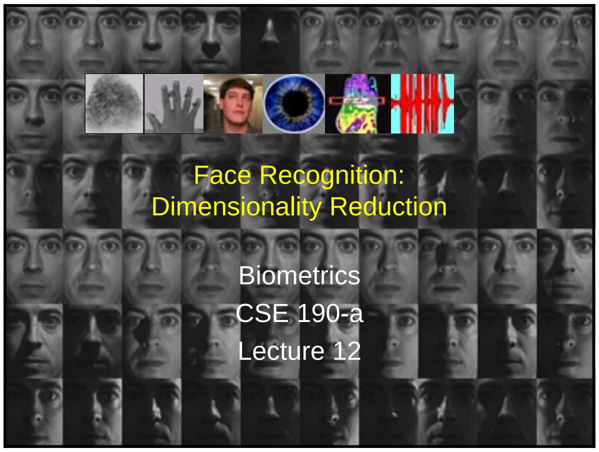


Face

Face Recognition

- Face is the most common biometric used by humans
- Applications range from static, mug-shot verification to a dynamic, uncontrolled face identification in a cluttered background
- Challenges:
  - automatically locate the face
  - recognize the face from a general view point under different illumination conditions, facial expressions, and aging effects

Authentication vs Identification

- Face Authentication/Verification (1:1 matching)

- Face Identification/recognition (1:N matching)

Applications

- Access Control

www.viisage.com

www.visionics.com

Applications

- Video Surveillance (On-line or off-line)

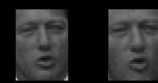
Face Scan at Airports

www.facesnap.de

Why is Face Recognition Hard?

Many faces of Madonna

Who are these people?



[Sinha and Poggio 1996]

Why is Face Recognition Hard?

Face Recognition Difficulties

- Identify similar faces (inter-class similarity)
- Accommodate intra-class variability due to:
  - head pose
  - illumination conditions
  - expressions
  - facial accessories
  - aging effects
- Cartoon faces

Inter-class Similarity

- Different persons may have very similar appearance

www.marykateandashley.com

Twins

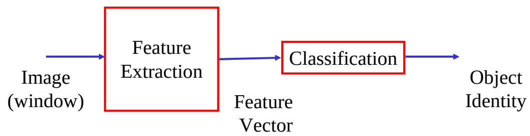
news.bbc.co.uk/hi/english/in_depth/americas/2000/us_elections

Father and son

Intra-class Variability

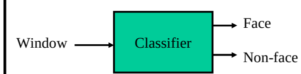
- Faces with intra-subject variations in pose, illumination, expression, accessories, color, occlusions, and brightness

Sketch of a Pattern Recognition Architecture



Example: Face Detection

- Scan window over image.
- Classify window as either:
  - Face
  - Non-face

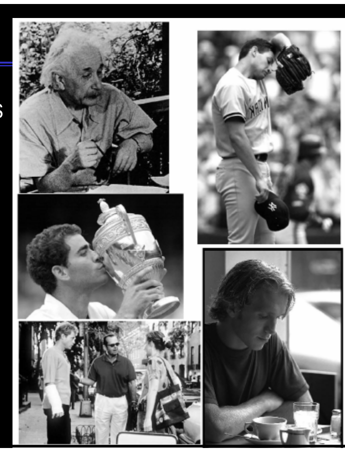


Detection Test Sets

Data Set	Location	Description
MIT Test Set [154]	http://www.cs.cmu.edu/~har	Two sets of high and low resolution gray scale images with multiple faces in complex background.
CMU Test Set [128]	http://www.cs.cmu.edu/~har	130 gray scale images with a total of 507 frontal faces.
CMU Profile Face Test Set [141]	ftp://eyes.ius.cs.cmu.edu/usr20/ ftp/testing_face_images.tar.gz	208 gray scale images with faces in profile views.
Kodak Data Set [94]	Eastman Kodak Corporation	Faces of multiple size, pose and under varying illumination in color images. Designed for face detection and recognition.

Profile views

Schneiderman's Test set



Face Detection: Experimental Results

Test sets: two CMU benchmark data sets
Test set 1: 125 images with 483 faces
Test set 2: 20 images with 136 faces

Method	Test Set 1		Test Set 2	
	Detection Rate	False Detections	Detection Rate	False Detections
Distribution based [154]	N/A	N/A	81.9%	13
Neural network [128]	92.5%	862	90.3%	42
Naive Bayes classifier [140]	93.0%	88	91.2%	12
Kullback relative information [24]	98.0%	12758	N/A	N/A
Support vector machine [107]	N/A	N/A	74.2%	20
Mixture of factor analyzers [175]	92.3%	82	89.4%	3
Fisher linear discriminant [175]	93.6%	74	91.5%	1
SNoW with primitive features [176]	94.2%	84	93.6%	3
SNoW with multi-scale features [176]	94.8%	78	94.1%	3
Inductive learning [38]	90%	N/A	N/A	N/A

[See also work by Viola & Jones, Rehg, more recent by Schneiderman]

Example: Finding skin
Non-parametric Representation of CCD

- Skin has a very small range of (intensity independent) colors, and little texture
  - Compute an intensity-independent color measure, check if color is in this range, check if there is little texture (median filter)
  - See this as a classifier - we can set up the tests by hand, or learn them.
  - get class conditional densities (histograms), priors from data (counting)
- Classifier is
  - if $p(\text{skin}|\boldsymbol{x}) > \theta$, classify as skin
  - if $p(\text{skin}|\boldsymbol{x}) < \theta$, classify as not skin
  - if $p(\text{skin}|\boldsymbol{x}) = \theta$, choose classes uniformly and at random

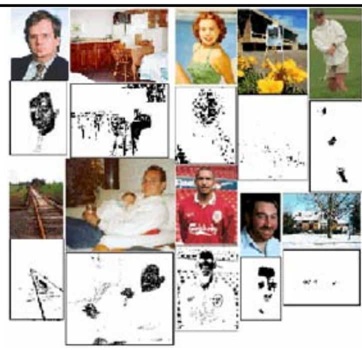
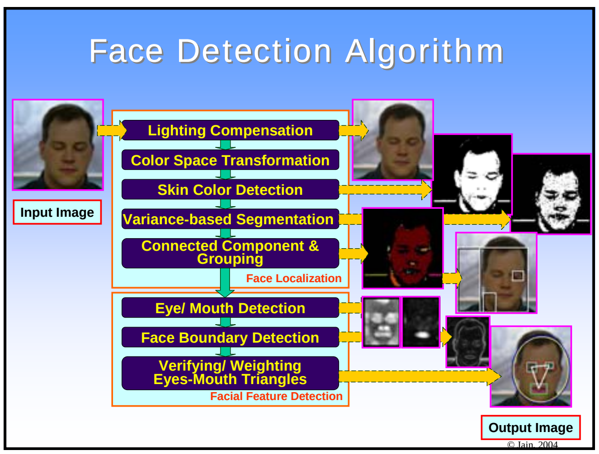
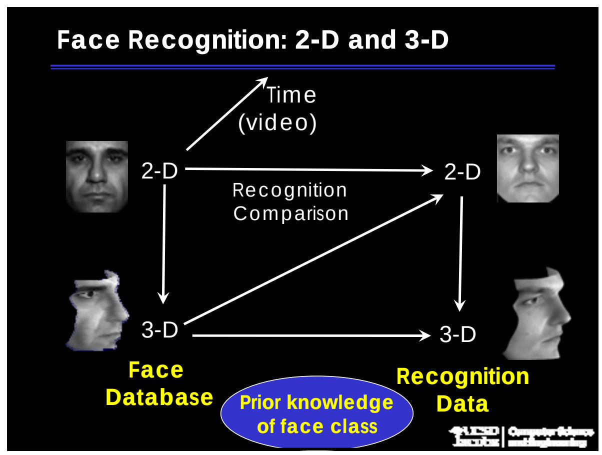


Figure from "Statistical color models with application to skin detection," M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE

Face Detection





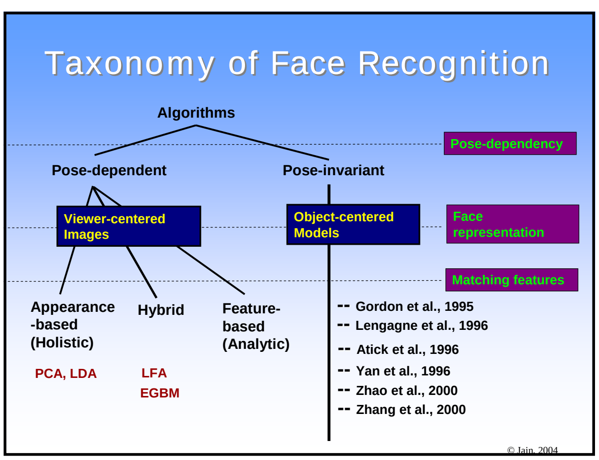
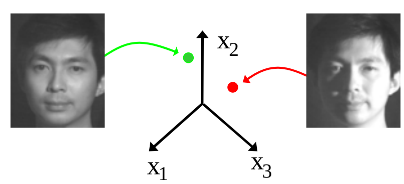


Image as a Feature Vector

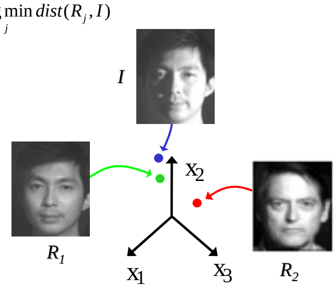


- Consider an n-pixel image to be a point in an n-dimensional space, $\mathbf{x} \in \mathbf{R}^n$.
- Each pixel value is a coordinate of $\mathbf{x}$.

Nearest Neighbor Classifier

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

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

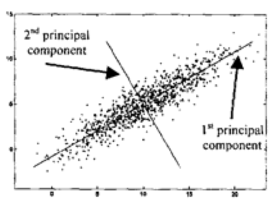


Comments

- 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

Eigenface (Turk, Pentland, 91) -1

- Use Principle Component Analysis (PCA) to determine the most discriminating features between images of faces.



Eigenfaces: linear projection

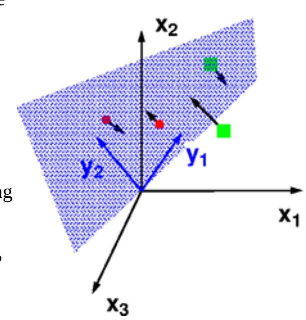
• An n-pixel image $\boldsymbol{x} \in \mathbf{R}^n$ can be projected to a low-dimensional feature space $\boldsymbol{y} \in \mathbf{R}^m$ by

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

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?



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

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

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

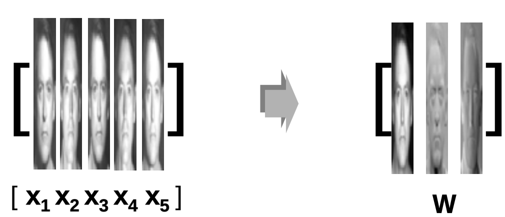
The unit eigenvectors of Σ — which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue — give a set of features with the following properties:

- 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)

Matrix Decompositions

- Definition: The factorization of a matrix $\boldsymbol{M}$ into two or more matrices $\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_n$, such that $\boldsymbol{M} = \boldsymbol{M}_1\boldsymbol{M}_2\ldots\boldsymbol{M}_n$.
- Many decompositions exist…
  - ***QR*** Decomposition
  - ***LU*** Decomposition
  - ***LDU*** Decomposition
  - Etc.

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 ($\sigma_1, \sigma_2, \ldots, \sigma_s$) 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:* $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$

SVD Properties

- In Matlab [u s v] = svd(A), and you can verify that: A=u*s*v'
- r=Rank($\boldsymbol{A}$) = # of non-zero singular values.
- $\boldsymbol{U}$, $\boldsymbol{V}$ give us orthonormal bases for the subspaces of $\boldsymbol{A}$:
  - 1st r columns of $\boldsymbol{U}$: Column space of $\boldsymbol{A}$
  - Last $m - r$ columns of $\boldsymbol{U}$: Left nullspace of $\boldsymbol{A}$
  - 1st r columns of $\boldsymbol{V}$: Row space of $\boldsymbol{A}$
  - last $n - r$ columns of $\boldsymbol{V}$: Nullspace of $\boldsymbol{A}$
- *For d<= r, the first d column of U provide the best d-dimensional basis for columns of A in least squares sense.*

Thin SVD

- 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 n][n x n][n x n]

- If m>n, then one can view Σ as:

$$\begin{bmatrix} \Sigma' \\ 0 \end{bmatrix}$$

- Where Σ'=diag($\sigma_1, \sigma_2, \ldots, \sigma_s$) with s=min(m,n), and lower matrix is (n-m by m) of zeros.
- Alternatively, you can write:

$$\boldsymbol{A} = \boldsymbol{U}'\Sigma'\boldsymbol{V}^T$$

- In Matlab, thin SVD is:[U S V] = svds(A)

Application: Pseudoinverse

- Given $\boldsymbol{y} = \boldsymbol{Ax}$, $\boldsymbol{x} = \boldsymbol{A}^+\mathbf{y}$
- For square $\boldsymbol{A}$, $\boldsymbol{A}^+ = \boldsymbol{A}^{-1}$
- For any $\boldsymbol{A}$…

$$\boldsymbol{A}^+ = \boldsymbol{V\Sigma}^{-1}\boldsymbol{U}^{\mathrm{T}}$$

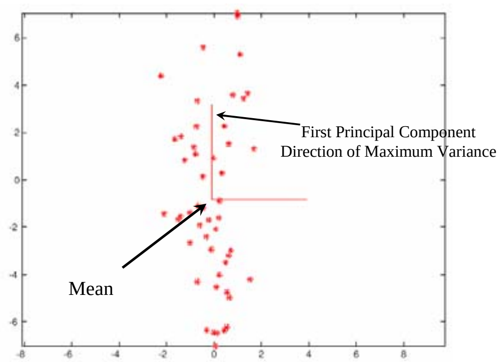
- $\boldsymbol{A}^+$ is called the ***pseudoinverse*** of $\boldsymbol{A}$.
- $\boldsymbol{x} = \boldsymbol{A}^+\mathbf{y}$ is the least-squares solution of $\boldsymbol{y} = \boldsymbol{Ax}$.
- Alternative to previous solution.

Performing PCA with SVD

- Singular values of A are the square roots of eigenvalues of both AA^T and A^TA & Columns of U are corresponding Eigenvectors
- And $\sum_{i=1}^{n} a_i a_i^T = [a_1 \ a_2 \ \cdots \ a_n][a_1 \ a_2 \ \cdots \ a_n]^T = AA^T$
- Covariance matrix is:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T$$

- So, ignoring 1/n subtract mean image μ from each input image, create data matrix, and perform thin SVD on the data matrix.



Eigenfaces

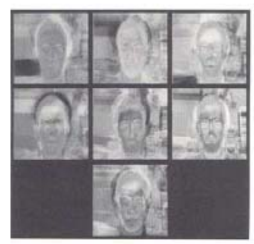
- Modeling
  1. Given a collection of n labeled training images,
  2. Compute mean image and covariance matrix.
  3. Compute k Eigenvectors (note that these are images) of covariance matrix corresponding to k largest Eigenvalues.
  4. Project the training images to the k-dimensional Eigenspace.
- Recognition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

Eigenfaces: Training Images

[Turk, Pentland 01

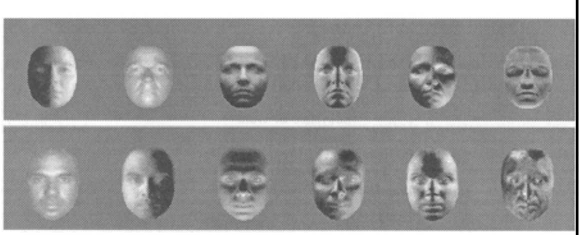
Eigenfaces

Mean Image



Basis Images

Variable Lighting

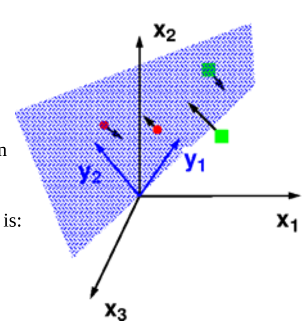


Projection, and reconstruction

- 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

$$y = Wx$$

- From $\boldsymbol{y} \in \mathbf{R}^m$, the reconstruction of the point is $\mathbf{W}^T\mathbf{y}$
- The error of the reconstruction is: $\|\mathbf{x}-\mathbf{W}^T\mathbf{W}\mathbf{x}\|$



Reconstruction using Eigenfaces

- Given image on left, project to Eigenspace, then reconstruct an image (right).