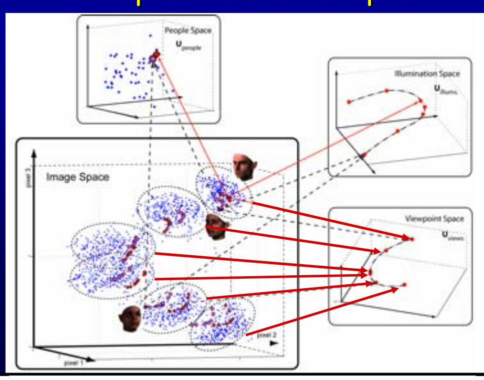
TensorFaces:

Multilinear Representation of Image Ensembles for Recognition and Compression

The Problem with Linear (PCA) Appearance Based Recognition Methods

- Eigenimages work best for recognition when only a single factor – e.g., object identity – is allowed to vary
- Natural images are the composite consequences of ***multiple factors*** (or modes) related to scene structure, illumination and imaging

Decomposition from Pixel Space to Factor Spaces



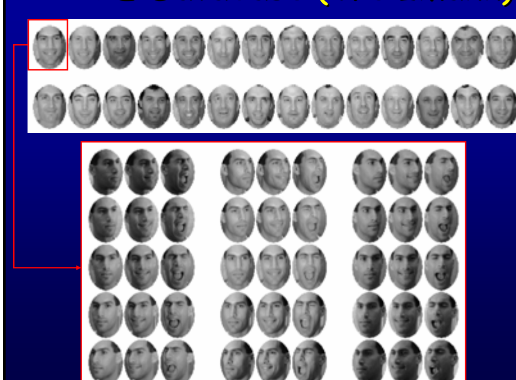
Multilinear Model Approach

- Nonlinear appearance based technique
- Appearance based model that explicitly accounts for each of the multiple factors inherent in image formation
- Multilinear algebra, the algebra of higher order tensors
- Applied to facial images, we call our tensor technique "TensorFaces" [Vasilescu & Terzopoulos, ECCV 02, ICPR 02]

TensorFaces vs Eigenfaces (PCA)

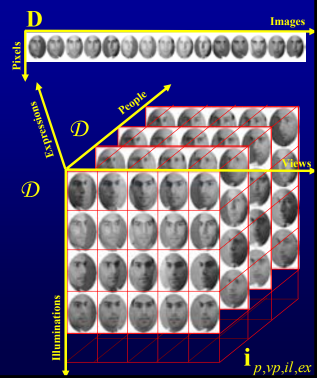
PIE Recognition Experiment	PCA	TensorFaces
Training: 23 people, 3 viewpoints (0,+34,-34), 4 illuminations **Testing:** 23 people, 2 viewpoints (+17,-17), 4 illuminations (center,left,right,left+right)	61%	**80%**
Training: 23 people, 5 viewpoints (0,+17, -17,+34,-34), 3 illuminations **Testing:** 23 people, 5 viewpoints (0,+17, -17,+34,-34), 4th illumination	27%	**88%**

PIE Database (Weizmann)



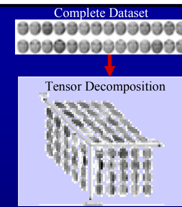
Data Organization

- Linear/PCA: **Data Matrix**
  - $R^{pixels \times images}$
  - a matrix of image vectors
- Multilinear: *Data Tensor*
  - $R^{people \times views \times illums \times express \times pixels}$
  - N-dimensional array
  - 28 people, 45 images/person
  - 5 views, 3 illuminations, 3 expressions per person



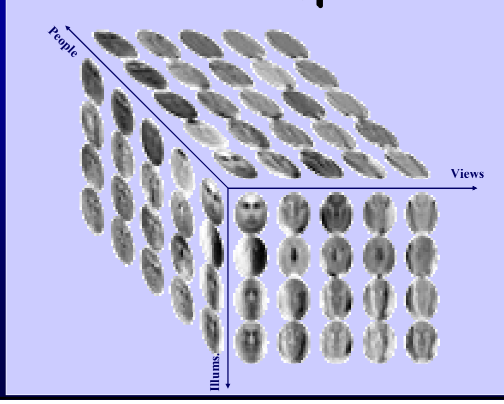
$\mathcal{D}$

$\mathbf{i}_{p,vp,il,ex}$



Learning Stage

Tensor Decomposition



Matrix Decomposition - SVD

- A matrix $\mathbf{D} \in IR^{I_1 x I_2}$ has a column and row space
- SVD orthogonalizes these spaces and decomposes $\mathbf{D}$

$$\mathbf{D} = \mathbf{U}_1 \Sigma \mathbf{U}_2^T$$

($\mathbf{U}_1$ contains the eigenfaces)

- Rewrite in terms of *mode-n products*

$$\mathbf{D} = \Sigma \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2$$

Tensor Decomposition

- $\mathcal{D}$ is a n-dimensional matrix, comprising N-spaces
- N-mode SVD is the natural generalization of SVD
- N-mode SVD orthogonalizes these spaces & decomposes $\mathcal{D}$ as the mode-n product of N-orthogonal spaces

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 \times_4 \dots \times_n \mathbf{U}_n$$

- $\mathcal{Z}$ core tensor; governs interaction between mode matrices
- $\mathbf{U}_n$, mode-n matrix, is the column space of $\mathbf{D}_{(n)}$

Multilinear (Tensor) Decomposition

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$$

$$= \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} \sigma_{r_1 r_2 r_3} \mathbf{u}_{1,r_1} \circ \mathbf{u}_{2,r_2} \circ \mathbf{u}_{3,r_3}$$

$$vec(\mathcal{D}) = (\mathbf{U}_3 \otimes \mathbf{U}_2 \otimes \mathbf{U}_1)\, vec(\mathcal{Z})$$

N-Mode SVD Algorithm

1. For n=1,…,N, compute matrix $\mathbf{U}_n$ by computing the SVD of the flattened matrix $\mathbf{D}_{(n)}$ and setting $\mathbf{U}_n$ to be the left matrix of the SVD.
2. Solve for the core tensor as follows

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$$

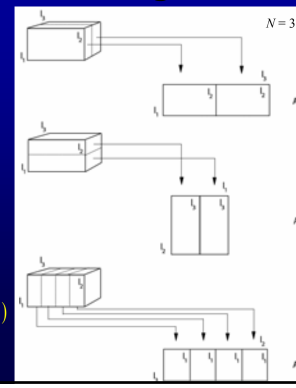
Multilinear (Tensor) Algebra

mode - *n* tensor *flattening*

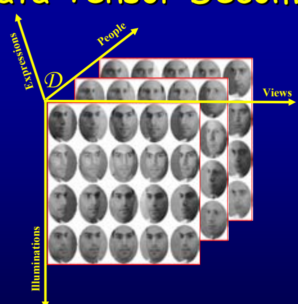
$$\mathcal{A} \in \Re^{I_1 \times I_2 \times \cdots I_N}$$

flattening = "matricize"

$$\mathbf{A}_{(n)} \in \Re^{I_n \times (I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N)}$$

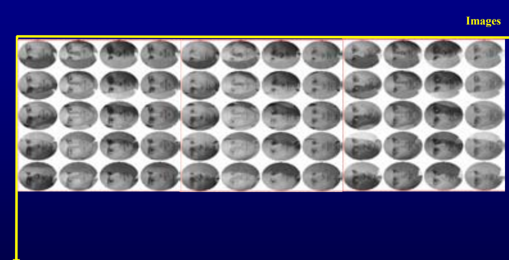


Facial Data Tensor Decomposition



$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{\text{people}} \times_2 \mathbf{U}_{\text{views}} \times_3 \mathbf{U}_{\text{illums.}} \times_4 \mathbf{U}_{\text{express}} \times_5 \mathbf{U}_{\text{pixels}}$$

Computing U_{views}



- $\mathbf{D}_{(\text{views})}$ - flatten $\mathcal{D}$ along the view point dimension
- $\mathbf{U}_{\text{views}}$ – orthogonalize the column space of $\mathbf{D}_{(\text{views})}$

Computing U_{pixels}

- $D_{(\text{pixels})}$ - flatten D along the pixel dimension
- U_{pixels} – orthogonalize $D_{(\text{pixels})}$
  – eigenimages

N-Mode SVD Algorithm

1. For n=1,…,N, compute matrix $\mathbf{U}_n$ by computing the SVD of the flattened matrix $\mathbf{D}_{(n)}$ and setting $\mathbf{U}_n$ to be the left matrix of the SVD.
2. Solve for the core tensor as follows

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$$

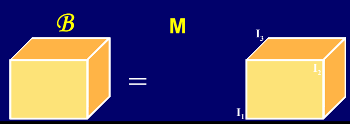
Mode-n Product

- Mode-n product is a generalization of the product of two matrices
- It is the product of a tensor with a matrix
- Mode-n product of $\mathcal{A} \in \Re^{I_1 \times \ldots \times I_n \times \ldots \times I_N}$ and $\mathbf{M} = \Re^{J_n \times I_n}$

$\mathcal{B} \in \Re^{I_1 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N}$

$$\mathcal{B} = \mathcal{A} \times_n \mathbf{M}$$

$$\left(\mathcal{A} \times_n \mathbf{M}\right)_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} m_{j_n i_n}$$



Multilinear (Tensor) Algebra

N-th order tensor $\mathcal{A} \in \Re^{I_1 \times I_2 \times \cdots I_N}$

matrix (2nd order tensor) $\mathbf{M} \in \Re^{J_n \times I_n}$

mode- n product:

$$\mathcal{B} = \mathcal{A} \times_n \mathbf{M} \quad \text{where} \quad \mathbf{B}_{(n)} = \mathbf{M}\,\mathbf{A}_{(n)}$$

Eigenfaces vs TensorFaces

- Multilinear Analysis / TensorFaces:

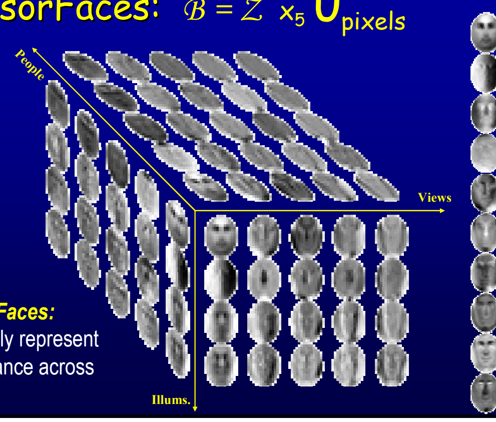
$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{\text{people}} \times_2 \mathbf{U}_{\text{views}} \times_3 \mathbf{U}_{\text{illums.}} \times_4 \mathbf{U}_{\text{express}} \times_5 \mathbf{U}_{\text{pixels}}$$

- Linear Analysis / Eigenfaces:

$$\underbrace{\mathbf{D}_{(\text{pixels})}}_{\text{data matrix}} = \underbrace{\mathbf{U}_{\text{pixels}}}_{\text{basis matrix}} \underbrace{\mathbf{Z}_{(\text{pixels})} \left(\mathbf{U}_{\text{express}} \otimes \mathbf{U}_{\text{illums.}} \otimes \mathbf{U}_{\text{views}} \otimes \mathbf{U}_{\text{people}}\right)^T}_{\text{coefficient matrix}}$$

- *TensorFaces subsumes Eigenfaces*

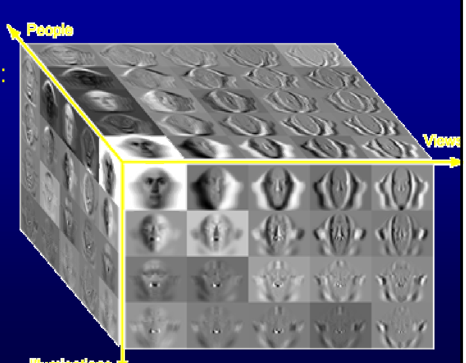
TensorFaces: $\mathcal{B} = \mathcal{Z} \times_5 \mathbf{U}_{\text{pixels}}$



TensorFaces:
explicitly represent covariance across factors

PCA:

TensorFaces:



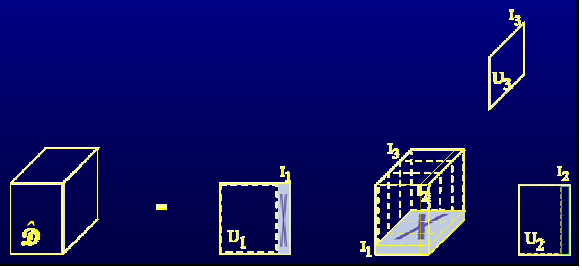
Strategic Data Compression = Perceptual Quality

- TensorFaces data reduction in illumination space primarily degrades illumination effects (cast shadows, highlights)
- PCA has *lower mean square error* but *higher perceptual error*

Original	TensorFaces	TensorFaces	PCA
	6 illum + 11 people param.	3 illum + 11 people param.	33 parameters
176 basis vectors	66 basis vectors	33 basis vectors	33 basis vectors

Dimensionality Reduction - Truncation

$$\|\mathcal{D} - \hat{\mathcal{D}}\|^2 \le \sum_{i_1=R_1}^{I_1}\sigma_{i_1}^2 + \sum_{i_2=R_2}^{I_2}\sigma_{i_2}^2 \cdots + \sum_{i_N=R_N}^{I_N}\sigma_{i_N}^2$$



Iterative Multilinear Model - Dimensionality Reduction

$$e = \|\mathcal{D} - \mathcal{Z}\times_1 \mathbf{U}_1 \times \ldots \times_n \mathbf{U}_n \times \ldots \times_N \mathbf{U}_N\| + \sum_{n=1}^{N} \lambda_n \|\mathbf{U}_n\mathbf{U}_n^T - \mathbf{I}\|$$

- Iterative data reduction approach:
  - Optimize mode per mode in an iterative way
  - Alternating Least Squares [Golub & Van Loan] improves data fit

Iterative Multilinear Model

1. Initialize $U_1^0, U_2^0, \ldots, U_N^0$:
   - Compute $U_1, U_2, \ldots, U_N$ using N-Mode SVD and
   - Truncate each mode matrix
2. Iterate:
   - $U_1^t = D \times_2 U_2^{t-1^T} \times_3 \cdots \times_N U_N^{t-1^T}$

     $U_1^t = \text{svd}(U^t_{1(1)})$
   - $U_2^t = D \times_1 U_1^{t^T} \times_3 U_3^{t-1^T} \cdots \times_N U_N^{t-1^T}$

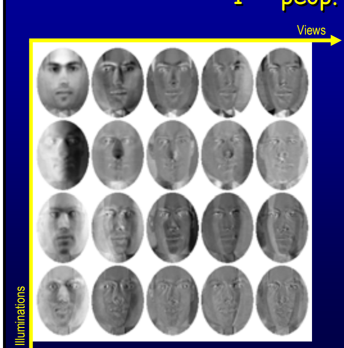
     $U_2^t = \text{svd}(U^t_{2(2)})$
3. $Z = D \times_1 U_1^{t^T} \times_2 U_2^{t^T} \times_3 \cdots \times_N U_N^{t^T}$

Construction of Projection Basis

- Multilinear decomposition allows for the construction of different basis depending on the application needs
  - Object Specific TensorFaces: person appearance model; eigenvectors span an individuals set of images
  - View Based TensorFaces
  - Recognition basis – basis maps an image into people parameter space, $\mathbf{U}_{people}$

Person Specific TensorFaces

$$\mathcal{B} = \mathcal{Z} \times_1 \mathbf{U}_{peop.} \times_5 \mathbf{U}_{pixels}$$



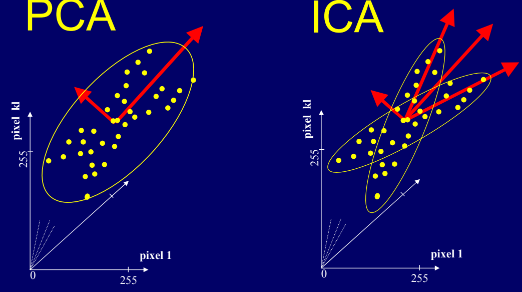
- Basis spanning one person's set of images

Perspective on Our Face Recognition Approach

	Linear Models	Our Nonlinear (Multilinear) Models
2nd -Order Statistics (covariance)	PCA Eigenfaces	Multilinear PCA TensorFaces
Higher -Order Statistics	ICA	Multilinear ICA Independent TensorFaces

Vasilescu & Terzopoulos, Learning 2004

PCA   ICA



$\mathbf{D} = \mathbf{U}\mathbf{S}\mathbf{V}^T$

basis matrix

coefficient matrix

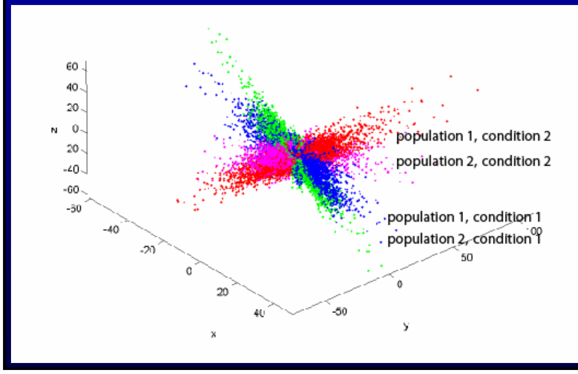
$\mathbf{D} = \mathbf{U}\ \mathbf{W}^T\ \mathbf{W}^{-T}\ \mathbf{S}\mathbf{V}^T$

$= \mathbf{K}\ \mathbf{C}$

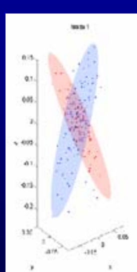
independent components

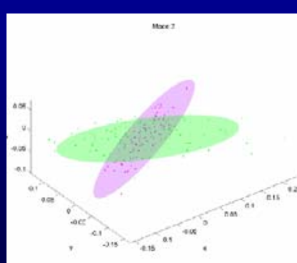
coefficient matrix

Toy Example: Observed Data



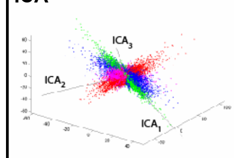
Toy Example: Hidden Variables





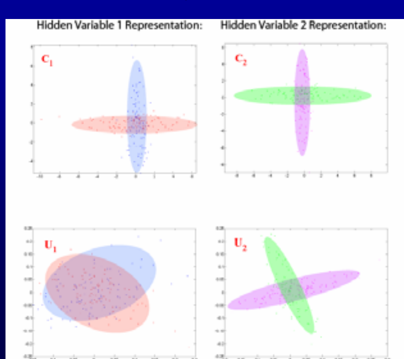
Bas

ICA



Hidden Variable Representation

- MICA
- MPCA



N-Mode ICA

1. For n=1,…,N, compute matrix $\mathbf{U}_n$ by computing the SVD of the flattened matrix $\mathbf{D}_{(n)}$ and setting $\mathbf{U}_n$ to be the left matrix of the SVD. Compute $\mathbf{W}_n^T$ using ICA. Our new mode matrix is $\mathbf{K}_n$

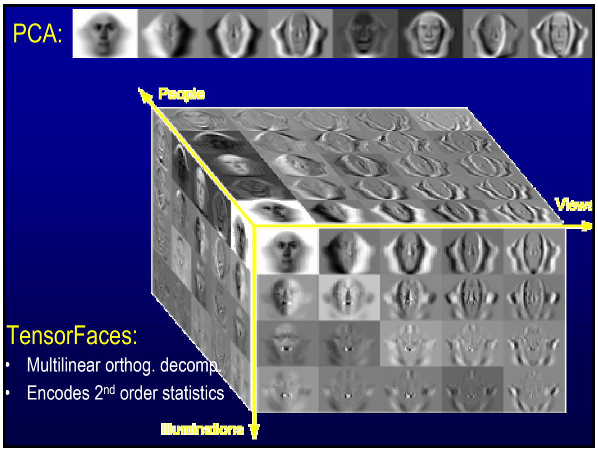
$$\mathbf{D}_{(n)} = \mathbf{U}_n \mathbf{Z}_{(n)} \mathbf{V}_n^T = \underbrace{\left(\mathbf{U}_n \mathbf{W}_n^T\right)} \mathbf{W}_n^{-T} \mathbf{Z}_{(n)} \mathbf{V}_n^T$$

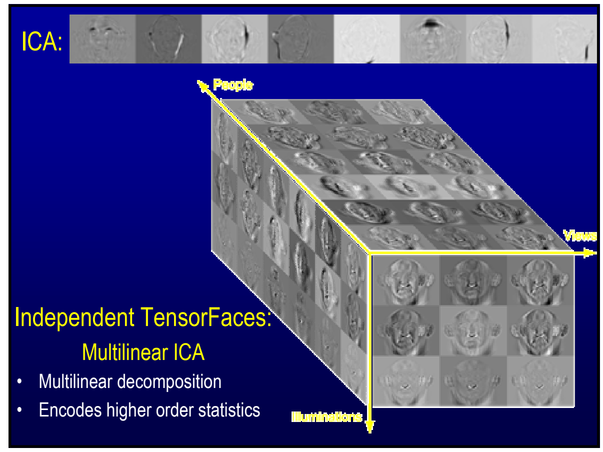
$$= \mathbf{K}_n \ \mathbf{W}_n^{-T} \mathbf{Z}_{(n)} \mathbf{V}_n^T$$

2. Solve for the core tensor as follows

$$\mathcal{S} = \mathcal{D} \times_1 \mathbf{K}_1^{-1} \times_2 \mathbf{K}_2^{-1} \times \cdots \times_n \mathbf{K}_n^{-1} \times \cdots \times_N \mathbf{K}_N^{-1}$$

$$\mathcal{S} = \mathcal{Z} \times_1 \mathbf{W}_1^{-T} \times_2 \mathbf{W}_2^{-T} \times \cdots \times_n \mathbf{W}_n^{-T} \times \cdots \times_N \mathbf{W}_N^{-T}$$


PCA:
People
Views
Illuminations
TensorFaces:
• Multilinear orthog. decomp.
• Encodes 2nd order statistics


ICA:
People
Views
Illuminations
Independent TensorFaces:
Multilinear ICA
• Multilinear decomposition
• Encodes higher order statistics