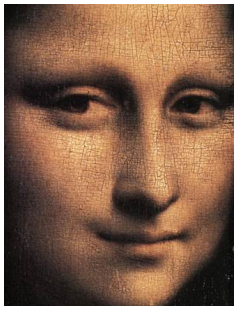
# Face Detection and Recognition

Reading: Chapter 18.10
and, optionally,
"Face Recognition using Eigenfaces" by M. Turk and A. Pentland



## Face Recognition Problem

database

Query face

## Face Verification Problem

- Face Verification (1:1 matching)

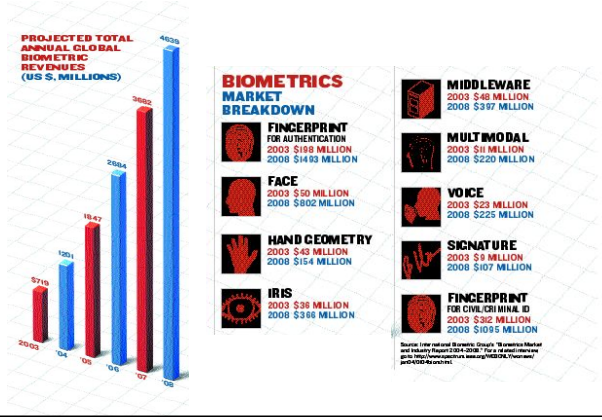
- Face Recognition (1:N matching)

## Application: Access Control


www.viisage.com

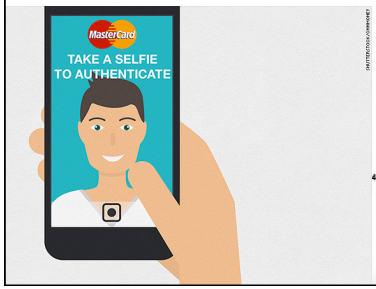
www.visionics.com

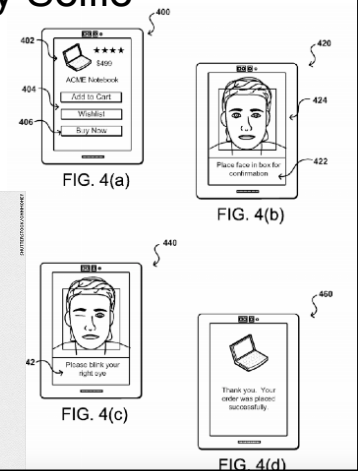
# Biometric Authentication




# Pay by Selfie

Amazon, Mastercard, Alibaba developing methods

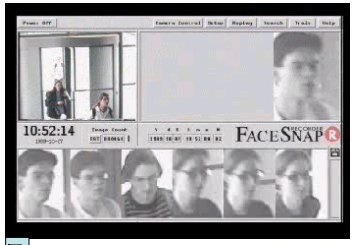



# Application: Video Surveillance

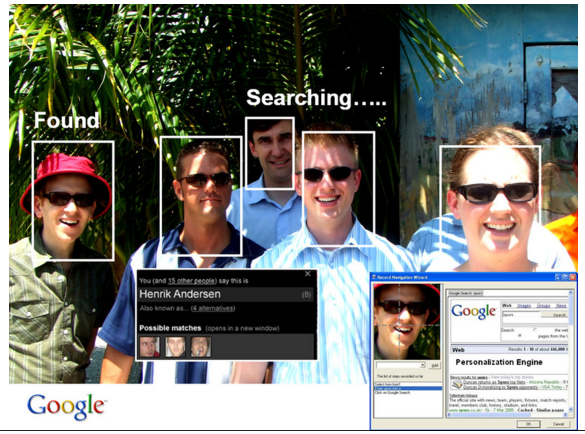
**Face Scan at Airports**

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.

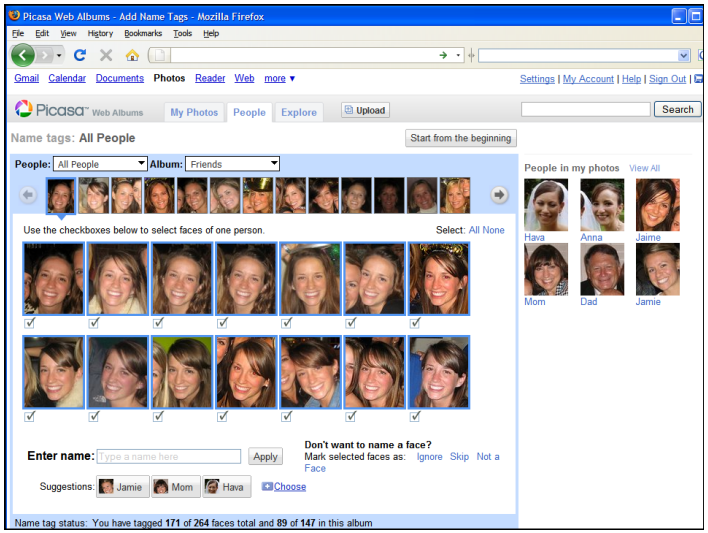



www.facesnap.de

# Application: Autotagging Photos in Facebook, Flickr, Picasa, iPhoto, …

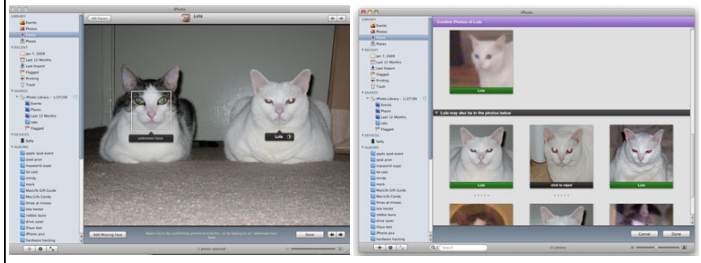



Google




# iPhoto

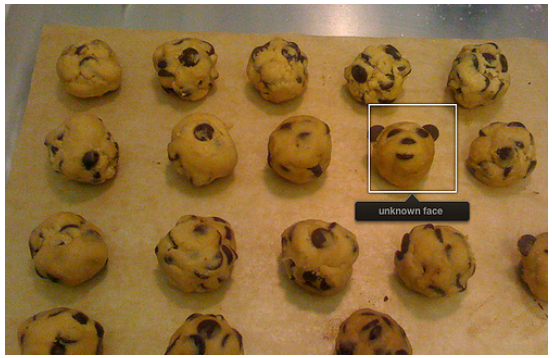
- Can be trained to recognize pets!



http://www.maclife.com/article/news/iphotos_faces_recognizes_cats

# iPhoto

- Things iPhoto thinks are faces




# Why is Face Recognition Hard?

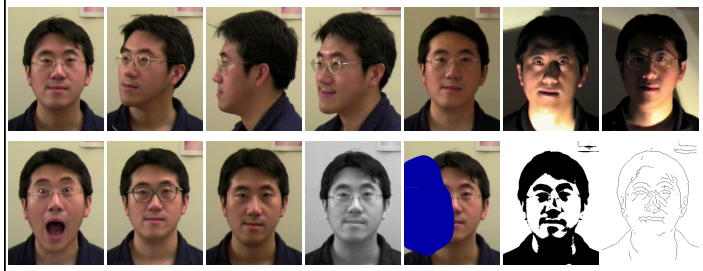
**The many faces of Madonna**

## Recognition should be **Invariant** to

- Lighting variation
- Head pose variation
- Different expressions
- Beards, disguises
- Glasses, occlusion
- Aging, weight gain
- …

## Intra-class Variability

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



## Inter-class Similarity

- Different people may have very similar appearance

www.marykateandashley.com

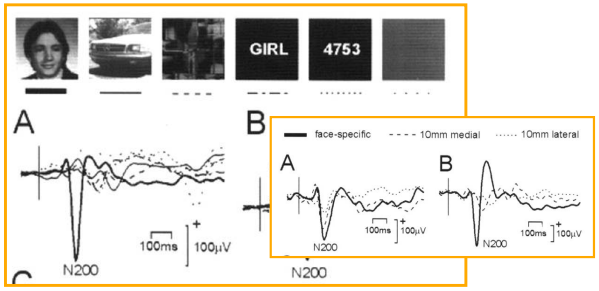
Twins

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

Father and son

## Face Detection in Humans

There are processes for face detection and recognition in the brain



# Blurred Faces ***are*** Recognizable


# Blurred Faces ***are*** Recognizable


Michael Jordan, Woody Allen, Goldie Hawn, Bill Clinton, Tom Hanks, Saddam Hussein, Elvis Presley, Jay Leno, Dustin Hoffman, Prince Charles, Cher, and Richard Nixon. The average recognition rate at this resolution is one-half.

# Upside-Down Faces are Recognizable

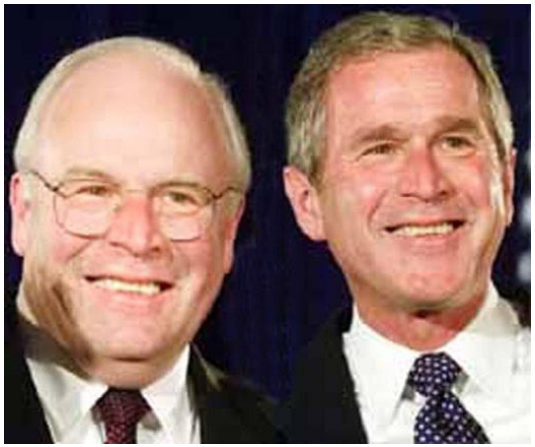
The “Margaret Thatcher Illusion”, by Peter Thompson

by L. Harmon and B. Julesz, *Scientific American*,1973

# Illumination and Shading Affect Interpretation

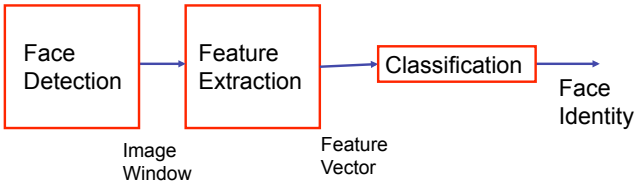
# Context is Important

P. Sinha and T. Poggio, I think I know that face, *Nature* **384**, 1996, 404.

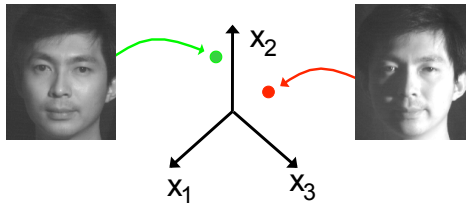


P. Sinha and T. Poggio, Last but not least, *Perception* **31**, 2002, 133.

# Face Recognition Architecture



# Image as a Feature Vector

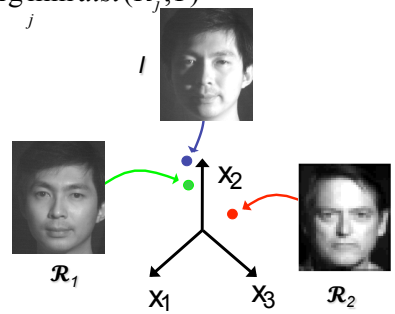



- Consider an *n*-pixel image to be a point in an *n*-dimensional "image space," $\mathbf{x} \in \mathcal{R}^n$
- Each pixel value is a coordinate of **x**
- Preprocess images so faces are cropped and (roughly) aligned (position, orientation, and scale)

# Nearest Neighbor Classifier

$\{ \mathcal{R}_j \}$ is a set of training images of frontal faces

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

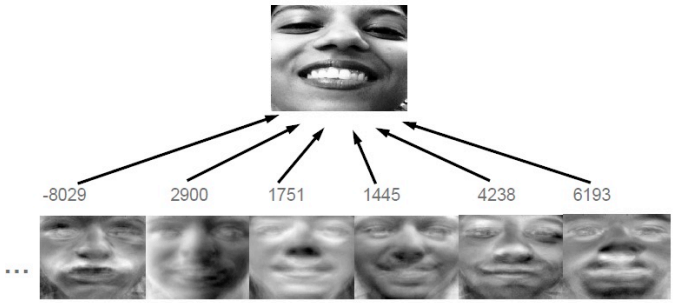



# Key Idea

- Expensive to compute nearest neighbor when each image is big (*n* dimensional space)
- Not all images are very likely – especially when we know that every image contains a face. That is, **images of faces are highly correlated, so compress them into a low-dimensional, linear subspace** that retains the key appearance characteristics of people's faces

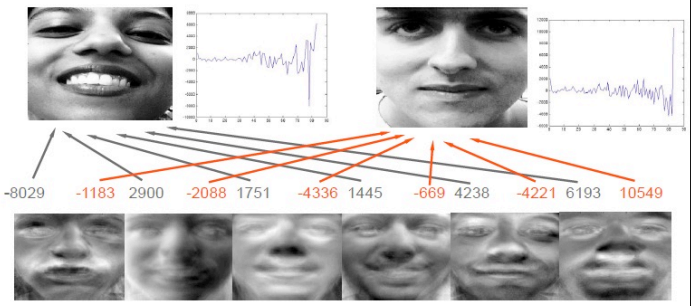
# Eigenface Representation

- Each face image is represented by a weighted (linear) combination of a small number of "component" or "basis" faces
- Each of these basis faces is called an "eigenface"



## Eigenface Representation

- These basis faces can be weighted differently to represent ***any*** face
- So, a vector of weights represents each face




## Learning Eigenfaces

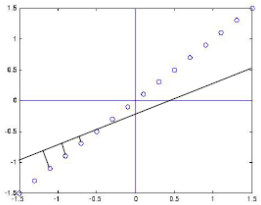
- How do we pick the set of basis faces?
- We take a set of real training faces
- Find the set of basis faces that best represent the ***differences*** between the training faces
- Use a statistical criterion for quantifying the "best representation of the differences"

## Eigenfaces (Turk and Pentland, 1991)

- The set of training face images is clustered in a "subspace" of the set of all images
- **Training Phase**
  - Find best subspace to reduce the dimensionality
  - Transform all training images into the subspace
- **Testing Phase**
  - Transform test image into the subspace and use a nearest-neighbor classifier to label it

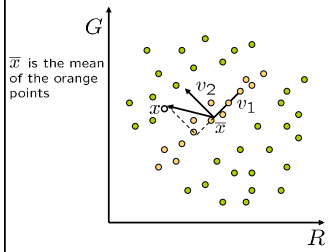
## Linear Subspaces

- Suppose we have points in 2D and we have a line through that space



- We can project each point onto that 1D line
- Represent a point by its position on the line

# Linear Subspaces

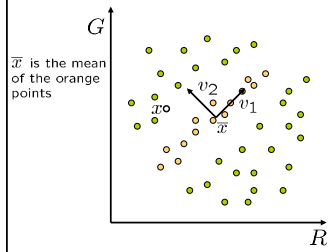



convert $x$ into $v_1$, $v_2$ coordinates:
(assume $\mathbf{v}_1$ and $\mathbf{v}_2$ are orthogonal unit vectors)

$$\mathbf{x} \rightarrow ((\mathbf{x} - \overline{x}) \cdot \mathbf{v_1}, (\mathbf{x} - \overline{x}) \cdot \mathbf{v_2})$$

- What does the $v_2$ coordinate measure?
  - Distance to line defined by $v_1$
- What does the $v_1$ coordinate measure?
  - Distance from $\overline{x}$ on line defined by $v_1$

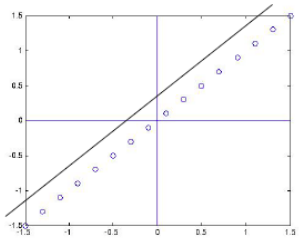
# Dimensionality Reduction

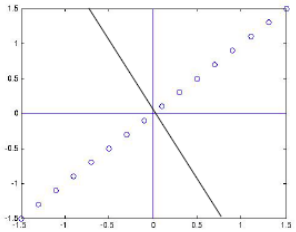



- We can represent the orange points well with *only* their $v_1$ coordinates (since $v_2$ coordinates are all ≈ 0)
- This makes it much cheaper to store and compare points
- A bigger deal for higher dimensional problems

# Linear Subspaces

Some lines will represent the data well and others not, depending on how well the projection separates the data points





# Principal Component Analysis (PCA)

- Problems arise when performing recognition in a high-dimensional space ("**curse of dimensionality**")
- Significant improvements can be achieved by first mapping the data into a ***lower-dimensional sub*space**

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_N \end{bmatrix} \text{--> } reduce\ dimensionality \text{--> } y = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} \quad (K << N)$$

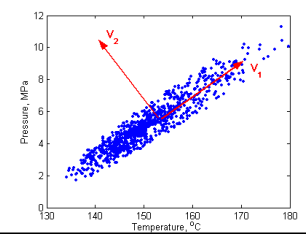
- **The goal of PCA is to reduce the dimensionality of the data while retaining the *largest variations* present in the original data**

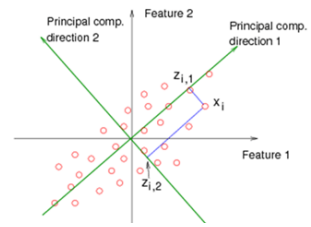
## Principal Component Analysis (PCA)

- Dimensionality reduction implies information will be lost
- How to determine the best lower dimensional subspace?
  - Maximize information content in the compressed data by **finding a set of *k* orthogonal vectors that account for as much of the data's variance as possible**
  - Best dimension = direction in *n*-D with *max variance*
  - 2nd best dimension = direction orthogonal to first with *max variance*

## Principal Component Analysis (PCA)

- **Geometric interpretation**
  - PCA projects the data along the directions where the data varies the ***most***
  - The vectors define a new coordinate system in which to represent each image




- The covariance between two pixels expresses how correlated they are (i.e., to what degree they "co-vary")
- The *n* x *n* covariance matrix for an image with *n* pixels expresses how correlated each pair of pixels is
- The eigenvectors of this matrix with the largest eigenvalues correspond to the pixels in which the data varies the most

## Eigenvectors

- An **eigenvector** is a vector, $u$, that satisfies

$$\lambda_i u_i = \mathbf{C} u_i$$

where **C** is a square matrix, and $\lambda$ is a scalar called the **eigenvalue**

- Example (for 2D data):

$$\mathbf{C} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \quad u = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

- So eigenvalue $\lambda$ = 4 for this eigenvector, $u$

## Principal Component Analysis (PCA)

- The best low-dimensional space can be determined by the "best" eigenvectors of the covariance matrix of the data, i.e., the eigenvectors corresponding to the largest eigenvalues – also called "**principal components**"
- Can be efficiently computed using the Singular Value Decomposition (SVD) algorithm

## Algorithm

- Each input image, $X_i$ , is an $n$-D column vector of all pixel values (in raster order)
- Compute "average face" image $A$ from all $M$ training images of all people:

$$A = \frac{1}{M}\sum_{i=1}^{M} X_i$$

- Normalize each training image, $X_i$, by subtracting the average face:

$$Y_i = X_i - A$$

## Algorithm

- Stack all training images together

$n$ x $M$ matrix

$$\mathbf{Y} = [Y_1 Y_2 \ldots Y_M]$$

- Compute the Covariance Matrix **C**:

$$\mathbf{C} = \mathbf{Y}\mathbf{Y}^{\mathrm{T}} = \frac{1}{M}\sum_i Y_i Y_i^{\mathrm{T}}$$

## Algorithm

- Compute eigenvalues and eigenvectors of **C** by solving

$$\lambda_i u_i = \mathbf{C} u_i$$

where the eigenvalues are

$$\lambda_1 > \lambda_2 > \ldots > \lambda_n$$

and the corresponding eigenvectors are

$u_1, u_2, \ldots, u_n$

# Algorithm

- Each $u_i$ is an $n$ x 1 eigenvector called an "eigenface" (to be cute!)
- Each $u_i$ is a vector/direction in "face space"

$$Y_i = w_1 u_1 + w_2 u_2 + \ldots + w_n u_n$$

$$X_i = \sum_{i=1}^{n} w_i u_i + A$$

- Image can be *exactly* reconstructed by a linear combination of *all* eigenvectors

# Algorithm

- Reduce dimensionality by using only the **best *k*** << *n* eigenvectors (i.e., the ones corresponding to the largest *k* eigenvalues

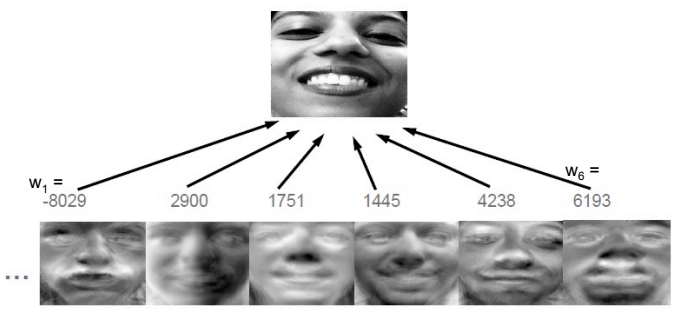
$$X_i \approx \sum_{i=1}^{k} w_i u_i + A$$

- Each image $X_i$ is approximated by a set of *k* "weights" $[w_{i1}, w_{i2}, \ldots, w_{ik}] = \boldsymbol{W}_i$ where

$$w_{ij} = u_j^{\mathrm{T}} (X_i - A)$$

# Eigenface Representation

Each face image is represented by a weighted combination of a small number of "component" or "basis" faces




# Using Eigenfaces

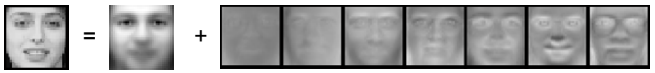
- **(Approximate) Reconstruction** of an image of a face from a set of weights
- **Recognition** of a person from a new face image

# Face Image Reconstruction

- Face **X** in "face space" coordinates:

$$\mathbf{x} \rightarrow [\mathbf{u}_1^{\mathrm{T}}(\mathbf{x}-\mu), \ldots, \mathbf{u}_k^{\mathrm{T}}(\mathbf{x}-\mu)]$$
$$= \quad w_1, \ldots, w_k$$

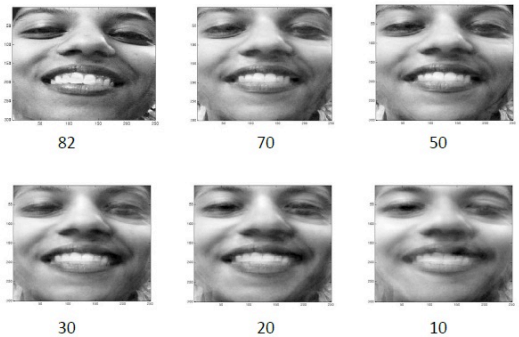
- Reconstruction:



$\hat{\mathbf{x}}$ = $A$ + $w_1u_1 + w_2u_2 + w_3u_3 + w_4u_4 + \ldots$

# Reconstruction

The more eigenfaces you use, the better the reconstruction, but even a small number gives good quality for matching




# Eigenfaces Recognition Algorithm

**Modeling (Training Phase)**

1. Given a collection of labeled training images
2. Compute mean image, $A$
3. Compute $k$ eigenvectors, $u_1, \ldots, u_k$, of covariance matrix corresponding to $k$ largest eigenvalues
4. Project each training image, $X_i$, to a point in $k$-dimensional "face space:"

   $$\text{for } j = 1, \ldots, k \text{ compute } w_{ij} = u_j^{\mathrm{T}}(X_i - A)$$

   $X_i$ projects to $\boldsymbol{W}_i = [w_{i1}, w_{i2}, \ldots, w_{ik}]$

# Eigenfaces Algorithm

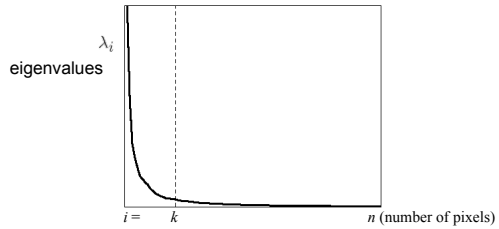
**Recognition (Testing Phase)**

1. Given a test image, $G$, project it into face space

   $$\text{for } j = 1, \ldots, k \text{ compute } w_j = u_j^{\mathrm{T}}(G - A)$$

2. Classify it as the class (person) that is closest to it (as long as its distance to the closest person is "close enough")

# Choosing *k*




- How many eigenfaces to use?
- Look at the decay of the eigenvalues
  - the eigenvalue tells you the amount of variance "in the direction" of that eigenface
  - ignore eigenfaces with low variance

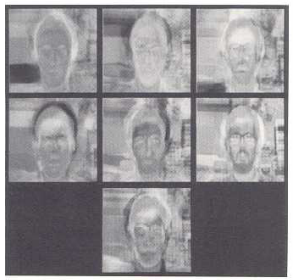
# Example: Training Images

Note: Faces must be approximately registered (translation, rotation, size, pose)!

[ Turk & Pentland, 2001]

# Eigenfaces

Average Image, *A*



7 eigenface images

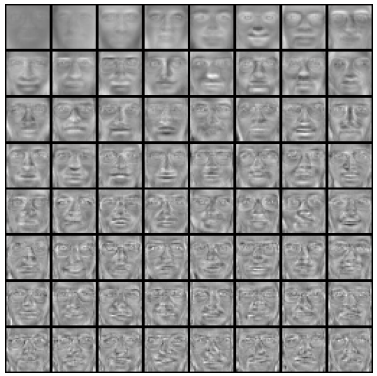
# Example

Training images

# Example

Top eigenvectors: $u_1,\ldots u_k$

Average: A



# Experimental Results

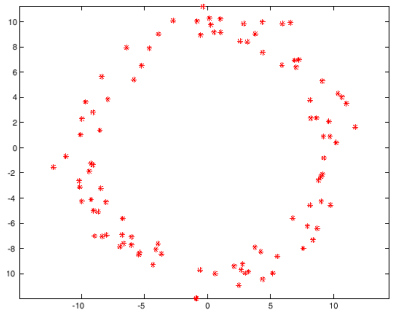
- Training set: 7,562 images of approximately 3,000 people
- $k$ = 20 eigenfaces computed from a sample of 128 images
- Test set accuracy on 200 faces was 95%

# Difficulties with PCA

- Projection may suppress important details
  - smallest variance directions may ***not*** be unimportant
- Method does not take discriminative task into account
  - we want to compute features that allow good discrimination
  - *not* necessarily the same as largest variance

# Limitations

- PCA assumes that the data has a Gaussian distribution




The shape of this dataset is not well described by its principal components

# Limitations

- Background (de-emphasize the outside of the face – e.g., by multiplying the input image by a 2D Gaussian window centered on the face)
- Lighting conditions (performance degrades with light changes)
- Scale (performance decreases quickly with changes to head size); possible solutions:
  - multi-scale eigenspaces
  - scale input image to multiple sizes
- Orientation (performance decreases but not as fast as with scale changes)
  - plane rotations can be handled
  - out-of-plane rotations are more difficult to handle

# Limitations

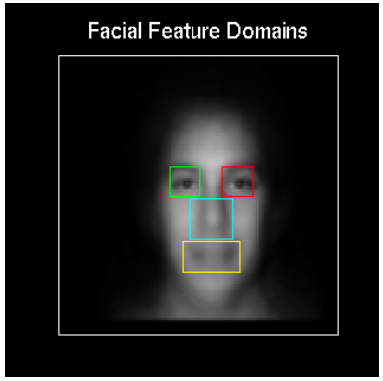
- Not robust to misalignment

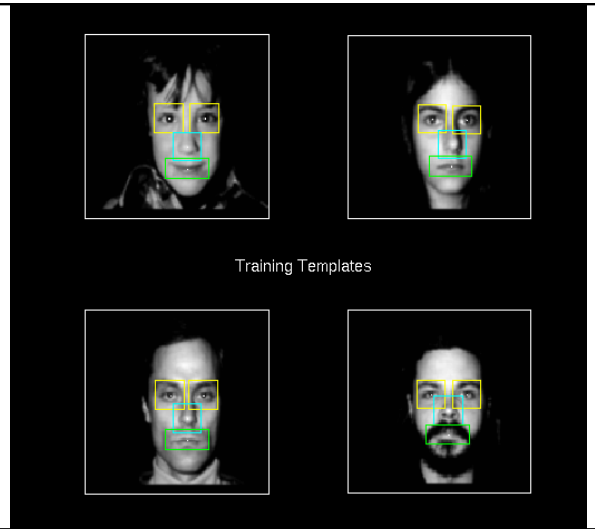


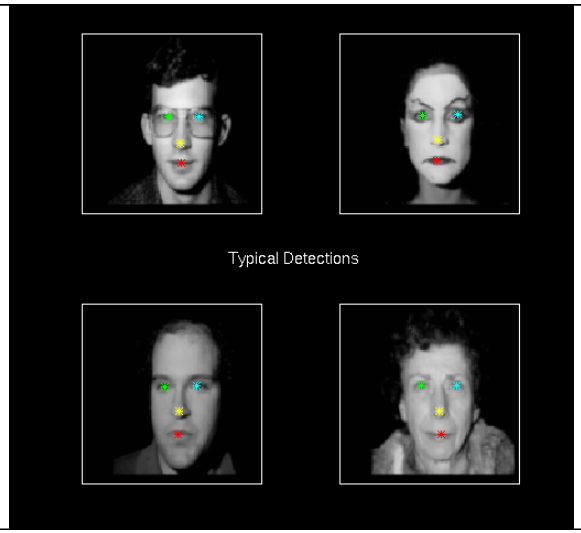


# Extension: Eigenfeatures

- Describe and encode a set of facial features: eigeneyes, eigennoses, eigenmouths
- Use for detecting facial features






# Recognition using Eigenfeatures