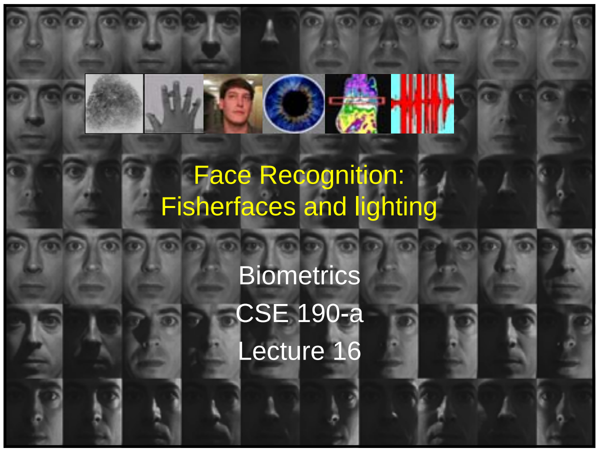


Face Recognition: Fisherfaces and lighting

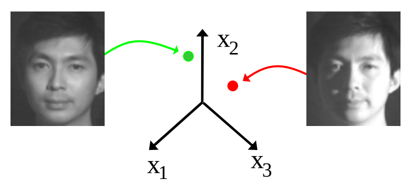
Biometrics
CSE 190-a
Lecture 16

Results of Hand Recognition!!

Vikram (CS1)	42%
Vikram (CS2) Nearest Neighbor	92%
Tom Bayesian	82%
Tom Nearest Neighbor	86%
Taurin (Statistical)	64%
Taurin (Nearest Neighbor)	60%
Warren (Bayesian)	32%
Warren (Nearest Neighbor)	42%
Alex	64%

Why is Face Recognition Hard?

Image as a Feature Vector

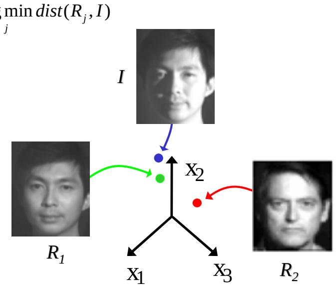


- Consider an n-pixel image to be a point in an n-dimensional space, $\mathbf{x} \in \mathbf{R}^n$.
- Each pixel value is a coordinate of **x**.

Nearest Neighbor Classifier

$\{ R_j \}$ are set of training images.

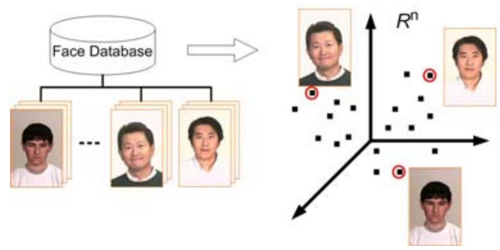
$$ID = \arg\min_j dist(R_j, I)$$



Comments

- Sometimes called “Template Matching”
- Variations on distance function (e.g. L_1, robust distances)
- Multiple templates per class- perhaps many training images per class.
- Expensive to compute k distances, especially when each image is big (N dimensional).
- May not generalize well to unseen examples of class.
- Some solutions:
  - Bayesian classification
  - Dimensionality reduction

- Face Space:
  - A set of face images construct a *face space* in R^n
  - Appearance-based methods analyze the distributions of individual faces in face space



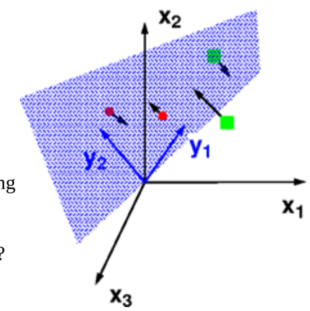
Eigenfaces: linear projection

- An n-pixel image $\mathbf{x} \in \mathbf{R}^n$ can be projected to a low-dimensional feature space $\mathbf{y} \in \mathbf{R}^m$ by

$$\mathbf{y} = W\mathbf{x}$$

where W is an n by m matrix.

- Recognition is performed using nearest neighbor in $\mathbf{R}^m$.
- How do we choose a good W?



Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of n feature vectors $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ in $\mathbb{R}^d$. Write

$$\boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{x}_i$$

$$\Sigma = \frac{1}{n-1}\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

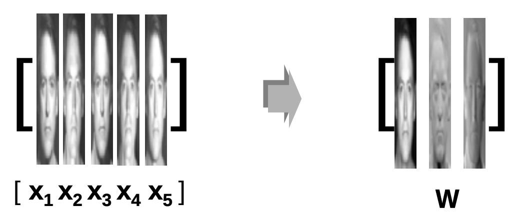
The unit eigenvectors of Σ — which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the k-dimensional set of linear features that preserves the most variance.

Algorithm 22.5: *Principal components analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.*

Some details: Use Singular value decomposition, "trick" described in text to compute basis when n<<d

How do you construct Eigenspace?



Construct data matrix by stacking vectorized images and then apply Singular Value Decomposition (SVD)

Singular Value Decomposition

Excellent ref: 'Matrix Computations," Golub, Van Loan

- Any m by n matrix $\boldsymbol{A}$ may be factored such that

$$\boldsymbol{A} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$$

[m x n] = [m x m][m x n][n x n]

- $\boldsymbol{U}$: m by m, orthogonal matrix
  - Columns of **U** are the eigenvectors of $\boldsymbol{AA}^T$
- $\boldsymbol{V}$: n by n, orthogonal matrix,
  - columns are the eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$
- **Σ:** m by n, diagonal with non-negative entries (σ_1, σ_2, …, σ_s) with s=min(m,n) are called the called the singular values
  - Singular values are the square roots of eigenvalues of both $\boldsymbol{AA}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$
  - *Result of SVD algorithm:* $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$

Eigenfaces

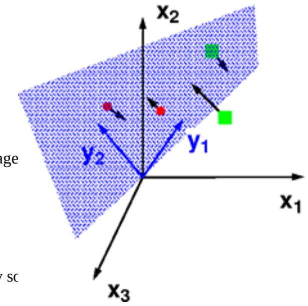
- Modeling
  1. Given a collection of n labeled training images,
  2. Compute mean image and covariance matrix.
  3. Compute k Eigenvectors (note that these are images) of covariance matrix corresponding to k largest Eigenvalues.
  4. Project the training images to the k-dimensional Eigenspace.
- Recognition
  1. Given a test image, project to Eigenspace.
  2. Perform classification to the projected training images.

Underlying assumptions

- Background is not cluttered (or else only looking at interior of object
- Lighting in test image is similar to that in training image.
- No occlusion
- Size of training image (window) same as window in test image.

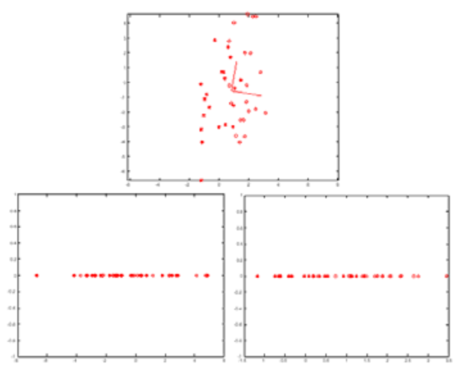
Face detection using "distance to face space"

- Scan a window ω across the image, and classify the window as face/not face as follows:
- Project window to subspace, and reconstruct as described earlier.
- Compute distance between ω and reconstruction.
- Local minima of distance over all image locations less than some treshold are taken as locations of faces.
- Repeat at different scales.
- Possibly normalize windows intensity sc that $|\omega| = 1$.



Difficulties with PCA

- Projection may suppress important detail
  - smallest variance directions may not be unimportant
- Method does not take discriminative task into account
  - typically, we wish to compute features that allow good discrimination
  - not the same as largest variance



Illumination Variability

"The variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity."
-- Moses, Adini, Ullman, ECCV '94

Fisherfaces: Class specific linear projection

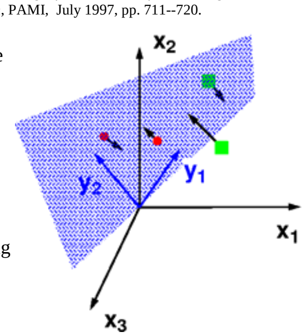
P. Belhumeur, J. Hespanha, D. Kriegman, *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*, PAMI, July 1997, pp. 711--720.

- An n-pixel image $\boldsymbol{x} \in \mathbf{R}^n$ can be projected to a low-dimensional feature space $\boldsymbol{y} \in \mathbf{R}^m$ by

$$\boldsymbol{y} = W\boldsymbol{x}$$

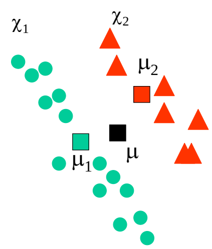
where W is an n by m matrix.

- Recognition is performed using nearest neighbor in $\mathbf{R}^m$.
- How do we choose a good W?

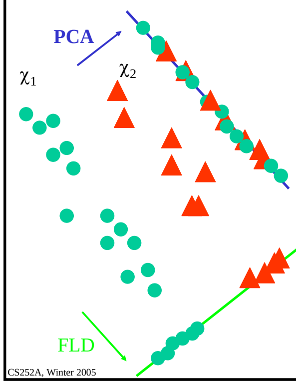


PCA & Fisher's Linear Discriminant

- Between-class scatter
  $$S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$
- Within-class scatter
  $$S_W = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu_i)(\mu_k - \mu_i)^T$$
- Total scatter
  $$S_T = \sum_{i=1}^{c} \sum_{x_k \in \chi_i} (x_k - \mu)(\mu_k - \mu)^T = S_B + S_W$$
- Where
  - c is the number of classes
  - μ_i is the mean of class χ_i
  - $|\chi_i|$ is number of samples of χ_i.



PCA & Fisher's Linear Discriminant



- PCA (Eigenfaces)
  $$W_{PCA} = \arg\max_W |W^T S_T W|$$
  Maximizes projected total scatter
- Fisher's Linear Discriminant
  $$W_{fld} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$
  Maximizes ratio of projected between-class to projected within-class scatter

Computing the Fisher Projection Matrix

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [\mathbf{w}_1 \ \mathbf{w}_2 \ \ldots \ \mathbf{w}_m] \quad (4)$$

where $\{\mathbf{w}_i \mid i = 1, 2, \ldots, m\}$ is the set of generalized eigenvectors of S_B and S_W corresponding to the m largest generalized eigenvalues $\{\lambda_i \mid i = 1, 2, \ldots, m\}$, i.e.,

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \qquad i = 1, 2, \ldots, m$$

• The w_i are orthonormal
• There are at most c-1 non-zero generalized Eigenvalues, so m <= c-1
• Can be computed with *eig* in Matlab

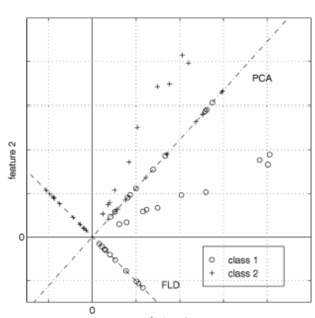
Fisherfaces

$$W = W_{fld} W_{PCA}$$
$$W_{PCA} = \arg\max_W |W^T S_T W|$$
$$W_{fld} = \arg\max_W \frac{|W^T W_{PCA}^T S_B W_{PCA} W|}{|W^T W_{PCA}^T S_W W_{PCA} W|}$$

• Since S_W is rank N-c, project training set to subspace spanned by first *N-c* principal components of the training set.
• Apply FLD to *N-c* dimensional subspace yielding *c-1* dimensional feature space.

• Fisher's Linear Discriminant projects away the within-class variation (lighting, expressions) found in training set.
• Fisher's Linear Discriminant preserves the separability of the classes.

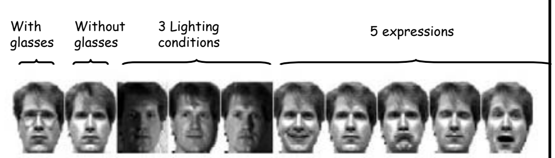
PCA vs. FLD



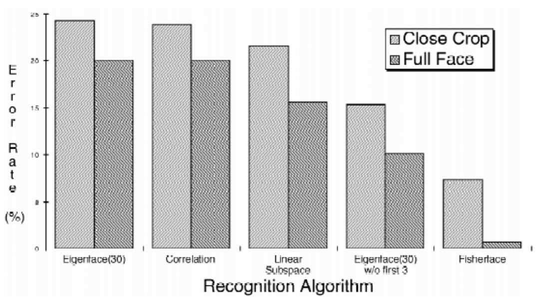
Experimental Results - 1

Variation in Facial Expression, Eyewear, and Lighting

- Input: 160 images of 16 people
- Train: 159 images
- Test: 1 image



Recognition Rates on Yale Face Database



Leave-one-out evaluation of PCA and LDA on the Yale Face Database [Belhumer, Hespanha, Kriegman 97]

Approach	Dim. of the subspace	Error rate (close crop)	Error rate (full face)
Eigenface (PCA)	30	24.4%	19.4%
Fisherface (LDA)	15	7.3%	0.6%

FLD For Glasses/No Glasses Recognition

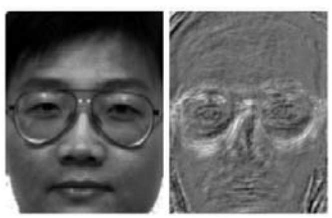
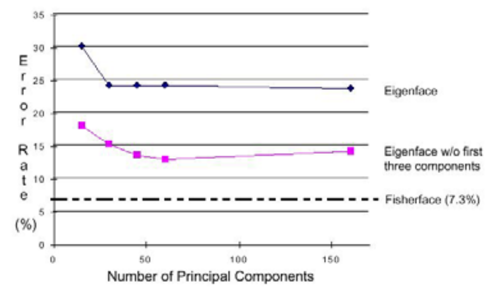


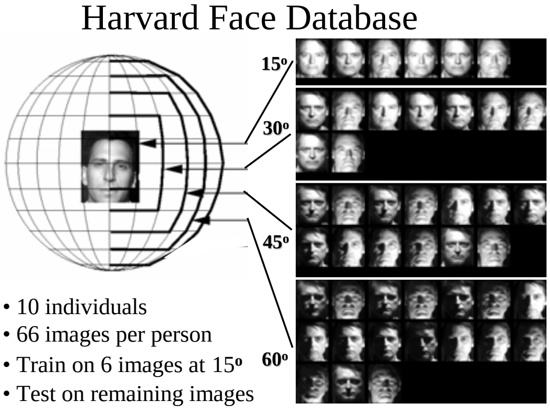
TABLE 1
COMPARATIVE RECOGNITION ERROR RATES FOR GLASSES/NO GLASSES RECOGNITION USING THE YALE DATABASE

Glasses Recognition		
Method	Reduced Space	Error Rate (%)
PCA	10	52.6
Fisherface	1	5.3

Experimental Results - 2

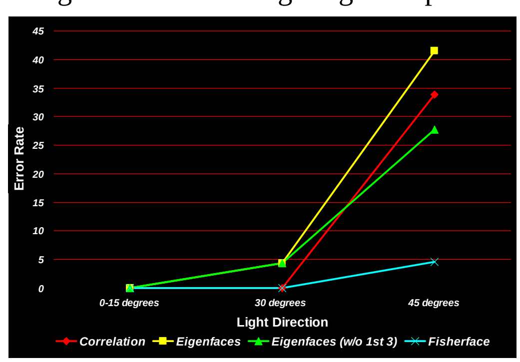


Harvard Face Database

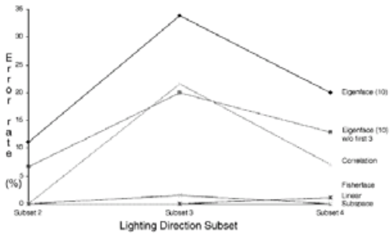


- 10 individuals
- 66 images per person
- Train on 6 images at 15º
- Test on remaining images

Recognition Results: Lighting Extrapolation



Interpolation in Illumination



(2D) Model-based
— Active Appearance Model

- Model Construction (linear)

labeled image   landmarks   shape-free texture

Active Appearance Model (AAM)

- Shape model

$$s = (x_1, y_1, ..., x_n, y_n)^T \qquad s = \bar{s} + P_s b_s$$

- Appearance model

$$g = (I_1, ..., I_m)^T \qquad g = \bar{g} + P_g b_g$$

- Combined model

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s P_s^T (s - \bar{s}) \\ P_g^T (g - \bar{g}) \end{pmatrix} \xrightarrow{\text{PCA}} b = Qc$$

AAM

- Model fitting
  - Minimize the objective function (gray level difference between the given image and the stored model

$$\Delta = |\delta I|^2$$

  - Searching by learning
    - Annotated model (true model parameters)
    - Relation: known model displacements ↔ observed difference vector
    - Use multivariate multiple regression to learn the relation and predict the displacement during searching

AAM

Initial   3 its   8 its   11 its   Converged   Original