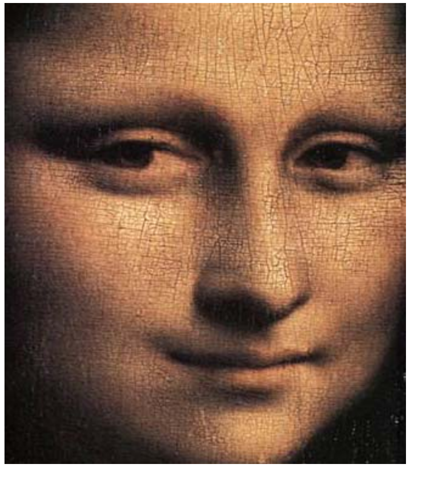
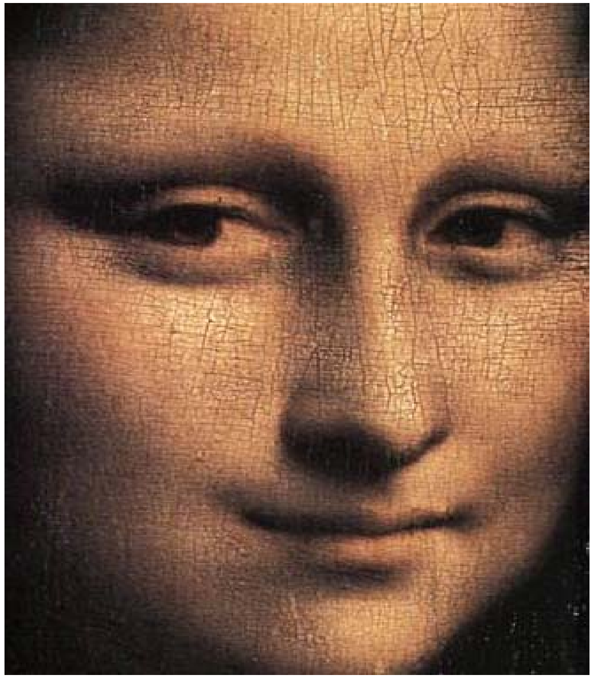
Multimedia Computing

Case Study:

Face Recognition



Face Image and Video Clip



Face Recognition

- Face is the most commonly used biometric characteristic 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

Verification vs. Identification

- Face Verification (1:1 matching)

- Face Identification (1:N matching)

Applications

- Access control

www.visionics.com

Applications

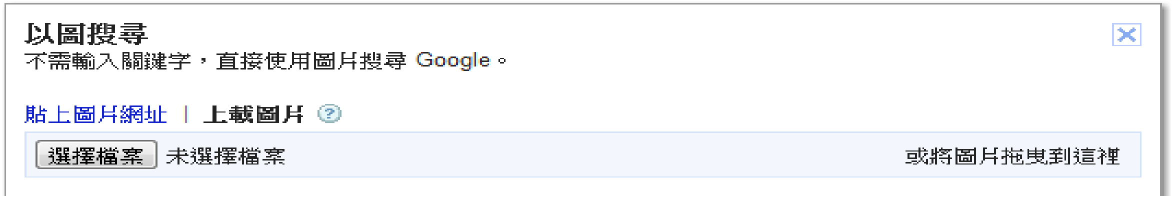
- Video surveillance (on-line or off-line)

Face Scan at Airports

The St. Petersburg-Clearwater Airport installed facial recognition systems at two security checkpoints in January. Six-foot tall towers (above) house cameras that snap pictures of passengers as they pass through magnetometers. The passengers' faces instantly are compared to a database of images of wanted criminals. Sheriff Everett Rice (above left) was one of the first people to pass through the new security system.

Applications

- Human-power search



這個圖片最有可能的推測結果：***leung chun ying***

Who are the two guys?

這個圖片最有可能的推測結果：***henry tang***

Applications

- Photo management

For many photos, they can be grouped by faces. (Picasa)

The friends are automatically recognized and tagged.

Find the pictures where you and your boyfriend/girlfriend are together.

Why is Face Recognition Hard?

- **Many faces of Madonna**

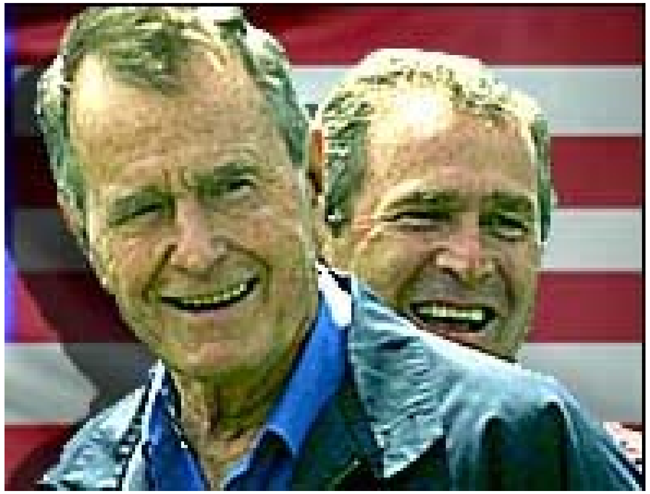
Face Recognition Difficulties

- 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



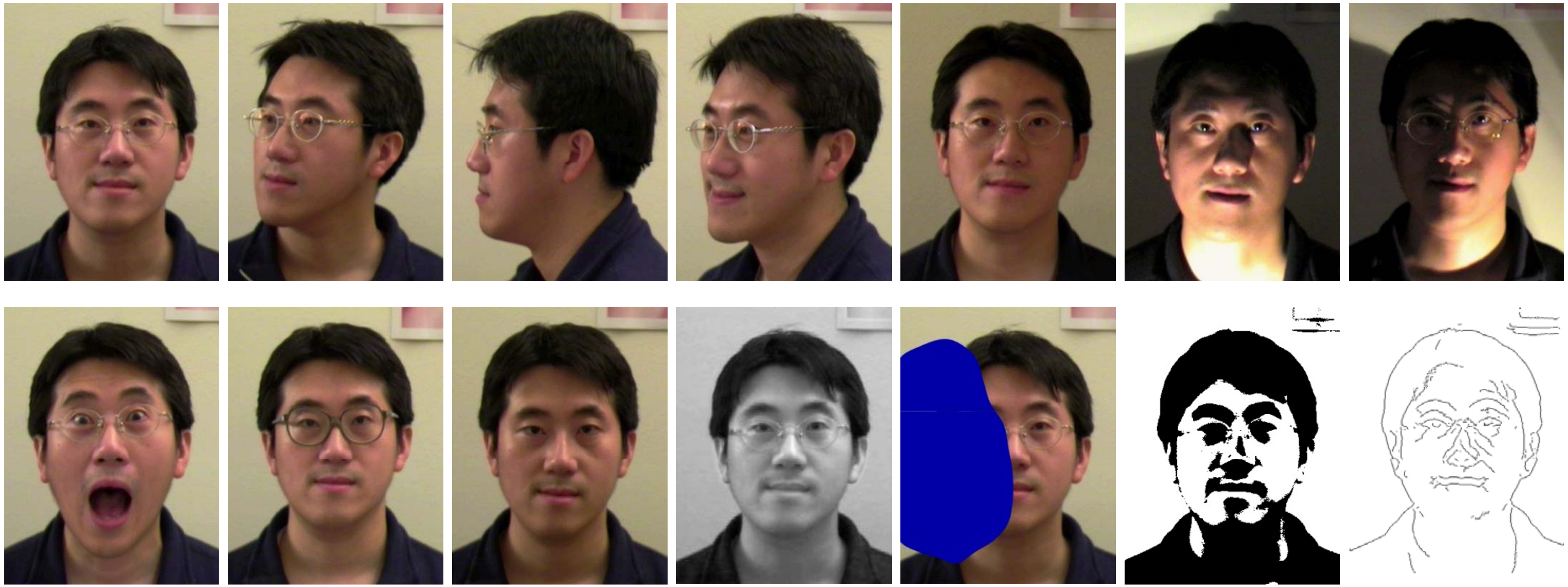
news.bbc.co.uk/hi/english/in_depth/americas/2000/us_elections

Twins

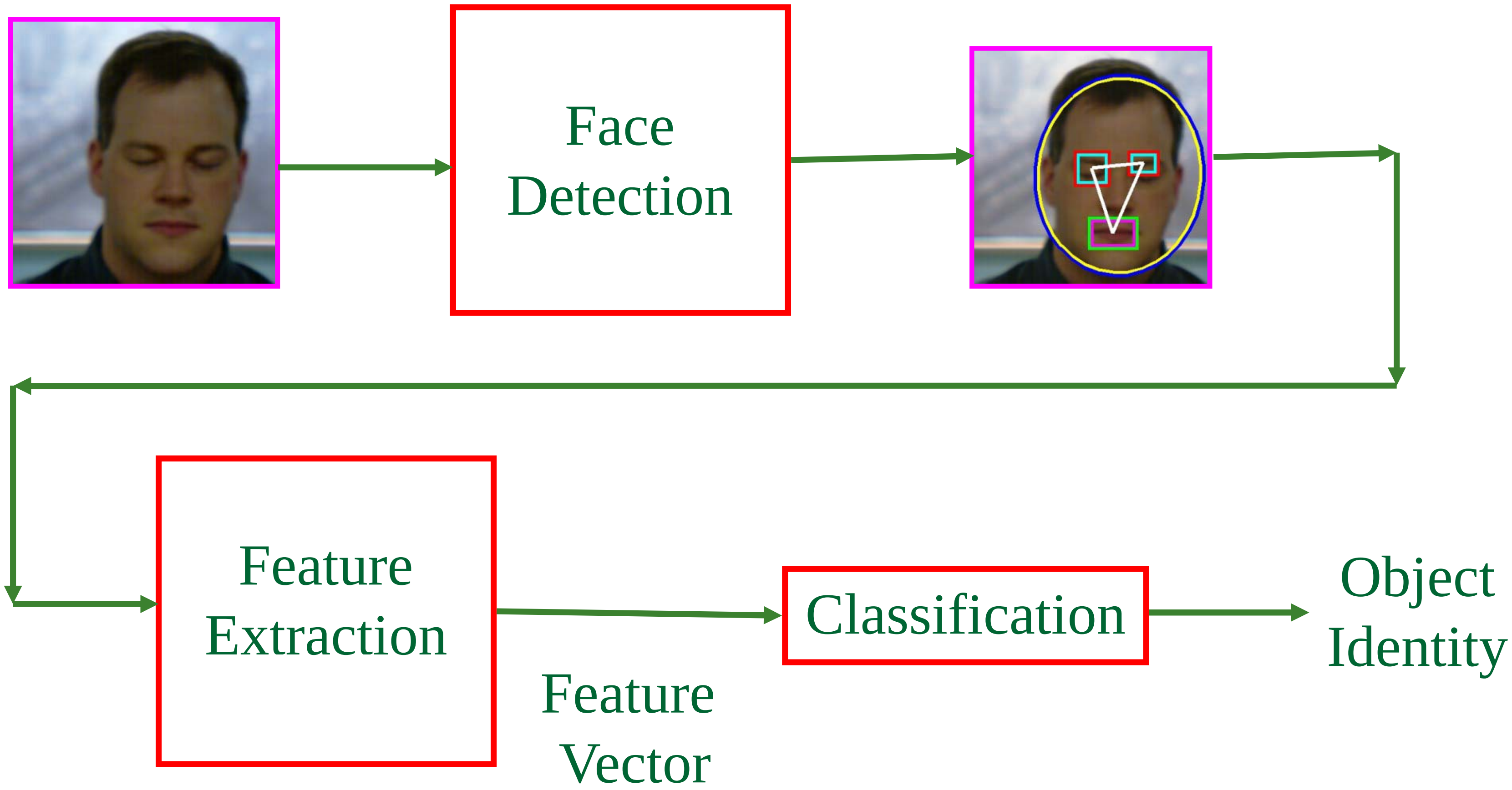
Father and son

Intra-class Variability

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



Face Recognition Architecture



Face Detection

- Statistics based methods
  - Subspace methods
  - NN (Neural Networks) methods (classification into face & non-face classes)
- Knowledge based methods
  - Distribution ruler of gray-value based features (e.g. gray values of eyes' area)
  - Contour ruler
  - Color
  - Movement information
  - Symmetry information

Face Detection

- Scan window over image

- Classify window as either:
  - Face
  - Non-face

Face Normalization

- Image is rotated to align the eyes (eye coordinates must be known).
- The image is scaled to make the distance between the eyes constant.
- The image is also cropped to a smaller size that is nearly just the face.
- Histogram equalization is used to smooth the distribution of gray values.
- The image is normalized so that the pixels have mean zero and standard deviation one.

Feature Extraction

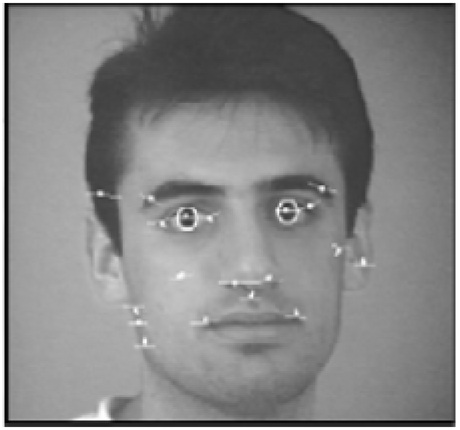
- Two main categories of feature extraction
  - Geometrical feature
    - Landmark points are found on face
    - Features are defined from these points
    - Example features like positions of eyes, nose, mouth and chin
  - Appearance based feature
    - Face is handled in holistic way
    - Features are extracted from the whole face image directly
    - Example methods like PCA or Eigenfaces and FLD/LDA or Fisherfaces

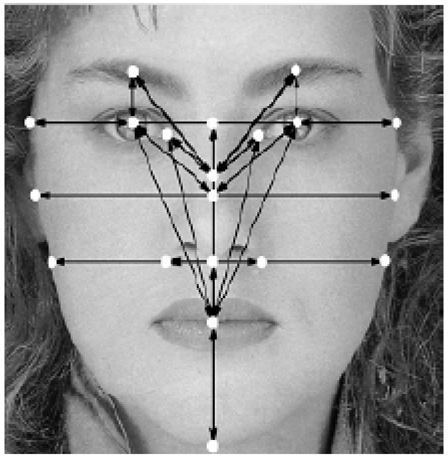
Geometrical Features

- Using geometric information of different parts of the face like eyes, nose, mouth, chin, cheekbones etc. as features of the face, for instance, distance between eyes, width of nose, etc.
- Position relationship between face parts, such as eyes, nose, mouth and chin, their shapes and sizes have strong contribution to face classification.

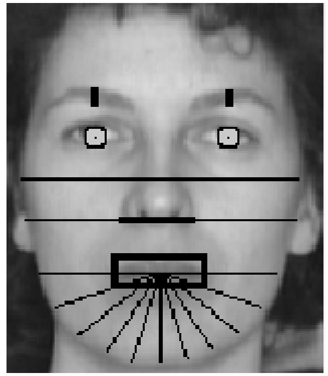
- Problem: geometrical features can not be calculated accurately, which affects the recognition capacity directly.

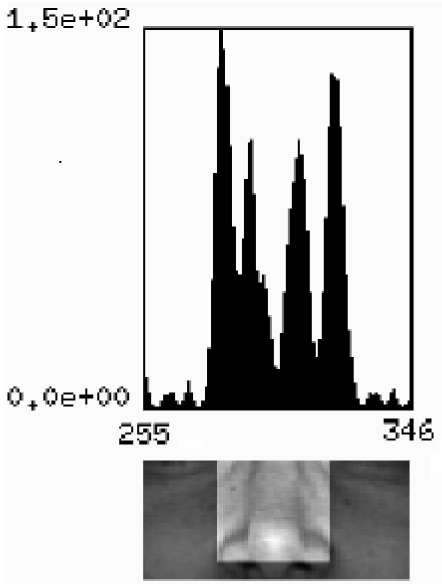
Geometrical Feature Extraction





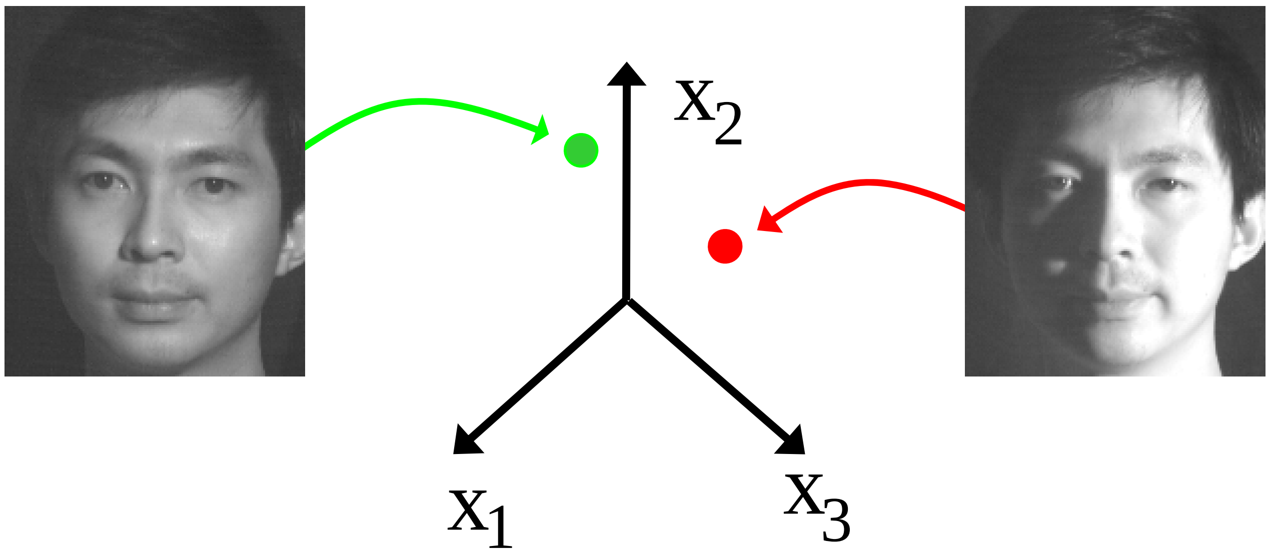
- Use vertical and horizontal integral projections of edge maps.
- The nose is found by searching for peaks in the vertical projection.
- A number of geometrical features can be defined and extracted.





Appearance based Feature Extraction

- Face image is taken as a vector in a high dimensional space and processed holistically.

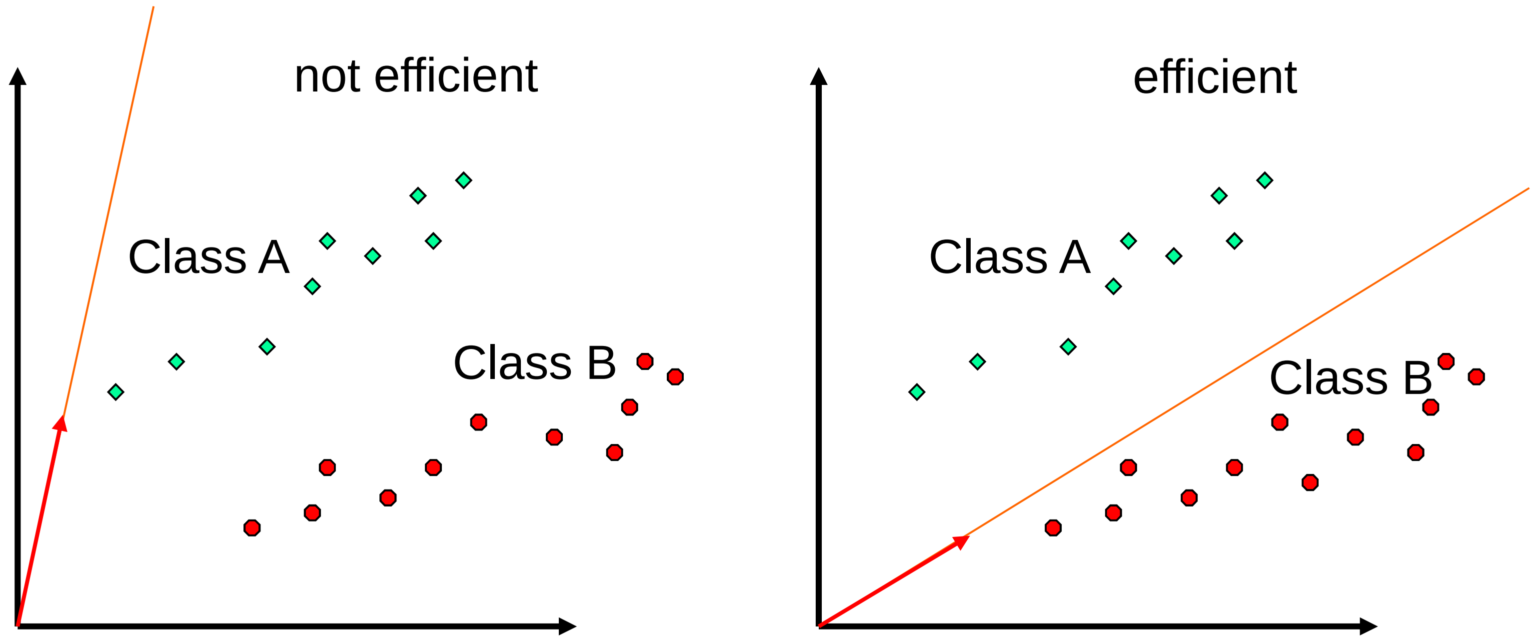


Example: Eigenfaces

- Developed in 1991 by M. Turk
- Based on PCA
- Relatively simple
- Fast
- Robust

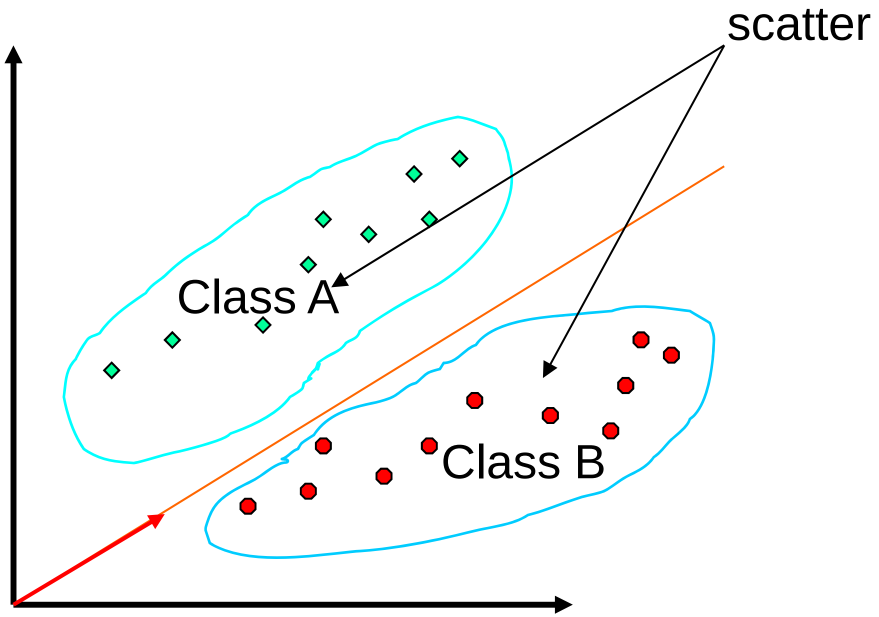
Eigenfaces

- PCA seeks directions that are efficient for representing the data



Eigenfaces

- PCA maximizes the total scatter



Eigenfaces

- PCA reduces the dimension of the data
- Speeds up the computational time

- Suppose
  - Square images ($N \times N$)
  - M is the number of images in the database
  - P is the number of persons in the database

Eigenfaces, the algorithm

- The database

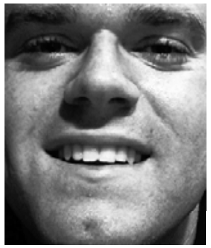
$$= \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{N^2} \end{pmatrix}$$

$$= \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N^2} \end{pmatrix}$$

$$= \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{N^2} \end{pmatrix}$$

$$= \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N^2} \end{pmatrix}$$

$$= \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N^2} \end{pmatrix}$$



$$= \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{N^2} \end{pmatrix}$$

$$= \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_{N^2} \end{pmatrix}$$

$$= \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{N^2} \end{pmatrix}$$

Eigenfaces, the algorithm

- We compute the average face

$$\vec{m} = \frac{1}{M}\begin{pmatrix} a_1 & +b_1 & +\cdots+h_1 \\ a_2 & +b_2 & +\cdots+h_2 \\ \vdots & \vdots & \vdots \\ a_{N^2} & +b_{N^2} & +\cdots+h_{N^2} \end{pmatrix}, \quad \text{where } M = 8$$

Eigenfaces, the algorithm

- Then subtract it from the training faces

$$\vec{a}_m = \begin{pmatrix} a_1 & - & m_1 \\ a_2 & - & m_2 \\ \vdots & & \vdots \\ a_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{b}_m = \begin{pmatrix} b_1 & - & m_1 \\ b_2 & - & m_2 \\ \vdots & & \vdots \\ b_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{c}_m = \begin{pmatrix} c_1 & - & m_1 \\ c_2 & - & m_2 \\ \vdots & & \vdots \\ c_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{d}_m = \begin{pmatrix} d_1 & - & m_1 \\ d_2 & - & m_2 \\ \vdots & & \vdots \\ d_{N^2} & - & m_{N^2} \end{pmatrix},$$

$$\vec{e}_m = \begin{pmatrix} e_1 & - & m_1 \\ e_2 & - & m_2 \\ \vdots & & \vdots \\ e_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{f}_m = \begin{pmatrix} f_1 & - & m_1 \\ f_2 & - & m_2 \\ \vdots & & \vdots \\ f_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{g}_m = \begin{pmatrix} g_1 & - & m_1 \\ g_2 & - & m_2 \\ \vdots & & \vdots \\ g_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{h}_m = \begin{pmatrix} h_1 & - & m_1 \\ h_2 & - & m_2 \\ \vdots & & \vdots \\ h_{N^2} & - & m_{N^2} \end{pmatrix}$$

Eigenfaces, the algorithm

- Now we build the matrix which is N^2 by M

$$A = \left[\vec{a}_m \ \vec{b}_m \ \vec{c}_m \ \vec{d}_m \ \vec{e}_m \ \vec{f}_m \ \vec{g}_m \ \vec{h}_m \right]$$

- The covariance matrix which is N^2 by N^2

$$Cov = AA^{\mathrm{T}}$$

Eigenfaces, the algorithm

- Find eigenvalues of the covariance matrix
  - The matrix is very large
  - The computational effort is very big

- We are interested in at most M eigenvalues
  - We can reduce the dimension of the matrix

Eigenfaces, the algorithm

- Compute another matrix which is M by M

$$L = A^{\mathrm{T}} A$$

- Find the M eigenvalues and eigenvectors
  - Eigenvectors of Cov and L are equivalent

- Build matrix V from the eigenvectors of L

Eigenfaces, the algorithm

- Eigenvectors of Cov are linear combination of image space with the eigenvectors of L

$$U = AV$$

- Eigenvectors represent the variation in the faces

Eigenfaces, the algorithm

- Compute for each face its projection onto the face space

$$\Omega_1 = U^{\mathrm{T}}\left(\vec{a}_m\right),\quad \Omega_2 = U^{\mathrm{T}}\left(\vec{b}_m\right),\quad \Omega_3 = U^{\mathrm{T}}\left(\vec{c}_m\right),\quad \Omega_4 = U^{\mathrm{T}}\left(\vec{d}_m\right),$$

$$\Omega_5 = U^{\mathrm{T}}\left(\vec{e}_m\right),\quad \Omega_6 = U^{\mathrm{T}}\left(\vec{f}_m\right),\quad \Omega_7 = U^{\mathrm{T}}\left(\vec{g}_m\right),\quad \Omega_8 = U^{\mathrm{T}}\left(\vec{h}_m\right)$$

Eigenfaces, the algorithm

- To recognize a face

 $= \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{N^2} \end{pmatrix}$

- Subtract the average face from it

$$\vec{r}_m = \begin{pmatrix} r_1 & - & m_1 \\ r_2 & - & m_2 \\ \vdots & & \vdots \\ r_{N^2} & - & m_{N^2} \end{pmatrix}$$

Eigenfaces, the algorithm

- Compute its projection onto the face space

$$\Omega = U^{\mathrm{T}} \left(\vec{r}_m \right)$$

- Compute the distance in the face space between the face and all known faces

$$\varepsilon_i^2 = \left\| \Omega - \Omega_i \right\|^2 \quad for\ i = 1..M$$

Eigenfaces, the algorithm

- Identification
  - Set a threshold *T*.
  - If all $\varepsilon_i \geq T$, then it's not in the database.
  - If $\theta = \min\{\varepsilon_i\} < T$, then it's in the database.
  - Suppose $\varepsilon_k = \theta$, then it's the k^{th} face.

Eigenfaces, the algorithm

- Problems with eigenfaces
  - Different illumination
  - Different head pose
  - Different alignment
  - Different facial expression