Component Analysis for Computer Vision

Fernando De la Torre

Carnegie Mellon
THE ROBOTICS INSTITUTE

Tutorial European Conference on Computer Vision
May 2006

Component Analysis

- Computer Vision & Image Processing
  - Structure from motion.
  - Spectral graph methods for segmentation.
  - Appearance and shape models.
  - Fundamental matrix estimation and calibration.
  - Compression.
  - Classification.
  - Dimensionality reduction and visualization.
- Signal Processing
  - Spectral estimation, system identification (e.g. Kalman filter), sensor array processing (e.g. cocktail problem, eco cancellation), blind source separation, …
- Computer Graphics
  - Compression (BRDF), synthesis,…
- Speech, bioinformatics, combinatorial problems.

Outline

- Introduction
- Generative models
  - Principal Component Analysis (PCA).
  - Non-negative Matrix Factorization (NMF).
  - Independent Component Analysis (ICA).
- Discriminative models
  - Linear Discriminant Analysis (LDA).
  - Oriented Component Analysis (OCA).
  - Canonical Correlation Analysis (CCA).
  - Relevant Component Analysis (RCA).
- Standard extensions of linear models
  - Latent variable models.
  - Tensor factorization.
  - Kernel methods.

Why Subspace Methods?

≡
f(p)

- Learning: High dimensional data lie in a low dimensional manifold.
- Estimation: Many cv problems (SFM, calibration, …) can be posed as subspace estimation.
- Better generalization (noise removal).
- Simple parameterization.
- Lower computational complexity.
- Closed form solution and global minimum.

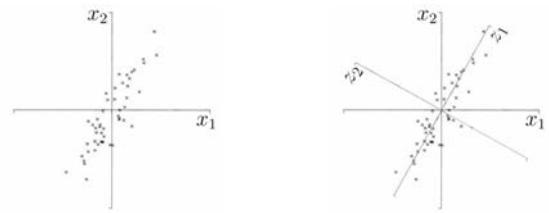
Generative Models

$$\mathbf{D} \approx \mathbf{BC}$$

- Principal Component Analysis/Singular Value Decomposition.
  1) Robust PCA/SVD.
  2) PCA with uncertainty and missing data.
  3) Parameterized PCA.
  4) PCA over continuous spaces.
  5) Filtered PCA.
  6) PCA of rotated images.
  7) Mixture of subspaces.
- Non-Negative Matrix Factorization.
  8) PCA and NMF for spectral clustering.
- Independent Component Analysis.

Principal Component Analysis (PCA)

(Pearson, 1901; Hotelling, 1933;Mardia et al., 1979; Jolliffe, 1986; Diamantaras, 1996)



- PCA finds the directions of maximum variation of the data based on linear correlation.
- PCA decorrelates the original variables.

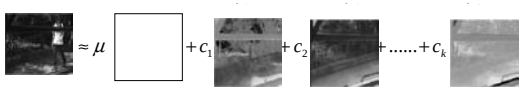
PCA

$$\mathbf{D} = \overbrace{[\mathbf{d}_1 \ \mathbf{d}_2 \ \dots \ \mathbf{d}_n]}^{\mathbf{n} = \text{images}} \approx \mathbf{BC} + \boldsymbol{\mu}\mathbf{1}_n^T$$

$\mathbf{d} =$ pixels

$\mathbf{D} \in \Re^{d \times n}$ $\mathbf{B} \in \Re^{d \times k}$

$\mathbf{C} \in \Re^{k \times n}$ $\boldsymbol{\mu} \in \Re^{d \times 1}$



•Assuming 0 mean data, the basis **B** that preserve the maximum variation of the signal is given by eigenvectors $\mathbf{DD}^T$.

$$\overbrace{\mathbf{DD}^T}^{d} \mathbf{B} = \mathbf{B}\boldsymbol{\Lambda}$$

Component Analysis for Computer Vision F. De la Torre **ECCV-06** 7

Snap-shot method & SVD

- If d>>n (e.g. images 100*100 vs. 300 samples) no $\mathbf{DD}^T$.
- $\mathbf{DD}^T$ and $\mathbf{D}^T\mathbf{D}$ have the same eigenvalues (energy) and related eigenvectors (by **D**).
- **B** is a linear combination of the data! (Sirovich, 1987)

$$\mathbf{DD}^T\mathbf{B} = \mathbf{B}\boldsymbol{\Lambda} \quad \mathbf{B} = \mathbf{D}\boldsymbol{\alpha} \quad \cancel{\mathbf{D}^T\mathbf{D}}\mathbf{D}^T\mathbf{D}\boldsymbol{\alpha} = \cancel{\mathbf{D}^T\mathbf{D}}\boldsymbol{\alpha}\boldsymbol{\Lambda}$$

- $[\boldsymbol{\alpha},\mathbf{L}]=\text{eig}(\mathbf{D}^T\mathbf{D})$ $\mathbf{B}=\mathbf{D}\,\boldsymbol{\alpha}(\text{diag}(\text{diag}(\mathbf{L})))^{-0.5}$
- SVD factorizes the data matrix **D** as:

(Beltrami, 1873; Schmidt, 1907; Golub & Loan, 1989)

$$\mathbf{DD}^T = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T \qquad \mathbf{D}^T\mathbf{D} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$$

PCA	SVD
$\mathbf{D} = \mathbf{BC}$	$\mathbf{D} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$
$\mathbf{B} \in \Re^{d \times k}$ $\mathbf{C} \in \Re^{k \times n}$	$\mathbf{U} \in \Re^{d \times k}$ $\boldsymbol{\Sigma} \in \Re^{k \times n}$ $\mathbf{V} \in \Re^{n \times n}$
$\mathbf{B}^T\mathbf{B} = \mathbf{I}$ $\mathbf{CC}^T = \boldsymbol{\Lambda}$	$\mathbf{U}^T\mathbf{U} = \mathbf{I}$ $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ $\boldsymbol{\Sigma}$ *diagonal*

Error function for PCA

- PCA minimizes the reconstruction error.

  (Eckardt & Young, 1936; Gabriel & Zamir, 1979; Baldi & Hornik, 1989; Shum et al., 1995; de la Torre & Black, 2003a)

$$E_1(\mathbf{B},\mathbf{C}) = \sum_{i=1}^{n} \left\| \mathbf{d}_i - \mathbf{B}\mathbf{c}_i \right\|_2^2 = \left\| \mathbf{D} - \mathbf{B}\mathbf{C} \right\|_F$$

- Not unique solution: $\mathbf{BRR}^{-1}\mathbf{C} = \mathbf{BC}$ $\quad \mathbf{R} \in \Re^{k \times k}$
- To obtain same PCA solution R has to satisfy:

$$\hat{\mathbf{B}} = \mathbf{BR} \qquad \hat{\mathbf{C}} = \mathbf{R}^{-1}\mathbf{C}$$

$$\hat{\mathbf{B}}^T\hat{\mathbf{B}} = \mathbf{I} \qquad \hat{\mathbf{C}}\hat{\mathbf{C}}^T = \Lambda$$

- **R** is computed as a generalized k×k eigenvalue problem.

  (de la Torre, 2006)

$$\left(\mathbf{C}\mathbf{C}^T\right)^{-1}\mathbf{R} = \mathbf{B}^T\mathbf{B}\mathbf{R}\Lambda^{-1}$$

PCA/SVD in computer vision

- PCA/SVD has been applied to:
  - Recognition (eigenfaces:Turk & Pentland, 1991; Sirovich & Kirby, 1987; Leonardis & Bischof, 2000; Gong et al., 2000; McKenna et al., 1997a)
  - Parameterized motion models (Yacoob & Black, 1999; Black et al., 2000; Black, 1999; Black & Jepson, 1998)
  - Appearance/shape models (Cootes & Taylor, 2001; Cootes et al., 1998; Pentland et al., 1994; Jones & Poggio, 1998; Casia & Sclaroff, 1999; Black & Jepson, 1998; Blanz & Vetter, 1999; Cootes et al., 1995; McKenna et al., 1997; de la Torre et al., 1998b; de la Torre et al., 1998b)
  - Dynamic appearance models (Soatto et al., 2001; Rao, 1997; Orriols & Binefa, 2001; Gong et al., 2000)
  - Structure from Motion (Tomasi & Kanade, 1992; Bregler et al., 2000; Sturm & Triggs, 1996; Brand, 2001)
  - Illumination based reconstruction (Hayakawa, 1994)
  - Visual servoing (Murase & Nayar, 1995; Murase & Nayar, 1994)
  - Visual correspondence (Zhang et al., 1995; Jones & Malik, 1992)
  - Camera motion estimation (Hartley, 1992; Hartley & Zisserman, 2000)

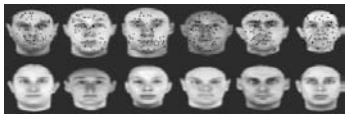
More PCA/SVD work

- PCA/SVD has been applied to:
  - Image watermarking (Liu & Tan, 2000)
  - Signal processing (Moonen & de Moor, 1995)
  - Neural approaches (Oja, 1982; Sanger, 1989; Xu, 1993)
  - Bilinear models (Tenenbaum & Freeman, 2000; Marimont & Wandell, 1992)
  - Direct extensions (Welling et al., 2003; Penev & Atick, 1996)
- And many more (google)…
  - Results **1 - 10** of about **1,870,000** for **"principal component analysis"**.
- Still work to do
  - Results **1 - 10** of about **65,300,000** for **"Britney spears"**.

1-Robust PCA

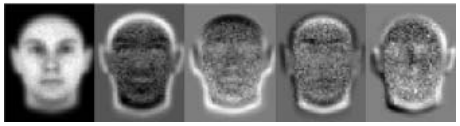
- Two types of outliers:

Sample outliers
(Xu & Yuille., 1995)



Intra-sample outliers
(de la Torre & Black, 2001b; Skocaj & Leonardis, 2003)

- Standard PCA solution (noisy data):

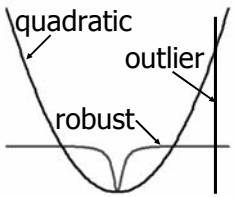


Robust PCA

- Using robust statistics:
  (de la Torre & Black, 2001b; de la Torre & Black, 2003a)

Pixel residual

$$E_{rpca}(\mathbf{B},\mathbf{C},\boldsymbol{\mu}) = \sum_{i=1}^{n}\sum_{p=1}^{d}\rho(d_{pi} - \mu_p - \sum_{j=1}^{k} b_{pj}c_{ji}, \sigma_p)$$



mean

Basis (**B**) & Coefficients(**c**)

Numerical problems

- No closed form solution in terms of an eigen-equation.
- Deflation approaches do not hold.

$$\mathbf{A}' = \mathbf{A} - \lambda_1 \mathbf{u}_1\mathbf{u}_1^T$$ → First eigenvector with highest eigenvalue.

$$\mathbf{A}'' = \mathbf{A}' - \lambda_2 \mathbf{u}_2\mathbf{u}_2^T$$ → Second eigenvector with highest eigenvalue.

$\ldots$

- In the robust case all the basis have to be computed simultaneously (including the mean).

How to optimize it?

$$E_{rpca}(\mathbf{B},\mathbf{C},\boldsymbol{\mu}) = \sum_{i=1}^{n}\sum_{p=1}^{d}\rho(d_{pi} - \mu_p - \sum_{j=1}^{k} b_{pj}c_{ji}, \sigma_p)$$

- Normalized Gradient descent

$$\mathbf{B}^{n+1} = \mathbf{B}^n - [\mathbf{H}_b]^{-1} \circ \frac{\partial E_{rpca}}{\partial \mathbf{B}} \qquad \mathbf{H}_\mathbf{b} = \max\ diag\ (\frac{\partial^2 E_{rpca}}{\partial \mathbf{b}_i \partial \mathbf{b}_i^T})$$

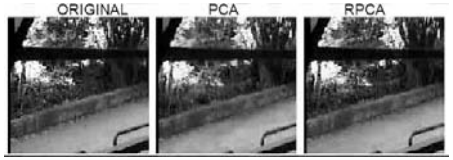
$$\mathbf{C}^{n+1} = \mathbf{C}^n - [\mathbf{H}_c]^{-1} \circ \frac{\partial E_{rpca}}{\partial \mathbf{C}} \qquad \mathbf{H}_\mathbf{c} = \max\ diag\ (\frac{\partial^2 E_{rpca}}{\partial \mathbf{c}_i \partial \mathbf{c}_i^T})$$

- Deterministic annealing methods to avoid local minima.
  (Blake & Zisserman, 1987)

Example

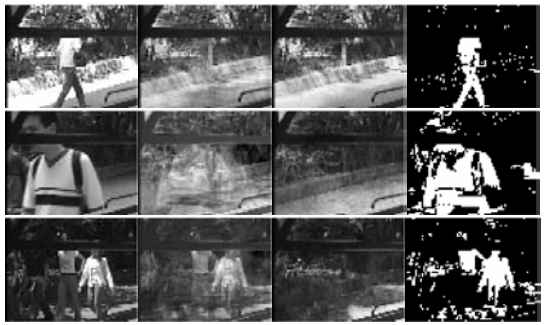
Statistical outlier

- Small region
- Short amount of time

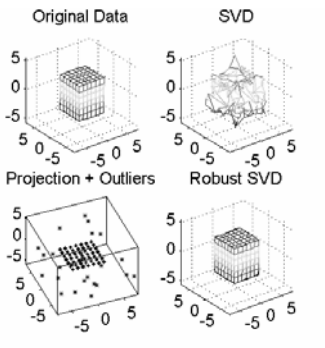


Robust PCA

Original PCA RPCA Outliers



Structure from Motion



Related RPCA work

- Robust estimation of coefficients
  (Black & Jepson, 1998; Leonardis & Bischof, 2000; Ke & Kanade, 2004)
- Robust estimation of basis and coefficients
  (Gabriel & Odoro, 1984; Croux & Filzmoser., 1981; Skocaj et al., 2002;Skocaj & Leonardis, 2003; de la Torre & Black, 2001b; de la Torre & Black, 2003a)
- Other Robust PCA techniques (sample outliers)
  (Campbell, 1980; Ruymagaart, 1981; Xu & Yuille., 1995)

2- PCA with uncertainty and missing data

- Adding uncertainty $E_2(\mathbf{B},\mathbf{C}) = \|\mathbf{W} \circ (\mathbf{D} - \mathbf{B}\mathbf{C})\|_F = \sum_{i=1}^{d}\sum_{j=1}^{n} w_{ij}(d_{ij} - \sum_{s=1}^{k} b_{is}c_{sj})^2$

- If weights are separable $\mathbf{W} = \mathbf{w}_r \mathbf{w}_c^T$ close form solution.

$$\mathbf{w}^c = \begin{pmatrix} w_1^c & w_2^c & \dots & w_n^c \end{pmatrix}$$

$$\mathbf{w}^r = \begin{pmatrix} w_1^r \\ w_2^r \\ \dots \\ w_d^r \end{pmatrix} \quad \begin{pmatrix} d_{11} & \cdots & \cdots & d_{1n} \\ d_{21} & \ddots & & d_{2n} \\ \vdots & & \ddots & \vdots \\ d_{d1} & \cdots & \cdots & d_{dn} \end{pmatrix} = \mathbf{D}$$

$$\mathbf{W} \in \Re^{d \times n}$$

$$w_{ij} \geq 0$$

$\circ$ *Hadamard product*

- Generalized SVD
  (Greenacre, 1984; Irani & Anandan, 2000;)

General case

- For arbitrary weights no closed-form solution.

$$E_2(\mathbf{B},\mathbf{C}) = \|\mathbf{W}\circ(\mathbf{D}-\mathbf{B}\mathbf{C})\|_F = \sum_{i=1}^{n}(\mathbf{d}_i-\mathbf{B}\mathbf{c}_i)^T diag(\mathbf{w}_i)(\mathbf{d}_i-\mathbf{B}\mathbf{c}_i) =$$

$$\sum_{p=1}^{d}(\mathbf{d}^p-\mathbf{C}^T\mathbf{b}^p)^T diag(\mathbf{w}^p)(\mathbf{d}^p-\mathbf{C}^T\mathbf{b}^p)$$ (Torre & Black, 2003a)

- Alternated least squares algorithms
  - Slow convergence, easy implementation.
- Damped Newton Algorithm
  - Fast convergence. (Buchanan & Fitzgibbon., 2005)

$$E_2(\mathbf{B},\mathbf{C}) = \|\mathbf{W}\circ(\mathbf{D}-\mathbf{B}\mathbf{C})\|_F + \lambda_1\|\mathbf{B}\|_F + \lambda_2\|\mathbf{C}\|_F$$

$$\mathbf{v} = \begin{bmatrix} vec(\mathbf{B}) \\ vec(\mathbf{C}) \end{bmatrix} \quad \mathbf{v}^{(n+1)} = \mathbf{v}^n - \left[\frac{[\partial^2 E_2]}{\partial^2 \mathbf{v}}\right]^{-1}\frac{\partial E_2}{\partial \mathbf{v}}$$

  - H definite positive: $\mathbf{H} = \frac{\partial^2 E_2}{\partial^2 \mathbf{v}} + \lambda \mathbf{I}$

repeat

$$\mathbf{H} = \frac{\partial^2 E_2}{\partial^2 \mathbf{v}} \quad \mathbf{g} = \frac{\partial E_2}{\partial \mathbf{v}}$$

repeat

$$\lambda = 10\lambda$$

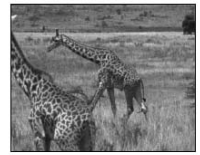
$$\mathbf{y} = \mathbf{x} - (\mathbf{H}+\lambda I)^{-1}\mathbf{g}$$

until $F(\mathbf{y}) < F(\mathbf{x})$

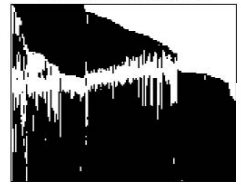
$$\mathbf{x} = \mathbf{y}; \lambda = \frac{\lambda}{10}$$

until convergence

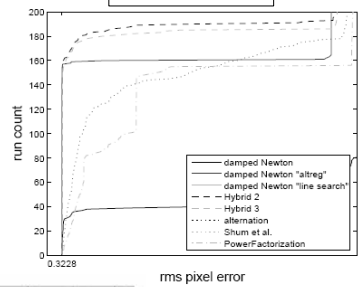
Experiments

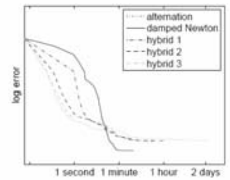


240×167: 70% known



Total: 500 runs





Related work

- Iterative (Wiberg, 1976; Shum et al., 1995; Morris & Kanade, 1998; Aans et al.,2002; Guerreiro & Aguiar, 2002)
- Closed-form (Aguiar & Moura, 1999; Irani & Anandan, 2000)
- Power factorization (Hartley & Schaalitzky, 2003)
- Bayesian estimation (L.Torresani & Bregler, 2004)

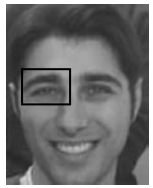
Incremental PCA

- (de la Torre et al., 1998b; Ross et al., 2004; Brand, 2002; Skocaj & Leonardis, 2003; Champagne & Liu., 1998; A. Levy, 2000)

3- Parameterized Component Analysis (PaCA) (de la Torre & Black, 2003b)

- Learn a subspace invariant to geometric transformations?

- Data has to be **geometrically** normalized
  - Tedious manual cropping.
  - Inaccuracies due to matching ambiguities.
  - Hard to achieve sub-pixel accuracy.



Error function for PaCA

$$E(\mathbf{B},\mathbf{C},\mathbf{a}) = \sum_{t=1}^{T} \left\| \mathbf{d}_t(\mathbf{f}(\mathbf{x},\mathbf{a}_t)) - \mathbf{B}\mathbf{c}_t \right\|^2_{\mathbf{W}_1} + p_1(\mathbf{a}) + p_2(\mathbf{c})$$

Motion (warping)

Basis (**B**) & coefficients (**c**)

Regularization

$$\sum_{t=1}^{T}\sum_{l=1}^{L} \lambda_1 \left\| \mathbf{c}_t - \mathbf{\Gamma}_c \mathbf{c}_{t-1} \right\|^2_{\mathbf{W}2} + \lambda_2 \left\| \mathbf{a}_t - \mathbf{\Gamma}_a \mathbf{a}_{t-1} \right\|^2_{\mathbf{W}3}$$

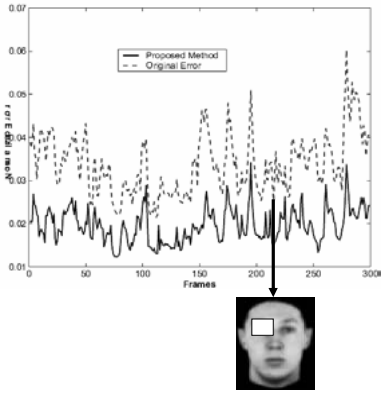
Solving the optimization problem

- Linearizing the motion (Bergen et al., 1992;Black & Jepson, 1998)

$$\mathbf{d}_t(\mathbf{f}(\mathbf{x},\mathbf{a}_t)) \approx \mathbf{d}_t(\mathbf{f}(\mathbf{x},\mathbf{a}_{0t})) + \mathbf{J}_t \Delta\mathbf{a}_t$$

- Normalized gradient descent w.r.t. all parameters + deterministic annealing.
  - Update for **c** (appearance) & **a** (motion).
  - Updated for **B** (appearance basis).
- It is a non-convex function.
  - Stochastic initialization (G.A). (Lanitis et al., 1995; de la Torre & Black, 2003b)
  - Multiresolution motion estimation framework.

EigenEye Learning



More on parameterized CA

- Probabilistic model
  - Search scales exponentially with the number of motion parameters (Frey & Jojic, 1999a; Frey & Jojic, 1999b; Williams & Titsias, 2004)
- Other continuous approaches.
  (Schewitzer, 1999; Rao, 1999; Shashua et al., 2002)
- Invariant clustering
  (Fitzgibbon & Zisserman, 2003)
- Non-rigid motion
  (Baker et al., 2004)
- Invariant recognition
  (Black & Jepson, 1998)
- Invariant support vector machines (Avidan, 2001)

4- PCA over continuous spaces

(Levin & Shashua, 2002)

- PCA assumes discrete samples, but in the limit:

$$\frac{1}{N}\sum_{i=1}^{N} \mathbf{d}_i \mathbf{d}_i^T \xrightarrow[N \to \infty]{} \int f(\mathbf{d})\mathbf{d}\mathbf{d}^T d\mathbf{d}$$

- Sometimes not uniform sampling of f(**d**).

Left illumination

Right illumination

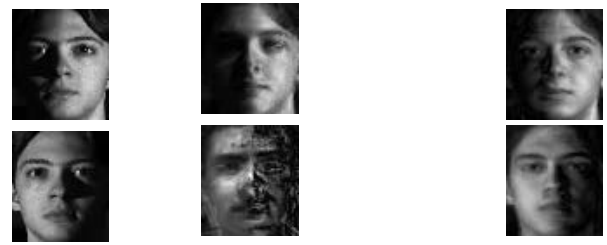
Bias solution

- 3 principal components (38 people)

Original

Bias

Unbiased



How to unbiased the solution?

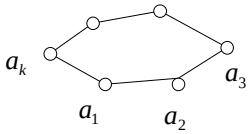
- Weighting the data.
  - Not clear which is the optimal weight. $E(\mathbf{B},\mathbf{C}) = \sum_{i=1}^{n} w_i \|\mathbf{d}_i - \mathbf{B}\mathbf{c}_i\|_F$
- More elegant approach.
  - Find the principal components over the data points represented by (dense) uniform sampling.
  - Integrating over the convex combination of the examples is less sensitive to the particular sample

PCA OVER POLYTOPS

- PCA over polytopes

$$Cov(W) = \frac{1}{V(W)} \int_{a \in W} aa^T da$$

Volume of polytops



- A simple 2D case, two points $\mathbf{a}_1$ and $\mathbf{a}_2$:
  - Regular PCA $\mathbf{a}_1\mathbf{a}_1^T + \mathbf{a}_2\mathbf{a}_2^T$
  - Integrating over a line $\lambda\mathbf{a}_1 + (1-\lambda)\mathbf{a}_2, 0 \le \lambda \le 1$

$$\max_{|u|=1} \int_{\mathbf{a}\in W} |\mathbf{a}^\top \mathbf{u}|^2 d\mathbf{a} = \mathbf{u}^\top \left[\int_{\mathbf{a}\in W} \mathbf{a}\mathbf{a}^\top d\mathbf{a}\right] \mathbf{u} \qquad \int_0^1 \lambda^2 d\lambda = \int_0^1 (1-\lambda)^2 d\lambda = \frac{1}{3}, \quad \int_0^1 \lambda(1-\lambda) d\lambda = \frac{1}{6},$$

$$\max_{|u|=1} \mathbf{u}^\top [\mathbf{a}_1\mathbf{a}_1^\top + \mathbf{a}_2\mathbf{a}_2^\top + \frac{1}{2}(\mathbf{a}_1\mathbf{a}_2^\top + \mathbf{a}_2\mathbf{a}_1^\top)]\mathbf{u} \qquad A\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} A^\top = A\Phi A^\top$$

In general

$$\Phi_k = \frac{1}{k(k+1)}\begin{bmatrix} 2 & 1 & \dots & 1 \\ 1 & 2 & \dots & 1 \\ . & . & . & . \\ 1 & 1 & \dots & 2 \end{bmatrix} = \frac{1}{k(k+1)}(\mathbf{I} + ee^T)$$

$$Cov(W) = \frac{1}{V(W)} \int_{a \in W} \mathbf{d}\mathbf{d}^T d\mathbf{d} = \frac{1}{k(k+1)} \mathbf{D}(\mathbf{I} + ee^T)\mathbf{D}^T$$

- Traditional PCA $\frac{1}{k}\mathbf{D}\mathbf{D}^T$
- Continuous PCA $\frac{1}{k(k+1)}\mathbf{D}(\mathbf{I} + \mathbf{e}\mathbf{e}^T)\mathbf{D}^T$

5-Filtered PCA

(Bischof et al., 2004; Wildenauer et al., 2002)

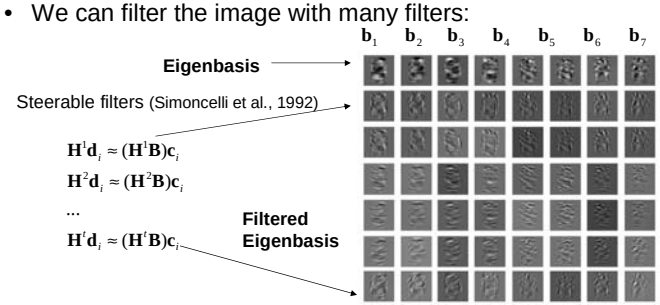
- How to construct eigenspaces robust:
  - Varying illumination.
  - Occlusion.
  - Noise.

- Filtered PCA.
  - **H** is a convolution matrix (block circulant structure)
  - The coefficients of $\mathbf{c}_i$ remain the same under a convolution.

$$\mathbf{H} \in \Re^{d \times d}$$
$$\mathbf{d}_i \in \Re^{d \times 1} \qquad \mathbf{d}_i \approx \mathbf{B}\mathbf{c}_i \Rightarrow \mathbf{H}\mathbf{d}_i \approx (\mathbf{H}\mathbf{B})\mathbf{c}_i$$

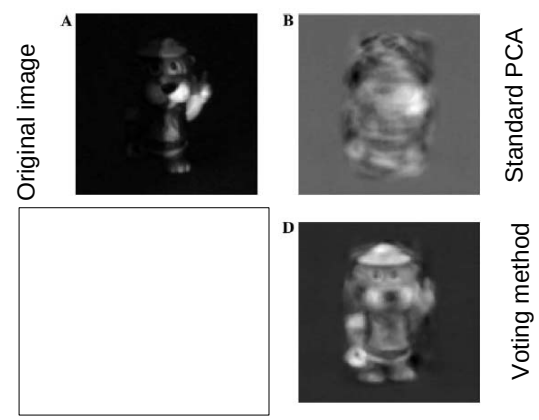
Filtered PCA

- We can filter the image with many filters:



- Given a new image, compute the filtered representation and robustly compute $\mathbf{c}_i$(eigenbasis known).

Experiments of Filtered PCA



More results

Obj.	1	2	3	4	5	6	7	8	9	10	%	Ang.
Robust filtered method—all eigenvectors used												
1	360	0	0	0	0	0	0	0	0	0	100.0	4.92
2	0	321	3	0	0	0	0	0	0	0	99.1	9.03
3	0	1	503	0	0	0	0	0	0	0	99.8	0.99
4	3	1	0	355	1	0	0	0	0	0	98.6	3.30
5	0	0	0	0	612	0	0	0	0	0	100.0	3.79
6	0	9	0	0	11	672	13	1	8	6	93.3	15.85
7	0	14	0	8	22	10	413	26	7	4	81.9	4.46
8	0	11	0	0	51	37	1	284	42	6	65.7	15.74
9	0	16	2	9	17	13	0	7	439	1	87.1	14.10
10	1	0	3	0	9	46	0	5	39	509	83.2	13.97
Avg.											90.6	8.71
Standard method—all eigenvectors used												
1	74	0	0	40	179	41	4	2	4	16	20.6	2.84
2	0	215	0	6	2	67	0	1	20	13	66.4	26.37
3	0	7	237	3	187	28	7	0	0	35	47.0	3.25
4	3	2	0	236	43	28	0	0	44	4	65.6	12.03
5	0	0	8	0	535	26	7	17	19	0	87.4	8.73
6	3	78	0	0	65	553	0	3	4	14	76.8	41.14
7	0	19	0	20	96	74	181	44	22	48	35.9	6.02
8	0	30	0	0	134	124	3	119	22	0	27.5	16.47
9	0	10	0	54	62	102	4	11	255	6	50.6	22.24
10	4	10	2	24	85	175	5	4	36	267	43.6	24.08
Avg.											54.2	19.48

Robust method

Standard method

Eigenfiltering for flexible Eigentracking

(de la Torre et al., 2000)

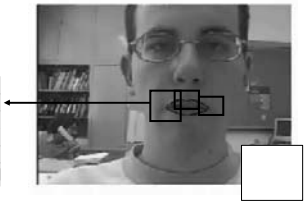
$$\prod_{i=1}^{N} k e^{-\frac{1}{2}(\mathbf{I}_F(\mathbf{x}_i)-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{I}_F(\mathbf{x}_i)-\boldsymbol{\mu}_i)}$$

Non-rigid parameters

$$\mathbf{x}_i = \mathbf{f}(\bar{x}_i, \mathbf{a}) + \sum_{i=1}^{k} b_i \mathbf{f}(\mathbf{u}_i^S, \mathbf{a})$$

Rigid parameters

$$\mathbf{I}_F(\mathbf{x}_i) = \begin{bmatrix} \mathbf{H}^1\mathbf{d}_i \\ \mathbf{H}^2\mathbf{d}_i \\ \dots \\ \mathbf{H}^t\mathbf{d}_i \end{bmatrix}$$



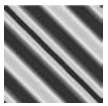
6- PCA of a set of Rotated Images

(Uenohara & Kanade, 1998; Jogan et al., 2003)

- Set of uniformly in-plane rotated versions of the same object (0 background).

- $\boldsymbol{\Gamma} = \mathbf{D}^T\mathbf{D}$ is a circulant symmetric Toeplitz matrix.

$$\mathbf{D}^T\mathbf{D} = \begin{bmatrix} \mathbf{d}_1^T\mathbf{d}_1 & \mathbf{d}_2^T\mathbf{d}_1 & \dots & \mathbf{d}_n^T\mathbf{d}_1 \\ \mathbf{d}_1^T\mathbf{d}_2 & \mathbf{d}_2^T\mathbf{d}_2 & \dots & \mathbf{d}_n^T\mathbf{d}_2 \\ \mathbf{d}_1^T\mathbf{d}_3 & \mathbf{d}_2^T\mathbf{d}_3 & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \mathbf{d}_1^T\mathbf{d}_n & & & \mathbf{d}_n^T\mathbf{d}_n \end{bmatrix} = \begin{bmatrix} d_0 & d_1 & & d_{n-1} \\ d_{n-1} & d_0 & d_1 & d_{n-2} \\ d_{n-2} & d_{n-1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ d_1 & & & d_0 \end{bmatrix}$$

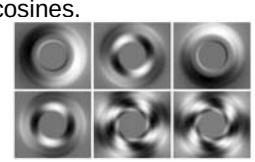


Eigenvectors of circulant matrices

- The eigenvectors of circulant matrices are the n basis of the Fourier matrix.

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{-\frac{2\pi i}{n}} & \dots & e^{-\frac{(n-1)*2\pi i}{n}} \\ 1 & e^{-\frac{2*2\pi}{n}} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 1 & e^{-\frac{(n-2)*2\pi i}{n}} & & e^{-\frac{(n-1)^2 2\pi i}{n}} \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} \sum_{k=0}^{n-1} d_o e^{-\frac{k*2\pi i}{n}} \\ \\ \sum_{k=0}^{n-1} d_{n-1} e^{-\frac{k*2\pi i}{n}} \end{bmatrix} \quad i = \sqrt{-1} \quad \boldsymbol{\Gamma}\mathbf{F} = \mathbf{F}\mathbf{V}$$

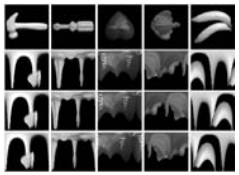
- Circulant and SYMMETRIC $d_i = d_{n-i}$. **F** is the basis of cosines.

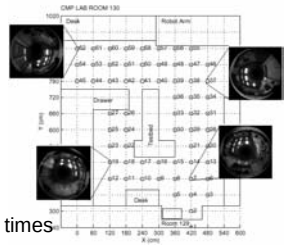


$eig(\mathbf{D}\mathbf{D}^T) \Longrightarrow$ DCT

Generalization to multiple templates

(Jogan et al., 2003)





- P different locations (objects), each shifted N times
- every Q_{ij} is circulant (but in general not symmetric!)

$$\mathbf{A} = \mathbf{D}^T\mathbf{D} = \begin{bmatrix} Q_{00} & \cdots & Q_{0(P-1)} \\ Q_{10} & \cdots & Q_{1(P-1)} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ Q_{P0} & & Q_{(P-1)(P-1)} \end{bmatrix}$$

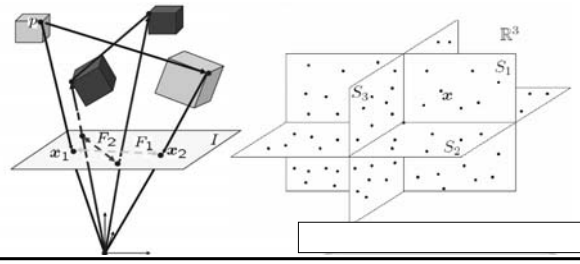
Same eigenvectors

•Eigencalculation of NO(P³) rather than O((NP)³)

7- Mixture of subspaces

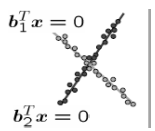
(Vidal et al., 2003)

- How to estimate (basis) of a mixture of subspaces and varying dimensions?
  - Multibody structure from motion
    - 2D & 3D Motion segmentation.



Generalized Principal Component Analysis

- Given points on multiple subspaces, identify:
  - Number of subspaces and their dimensions
  - Basis of each subspace
  - Segmentation of data points.
- Classical chicken-egg problem.
- GPCA finds an algebraic geometric solution to the mixture of subspaces.
  - Number of subspaces= degrees of polynomial
  - Subspaces basis= derivatives of polynomial.



Example: clustering in 1D

$x = b_1 \quad x = b_2$

$x = b_1$ or $x = b_2$

$(x - b_1)(x - b_2) = 0$

$$x^2 - (b_1 + b_2)x + b_1 b_2 = 0$$

$$\underbrace{\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 1 \\ -(b_1 + b_2) \\ b_1 b_2 \end{bmatrix}}_{\boldsymbol{c}} = 0$$

- Number of groups?

$$\text{rank}(P) = 1: \text{ one group only}$$
$$\text{rank}(P) = 2: \text{ two groups}$$

Example 2

$x = b_1$ or $x = b_2 \cdots x = b_n$

$$p_n(x) = (x-b_1)\cdots(x-b_n) = 0$$

$$p_n(x) = x^n + c_1 x^{n-1} + \cdots c_n = 0$$

$$p_n(x) = \begin{bmatrix} x^n & \cdots & x & 1 \end{bmatrix} \boldsymbol{c} = 0$$

$$P_n \boldsymbol{c} = \underbrace{\begin{bmatrix} x_1^n & \cdots & x_1 & 1 \\ x_2^n & \cdots & x_2 & 1 \\ \vdots & & \vdots & \vdots \\ x_N^n & \cdots & x_N & 1 \end{bmatrix}}_{P_n \in \mathbb{R}^{N \times (n+1)}} \boldsymbol{c} = 0$$

How to compute n, c, b's?

- Number of clusters

$$n \doteq \min\{i : rank(P_i) = i\}$$

- Cluster centers

  Roots of $p_n(x)$

- Solution is unique if

  $N_{points} \geq n_{groups}$

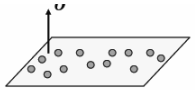
- Solution is closed form if

  $n_{groups} \leq 4$

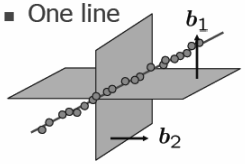
GPCA

- One plane

  $\boldsymbol{b}^T \boldsymbol{x} = b_1 x_1 + b_2 x_2 + b_3 x_3 = 0$



- One line



$$\boldsymbol{b}_1^T \boldsymbol{x} = b_1 x_1 + b_2 x_2 + b_3 x_3 = 0$$
$$\boldsymbol{b}_2^T \boldsymbol{x} = b_4 x_1 + b_5 x_2 + b_6 x_3 = 0$$

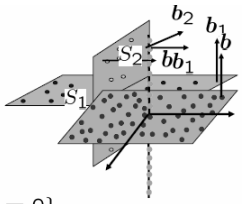
- One subspace can be represented with
  - Set of linear equations $S = \{\boldsymbol{x} : B^T \boldsymbol{x} = 0\}$
  - Set of polynomials of degree 1

GPCA

- Two planes

  $(\boldsymbol{b}_1^T \boldsymbol{x} = 0)$ or $(\boldsymbol{b}_2^T \boldsymbol{x} = 0)$

  $$p_2(\boldsymbol{x}) = (\boldsymbol{b}_1^T \boldsymbol{x})(\boldsymbol{b}_2^T \boldsymbol{x}) = 0$$



- One plane and one line
  - Plane: $S_1 = \{\boldsymbol{x} : \boldsymbol{b}^T \boldsymbol{x} = 0\}$
  - Line: $S_2 = \{\boldsymbol{x} : \boldsymbol{b}_1^T \boldsymbol{x} = \boldsymbol{b}_2^T \boldsymbol{x} = 0\}$

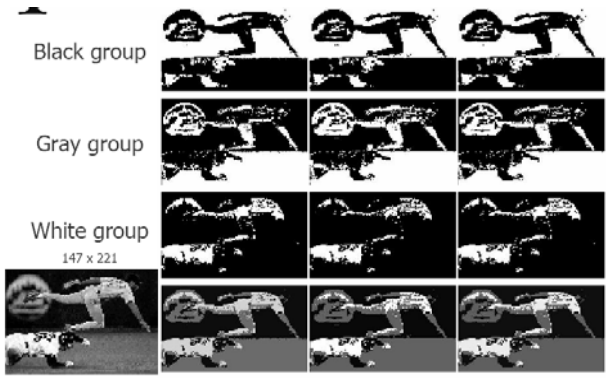
  $S_1 \cup S_2 = \{\boldsymbol{x} : (\boldsymbol{b}^T \boldsymbol{x} = 0)$ or $(\boldsymbol{b}_1^T \boldsymbol{x} = \boldsymbol{b}_2^T \boldsymbol{x} = 0\}$

  De Morgan's rule

  $S_1 \cup S_2 = \{\boldsymbol{x} : (\boldsymbol{b}^T \boldsymbol{x})(\boldsymbol{b}_1^T \boldsymbol{x}) = 0$ and $(\boldsymbol{b}^T \boldsymbol{x})(\boldsymbol{b}_2^T \boldsymbol{x}) = 0\}$

- A union of *n* subspaces can be represented with a set of homogeneous polynomials of degree *n*

Examples

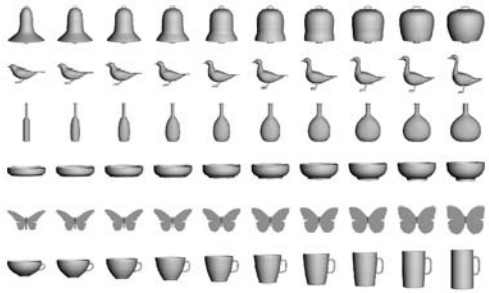


	(a) Kmeans	(b) EM	(c) Polysegment
Time:	2 (sec)	10 (sec)	0.4 (sec)

Multiple eigenspaces

(Leonardis et al., 2002)

- Goal: group visually similar images into categories in an unsupervised and self-organising way.



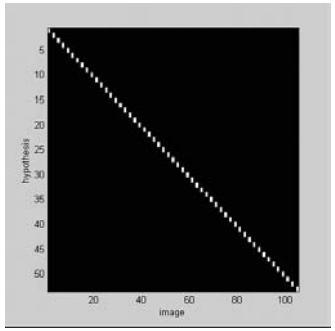
Multiple eigenspaces – idea

- Group the images and construct multiple eigenspaces such that:
  - Mean reconstruction error is always below a threshold
  - The dimensionalities of eigenspaces are as small as possible
- Start from a seed and then grow hypotheses
- Grow and select paradigm:
  - Simultaneously and independently grow multiple competing hypotheses
  - Select the best hypotheses at the end

Multiple eigenspace - example

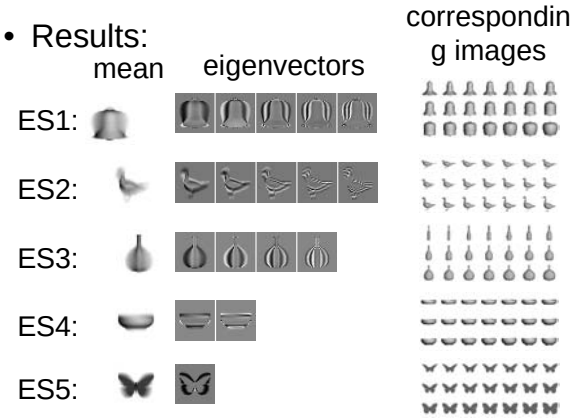
- 5 categories, 21 images from each

Growing :

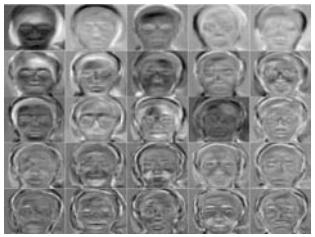


Multiple eigenspace - example

- Results:

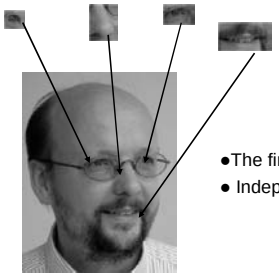


“Intercorrelations among variables are the bane of the multivariate researcher’s struggle for meaning”



Cooley and Lohnes, 1971

Part-based representation



- The firing rates of neurons are never negative
- Independent representations.

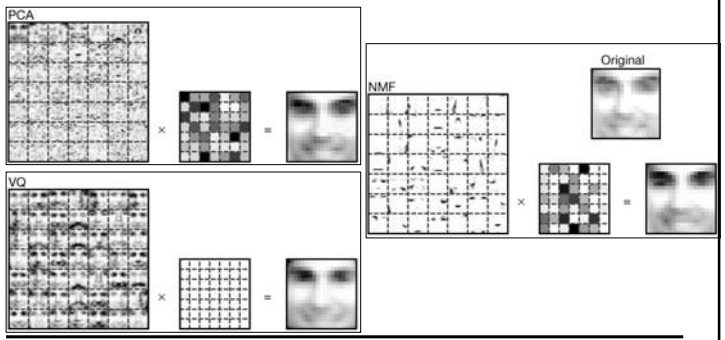
NMF & ICA

Non-negative Matrix Factorization

- Positive factorization.

$$E(\mathbf{B}, \mathbf{C}) = \| \mathbf{D} - \mathbf{BC} \|_F \quad \mathbf{B}, \mathbf{C} \geq 0$$

- Leads to part-based representation.



Nonnegative factorization algorithm

(Lee & Seung, 1999;Lee & Seung, 2000)

$$\min_{\mathbf{W}\geq 0, \mathbf{V}\geq 0} F = \sum_{ij} \left| d_{ij} - (\mathbf{BC})_{ij} \right|^2$$

Derivatives:

$$\frac{\partial F}{\partial \mathbf{C}_{ij}} = (\mathbf{B}^{\mathbf{T}}\mathbf{BC})_{ij} - (\mathbf{B}^{\mathbf{T}}\mathbf{C})_{ij}$$

$$\frac{\partial F}{\partial \mathbf{B}_{ij}} = (\mathbf{BCC}^T)_{ij} - (\mathbf{DC}^T)_{ij}$$

Inference:

$$\mathbf{C}_{ij} \leftarrow \mathbf{C}_{ij} \frac{(\mathbf{B}^{\mathbf{T}}\mathbf{D})_{ij}}{(\mathbf{B}^{\mathbf{T}}\mathbf{BV})_{ij}}$$

Learning:

$$\mathbf{B}_{ij} \leftarrow \mathbf{B}_{ij} \frac{(\mathbf{DC}^T)_{ij}}{(\mathbf{BCC}^T)_{ij}}$$

- Multiplicative algorithm can be interpreted as diagonally rescaled gradient descent.

8- PCA & NMF for clustering

(Zha et al., 2001;Ding & He, 2004; de la Torre & Kanade, 2006)

- VQ, k-means $E(\mathbf{G},\mathbf{B}) = \| \mathbf{D} - \mathbf{B}\mathbf{G}^T \|_F = \sum_{i=1}^{c} \sum_{j \in C_i} \| \mathbf{d}_j - \mathbf{b}_i \|$

$$\mathbf{G}^T = \begin{bmatrix} 1 & \dots & 0 \\ 0 & \dots & 1 \\ 0 & \dots & 0 \end{bmatrix} \quad g_{ij} \in \{0,1\} \quad \mathbf{G}\mathbf{1}_k = \mathbf{1}_n \quad \mathbf{G} \in \Re^{n \times k}$$

- After eliminating (**B**)

$$E(\mathbf{G}) = \| \mathbf{D} - \mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T \|_F = tr(\mathbf{D}^T\mathbf{D}) - tr((\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G}) \geq \sum_{i=c+1}^{\min(d,n)} \lambda_i$$

- Relaxing binary constraints

$$E(\mathbf{G}) \propto tr((\mathbf{G}^T\mathbf{G})^{-1}(\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G}))$$

- Eigenvectors (PCA) of $\mathbf{D}^T\mathbf{D}$ are the optimal continuous solution of the indicator variable **G**.

Clustering with NMF

(Zass & Shashua, 2005; Ding et al., 2005)

$$E(\mathbf{G},\mathbf{B}) = \| \mathbf{\Gamma} - \mathbf{B}\mathbf{G}^T \|_F \quad \mathbf{\Gamma} = [\varphi(\mathbf{d}_1)\ \varphi(\mathbf{d}_2)\ \dots\ \varphi(\mathbf{d}_n)]$$

- Soft clustering and non-negative matrix factorization:

$$\| \mathbf{\Gamma}^T\mathbf{\Gamma} - \mathbf{G}\mathbf{G}^T \|_F \quad \mathbf{G} \geq 0 \quad \mathbf{F} = \mathbf{G}^T\mathbf{G} = \mathbf{I} \quad \mathbf{F}1 = 1 \quad \mathbf{F}^T 1 = 1$$

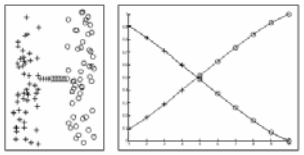
Affinity matrix

$$\mathbf{G} = \begin{bmatrix} \frac{1}{\sqrt{n_1}}\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \frac{1}{\sqrt{n_2}}\begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} & \cdots & \frac{1}{\sqrt{n_c}}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \end{bmatrix}$$

- Previous normalization $\min_{\mathbf{K}} \| \mathbf{\Gamma}^T\mathbf{\Gamma} - \mathbf{K} \|_F \quad s.t. \ \mathbf{K}1 = \mathbf{K}^T 1 = \beta 1 \ \mathbf{K} = \mathbf{K}^T$
- Gradient descent with multiplicative updates.

Examples

- Soft clustering

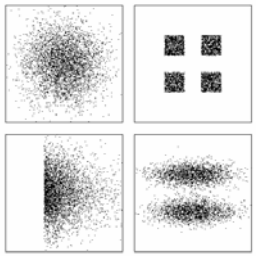


R_{ij}=1 if $\mathbf{d}_i$ and $\mathbf{d}_j$ are cluster together.
R_{ij}=-1 if $\mathbf{d}_i$ and $\mathbf{d}_j$ are not cluster together.

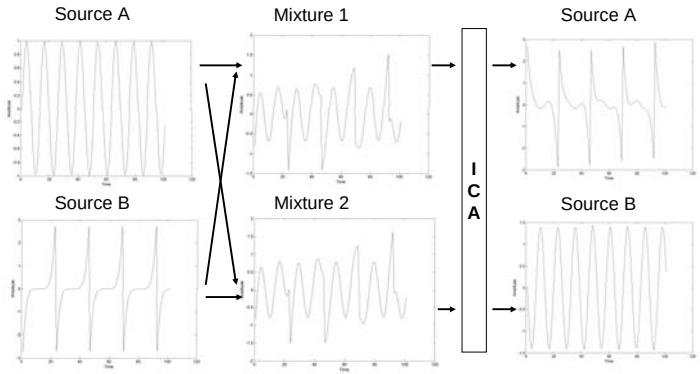
- Adding side-information $\mathbf{K}' = \mathbf{K} + \alpha\mathbf{R}$ (Zass & Shashua, 2005)

Independent Component Analysis

- We need more than second order statistics to represent the signal.



ICA & Signals



ICA

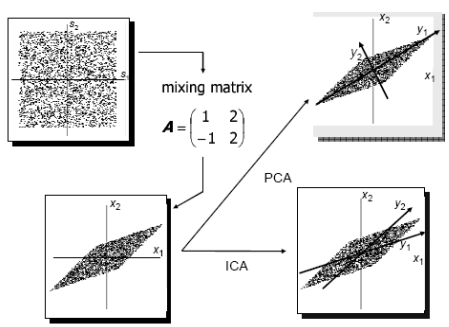
(Hyvrinen et al., 2001)

$$\mathbf{D} = \mathbf{BC} \quad \mathbf{C} \approx \mathbf{S} = \mathbf{WD} \quad \mathbf{W} \approx \mathbf{B}^{-1}$$

- Look for s_i that are independent.
- PCA finds uncorrelated variables, the independent components have non Gaussian distributions.
- Uncorrelated $E(s_i s_j) = E(s_i)E(s_j)$
- Independent $E(g(s_i)f(s_j)) = E(g(s_i))E(f(s_j))$ for any non-linear f,g

PCA ICA

ICA vs PCA



Many optimization criteria

- Minimize high order moments: e.g. kurtosis

$$\text{kurt}(\boldsymbol{W}) = E\{s^4\} - 3(E\{s^2\})^2$$

- Many other information criteria.
- Also an error function: (Olhausen & Field, 1996)

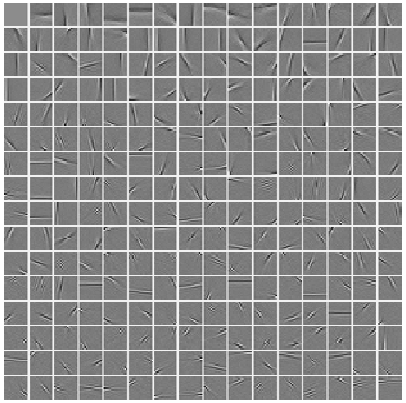
$$\sum_{i=1}^{n} \|\mathbf{d}_i - \mathbf{B}\mathbf{c}_i\| + \sum_{i=1}^{n} S(\mathbf{c}_i)$$

Sparseness (e.g. S=| |)

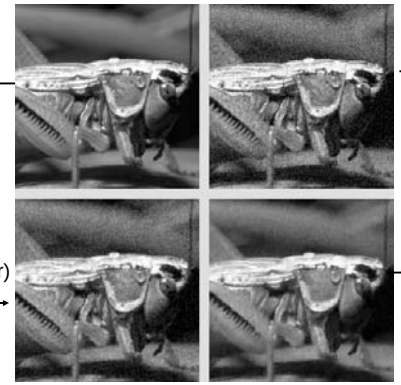
- Other sparse PCA.

(Chennubhotla & Jepson, 2001b; Zou et al., 2005; dAspremont et al., 2004;)

Basis of natural images



Denoising

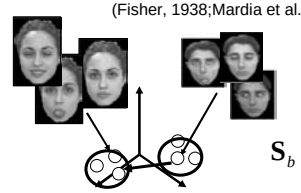


Discriminative Models

- Linear Discriminant Analysis (LDA)
  - 9) Multimodal Oriented Discriminant Analysis.
  - 10) Discriminative Cluster Analysis.
  - 11) Robust Linear Discriminant Analysis.
- Oriented Component Analysis (OCA)
  - 12) Representational Oriented Component Analysis.
- Canonical Correlation Analysis (CCA)
  - 13) Dynamical Coupled Component Analysis.
  - 14) CCA and Mobile robotics applications.
- Relevance Component Analysis (RCA)

Linear Discriminant Analysis (LDA)

(Fisher, 1938;Mardia et al., 1979; Bishop, 1995)



$$\mathbf{S}_b = \sum_{i=1}^{C}\sum_{j=1}^{C}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T$$

$$\mathbf{S}_t = \mathbf{D}\mathbf{D}^T = \sum_{i=1}^{n}\mathbf{d}_i\mathbf{d}_i^T$$

$$J(\mathbf{B}) = \frac{|\mathbf{B}^T\mathbf{S}_b\mathbf{B}|}{|\mathbf{B}^T\mathbf{S}_t\mathbf{B}|}$$

$$\mathbf{S}_b\mathbf{B} = \mathbf{S}_t\mathbf{B}\boldsymbol{\Lambda}$$

$$\mathbf{S}_w = \sum_{j=1}^{c}\sum_{i=1}^{C_j}(\mathbf{d}_i - \boldsymbol{\mu}_j)(\mathbf{d}_i - \boldsymbol{\mu}_j)^T$$

- Optimal linear dimensionality reduction if classes are Gaussian with equal covariance matrix.

Matrix formulation

(de la Torre & Kanade, 2005b)

$$\mathbf{S}_t = \frac{1}{n-1}\sum_{j=1}^{n}(\mathbf{d}_j - \mathbf{m})(\mathbf{d}_j - \mathbf{m})^T = \frac{1}{n-1}\mathbf{D}\mathbf{P}_1\mathbf{D}^T$$

$$\mathbf{S}_w = \frac{1}{n-1}\sum_{i=1}^{c}\sum_{\mathbf{d}_j \in C_i}(\mathbf{d}_j - \mathbf{m}_i)(\mathbf{d}_j - \mathbf{m}_i)^T = \frac{1}{n-1}\mathbf{D}\mathbf{P}_2\mathbf{D}^T$$

$$\mathbf{S}_b = \sum_{i=1}^{c}\frac{n_i}{n-1}(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = \frac{1}{n-1}\mathbf{D}\mathbf{P}_3\mathbf{D}^T$$

$$\mathbf{P}_1 = \mathbf{I} - \frac{1}{n}1_n 1_n^T \qquad \mathbf{P}_2 = \mathbf{I} - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$$

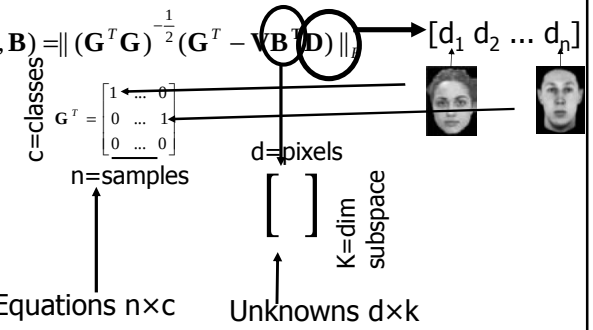
$$\mathbf{P}_3 = \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T - \frac{1}{n}1_n 1_n^T$$

- **m** overall mean, $\mathbf{m}_i$ mean for class i.
- $\mathbf{P}_i$ is a projection matrix (i.e. $\mathbf{P}_i = \mathbf{P}_i^2 \qquad \mathbf{P}_i = \mathbf{P}_i^T$)

Error function for LDA

(de la Torre & Kanade, 2006)

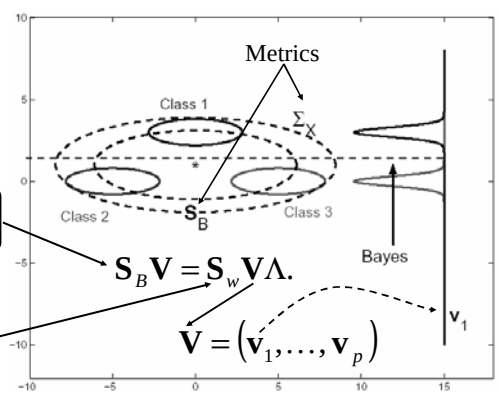
$$E(\mathbf{V},\mathbf{B}) = \| (\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}(\mathbf{G}^T - \mathbf{V}\mathbf{B}^T\mathbf{D}) \|_F$$



- d>>n an UNDETERMINED system of equations! (over-fitting)

Where are linear methods applicable?

(Martinez & Zhu., 2005)



Between-class scatter matrix

Covariance matrix

$$\mathbf{S}_B\mathbf{V} = \mathbf{S}_w\mathbf{V}\Lambda.$$

$$\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_p)$$

When is LDA good?

(Martinez & Zhu., 2005)

- The *discriminant power* of the generalized eigenvalue decomposition equation

$$\mathbf{M}_A\mathbf{V} = \mathbf{M}_B\mathbf{V}\Lambda$$

is $tr\left(\mathbf{M}_B^{-1}\mathbf{M}_A\right)$, which is the same as

relative variance

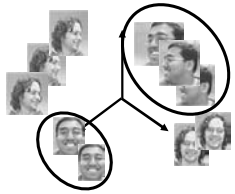
$$K = \sum_{i=1}^{q}\sum_{j=1}^{p}\frac{\lambda_{A_i}}{\lambda_{B_j}}\left(\mathbf{b}_j^T\mathbf{a}_i\right)^2.$$

- Large values of K indicates instability in the results.
- If for some (i,j) max $(\mathbf{b}_j\mathbf{a}_i)^2$ close to 1 the solution might be unstable.

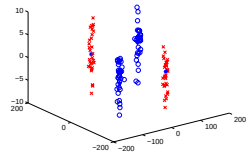
9- Multimodal Oriented Component Analysis (MODA)

(de la Torre & Kanade, 2005a)

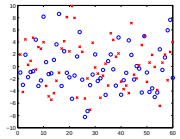
- How to extend LDA
  - Model class covariances.
  - Multimodal classes.
  - Deal efficiently with huge covariance matrices (e.g. 100*100).



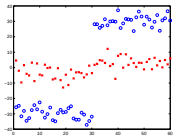
Multimodality



LDA



MODA



Multimodal Oriented Discriminant Analysis

- **B** that MAXIMIZES the Kullback-Leibler divergence between clusters among classes.

$$\sum_{i=1}^{Classes}\sum_{j\neq i}^{classes}\sum_{r_1\in C_i}\sum_{r_2\in C_j} tr(\mathbf{B}^T(\boldsymbol{\Sigma}_j^{r_2}\boldsymbol{\Sigma}_i^{r_1}{}^{-1}+\boldsymbol{\Sigma}_j^{r_2}{}^{-1}\boldsymbol{\Sigma}_i^{r_1}+(\boldsymbol{\mu}_i^{r_1}-\boldsymbol{\mu}_j^{r_2})(\boldsymbol{\Sigma}_j^{r_1}{}^{-1}+\boldsymbol{\Sigma}_i^{r_2}{}^{-1})(\boldsymbol{\mu}_i^{r_1}-\boldsymbol{\mu}_j^{r_2})^T)\mathbf{B})$$

- 1 mode per class and equal covariances equivalent to LDA.

Optimization

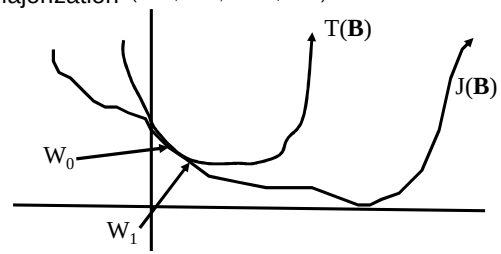
- Hard optimization problem

$$J(\mathbf{B}) = -\sum_{i=1} tr((\mathbf{B}^T\boldsymbol{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{A}_i\mathbf{B}))$$

- Iterative Majorization (Kiers, 1995; Leeuw, 1994)

$T(\mathbf{B}) \geq J(\mathbf{B}) \;\; \forall \mathbf{B}$

$T(\mathbf{B}_0) = J(\mathbf{B}_0)$



Majorization

$$T(\mathbf{B}) = \sum_{i=1}^{classes} \|(\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{-\frac{1}{2}}\mathbf{B}^T\mathbf{A}_i^{-\frac{1}{2}} - (\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{\frac{1}{2}}(\mathbf{B}_0^T\mathbf{\Sigma}_i\mathbf{B}_0)\mathbf{B}^T\mathbf{A}_i^{\frac{1}{2}}\|$$

$$\geq -\sum_{i=1}^{classes} tr((\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{A}_i\mathbf{B}))$$

- Slow convergence, first gradient descent:

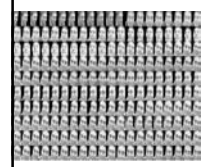
$$\mathbf{B}^{(n+1)} = \mathbf{B}^{(n)} - \eta\frac{\partial E_2(\mathbf{B}^{(n)})}{\partial \mathbf{B}}$$

$$\frac{\partial E_2}{\partial \mathbf{B}} = \sum_i \mathbf{A}_i\mathbf{B}^{(n)}(\mathbf{B}^{(n)T}\mathbf{\Sigma}_i\mathbf{B}^{(n)})^{-1} - \mathbf{\Sigma}_i\mathbf{B}^{(n)}(\mathbf{B}^{(n)T}\mathbf{\Sigma}_i\mathbf{B}^{(n)})^{-1}(\mathbf{B}^{(n)T}\mathbf{A}_i\mathbf{B}^{(n)})^{-1}(\mathbf{B}^{(n)T}\mathbf{\Sigma}_i\mathbf{B}^{(n)})^{-1}$$

Face recognition from video

(de la Torre et al., 2005b)

- Challenges
  - Low quality small images (40-50 pixels).
  - Changes in expression/pose/occlusion/illumination.
  - Real time and scalable to several users.

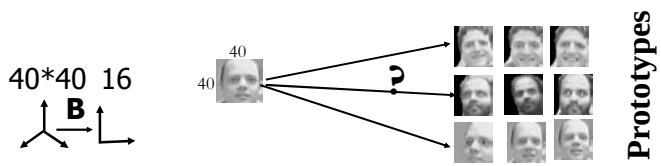


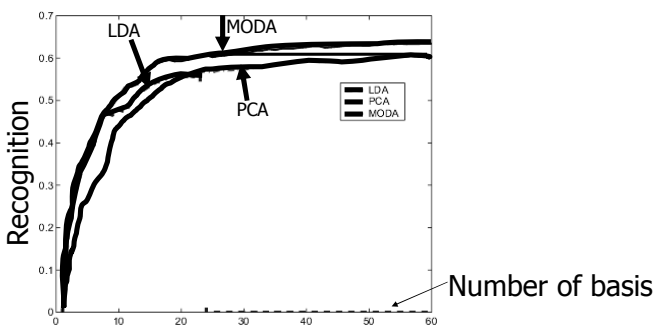
Photometric normalization

Geometric normalization

Clustering.

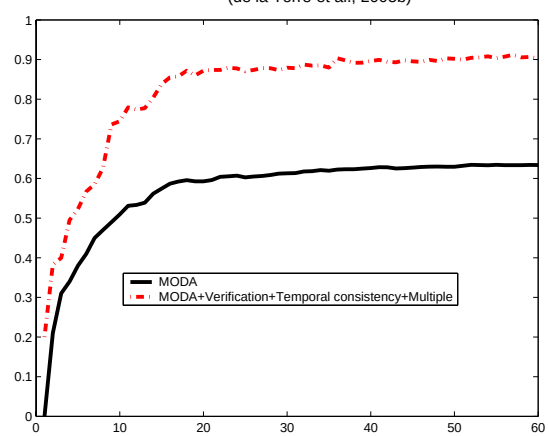
Prototypes.





Adding space-time constraints

(de la Torre et al., 2005b)



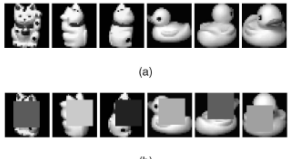
10- Robust Linear Discriminant Analysis

(Fidler et al., 2006)

- How to construct robust discriminative models?

No noise model.

$$\mathbf{G}=\mathbf{B}^{g^T}\mathbf{D}$$



(a)

- Remove outliers in the space of generative models and learn linear a mapping to discriminative models.
- Assume (d>>n) and all the samples independent.

$\mathbf{D}=\mathbf{B}^g\mathbf{C}$ *generative* $\quad \mathbf{D}\in\Re^{d\times n}\quad \mathbf{B}^g\in\Re^{d\times n}\quad \mathbf{C}\in\Re^{n\times n}$ samples

$\mathbf{G}=\mathbf{B}^{d^T}\mathbf{D}$ *discriminative* $\quad \mathbf{B}^d\in\Re^{d\times n}\quad \mathbf{G}\in\Re^{n\times c}$ classes

pixels

Robust LDA

- $\mathbf{B}^d$ as linear combination of the basis of $\mathbf{B}^g$.

$$\mathbf{B}^d=\mathbf{B}^g\mathbf{V}=\left[\mathbf{B}^g_{1k}\ \mathbf{B}^g_{n-k}\right]\begin{bmatrix}\mathbf{V}_{1:k}\\ \mathbf{V}_{n-k}\end{bmatrix}\qquad \mathbf{V}\in\Re^{n\times c}$$

PCA coefficients

$$\mathbf{G}=\mathbf{B}^{d^T}\mathbf{D}=\mathbf{V}^T\mathbf{B}^{g^T}\mathbf{D}=\mathbf{V}^T\mathbf{C}=\mathbf{V}^T_{1:k}\mathbf{C}_{1:k}+\mathbf{V}^T_{n-k:n}\mathbf{C}_{n-k:n}$$

- If just $\mathbf{V}^T_{1:k}\mathbf{C}_{1:k}$ discriminative power can be lost.
- Add c basis to fully recover the LDA solution.

$$\mathbf{W}=\mathbf{B}^g_{n-k:n}\mathbf{V}_{n-k:n}\in\Re^{d\times c}\qquad \tilde{\mathbf{W}}=\mathbf{W}(\mathbf{W}^T\mathbf{W})^{-\frac{1}{2}}$$

$$\hat{\mathbf{B}}^g=[\mathbf{B}^g_{1:k}\ \ \tilde{\mathbf{W}}]\qquad \hat{\mathbf{V}}=\begin{bmatrix}\mathbf{V}_{1:k}\\ (\mathbf{W}^T\mathbf{W})^{\frac{1}{2}}\end{bmatrix}\quad\Rightarrow\quad \mathbf{G}=\hat{\mathbf{V}}^T\hat{\mathbf{B}}^{g^T}\mathbf{D}$$

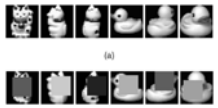
No lost of information

Robust PCA

- Given training data find $\mathbf{B}^d,\ ,\mathbf{V},\ ,\hat{\mathbf{B}}^g=[\mathbf{B}^g_{1:k}\ \ \tilde{\mathbf{W}}]$
- Find robustly the coefficients in the $\hat{\mathbf{B}}^g$ basis such that: $\mathbf{d}_i\approx\hat{\mathbf{B}}^g\mathbf{c}_i$ and then estimate **G.**
- Robust estimation is achieved by hypothesize-and-test paradigm.

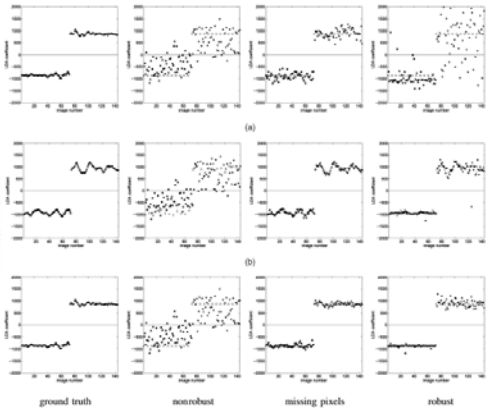
$$\begin{bmatrix}d_{1i}\\ \\ \\ d_{di}\end{bmatrix}\approx\begin{bmatrix}b_{11}c_1+\ldots+b_{ik}c_k\\ \\ \\ b_{d1}c_d+\ldots+b_{dk}c_k\end{bmatrix}$$

Experimental results



(a)

	Fisher criterion				MARE
	ground truth	non-robust	missing pixels	robust	ground truth
LDA	4.84	0.04	0.45	0.03	40.17
LDA+nK	1.47	0.05	1.08	0.39	17.29
LDA+PCA	4.84	0.04	2.65	1.06	17.65



More examples

	Fisher criterion		recognition rate (%)	
	non-robust	robust	non-robust	robust
sun glasses	0.28	0.99	43.00	84.00
scarf	0.46	1.20	59.00	93.00
scream	0.85	0.86	87.00	87.00

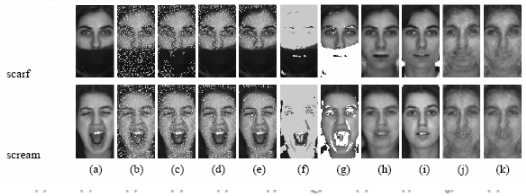


Fig. 8. Results of the pixel selection procedure. (a) test image, (b) random initialization in the pixel selection procedure (light-gray pixels denote the selected pixels), (c)-(e) a few stages of the α-trimming method, (f) all compatible points, (g) the finally selected pixels used for coefficient calculation (denoted with the gray-value of the original test image), (h) reconstructed image, (i) training image of the person in (a), (j) reconstructed image obtained with the standard LDA method, (k) reconstructed image obtained using naive ($k = 0$) robust LDA method (see text for details).

11-Discriminative Cluster Analysis (DCA)

(de la Torre & Kanade, 2006)

- Generative clustering (e.g. k-means):

$$E(\mathbf{G},\mathbf{B}) = \| \mathbf{D} - \mathbf{B}\mathbf{G}^T \|_F = \sum_{i=1}^{c} \sum_{j \in C_i} \| \mathbf{d}_j - \mathbf{b}_i \|$$

$$g_{ij} \in \{0,1\} \qquad \mathbf{G}\mathbf{1}_k = \mathbf{1}_n$$

  - Not efficient for high dimensional data.
  - Multiple local minima.

- Discriminative clustering (de la Torre & Kanade, 2006):

$$E(\mathbf{V},\mathbf{B},\mathbf{G}) = \| (\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}} (\mathbf{G}^T - \mathbf{V}\mathbf{B}^T\mathbf{D}) \|_F$$

  - Simultaneous dimensionality reduction and clustering.

Optimization

- Eliminate **V**

$$E(\mathbf{B},\mathbf{G}) \propto tr((\mathbf{B}^T \underbrace{\mathbf{D}\mathbf{D}^T}_{\mathbf{S}_t} \mathbf{B})^{-1} (\mathbf{B}^T \underbrace{\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T}_{\mathbf{S}_b} \mathbf{B}))$$

- Optimize for **B**

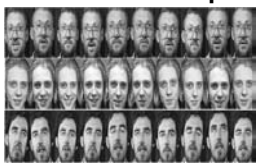
$$\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{B} = \mathbf{D}\mathbf{D}^T\mathbf{B}\Lambda$$

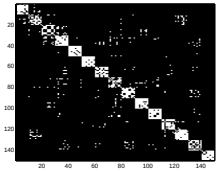
- Optimize for **G** $\quad \mathbf{A} = \mathbf{C}^T(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C} \qquad \mathbf{C} = \mathbf{B}^T\mathbf{D}$

$$\mathbf{G} = \mathbf{V} \circ \mathbf{V} \qquad \mathbf{V}^{(n+1)} = \mathbf{V}^{(n)} - \eta \frac{\partial E}{\partial \mathbf{V}}$$

$$\frac{\partial E}{\partial \mathbf{V}} = (\mathbf{I}_C - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T)\mathbf{A}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}$$

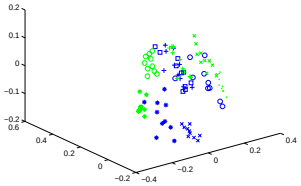
Experiments

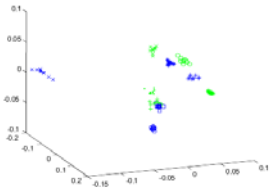




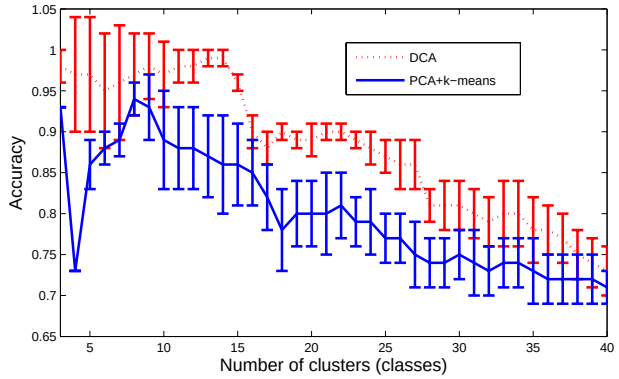
PCA

DCA

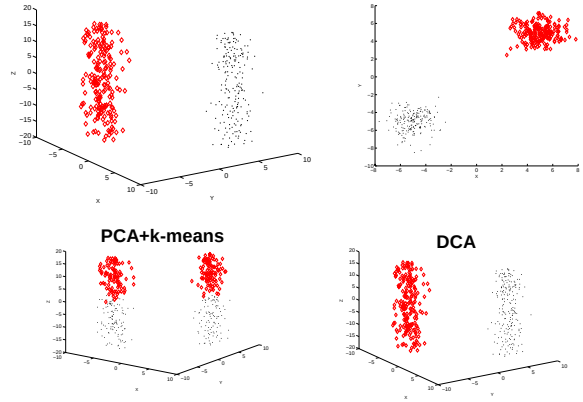




DCA vs. PCA+k-means



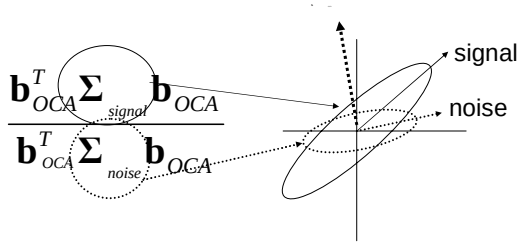
Toy problem



Related LDA work

- Face recognition (Belhumeur et al., 1997;Zhao, 2000;Martinez & Kak, 2003)
- Small sample problem (Chen et al., 2000; Yu & Yang, 2001)
- Mixture (Hastie et al., 1995; Zhu & Martinez, 2006;)
- Neural approaches (Gallinari et al., 1991; Lowe & Webb, 1991)
- Heteroscedastic Discriminant Analysis (Kumar & Andreou, 1998; Fukunaga, 1990; Mardia et al., 1979; Saon et al., 2000;)

Oriented Component Analysis (OCA)

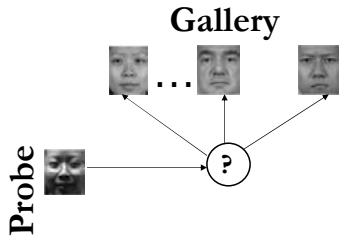


$$\frac{\mathbf{b}_{OCA}^T \boldsymbol{\Sigma}_{signal} \mathbf{b}_{OCA}}{\mathbf{b}_{OCA}^T \boldsymbol{\Sigma}_{noise} \mathbf{b}_{OCA}}$$

- Generalized eigenvalue problem: $\boldsymbol{\Sigma}_i \mathbf{b}_k = \boldsymbol{\Sigma}_e \mathbf{b}_k \lambda$
- $\mathbf{b}_{oca}$ is steered by the distribution of noise.

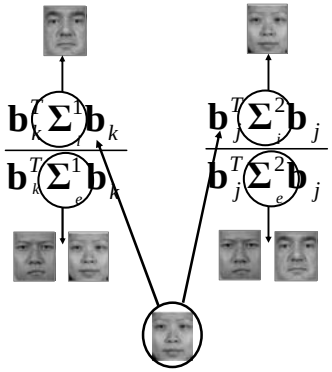
12- Representational Oriented Component Analysis (ROCA)

(de la Torre et al., 2005a)



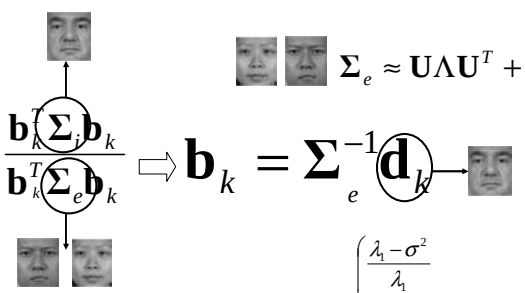
- Challenges
  - Just 1 training image.
  - Changes in appearance, expression and illumination.

OCA



Interpretation of OCA

(de la Torre et al., 2005a)



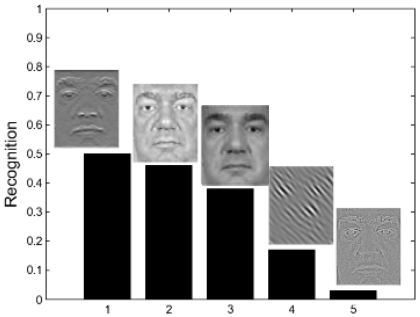
$$\boldsymbol{\Sigma}_e \approx \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T + \sigma^2\mathbf{I}$$

$$\frac{\mathbf{b}_k^T \boldsymbol{\Sigma}_i \mathbf{b}_k}{\mathbf{b}_k^T \boldsymbol{\Sigma}_e \mathbf{b}_k} \Rightarrow \mathbf{b}_k = \boldsymbol{\Sigma}_e^{-1}\mathbf{d}_k$$

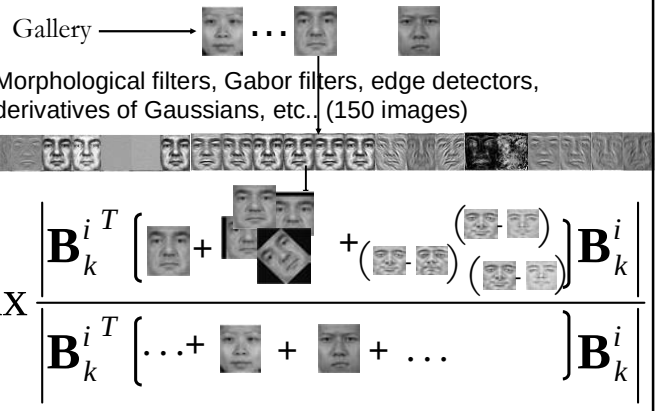
$$\mathbf{b}_k = \boldsymbol{\Sigma}_e^{-1}\mathbf{d}_k \propto (\mathbf{I} - \mathbf{U}\begin{pmatrix} \frac{\lambda_1 - \sigma^2}{\lambda_1} & & \\ & \frac{\lambda_2 - \sigma^2}{\lambda_2} & \\ & & \ldots \end{pmatrix}\mathbf{U}^T)\mathbf{d}_k$$

Does representation matter?

- Unstable classifier with large probability of misclassification (prone to over-fitting).



Combining several representations



- Morphological filters, Gabor filters, edge detectors, derivatives of Gaussians, etc.. (150 images)

$$\max_{\mathbf{B}_k^i} \frac{\left| \mathbf{B}_k^{i\,T} \left[\ldots \right] \mathbf{B}_k^i \right|}{\left| \mathbf{B}_k^{i\,T} \left[\ldots + \ldots + \ldots \right] \mathbf{B}_k^i \right|}$$

Solving Generalized Eigenvalue

- Rank deficient matrices.
- Not numerically stable algorithms/bad generalization (over fitting)

$$\left\| \mathbf{\Sigma}_i - \mathbf{U}_i \mathbf{U}_i^T - \sigma^2 \mathbf{I} \right\|_F \qquad \mathbf{\Sigma}_i \approx \mathbf{U}_i \mathbf{U}_i^T + \sigma^2 \mathbf{I}$$

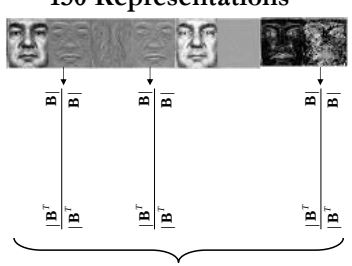
$$\max_{\mathbf{B}_k^i} \frac{\left| \mathbf{B}_k^{i\,T} \mathbf{A} \mathbf{B}_k^i \right|}{\left| \mathbf{B}_k^{i\,T} \mathbf{C} \mathbf{B}_k^i \right|} \rightarrow 10.000 \times 10.000$$

- Modified Subspace Iteration.

$$\mathbf{C}\hat{\mathbf{V}}_{k+1} = \mathbf{A}\mathbf{V}_k \quad \hat{\mathbf{V}}_{k+1} = \hat{\mathbf{V}}_{k+1} / max(\hat{\mathbf{V}}_{k+1})$$
$$\mathbf{S} = \hat{\mathbf{V}}_{k+1}^T \mathbf{A} \hat{\mathbf{V}}_{k+1} \quad \mathbf{T} = \hat{\mathbf{V}}_{k+1}^T \mathbf{C} \hat{\mathbf{V}}_{k+1}$$
$$\mathbf{S}\mathbf{W} = \mathbf{C}\mathbf{W}\mathbf{\Delta}$$
$$\mathbf{V}_{k+1} = \hat{\mathbf{V}}_{k+1}\mathbf{W}$$

Combining Classifiers

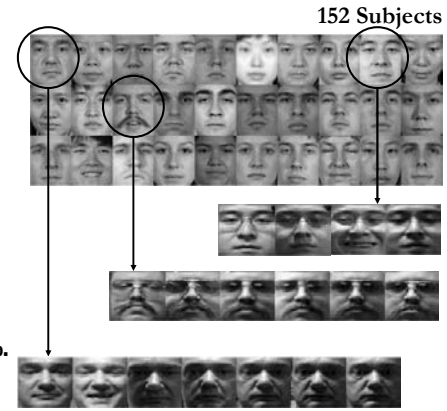
150 Representations



Combining classifiers
* Normalization
* Sum.

Results

- PCA –
  - Mah **10%**
  - Euclidean **16%**
- NN-- **29%**
- Face it-- **40.9%**
- ROCA
  - Max of individual classifiers **47%.**
  - Combining **62.1%.**

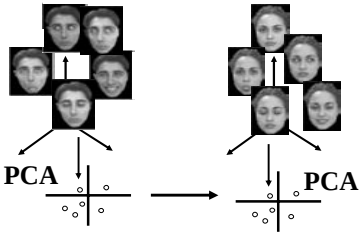


13- Dynamic Coupled Component Analysis (DCCA) (de la Torre & Black, 2001a)

- Learning the coupling.
- High dimensional data.
- Limited training data.

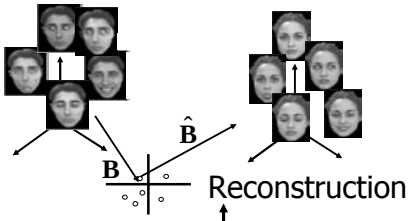
Solutions?

- PCA independently and general mapping



- Signals dependent signals with small energy can be lost.

DCCA



Reconstruction

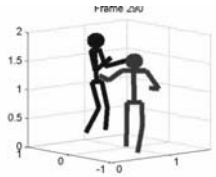
$$E_{cca}(\mathbf{B},\hat{\mathbf{B}},\mathbf{A},\mathbf{C},\boldsymbol{\mu},\hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n}\left\|\hat{\mathbf{d}}_i - \hat{\boldsymbol{\mu}} - \hat{\mathbf{B}}\mathbf{c}_i\right\|^2_{\mathbf{W}1_i}$$

Projection

$$+\lambda\sum_{i=1}^{n}\left\|\mathbf{c}_i - \mathbf{B}^T(\mathbf{d}_i - \boldsymbol{\mu})\right\|^2_{\mathbf{W}2_i} + \lambda_2\sum_{i=1}^{n}\left\|\mathbf{c}_i - \mathbf{A}\mathbf{c}_{i-1}\right\|^2_{\mathbf{W}3_i}$$

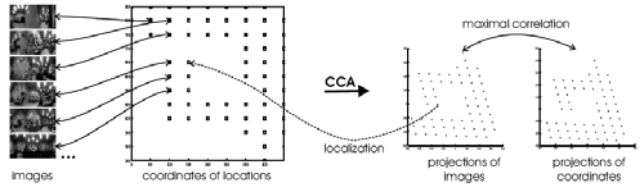
Dynamics

Dynamic Coupled Component Analysis



Robot localization with Canonical Correlation Analysis

(Skocaj & Leonardis, 2000)



Canonical Correlation Analysis (CCA)

(Mardia et al., 1979; Borga)

- Learn relations between multiple data sets? (e.g. find features in one set related to another data set)
- Given two sets $\mathbf{X} \in \Re^{d_1 \times n}$ *and* $\mathbf{Y} \in \Re^{d_2 \times n}$, CCA finds the pair of directions $\mathbf{w}_x$ and $\mathbf{w}_y$ that maximize the correlation between the projections (assume zero mean data)

$$\rho = \frac{\mathbf{w}_x^T \mathbf{X}^T \mathbf{Y} \mathbf{w}_y}{\sqrt{\mathbf{w}_x^T \mathbf{X}^T \mathbf{X} \mathbf{w}_x \mathbf{w}_y^T \mathbf{Y}^T \mathbf{Y} \mathbf{w}_y^T}}$$

- Several ways of optimizing it:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{X}^T\mathbf{Y} \\ \mathbf{X}^T\mathbf{Y} & \mathbf{0} \end{bmatrix} \in \Re^{(d_1+d_2)\times(d_1+d_2)}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}^T\mathbf{Y} \end{bmatrix} \in \Re^{(d_1+d_2)\times(d_1+d_2)} \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix}$$

- An stationary point of r is the solution to CCA.

$$\mathbf{A}\mathbf{w} = \lambda \mathbf{B}\mathbf{w}$$

More on Canonical Correlation Analysis

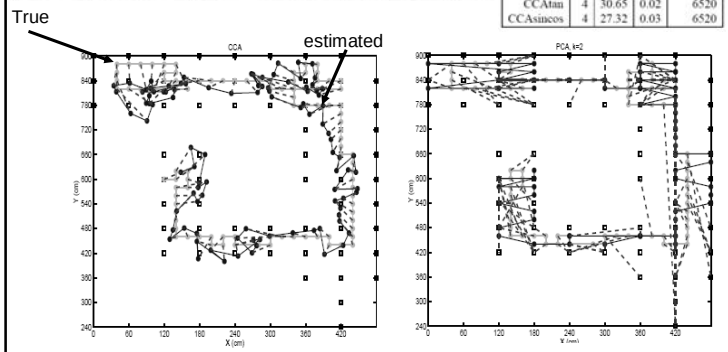
- If d_1>>n using the kernel trick efficient ways of solving it.
- Maximum number of canonical correlation vectors min(d_1,d_2)
- Learn a linear mapping between **Y** and the projection of **X** into canonical components.

$$\min_{\mathbf{F}} \| \mathbf{Y} - \mathbf{F}(\mathbf{W}_x^T \mathbf{X}) \|_F \qquad \mathbf{F} = \mathbf{Y}(\mathbf{W}_x \mathbf{X})^+$$

Examples

- Training images

(a) 2DOF	k	error	secs	elements
CCA	2	34.77	0.02	3906
PCA	2	77.37	0.21	4148
PCA	10	26.16	0.23	15044
PCAint	10	13.04	2.77	24500
KCCA	10	26.65	0.23	15044
KCCAint	10	13.12	2.81	24500
CCAsin	2	58.24	0.02	3906
CCAcos	2	34.33	0.02	3906
CCAtan	4	30.65	0.02	6520
CCAsincos	4	27.32	0.03	6520



Relevant Component Analysis

(Shental et al., 2002)

- Adding side-information

Algorithm 1 The RCA algorithm

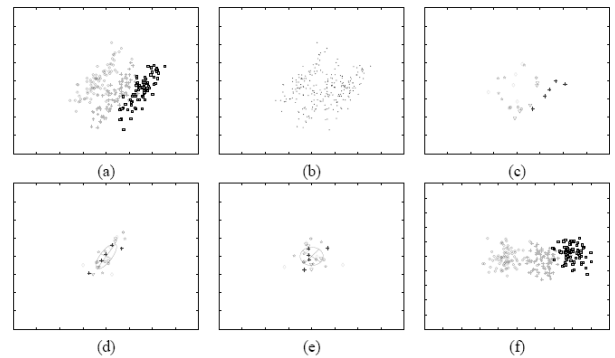
Given a dataset $\{x_i\}_{i=1}^{N}$ and k chunklets $C_j = \{x_{ji}\}_{i=1}^{n_j}$ $j = 1 \ldots k$, do

1. For each chunklet C_j, subtract the chunklet's mean from all the points it contains (Fig. 1d).
2. Compute the covariance matrix of all the centered data-points in chunklets (Fig. 1d). Assume a total of p points in k chunklets, where chunklet C_j consists of points $\{x_{ji}\}_{i=1}^{n_j}$ and its mean is $\hat{m}_j$. RCA computes the following matrix:

$$\hat{C} = \frac{1}{p}\sum_{j=1}^{k}\sum_{i=1}^{n_j}(x_{ji} - \hat{m}_j)(x_{ji} - \hat{m}_j)^t \qquad (1)$$

3. Compute the whitening transformation $W = \hat{C}^{-\frac{1}{2}}$ associated with this covariance matrix (Fig. 1e), and apply it to the original data points: $x_{new} = Wx$ (Fig. 1f). Alternatively, use the inverse of $\hat{C}$ as a Mahalanobis distance.

Example



Standard extensions

- Latent Variable Models
- Tensor Factorization
  - 2D PCA/LDA.
  - Higher order extension.
- Kernel Methods

Factor Analysis

- A Gaussian distribution on the coefficients and noise is added to PCA➔ Factor Analysis. (Mardia et al., 1979)

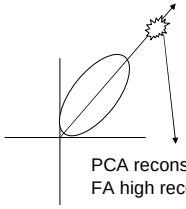
$$\mathbf{d} = \boldsymbol{\mu} + \mathbf{B}\mathbf{c} + \boldsymbol{\eta}$$

$$p(\mathbf{c}) = N(\mathbf{c} \mid \mathbf{0}, \mathbf{I}_k) \qquad p(\mathbf{d} \mid \mathbf{c}, \mathbf{B}) = N(\mathbf{d} \mid \boldsymbol{\mu} + \mathbf{B}\mathbf{c}, \Psi)$$

$$p(\boldsymbol{\eta}) = N(\mathbf{c} \mid \mathbf{0}, \Psi) \qquad \Psi = diag(\eta_1, \eta_2, \ldots, \eta_d)$$

$$\mathrm{cov}(\mathbf{d}) = E((\mathbf{d} - \boldsymbol{\mu})(\mathbf{d} - \boldsymbol{\mu})^T) = \mathbf{B}\mathbf{B}^T + \Psi$$

- Inference (Roweis & Ghahramani, 1999;Tipping & Bishop, 1999a)



$p(\mathbf{c}, \mathbf{d})$ Jointly Gaussian

$$p(\mathbf{c} \mid \mathbf{d}) = N(\mathbf{c} \mid \mathbf{m}, \mathbf{V})$$

$$\mathbf{m} = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T + \Psi)^{-1}(\mathbf{d} - \boldsymbol{\mu})$$

$$\mathbf{V} = (\mathbf{I} + \mathbf{B}^T\Psi^{-1}\mathbf{B})^{-1}$$

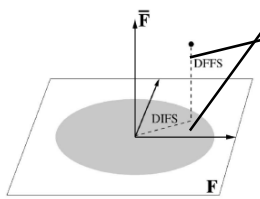
PCA reconstruction low error.
FA high reconstruction error (low likelihood).

Ppca

- If $\Psi = E(\boldsymbol{\eta}\boldsymbol{\eta}^T) = \varepsilon \mathbf{I}_d$ PPCA.
- If $\varepsilon \to 0$ is equivalent to PCA. $\varepsilon \to 0 \quad \mathbf{B}^T(\mathbf{B}\mathbf{B}^T + \Psi)^{-1} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$
- Probabilistic visual learning (Moghaddam & Pentland, 1997;)

$$p(\mathbf{d}) = \int p(\mathbf{d}\,|\,\mathbf{c})\,p(\mathbf{c})\,d\mathbf{c} = \frac{e^{-\frac{1}{2}(\mathbf{d}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{d}-\boldsymbol{\mu})}}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} = \frac{e^{-\frac{1}{2}(\mathbf{d}-\boldsymbol{\mu})^T(\mathbf{B}\mathbf{B}^T+\varepsilon\mathbf{I})^{-1}(\mathbf{d}-\boldsymbol{\mu})}}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} = \left[\frac{e^{-\frac{1}{2}\sum_{i=1}^{k}\frac{c_i^2}{\lambda_i}}}{(2\pi)^{\frac{d}{2}}\prod_{i=1}^{k}\lambda_i^{1/2}}\right]\left[\frac{e^{-\frac{\varepsilon^2(\mathbf{d})}{2\rho}}}{(2\pi\rho)^{\frac{(d-k)}{2}}}\right]$$

$$\mathbf{c}_i = \mathbf{B}^T\mathbf{d}_i$$



A least squares interpretation

- Directly minimizing

Sample covariance

$$E(\mathbf{B},\Lambda,\sigma) = \left\|\boldsymbol{\Sigma} - \mathbf{B}\Lambda\mathbf{B}^T - \sigma^2\mathbf{I}_d\right\|_F$$

$$\mathbf{B}^T\mathbf{B} = \mathbf{I}_d \quad \Lambda \ \textit{is diagonal}$$

derives in the same solution as PPCA (de la Torre & Kanade, 2005b).

$\boldsymbol{\Sigma}$ d×d $\qquad \mathbf{B}\mathbf{B}^T + \sigma^2\mathbf{I}$

$$\frac{d(d+1)}{2} \approx \frac{d^2}{2} \qquad \frac{k(2d-k+1)}{2} \approx kd$$

More on PPCA

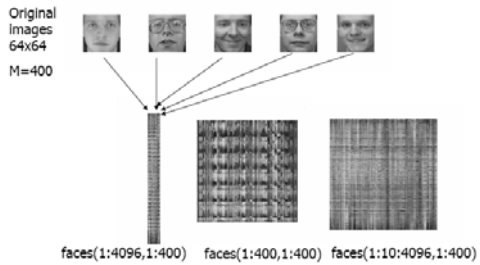
- Extension to mixtures of Ppca (mixture of subspaces). (Tipping & Bishop, 1999b; Black et al., 1998; Jebara et al., 1998)
- Tracking (Yang et al., 1999; Yang et al., 2000a; Lee et al., 2005; de la Torre et al., 2000b)
- Recognition/Detection (Moghaddam et al., 2000; Shakhnarovich & Moghaddam, 2004; Everingham & Zisserman, 2006)
- PCA for the exponential family (collins et al., 2001)

Tensor Decomposition

- 2D PCA/LDA
- General tensor factorization.

2D PCA/SVD

- Vectorizing images do not preserve 2D properties and spatial properties are lost.
- Many ways of stacking images into matrices



SVD versus 2D SVD

(Ye, 2004, Ding & Ye, 2006)

$$E(\mathbf{L},\mathbf{R},\{\mathbf{M}_i\}) = \sum_{i=1}^{n} \left\| \mathbf{D}_i - \mathbf{L}\mathbf{M}_i\mathbf{R}^T \right\|_F$$

$$\mathbf{D}_i \in \Re^{r\times c} \quad \mathbf{L} \in \Re^{r\times l_1} \quad \mathbf{R} \in \Re^{c\times l_2} \quad \mathbf{M}_i \in \Re^{l_1\times l_2}$$

$$\mathbf{L}^T\mathbf{L} = \mathbf{I} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}$$

- Compression ratio: $\frac{nrc}{rl_1 + cl_2 + nl_2l_1}$
- Recognition. $\| \mathbf{D}_i - \mathbf{D}_j \| = \| \mathbf{L}(\mathbf{M}_i - \mathbf{M}_j)\mathbf{R} \| \approx \| (\mathbf{M}_i - \mathbf{M}_j) \|$ $\quad \frac{rc}{l_2 l_1}$
- 2D SVD smaller computational cost (space & time)
- Same or better reconstruction (for same number of parameters)

Optimization

- No closed form solution.
- Alternate

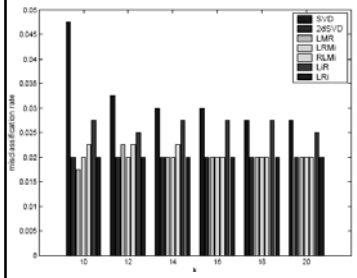
$$\text{Iterate until convergence} \left\{ \begin{array}{l} \left.\begin{array}{l} (\sum_{i=1}^{n} \mathbf{D}_i\mathbf{R}\mathbf{R}^T\mathbf{D}_i^T)\mathbf{L} = \mathbf{L}\Lambda_1 \\ (\sum_{i=1}^{n} \mathbf{D}_i^T\mathbf{L}\mathbf{L}^T\mathbf{D}_i)\mathbf{R} = \mathbf{R}\Lambda_2 \end{array}\right\} \text{Iterate until convergence} \\ \mathbf{M}_i = \mathbf{L}^T\mathbf{D}_i\mathbf{R} \end{array} \right.$$

2D PCA/SVD

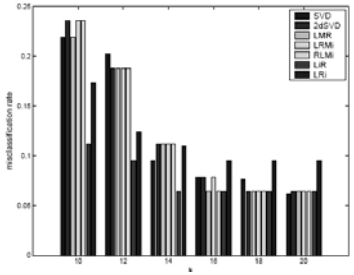
- SVD (k=15), storage 160560
- 2DSVD (l_1=l_2=15), storage 93060

Recognition

ORL



AR



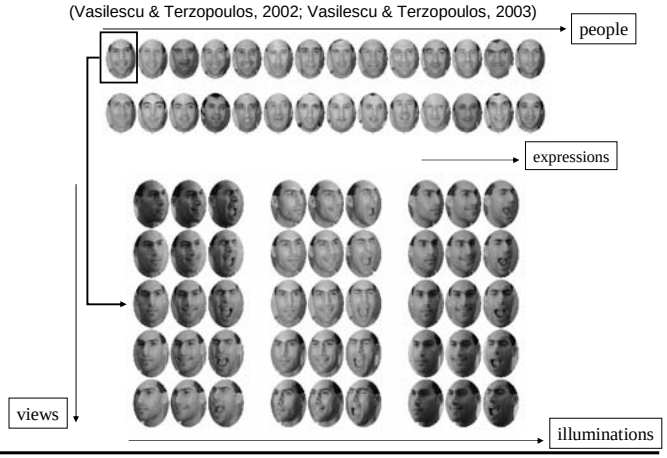
Eigenfaces

- Facial images (identity change)

- Eigenfaces bases vectors capture the variability in facial appearance (do not decouple pose, illumination, …)

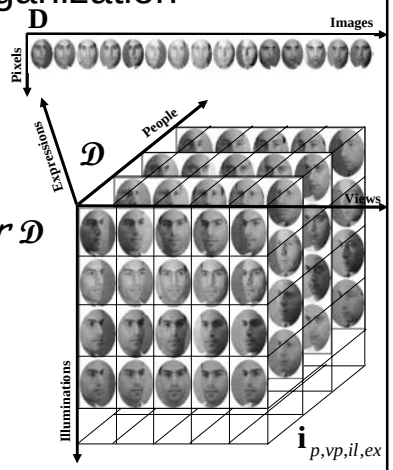
Tensor faces

(Vasilescu & Terzopoulos, 2002; Vasilescu & Terzopoulos, 2003)



Data Organization

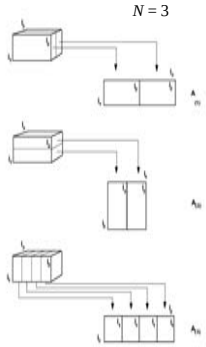
- Linear/PCA: **Data Matrix**
  - $R^{\text{pixels x images}}$
  - a matrix of image vectors
- Multilinear: *Data Tensor* $\mathcal{D}$
  - $R^{\text{people x views x illums x express x pixels}}$
  - N-dimensional matrix
  - 28 people, 45 images/person
  - 5 views, 3 illuminations, 3 expressions per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



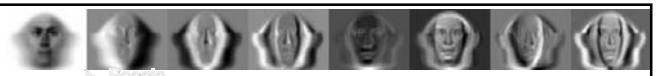
N-Mode SVD Algorithm

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

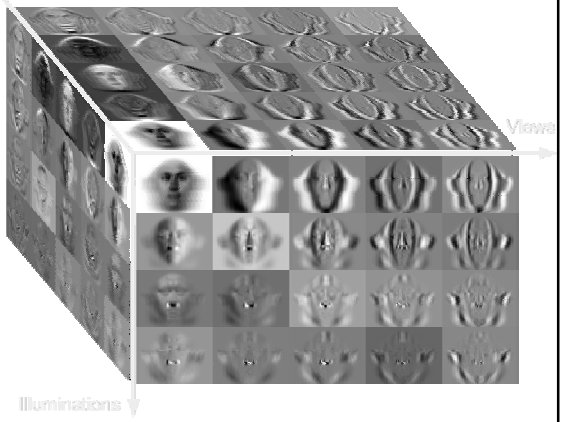
$N = 3$



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

176 basis vectors

TensorFaces

6 illum + 11 people param.

66 basis vectors

TensorFaces	**PCA**
Mean Sq. Err. = **409.15**	Mean Sq. Err. = **85.75**
3 illum + 11 people param.	33 parameters
33 basis vectors	33 basis vectors

Results

Data Set - 16,875 images	Training Images - 2,700	Test Images:
– 75 people	– 75 people	– 75 people
– 15 viewpoints	– 6 viewpoints	– 9 viewpoints
– 15 illuminations	– 6 illuminations	– 9 illums

Linear Models		Multilinear Models	
PCA	ICA	TensorFaces	Independent TensorFaces
83%	89%	93%	97%

Related work

- Tensor factorization (O'Leary & Peleg, 1983;Shashua & Levin, 2001; Paatero & Tapper, 1994;Shashua & Hazan, 2005)
- 2D PCA (Kong & Wang, 2005; Ding & Ye, 2006; Zhang et al., 2006; Zhang & Zhou, 2005; Yang et al., 2004b)
- 2D LDA (Ye, 2005; Ye et al., 2005; Liu et al., 1993)

Kernel Methods

- Given in the tutorial.