M. Alex O. Vasilescu and
Demetri Terzopoulos

Multilinear (Tensor) Image Synthesis, Analysis, and Recognition

Linear algebra, the algebra of vectors and matrices, has traditionally been a veritable workhorse in image processing. Linear algebraic methods such as principal components analysis (PCA) and its refinement known as independent components analysis (ICA) model single-factor linear variation in image formation or the linear combination of multiple sources.

In this exploratory signal processing article, we review a novel, multilinear (tensor) algebraic framework for image processing, particularly for the synthesis, analysis, and recognition of images. In particular, we will discuss multilinear generalizations of PCA and ICA and present new applications of these tensorial methods to image-based rendering and the analysis and recognition of facial image ensembles.

MULTILINEAR VS LINEAR METHODS

Ordinary images result from the interaction of multiple factors related to scene structure, illumination, and imaging. For example, facial images are determined by facial geometry (person, expression), the pose of the head relative to the camera, the lighting conditions, and the camera employed.

Linear methods, including PCA and ICA, are not well suited to the representation of multifactor image ensembles; they are better treated using nonlinear methods, specifically those based on multilinear algebra [1].

Multilinear algebra involves the natural generalization of matrices. Whereas matrices are linear operators defined over a vector space, these generalizations, referred to as *tensors*, define multilinear operators over a *set* of vector spaces. Hence, multilinear algebra, the algebra of higher-order tensors, subsumes linear algebra and matrices/vectors/scalars as a special case. Multilinear algebra serves as a unifying mathematical framework suitable for addressing a variety of challenging problems in image science and visual computing.

The multilinear algebraic framework can be applied to the synthesis, analysis, and recognition of images. Within this mathematical framework, the image ensemble of interest is represented as a higher-order tensor, which must be decomposed in order to separate and parsimoniously represent the constituent factors.

NOTATION

Throughout this article, we will denote scalars by italic lowercase letters $(a, b, \ldots)$, vectors by bold lowercase letters $(\mathbf{a}, \mathbf{b}, \ldots)$, matrices by bold uppercase letters $(\mathbf{A}, \mathbf{B}, \ldots)$, and higher-order tensors by calligraphic uppercase letters $(\mathcal{A}, \mathcal{B}, \ldots)$.

LINEAR PCA AND ICA

The PCA of an ensemble of I images, each comprising J pixels, is computed by performing a singular value decomposition (SVD) on a $J \times I$ *data matrix* $\mathbf{D}$. The columns of $\mathbf{D}$ represent images obtained by subtracting the mean image of the ensemble from each input image and "vectorizing" it by consistently arranging the J pixels into a column vector. Regarding entire images as vectors or points in a high, J-dimensional space enables PCA to model the full second-order image statistics of all pairs of pixels in the image.

The matrix $\mathbf{D}$ has two associated vector spaces, a row space and a column space. In a factor analysis of $\mathbf{D}$, the SVD orthogonalizes these two spaces and decomposes the matrix as $\mathbf{D} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, where the orthonormal matrix $\mathbf{U}$ represents the column space, $\boldsymbol{\Sigma}$ is a diagonal matrix whose nonincreasing, nonnegative entries are referred to as the singular values of $\mathbf{D}$, and the orthonormal matrix $\mathbf{V}$ represents the row space.

The column vectors of $\mathbf{U}$, or singular vectors, are also called the *principal component* (or Karhunen-Loeve) directions of $\mathbf{D}$. Optimal dimensionality reduction in matrix PCA is obtained by truncating the singular value decomposition (i.e., deleting the singular vectors associated with the smallest singular values).

The ICA of multivariate data computes second- and higher-order pixel statistics by seeking a sequence of projections such that the projected data appear as far from Gaussian as possible [2]. ICA starts essentially from the PCA solution and computes an invertible matrix which transforms the principal components into *independent components*.

MULTILINEAR GENERALIZATIONS

In the multilinear approach, an image ensemble is organized as a higher-order data tensor that must be decomposed in order to separate its constituent factors and make them explicit.

TENSORS

A *tensor* is a higher-order generalization of a matrix (second-order tensor), vector (first-order tensor), and scalar (zeroth-order tensor). Whereas matrices define linear mappings over a vector space, tensors define multilinear mappings over a set of vector spaces. The *order* of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is N.

Digital Object Identifier 10.1109/MSP.2007.906024

1053-5888/07/$25.00©2007IEEE

TENSOR FLATTENING

The mode-n vectors or "fibers" of an Nth-order tensor $\mathcal{A}$ are the I_n-dimensional vectors obtained by varying index i_n from $1 \le i_n \le I_n$ while keeping the other indices fixed.

Flattening tensors into matrices enables us to express tensor operations in terms of matrix operations. The mode-n fibers are the column vectors of matrix $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} \ldots I_N I_1 I_2 \ldots I_{n-1})}$ that results by *mode-n flattening* the tensor $\mathcal{A}$. The index order $i_{n+1}, \ldots, i_N, i_1, i_2, \ldots, i_{n-1}$ reflects the left-to-right ordering of the fibers in $\mathbf{A}_{(n)}$. Mode-n flattening is illustrated in Figure 1 for the case $N = 3$.

MODE-N PRODUCT

The mode-n product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n \times \ldots \times I_N}$ by a matrix $\mathbf{M} \in \mathbb{R}^{J_n \times I_n}$, is denoted by $\mathcal{B} = \mathcal{A} \times_n \mathbf{M}$. It can be expressed in terms of the flattened tensors as $\mathbf{B}_{(n)} = \mathbf{M}\mathbf{A}_{(n)}$. Note that the conventional SVD, given by $\mathbf{D} = \mathbf{U}\Sigma\mathbf{V}^T$, can be rewritten as $\mathbf{D} = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V}$ using mode-n products.

N-MODE SVD

A tensor $\mathcal{D}$ of order $N > 2$ has N associated spaces. The N-mode SVD is a "generalization" of conventional (two mode) SVD, which orthogonalizes these N spaces, decomposing the tensor as follows [3], [4]:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \ldots \times_n \mathbf{U}_n \ldots \times_N \mathbf{U}_N, \quad (1)$$

with $\mathcal{Z}$ referred to as the *core tensor* and $\mathbf{U}_1, \ldots, \mathbf{U}_N$ as *mode matrices*. Mode matrix $\mathbf{U}_n$ contains the orthonormal vectors spanning the column space of matrix $\mathbf{D}_{(n)}$ resulting from the mode-n flattening of $\mathcal{D}$. The core tensor governs the interaction between the mode matrices. It is analogous to the diagonal singular value matrix Σ in conventional matrix SVD, although it does not have a simple, diagonal structure.

This decomposition is illustrated in Figure 2 for $N = 3$. In the figure, $\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$. Deleting the last mode-1 singular vector of $\mathbf{U}_1$ incurs an approximation error equal to the Frobenius norm of the (grey) subtensor of $\mathcal{Z}$ whose row vectors would normally multiply the singular vector in the mode-1 product $\mathcal{Z} \times_1 \mathbf{U}_1$.

The N-mode SVD algorithm for decomposing $\mathcal{D}$ according to (1) is a multilinear extension of the conventional matrix SVD. For $n = 1, \ldots, N$, we compute matrix $\mathbf{U}_n$ in (1) by computing the SVD of the flattened matrix $\mathbf{D}_{(n)}$ and setting $\mathbf{U}_n$ to be the left matrix of the SVD. Finally, we can solve for the core tensor as $\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$.

MULTILINEAR PCA

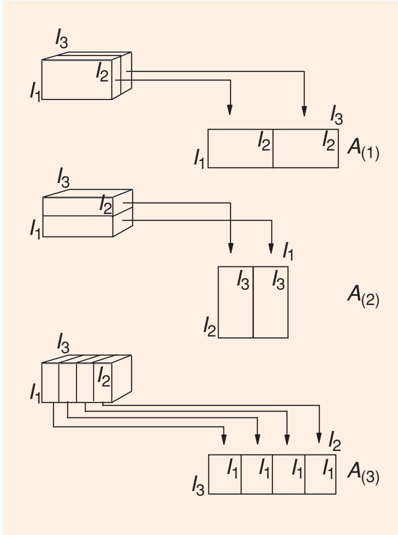
The N-mode SVD is the basis of multilinear PCA (MPCA). There is no trivial multilinear counterpart to dimensionality reduction in the linear case. A useful generalization in the tensor case involves an optimal rank-$(R_1, R_2, \ldots, R_N)$ approximation that iteratively optimizes each of the modes of the given tensor, where each optimization step involves a best reduced-rank approximation of a positive semi-definite symmetric matrix [3], [4]. This technique is a higher-order extension of the orthogonal iteration for matrices.

MULTILINEAR ICA

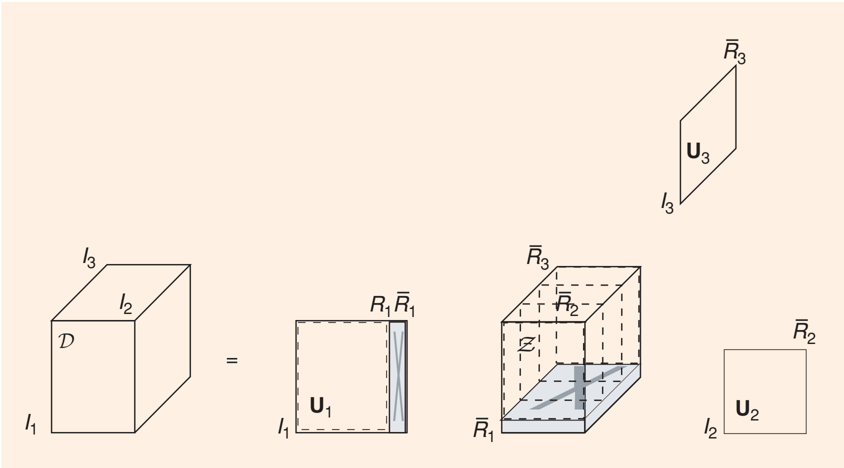
A multilinear ICA (MICA) algorithm was proposed in [5]. Analogously to (1), multilinear ICA is obtained by decomposing the data tensor $\mathcal{D}$ as the mode-n product of N mode matrices $\mathbf{C}_n$ and a core tensor $\mathcal{S}$. Analogously to the case of MPCA, optimal dimensionality reduction in MICA is achieved by optimizing mode per mode using a straightforward variant of the N-mode orthogonal iteration algorithm. The independent components for each mode are computed iteratively using alternating least squares, by solving for $\mathbf{C}_n$ while holding all the other mode matrices fixed.

APPLICATION TO IMAGE SYNTHESIS

An essential problem in computer graphics is image synthesis or rendering. The appearance of rendered surfaces is generally determined by a complex interaction of multiple factors related to scene



[FIG1] Flattening a third-order tensor. The tensor can be flattened in three ways to obtain matrices comprising its mode-1, mode-2, and mode-3 vectors.



[FIG2] The N-mode SVD for $N = 3$.

geometry, illumination, and imaging. In particular, the *bidirectional texture function* (BTF) [6] captures the appearance of extended, textured surfaces, including spatially varying reflectance, surface mesostructure (i.e., three-dimensional (3-D) texture caused by local height variation over rough surfaces), subsurface scattering, and other visually relevant phenomena over a region of the surface.

The BTF is a function of six variables $(x, y, \theta_v, \phi_v, \theta_i, \phi_i)$, where (x, y) are surface parametric (texel) coordinates and where (θ_v, ϕ_v) is the view direction and (θ_i, ϕ_i) is the illumination direction (a.k.a. the photometric angles). Given only sparsely sampled BTF data, the problem of rendering the appearance of a textured surface viewed from an arbitrary vantage point under arbitrary illumination is a problem in image-based rendering.

Linear PCA has conventionally been the BTF representation method of choice. A major limitation of PCA is that it captures the overall variation in the image ensemble without explicitly distinguishing what proportion is attributable to each of the relevant factors—illumination change, viewpoint change, etc.

TensorTextures: MULTILINEAR IMAGE-BASED RENDERING

By contrast, our multilinear framework for image-based rendering of textured surfaces from sparsely sampled data prescribes a more sophisticated, tensor decomposition that further analyzes this overall variation into individually encoded constituent factors using a novel set of basis functions. The resulting method is referred to as TensorTextures [7].

Given an ensemble of sample images of a textured surface, the TensorTextures algorithm first learns (in an offline analysis stage) a generative model that accurately approximates the BTF. Then (in the online synthesis stage) the learned model serves in rendering the appearance of the textured surface under arbitrary view and illumination conditions.

We define an image data tensor $\mathcal{D} \in \mathbb{R}^{T \times I \times V}$, where T is the number of texels in each texture image sample and where V and I are, respectively, the number of different viewing and illumination conditions associated with the image acquisition process.

Consider the synthetic scene of scattered coins shown in Figure 3. Although the coins in the treasure chest shown in Figure 3(a) appear to have considerable 3-D relief as we vary the view direction (images 1–3) and illumination direction (images 3–5), this is in fact a TensorTexture mapped onto a perfectly planar surface. The TensorTextures model has learned a compact representation of the variation in appearance of the surface under changes in viewpoint and illumination.

A total of 777 sample RGB images of the scene were acquired from $V = 37$ different view directions over the viewing hemisphere shown in Figure 3(b), each of which is illuminated by a light source oriented in $I = 21$ different directions over the illumination hemisphere shown in Figure 3(c).

We organize the ensemble of acquired images as a third-order tensor $\mathcal{D}$ with view, illumination, and texel modes, a portion of which is shown in Figure 3(d) (each tensor element is shown as a regular image rather than as a vector of texels). We apply the N-mode SVD algorithm to decompose this tensor as follows:

$$\mathcal{D} = \underbrace{\mathcal{Z} \times_1 \mathbf{U}_{\text{texels}}}_{\mathcal{T}} \times_2 \mathbf{U}_{\text{illums}} \times_3 \mathbf{U}_{\text{views}}, \tag{2}$$

the product of three orthonormal mode matrices and a core tensor $\mathcal{Z}$ that governs the interaction between the different modes. The column vectors of $\mathbf{U}_{\text{views}}$ span the view space, while its rows encode an illumination and texel invariant representation for each of the different views. The column vectors of $\mathbf{U}_{\text{illums}}$ span the illumination space, while its rows encode a view and texel invariant representations for each of the different illuminations. The TensorTextures representation $\mathcal{T}$, a portion of which is illustrated in Figure 3(e), is the extended core $\mathcal{T} = \mathcal{Z} \times_1 \mathbf{U}_{\text{texels}}$ and it is efficiently computed as $\mathcal{T} = \mathcal{D} \times_2 \mathbf{U}_{\text{illums}}^T \times_3 \mathbf{U}_{\text{views}}^T$.

TensorTextures characterizes how viewing parameters and illumination parameters interact and multiplicatively modulate the appearance of a surface under variation in view direction (θ_v, ϕ_v), illumination direction (θ_i, ϕ_i), and position (x, y) over the surface. Hence, to render an image $\mathbf{d}$, we compute

$$\mathbf{d} = \mathcal{T} \times_2 \mathbf{l}^T \times_3 \mathbf{v}^T, \tag{3}$$

where $\mathbf{v}$ and $\mathbf{l}$ are, respectively, the view and illumination representation vectors associated with the desired view and illumination directions. These will in general be *novel* directions, in the sense that they will differ from the *observed* directions associated with the sample images in the ensemble.

APPLICATION TO IMAGE ANALYSIS AND RECOGNITION

People possess a remarkable ability to recognize objects from their appearance, especially human faces despite considerable variation of (expressive) facial geometries, head poses, and lighting conditions. A highly researched class of methods in computer vision are known as appearance-based methods. They have been applied to images of arbitrary objects, but have attracted the greatest attention in the context of human facial images. Given a database of suitably labeled training images of numerous individuals, the approach aspires to learn parsimonious appearance-based representations of the image ensemble, which may be used for facial image compression and/or for facial image recognition [8].

Linear PCA has been at the core of the dominant appearance-based methods, such as the well-known "eigenfaces" face recognition method [9]. As stated earlier, however, PCA can model only single-factor variations in image ensembles. Hence, this linear method and its variants adequately address face recognition only under tightly constrained conditions—e.g., frontal images, fixed lightsources, fixed expression—where person identity is the only factor that is allowed to vary.

TensorFaces: Multilinear Facial Image Analysis and Recognition

By contrast, our multilinear approach confronts the fact that facial images result from the interaction of multiple factors, among them different facial geometries and expressions, viewpoints, and illumination conditions. The resulting method, referred to as TensorFaces [1], yields significantly better recognition rates relative to the standard, linear methods when applied to appearance-based face recognition under unconstrained conditions.

Figure 4 illustrates our technique using gray-level facial images of 75 subjects. Each subject is imaged from 15 different viewpoints ($\theta = -35^\circ$ to $+35^\circ$ in 5° steps on the horizontal plane $\phi = 0^\circ$) under 15 different illuminations ($\theta = -35^\circ$ to $+35^\circ$ in 5° steps on an inclined plane $\phi = 45^\circ$). Figure 4(b) shows the set of 225 images for one of the subjects, with viewpoints arrayed horizontally and illuminations arrayed vertically. Each image has 8,560 pixels. The image set was rendered from a 3-D scan of the subject shown boxed in Figure 4(a). The 75 head scans shown in the figure were acquired using a Cyberware 3030PS laser scanner and are part of the 3-D morphable faces database created at the University of Freiburg.

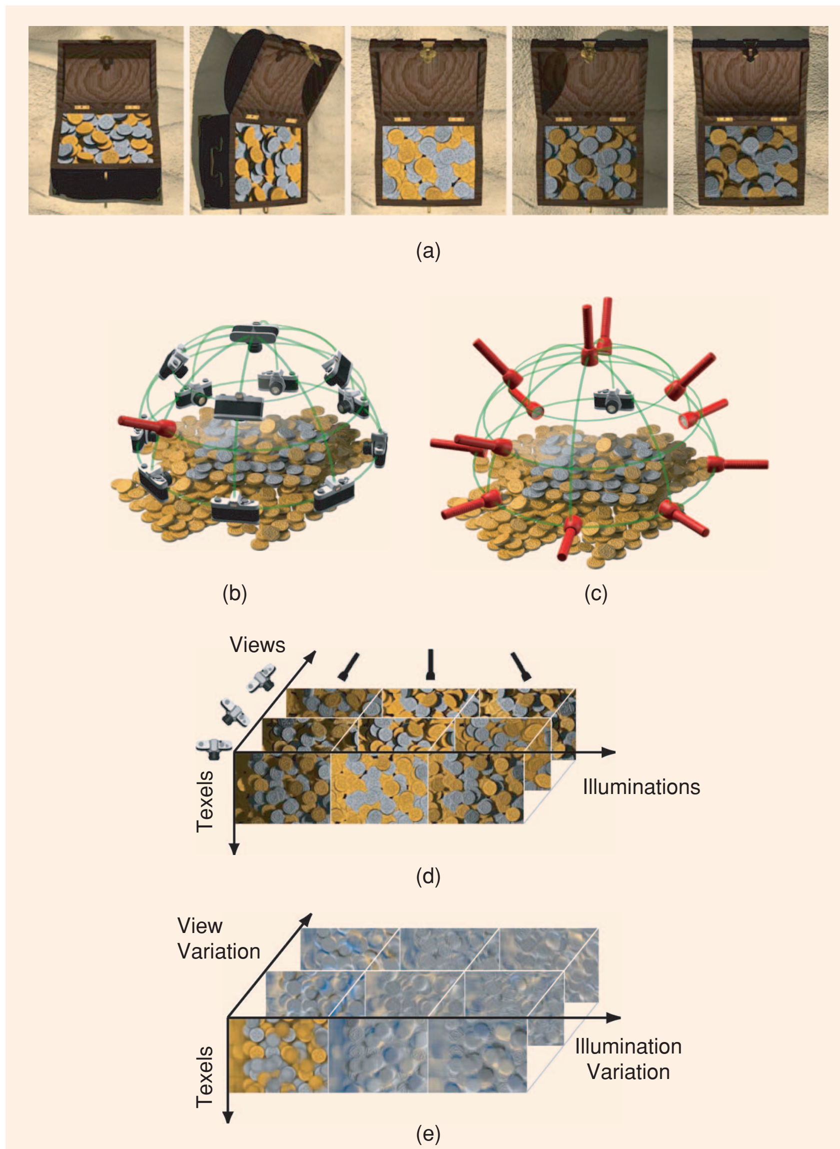
We select an ensemble of training images from the dataset of Figure 4 comprising the 36 dash-boxed images for each person. Our facial image data tensor $\mathcal{D}$, a portion of which is illustrated in Figure 4(c), has dimensions $8,560 \times 6 \times 6 \times 75$ (each tensor element is shown as a regular image rather than as a vector of pixels). Applying multilinear analysis to $\mathcal{D}$, using the N-mode SVD algorithm with $N = 4$, we obtain

$$\mathcal{D} = \underbrace{\mathcal{Z} \times_1 \mathbf{U}_{\text{pixels}}}_{\mathcal{T}} \times_2 \mathbf{U}_{\text{illums}} \times_3 \mathbf{U}_{\text{views}} \times_4 \mathbf{U}_{\text{people}}, \qquad (4)$$

where the $8,560 \times 6 \times 6 \times 75$ extended core tensor $\mathcal{T} = \mathcal{Z} \times_1 \mathbf{U}_{\text{pixels}}$, called TensorFaces, governs the interaction between the factors represented in the three mode matrices: The 6×6 mode matrix $\mathbf{U}_{\text{illums}}$ spans the space of illumination parameters, the 6×6 mode matrix $\mathbf{U}_{\text{views}}$ spans the space of viewpoint parameters, and the 75×75 mode matrix $\mathbf{U}_{\text{people}}$ spans the space of people parameters. The $8,560 \times 2,700$ mode matrix $\mathbf{U}_{\text{pixels}}$ orthonormally spans the space of images, but it need never be computed in practice. TensorFaces is computed as $\mathcal{T} = \mathcal{D} \times_2 \mathbf{U}^T_{\text{illums}} \times_3 \mathbf{U}^T_{\text{views}} \times_4 \mathbf{U}^T_{\text{people}}$.

This facial image database comprises 36 images per person that vary with viewpoint and illumination. PCA represents each image with one coefficient vector while each person is represented by a set of 36 coefficient vectors, one for each image in which the person appears. The length of each PCA coefficient vector is $6 \times 6 \times 75 = 2,700$.

By contrast, each image in the multilinear analysis is represented with a set



[FIG3] An example of TensorTextures. The chest contains a TensorTexture mapped onto a planar surface (a). Images are acquired from several different view directions over the viewing hemisphere (b) and, for each viewpoint, under several different illumination conditions over the illumination hemisphere (c). The ensemble of acquired images is organized in a third-order tensor with view, illumination, and texel modes (d). Although the contents of the texel mode are vectors of RGB texel values, for clarity they are displayed as two-dimensional images. (e) A partial visualization of the 37×21 TensorTextures bases of the coins image ensemble.

of coefficient vectors representing the illumination, viewpoint, and person modes that generated the image. However, each person, regardless of viewpoint, illumination, and expression, is represented by a single coefficient vector of dimension 75 relative to the set of bases defined by the TensorFaces $\mathcal{T}$. This many-to-one mapping is useful for face recognition. Recognition can be accomplished using a multilinear projection algorithm [10], which projects an unlabeled test image into multiple constituent mode spaces to simultaneously infer its person, illumination, and viewpoint mode labels.

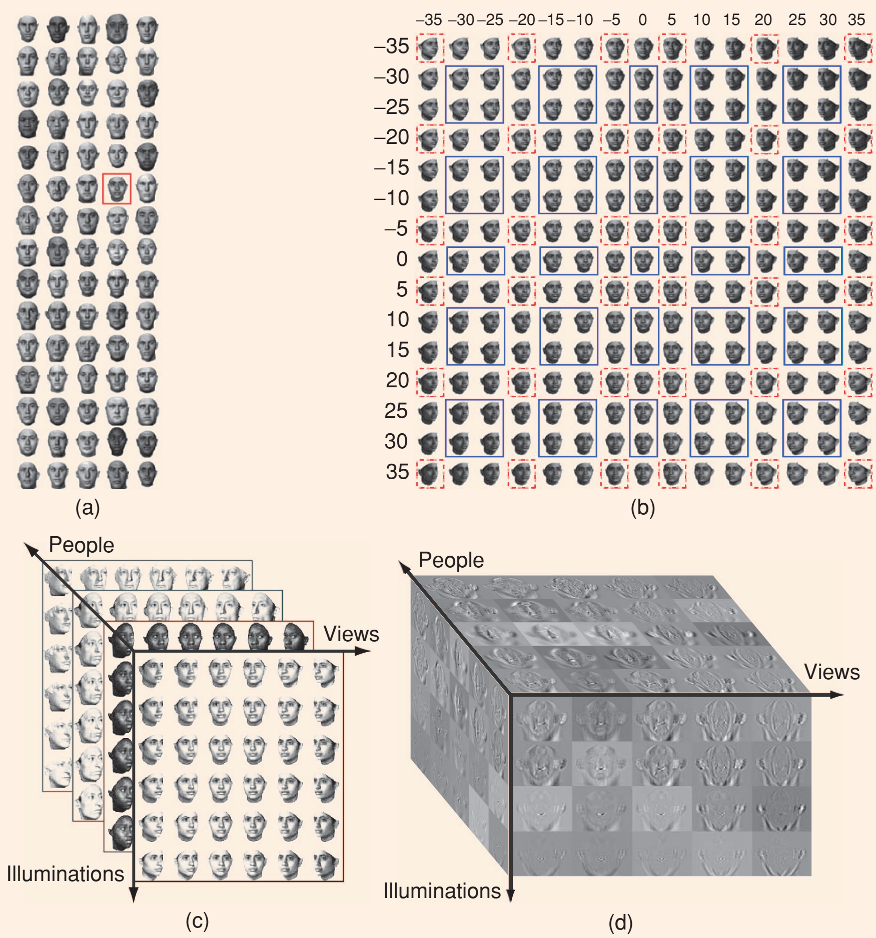
We applied the MPCA and MICA algorithms in face recognition experiments with 16,875 images captured from the University of Freiberg 3-D Morphable Faces Database. Using the bases illustrated in Figure 4(d), which were computed from a training ensemble of 2,700 images, MICA yields better recognition rates (98.14%) than PCA (eigenfaces) (83.9%), conventional ICA (89.5%) and even multilinear PCA (93.4%) in scenarios involving the recognition of test subjects whose faces were imaged in previously unseen viewpoints and illuminations.

CONCLUSION

We have presented a multilinear algebraic framework for image sythesis, analysis, and recognition, which employs a tensor (N-mode) extension of the conventional matrix SVD. This leads to a multilinear generalization of PCA and a novel multilinear generalization of ICA. We have also discussed important applications that benefit, such as image-based rendering (specifically the multilinear synthesis of images of textured surfaces for varying viewpoint and illumination), as well as multilinear analysis and recognition of facial images under variable face shape, view, and illumination conditions. These new multilinear algebraic (tensor) methods outperform their conventional linear algebraic (matrix) counterparts.

ACKNOWLEDGMENTS

This work was funded in part by the Technical Support Working Group <http://tswg.gov> through the U.S. Department of Defense's Combating Terrorism Technology Support Program. The 3-D Morphable faces database was obtained courtesy of Prof. Sudeep Sarkar of the University of South Florida (USF) as part of the USF HumanID 3-D Database.



[FIG4] A facial image dataset. (a) 3-D scans of 75 subjects, recorded using a Cyberware 3030PS laser scanner as part of the University of Freiburg 3-D morphable faces database. (b) Facial images for a subject [boxed head in (a)], viewed from 15 different viewpoints (across) under 15 different illuminations (down). In our recognition experiments, the 36 dash-boxed images served as training images and the 81 solid-boxed images served as test images. (c) A portion of the fourth-order data tensor $\mathcal{D}$ for the image ensemble formed from the dash-boxed images of each subject in (a); only four of the subjects are shown. (d) A partial visualization of the MICA representation of $\mathcal{D}$.

AUTHORS

M. Alex O. Vasilescu (maov@media.mit.edu) is a research scientist at the MIT Media Lab, Massachusetts Institute of Technology, Cambridge, and an assistant professor of computer science, at the State University of New York, Stony Brook. Her research interests include computer vision, computer graphics, and machine learning. In 2003 she was named as one of the Top 100 Young Scientists by MIT's *Technology Review* magazine.

Demetri Terzopoulos (dt@cs.ucla.edu) is the Chancellor's Professor of Computer Science at the University of California, Los Angeles. A Fellow of the IEEE and the Royal Society of Canada, he has received numerous awards for his research in computer graphics, computer vision, medical image analysis, computer-aided design, and artificial intelligence/life, among them an Academy Award for Technical Achievement from the Academy of Motion Picture Arts and Sciences.

REFERENCES

[1] M. Vasilescu and D. Terzopoulos, "Multilinear analysis for facial image recognition," in *Proc. Int. Conf. Pattern Recognition*, Quebec City, PQ, Aug. 2002, vol. 2, pp. 511–514.

[2] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[3] P. Kroonenberg and J. de Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, no. 1, pp. 69–97, 1980.

[4] L. de Lathauwer, B. de Moor, and J. Vandewalle, "On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors," *SIAM J. Matrix Analysis Applicat.*, vol. 21, no. 4, pp. 1324–1342, 2000.

[5] M. Vasilescu and D. Terzopoulos, "Multilinear independent components analysis," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. I, San Diego, CA, June 2005, pp. 547–553.

[6] K. Dana, B. van Ginneken, S. Nayar, and J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Trans.Graphics*, vol. 18, no. 1, pp. 1–34, 1999.

[7] M. Vasilescu and D. Terzopoulos, "TensorTextures: Multilinear image-based rendering," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 336–342, Aug. 2004.

[8] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, no. 5, pp. 705–740, May 1995.

[9] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[10] M. Vasilescu and D. Terzopoulos, "Multilinear projection for recognition," in *Proc. 11th IEEE Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[SP]