otimes w_s \quad (u_s \in X,\ v_s \in Y,\ w_s \in W),$$

with *minimal* possible $r$. The least $r$ for which such a representation exists is called the rank of $t$ (see [12, 2, 14, 7] for motivation and background). For example, in the computer vision literature the tensor rank decomposition was introduced in the context of efficient evaluation of a collection of filters (a filter bank) which are applied simultaneously across the image [15, 17]. The triadic decomposition was performed in an interleaving manner where two of the three vector variables would remain fixed while minimizing (in a least-squares sense) the value of the third vector-variables, then cycle for the remaining variables. Although this approach does converge to a local minimum, there is no guarantee as to the quality of the solution due to the dependence on the starting point.

We wish to introduce the tensor rank decomposition in the context of image coding and while doing so constrain the solution so that certain desirable properties are *guaranteed* to hold. As a first step, it would be advantageous to rewrite the triadic decomposition above in a way which appears like a natural extension of the SVD decomposition of matrices, as shown below.

An equivalent formulation of the triadic decomposition problem is to consider the $\dim W = p$ slices of $t$ as a collection of $n \times m$ matrices $A_1, ..., A_p$, thus, the tensor-rank task is to find a collection of rank-1 matrices $\tau_1, \tau_2, ..., \tau_r$, of smallest possible $r$, whose linear span includes the matrices $A_1, ..., A_p$, i.e.,

$$A_i = \sum_{j=1}^{r} \lambda_{ij} \tau_j.$$

The later formulation lends itself to a multi-matrix extension of SVD as follows. Let $\tau_i = u_i v_i^\top$, and let $U$ be an $n \times r$ matrix whose columns consist of $u_1, ..., u_r$, let $V$ be an $m \times r$ matrix whose columns consist of $v_1, ..., v_r$, and let $D_i$ be $r \times r$ diagonal matrices $i = 1, ..., p$. We have:

$$A_i = U D_i V^\top \quad i = 1, ..., p.$$

The trivial case $p = 1$ is simply the SVD of a matrix $A = UDV^\top$, and because of the multitude of solutions for the decomposition into rank-1 forms one can enforce additional orthogonality constraints $u_i^\top u_j = 0, v_i^\top v_j = 0$. The diagonal elements of $D$ are the "singular" values $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_r$. The SVD decomposition $A = \sum \lambda_i u_i v_i^\top$ forms a spectral decomposition in which the lower spatial frequencies are represented in the sum of the first terms while the higher frequencies are added with increasing terms.

The case $p = 2$, i.e., the tensor has two slices $A_1, A_2$, is also well understood [13]. For simplicity, assume $n = m$ and $A \neq B$ are full rank matrices. Consider $\lambda$ and vector $x$ such that:

$$(A_1 + \lambda A_2)x = U(D_1 + \lambda D_2)V^\top x = 0$$

Therefore, the columns of $V^{-\top}$ are the generalized eigenvectors of the pair $A_1, A_2$ and the columns of $U^{-\top}$ are the generalized eigenvectors of the pair $A_1^\top, A_2^\top$ and the generalized eigenvalues $\lambda_i = -\lambda_{1i}/\lambda_{2i}$.

It is worthwhile noting that even though the matrices $A_1, A_2$ may have nothing in common (just two arbitrary matrices), the rank of the 2-slice tensor is bounded from above by $n$ which is much smaller than the sum of the individual ranks ($2n$) — this is what makes the tensor-rank problem interesting to begin with. Therefore, the tensor rank provides the most compact representation of a collection of matrices. Unfortunately, these two cases are not typical for the general case $p > 2$ as it has been shown to belong to the family of NP-complete problems [11].

We will introduce two approaches to the multi-image rank-1 decomposition. The first approach is a closed-form solution assuming $r = p$, i.e., the number of rank-1 matrices $r$ is to be equal to the number $p$ of input matrices. We will see from experiments that the decomposition provides very good approximations (far superior in terms of compression to PCA, for example). The second approach is general but is based on a "greedy" policy explicitly targeting a spatio-temporal decomposition inspired by the spectral decomposition of SVD.

The greedy approach is designed in a such a way which guarantees to fall back onto the SVD (of a single matrix) when the input matrices $A_1, ..., A_p$ are multiple copies of the same matrix $A$. This property provides the coding scheme the phase-alignment extension to the spectral decomposition of a single matrix: when all the matrices are equal, the coding scheme will provide a decomposition into a set of rank-1 matrices which optimize the spatial redundancy (i.e., the SVD of the matrix $A$). As changes are being introduced among the input matrices the coding scheme will add rank-1 elements to account for the changes in phase-alignment (temporal redundancy). Results on real-imagery in Section 5 demonstrate a good trade-off between spatial and temporal coding with much higher compression rates compared to PCA (of a collection of matrices) alone, or SVD (applied to each matrix separately).

# 3 Closed-form Solution $r = p$

Assume that the number of rank-1 matrices $r$ is *known* to be equal to the number of input matrices $p$. Also assume that the set of input matrices are linearly independent (otherwise replace them with their principle components). Thus the setup of the problem is that we are given $n \times m$ linearly independent matrices $A_1, ..., A_p$ and we are looking for $p$ rank-1 matrices $\tau_1, ..., \tau_p$ which best span the input matrices. This situation is interesting because one can obtain a closed form solution, it works in practice very well, and one can extend the approach to recover in this manner $kp$ (k-multiple of $p$ rank-1 matrices) rank-1 matrices.

Since $r = p$, and the set of input matrices is linearly independent, the requirement that $A_i$ be spanned by $\{\tau_i\}$ is equivalent to the requirement that each $\tau_i$ be spanned by the set $\{A_i\}$. Therefore, we obtain the constraint:

$$\text{rank}(\sum_{i=1}^{p} y_i A_i) = 1,$$

for some scalars $\boldsymbol{y} = (y_1, ..., y_p)$. Let $\boldsymbol{y}_1, ..., \boldsymbol{y}_p$ be the $p$ solutions (we are looking for) for this system of constraints. Since every $2 \times 2$ minor of $\sum_{i=1}^{p} y_i A_i$ must vanish, we obtain a linear system of $\binom{n}{2}\binom{m}{2}$ (not all independent) and $\binom{p}{2}$ equations on $\hat{y} = \boldsymbol{y} \otimes \boldsymbol{y}$ ($p^2$ unknowns). Let $E$ be the estimation matrix, i.e., $E\hat{y} = 0$. Since we know there are exactly $p$ solutions for $\hat{y}$, let $v_1, ..., v_p$ be vectors in the null space of $E$, i.e., $Ev_i = 0$. Let $V_i$ the $p \times p$ matrix associated with $v_i$ (recall that $\hat{y} = \boldsymbol{y} \otimes \boldsymbol{y}$ can be presented also as an $p \times p$ matrix). Let $Y = [\boldsymbol{y}_1, ..., \boldsymbol{y}_p]$ be the (unknown) $p \times p$ matrix whose columns are the solution vectors. Because the sets $\{v\}_i$ and $\{\boldsymbol{y}\}_i$ span the same subspace, we must have: $V_i = Y D_i Y^\top$, where $D_i$ is a diagonal matrix. Thus, taking any two elements of the null space of $E$, say $V_1, V_2$ we have:

$$0 = (V_1 - \lambda V_2)x = Y(D_1 - \lambda D_2)Y^\top x,$$

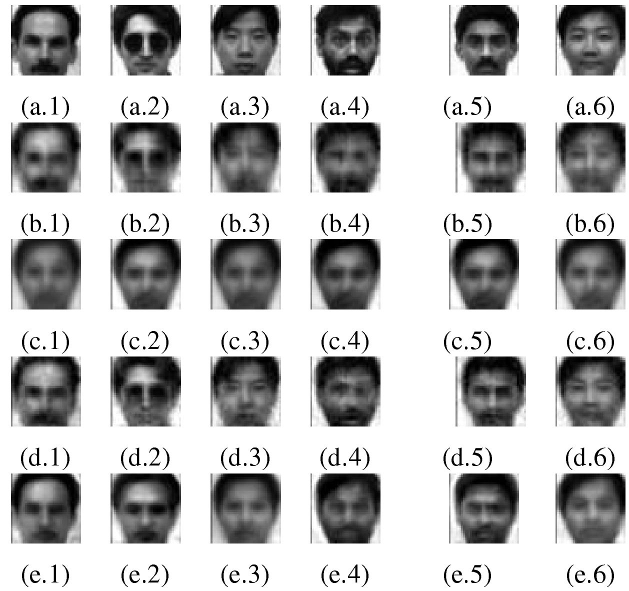


Figure 1: Demonstration of the closed-form approach, where the number of rank-1 elements is equal to a multiple of the number of input matrices. The image database is from Yale. Columns (1-4) presents sample of images that were part of the training set, Columns (5-6) present the results for novel images, that were not part of the training set. (a)- A sample from the original set of 40 training images (a.5 and a.6 were not part of the training set). (b)- Projection of the images in (a.1-1.6) onto the subspace spanned by the 40 rank-1 matrices. (c)- PCA with 3 principal components (the same compression ratio as b). (d)- The $r = p$ algorithm with 2 additional iteration on the residual images (total of 120 rank-1 elements). (e)- PCA with 9 principal components (the same compression ratio as d). The reconstruction results of the multi-image rank-1 are generally better than PCA: note the glasses in columns 2 which were not picked up in the PCA and the recognizability of the face in column 3.

hence, the columns of $Y$ are the generalized eigenvectors of the pair of matrices $V_1, V_2$. Thus, we have obtained the solution vectors $\boldsymbol{y}_1, ..., \boldsymbol{y}_p$ which are the coefficients of the linear superpositions of the original input matrix in order to obtain the rank-1 matrices $\tau_1, ..., \tau_p$.

The scheme above is an algorithm for obtaining the $p$ rank-1 matrices, provided we *know* that they exist. In practice, this scheme provides an approximation to the best[1] set of $p$ rank-1 matrices given an input set of $p$ matrices. Fig. 1 demonstrates the algorithm on a set of 40 images of $50 \times 50$ of human faces. The first row illustrates a sample of the images used for generating the rank-1 matrices, and also a number of novel images not used in the algorithm. The second row shows the reconstructed images (projection of input images onto the subspace spanned by the 40 rank-1 matrices), and also the reconstructed novel images. In terms of compression, each rank-1 matrix contains 100 scalars (outer-product of two vectors), therefore the set of 40 input images are represented by $40 \times 100 + 40 \times 40 = 5,600$ scalars (instead of $40 \times 50 \times 50 = 100,000$). Compared to PCA, each principal component is represented by $2,500$ scalars, thus if we represent the input set by $k$ principal components, then the total space required is $2,500k + 40k$, thus $k = 3$ would provide comparable space to the rank-1 scheme. The third row of Fig. 1 shows the reconstruction of the input set and the novel images using 3 principal components — there is a significant difference in quality. Row 4 shows the reconstruction using 120 rank-1 matrices: this is done by subtracting the reconstructed images from the input set and running the algorithm again on the residual matrices (twice) — in this manner one can obtain any multiple of $p$ rank-1 matrices. Finally, Row 5 shows the reconstruction using 9 Principal Components — again there is a noticeable difference such as d.2 compared to e.2.

# 4 The Greedy Approach for the General Case

The closed-form algorithm, assuming $r = p$, is designed to produce a predetermined number of rank-1 matrices provided that indeed such a number exists. Below, we take a different approach and introduce a "greedy" policy for generating a decomposition of the input set into rank-1 matrices, one at a time. The scheme is designed such that it provides an *extension* of the SVD of a single matrix to multiple matrices by the fact that when the set of input matrices consists of multiple copies of the same matrix, the greedy algorithm produces exactly the spectral decomposition of SVD (Claim 2). We prove the convergence (and rate) of the process and demonstrate its effectiveness in practice.

[1] Formalizing what is meant by "best" is not obvious and is left for future research.

The greedy approach is based on the following property of SVD (of a single matrix $A$): Let $A = UDV^T$ be the SVD of $A$ of rank $r$. If $k < r$ and

$$X_k = \sum_{i=1}^{k} \lambda_i u_i v_i^\top$$

then

$$X_k = \arg \min_{rank(B)=k} \|A - B\|_F^2$$

which forms the basis of the spectral decomposition property of SVD. As an extension of this, given matrices $A_1, ..., A_p$ we wish to find unit vectors $u, v$ and scalars $\lambda_1, ..., \lambda_p$ such that

$$\sum_{i=1}^{p} \|A_i - \lambda_i u v^\top\|_F^2$$

is minimal. Note that the Frobenius norm is defined as $\|A\|_F^2 = \sum_{i,j} a_{ij}^2$. Assume we have found those, then $\tau_1 = uv^\top$ would be the first rank-1 matrix in the decomposition of the set into $\tau_1, ..., \tau_r$. We then replace the original matrices with $A_i' = A_i - \lambda_i \tau_1$ and repeat the greedy step above on $A_1', ..., A_p'$ to find $\tau_2$ and a new set of scalars $\lambda_i'$, and so forth. We will discuss the convergence issue later, while next we will derive the greedy step.

**Claim 1** *The vectors $u, v$, $|u| = |v| = 1$, which minimize the expression $\sum_{i=1}^{p} \|A_i - \lambda_i uv^\top\|_F^2$ maximize the expression*

$$\sum_{i=1}^{p} (v^\top A_i^\top u)^2.$$

**Proof:** Noting that $\|A\|_F^2 = trace(AA^\top)$ and that $trace(AB) = trace(BA)$, we have

$$\begin{aligned}
\|A_i - \lambda_i uv^\top\|_F^2 &= trace(A_i A_i^\top) - 2\lambda_i trace(A_i v u^\top) \\
&+ \lambda_i^2 trace(uv^\top v u^\top) \\
&= \|A_i\|_F^2 - 2\lambda_i v^\top A_i^\top u + \lambda_i^2
\end{aligned}$$

Where the last line follows from $trace(ab^\top) = a^\top b$. A necessary condition for an extremum on $\lambda_i$ is the vanishing partial derivative:

$$\frac{\partial}{\partial \lambda_i}(\|A_i - \lambda_i uv^\top\|_F^2) = 2(\lambda_i - v^\top A_i^\top u) = 0,$$

Therefore the minimum is achieved when $\lambda_i = v^\top A_i^\top u$. Substituting for $\lambda_i$ we obtain

$$\|A_i - \lambda_i uv^\top\|_F^2 = \|A_i\|_F^2 - (v^\top A_i^\top u)^2$$

from which the claim follows. □

Therefore, in order to find $\tau_1 = uv^\top$ we must maximize $\sum_{i=1}^{p} (v^\top A_i^\top u)^2$. We propose an iterative approach for a local optimum, based on the gradient descent principle. In practice, this approach turns out to be very efficient and to provide very good results within a small number of iterations. As a first guess, we look for a unit vector $u$ which maximizes $\sum_i \|A_i^\top u\|_2^2$, thus $u$ is the eigenvector associated with the largest eigenvalue of the $n \times n$ matrix $\sum_{i=1}^p A_i A_i^\top$. In the following step, we look for a unit vector $v$ which maximizes $\sum_{i=1}^p (v^\top A_i^\top u)^2$ (given $u$ we have found in the previous step). Let $\hat{A}$ be the $m \times p$ matrix whose columns are $A_i^\top u$. Thus we seek unit vector $v$ which maximizes $\|\hat{A}^\top v\|^2$ and $v$ is therefore the eigenvector associated with the largest eigenvalue of the $m \times m$ matrix $\hat{A}\hat{A}^\top$. The process is then continued iteratively. Given $v$ from the previous step we look for the best $u$— the one maximizing $\sum_{i=1}^p (v^\top A_i^\top u)^2$, and so on.

We summarize the greedy algorithm below:

1. Given $n \times m$ matrices $A_1, ..., A_p$ we would like to find rank-1 matrices $\tau_1, ..., \tau_r$ which span the input set.
2. Perform the following for $j = 1, ..., r$:
3. Let $u$ be the eigenvector associated with the largest eigenvalue of the $n \times n$ matrix $\sum_{i=1}^p A_i A_i^\top$.
4. Let $\hat{A} = [A_1^\top u, ..., A_p^\top u]$ be an $m \times p$ matrix. Let $v$ be the eigenvector associated with the largest eigenvalue of the $m \times m$ matrix $\hat{A}\hat{A}^\top$.
5. Repeat few times:
   (a) Let $\hat{A} = [A_1 v, ..., A_p v]^\top$ be an $p \times n$ matrix. Let $u$ be the eigenvector associated with the largest eigenvalue of the $n \times n$ matrix $\hat{A}^\top \hat{A}$.
   (b) Let $\hat{A} = [A_1^\top u, ..., A_p^\top u]$ be an $m \times p$ matrix. Let $v$ be the eigenvector associated with the largest eigenvalue of the $m \times m$ matrix $\hat{A}\hat{A}^\top$.
6. $\tau_j = uv^\top$.
7. Replace $A_i$ with $A_i - (v^\top A_i^\top u)uv^\top$.
8. Go to Step 3.

The process ends when the sum of norms of the residual matrices is below some threshold (see convergence details in the sections below).

This process converges to a local minima (proof detailed in the next section) thus therefore is not guaranteed to be the global one (not in the sense of the smallest number of rank-1 elements and not in the sense of best approximation per given number of rank-1 elements). As this is a greedy approach, it suffers from the shortcoming that previous decisions (selection of rank-1 elements) are not re-evaluated as the process unfolds. However, this specific greedy rule has a critical feature which makes it useful for image coding and that it is guaranteed to produce the spectral decomposition of SVD when all the matrices are equal to each other $W = A_1, ..., A_p$ ($p \geq 1$); i.e., the rank-1 elements produced in this manner are exactly those produced by SVD of the matrix $W$. This feature holds both due to the greedy policy which selects the closest rank-1 element to the residual matrix, and the choice of the optimization approach for $\sum_{i=1}^p (v^\top A_i^\top u)^2$. Note that the optimization approach (steps 3,4 in the algorithm) converge to a local minima in general, but in the case of all matrices being equal to each other, one obtains the SVD decomposition. This is proven next.

**Claim 2** *The greedy algorithm reduces to an SVD when* $A_1 = ... = A_p = W$ *for* $p \geq 1$.

**Proof:** It is sufficient to show that in Step 3 $u$ is the largest eigenvector of $WW^\top$ (which follows immediately) and that in Step 4 $v$ is the largest eigenvector of $W^\top W$. In Step 4, $v$ is the largest eigenvector of $(W^\top u)(W^\top u)^\top$. Note that if $WW^\top u = \lambda u$ then $W^\top u$ is the largest eigenvector of $W^\top W$ (simply multiply by $W^\top$ on the left on both sides). Therefore, $(W^\top u)(W^\top u)^\top = vv^\top$ and $v$ is the largest eigenvector of $W^\top W$. Finally, note that the iterations in step 5 do not change the values of $u, v$ as they are a stationary point in the process (following the same arguments as above).$\square$

As a result of the claim above, the algorithm extends the SVD procedure to multiple images (matrices). If a single matrix (or a set of identical or very similar matrices) is given, the algorithm provides a spectral decomposition which is largely based on the spatial domain. As the images in the set differ from each other the spectral decomposition will contain the temporal (phase alignment) domain as well. These properties will be demonstrated in the experimental section later on.

## 4.1 Convergence

The algorithm above is guaranteed to converge. We analyze below the worst-case convergence property:

**Claim 3** *Let* $A_i^{(r)}$ *be the residual of the i'th matrix, after approximating the input images set with* $r$ *rank-1 matrices, then:*

$$\sum_{i=1}^{p} \|A_i^{(r)}\|_F^2 \leq (1 - 1/(NM))^r \sum_{i=1}^{p} \|A_i\|_F^2$$

*where* $M = min(m, p), N = min(mp, n)$.

For proof see the extended version of this paper in http://www.cs.huji.ac.il/∼ shashua/papers/rank1-full.pdf.

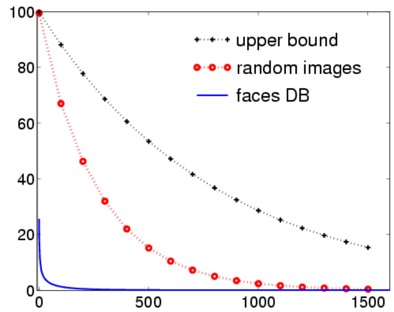


Figure 2: The theoretical upper bound on convergence rate versus the actual convergence (measured empirically) on a set of random matrices and on a set of face images (Yale database). One can see that the actual convergence is much faster than the upper bound. X-axis is the number of iterations and Y-axis is the residual percents (the ratio $\sum_{i=1}^{p} ||A_i^{(r)}||_F^2 / \sum_{i=1}^{p} ||A_i||_F^2$).

Note that this claim provides only an upper bound on the convergence rate of the algorithm — in practice the convergence may be much faster. Fig. 2 displays the theoretical convergence rate versus empirical convergence rates of actual examples: a set of random matrices, and a set of face image (Yale database). One can see that the actual convergence is much faster than the theoretical upper bound.

# 5 Experiments

The multi-image rank-1 decomposition captures both the spatial and temporal redundancies of the image set. To better understand how these two dimensions (spatial and temporal) take part in the decomposition we consider the following experiments.

Consider a sequence of images depicting a movie of 5 frames — three of which are shown in the first row of Fig. 3. The phase (temporal) alignment is very strong as the scene does not change much from frame to frame, yet there are also spatial redundancies. The second row shows the reconstructed images from 29 rank-1 elements. We compare the result to PCA in row 3 in which the number of principal components is selected such that we achieve the same compression ratio (which was roughly 1:7). Then, compare the result to single image SVD with the same compression ratio of 1:7. It is evident that the 29 rank-1 matrices have captured both the spatial and temporal redundancies, for if no temporal redundancy was being captured then row 2 and 4 should be very similar (which they are not) and if no spatial redundancy were being captured then row 2 and row 3 would have been similar, which again they are not.

Consider the other end of the extreme in which the training set has very little temporal alignment. This is shown in Fig. 4 depicting a collection of toys (Columbia U.

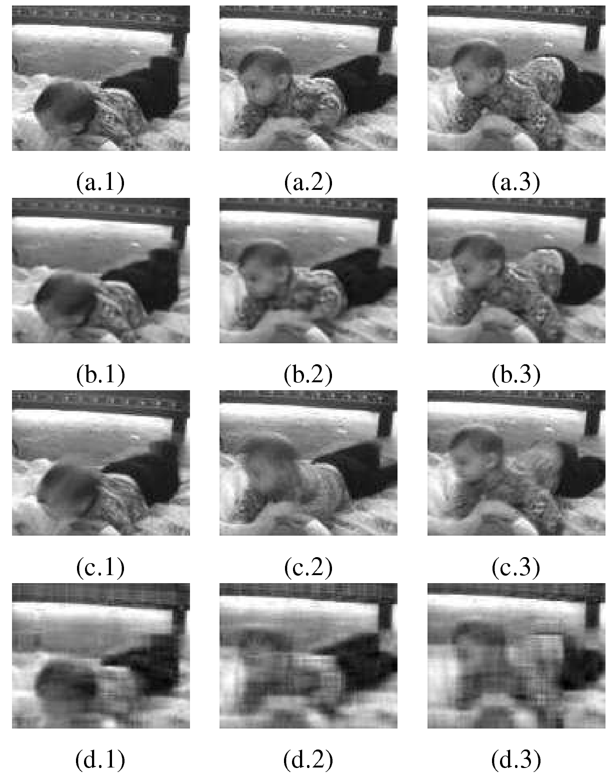


Figure 3: Testing the efficiency of spatial and temporal redundancy coding by the greedy method. (a)- three frames from a movie of 5 frames. (b)- Reconstruction from 29 rank-1 elements (compression ratio of 1:7) (c) Reconstruction using PCA with the same compression ratio. (d)- Reconstruction using single image SVD, with the same compression ratio. If the multi-image decomposition was not efficiently coding the temporal redundancy, then rows b,d would look similar. Likewise, if it were not efficiently coding the spatial redundancy then rows b,c would look similar to each other.

database). Row 1 shows a sample out of 20 training images. Row 2 shows the reconstruction using 500 rank-1 elements (achieving a compression of 1:2.3). Row 3 shows the reconstruction using PCA with 9 principal components (same compression ratio) — and as one can see the lack of temporal alignment in the image set has a detrimental effect on the reconstruction (as expected from PCA). Row 4 shows the reconstruction using single-image SVD with the same compression ratio of 1:2.3. As one can see, rows 2 and 4 look identical, therefore what has been picked by the multi-image decomposition is mostly the spatial redundancy.

Finally, consider the situation in-between the two extreme situations depicted by a collection of face images (frontal) of various people (Yale U. database), shown in Fig. 5. The multi-view picks up both dimensions (spatial, temporal) as the comparison with single image SVD (rows 2,4) and with PCA (rows 2,3) is significantly favorable for the multi-image rank-1 decomposition.

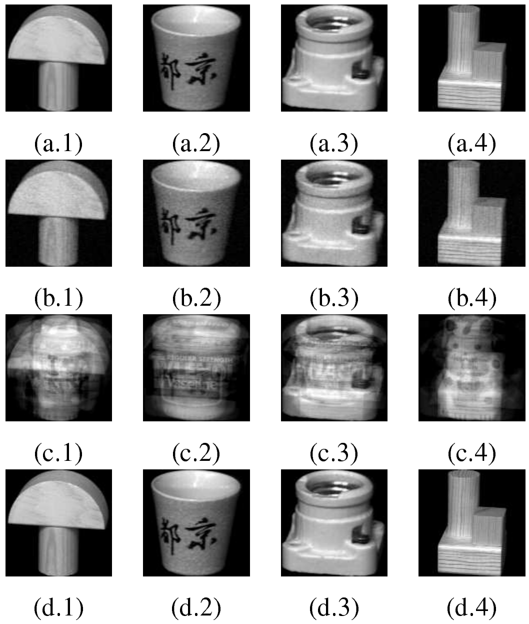


Figure 4: Testing efficiency of coding when the set of training images lack temporal (phase) alignment. (a)- Sample from a set of 20 training images (Columbia U. database). (b)- Reconstruction from 500 rank-1 elements (compression ratio 1:2.3). (c) PCA with the same compression ratio (9 PCs). (d)- Single view SVD, with the same compression ratio. Note that rows b,d are very similar, suggesting that the multi-view decomposition picked up the spatial redundancy in an efficient, i.e., the fact that multiple images were used did not have a noticeable detrimental effect.

## 5.1 Classification

We conducted another set of experiments to measure the utility of the multi-view rank-1 coding for purposes of object classification. Consider again the last two columns of Fig. 5. The two images in Row 1 are novel images which did not participate in the training set and therefore the rank-1 decomposition was not optimized to generate them. Thus the quality of reconstruction we observe in row 2 (especially compared to rows 3,4) indicates a certain *generalization* capability which we would like to evaluate in a more systematic manner in this section.

We have experimented with a variety of databases, faces, vehicles, small-sized training sets and very large sized training sets, and concluded that the superiority of the technique over others comes to bear when the training set is very small. We compare the classification ability to PCA. Note that we have conducted experiments where the coefficients of the reconstruction are used as a feature vector in an SVM classifier, and considered Fisher's LDA [9] as well, but for small training sets only PCA and LDA are relevant, and it has been shown already that PCA outperforms LDA when the training set is small [16], thus it is sufficient to conduct the comparison with PCA alone.

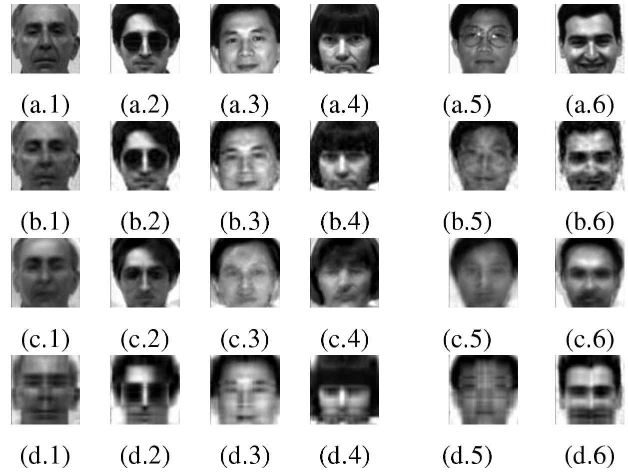


Figure 5: Testing efficiency of coding for training sets in which the temporal redundancy exists, is noticeable, but is weak compared to that seen in a movie-sequence. The training set consisted of 150 images (80 $\times$ 80) and the multi-image decomposition was into 250 rank-1 elements (thus achieving a 1:12 compression ratio). Columns (1-4) presents sample of images that were part of the training set, Columns (5-6) present the results for novel images, that were not part of the training set. (a)- (1-4) sample of the training set. (b)- Reconstruction from the 250 rank-1 basis. (c) PCA with the same compression ratio. (d)- Single view SVD, with the same compression ratio.

We have experimented with a number of databases with consistent results. Here we detail the experiment with the AR-face database as described in [16] (and was kindly provided to us after pre-alignment processing by the authors). The database consists of 50 people with 14 images per person. The 14 images are divided into two groups depicting the same conditions (lighting, facial expressions) but taken two weeks apart. In each group of 7, 3 of the images vary only according to change of lighting, and the remaining 4 have a fixed lighting but vary in facial expressions (see Fig. 6). We conducted two experiments, the first with the group containing only facial expressions (8 images per person), and second with the entire set of 14 images per person. The size of the images was $60 \times 85$. The size of the training set was set to the range $k = 1, 2, 3$. For each trial (per choice of $k$) we created a rank-1 subspace (consisting of between 50-70 rank-1 elements) from the $k$ training images per person (thus obtaining 50 subspaces). Then, each image of the $14 \times 50$ (or $8 \times 50$) images was projected onto the 50 subspaces and the distance (Frobenius norm) between the image and its projection (to each of the subspaces) was recorded. The match was decided upon based on the smallest distance. The percentage correct over the variation of facial expressions only (8 images per person) were $65.43\%, 86\%, 97.6\%$ for $k = 1, 2, 3$, compared to $61.43\%, 80.7\%, 95.2\%$ for the PCA (linear subspace) method. When all the 14 images (per person) are used for

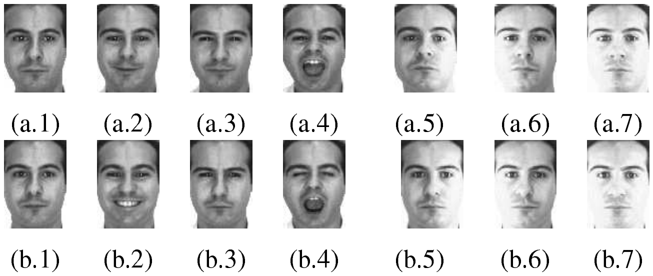


Figure 6: A sample of images from the AR-face database. (a.1-4)- Different facial expressions. (a. 5-7) change of lighting (b)- Same session taken two weeks later. Database consists of 50 sets like this one, corresponding to different faces.

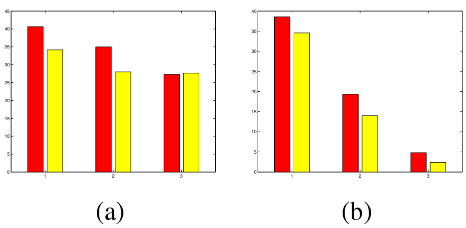


Figure 7: Classification error percents on the AR-face data base. The linear subspace method — dark bars, and the multi-image rank-1 are displayed in bright bars. (a)- Facial expressions and different lighting (14 images per person). (b)- Facial expression alone (8 images per person). The numbers themselves are detailed in the text.

testing the percentage correct becomes: $65.8\%, 72\%, 72\%$ compared to the PCA with $59.4\%, 65\%, 72\%$. In both cases, the multi-image rank-1 significantly outperforms the linear subspace approach. Fig. 7 shows the results graphically.

# 6 Summary

We have introduced a novel multi-image decomposition into basis elements based on the tensor-rank principle. The method is designed to capture both the spatial and temporal redundancies in a training set of images. For example, when the training set consists of a single image or a collection of identical images the decomposition reduces to an SVD decomposition (thus coding the spatial redundancy). When the images within the training set vary, the decomposition picks up the temporal (phase alignment) redundancy as well. The detailed experimental component of this work was designed to highlight various aspects of the efficiency of the technique in capturing these two dimensions (spatial, temporal), and to make the case that the technique has strong capabilities for classification purposes with small training sets — for example, we have shown it significantly outperforms the linear subspace (PCA) method.

# References

[1] J.J. Atick, P.A. Griffin, and N.A. Redlich. Statistical approach to shape-from-shading: deriving 3d face surfaces from single 2d images. *Neural Computation*, 1997.

[2] M.D. Atkinson and N.M. Stephens. On the maximal multiplicative complexity of a family of bilinear forms. *Linear Algebra and Its Applications*, 27:1–8, 1979.

[3] H.B. Barlow. Unsupervised learning. *Neural Computation*, 1:295–311, 1989.

[4] P.N Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. In *Proceedings of the European Conference on Computer Vision*, 1996.

[5] A.J. Bell and T.J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. In *Neural Computation 7(6)*, pages 1129-1159, 1995.

[6] Michael J. Black and Allan D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. In *Proceedings of the European Conference on Computer Vision*, pages 329–342, Cambridge, England, 1996.

[7] A. Borodin and I. Munro. *Computational Complexity of Algebraic and Numeric Problems*. American Elsevier, New York, 1975.

[8] P. Comon. Independent component analysis, a new concept? In *Signal processing 36(3)*, pages 11-20, 1994.

[9] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. John Wiley, New York, 1973.

[10] D.J. Field. what is the goal of sensory coding? In *Neural Computation 6* pages 559-601, 1994.

[11] J. Hastad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

[12] T.D. Howell. Global properties of tensor rank. *Linear Algebra and Its Applications*, 22:9–23, 1978.

[13] J. Ja'Ja. Optimal evaluation of pairs of bilinear forms. In *Proc. of the 10th Annual ACM Sym. of Theory of Computing*, pages 173–182, 1978.

[14] T. Lickteig. Typical tensorial rank. *Linear Algebra and Its Applications*, 69:95–120, 1985.

[15] R. Manduchi, P. Perona and D. Shy. Efficient implementation of deformable filter banks. *IEEE Trans. on Signal Processing*, 46(4):1168–1173, 1998.

[16] A.M. Martinez and A.C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):228–233, 2001.

[17] P. Perona. deformable kernels for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):488-499, 1995.

[18] M.Turk and A.Pentland. Eigen faces for recognition. *J. of Cognitive Neuroscience*, 3(1), 1991.

[19] H. Murase and S.K. Nayar. Learning and recognition of 3D objects from appearance. In *IEEE 2nd Qualitative Vision Workshop*, pages 39–50, New York, NY, June 1993.

[20] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(13), 1996.

[21] L. Sirovich and M. Kirby. Low dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America*, 4(3):519–524, 1987.

[22] V.N. Vapnik. *The nature of statistical learning*. Springer, 1995.