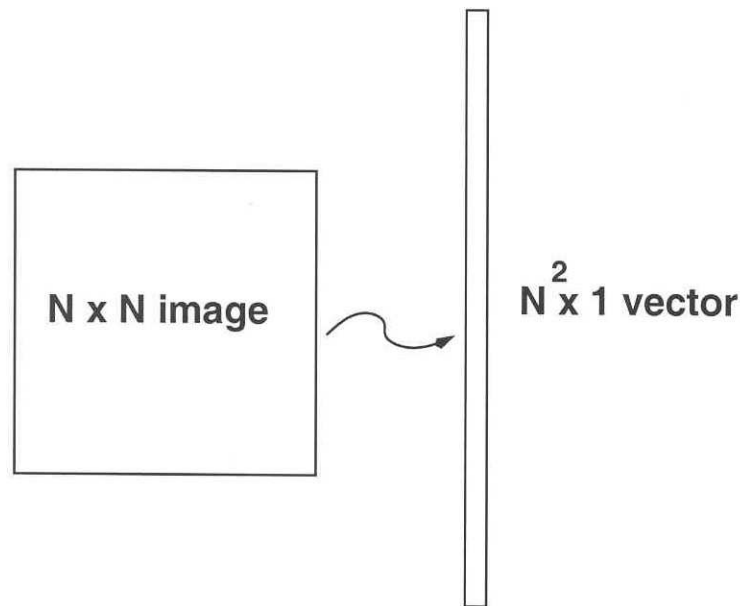


# Eigenfaces for Face Detection/Recognition

(M. Turk and A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991, hard copy)

## • Face Recognition

- The simplest approach is to think of it as a template matching problem:



- Problems arise when performing recognition in a high-dimensional space.
- Significant improvements can be achieved by first mapping the data into a *lower-dimensionality* space.
- How to find this lower-dimensional space?

## • Main idea behind eigenfaces

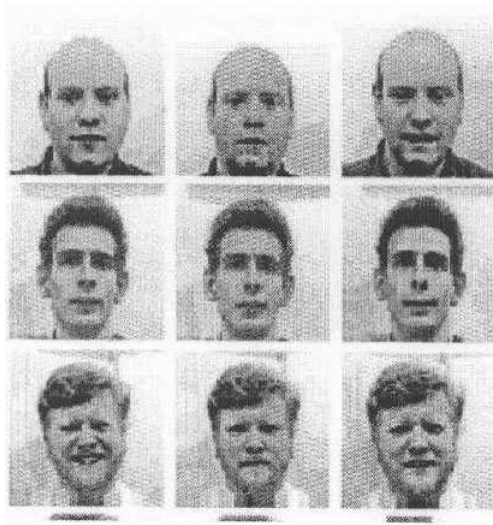
- Suppose  $\Gamma$  is an  $N^2 \times 1$  vector, corresponding to an  $N \times N$  face image  $I$ .
- The idea is to represent  $\Gamma$  ( $\Phi = \Gamma$  - mean face) into a low-dimensional space:

$$\hat{\Phi} - mean = w_1 u_1 + w_2 u_2 + \cdots w_K u_K \quad (K \ll N^2)$$

## Computation of the eigenfaces

Step 1: obtain face images  $I_1, I_2, \dots, I_M$  (training faces)

(**very important**: the face images must be *centered* and of the same *size*)



Step 2: represent every image  $I_i$  as a vector  $\Gamma_i$

Step 3: compute the average face vector  $\Psi$ :

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

Step 4: subtract the mean face:

$$\Phi_i = \Gamma_i - \Psi$$

Step 5: compute the covariance matrix  $C$ :

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (N^2 \times N^2 \text{ matrix})$$

$$\text{where } A = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_M] \quad (N^2 \times M \text{ matrix})$$

Step 6: compute the eigenvectors  $u_i$  of  $AA^T$

The matrix  $AA^T$  is very large --> not practical !!

Step 6.1: consider the matrix  $A^T A$  ( $M \times M$  matrix)

Step 6.2: compute the eigenvectors  $v_i$  of  $A^T A$

$$A^T A v_i = \mu_i v_i$$

What is the relationship between  $u_i$  and  $v_i$ ?

$$A^T A v_i = \mu_i v_i \Rightarrow AA^T A v_i = \mu_i A v_i \Rightarrow$$

$$CA v_i = \mu_i A v_i \text{ or } C u_i = \mu_i u_i \text{ where } u_i = A v_i$$

Thus,  $AA^T$  and  $A^T A$  have the same eigenvalues and their eigenvectors are related as follows:  $u_i = A v_i$  !!

Note 1:  $AA^T$  can have up to  $N^2$  eigenvalues and eigenvectors.

Note 2:  $A^T A$  can have up to  $M$  eigenvalues and eigenvectors.

Note 3: The  $M$  eigenvalues of  $A^T A$  (along with their corresponding eigenvectors) correspond to the  $M$  *largest* eigenvalues of  $AA^T$  (along with their corresponding eigenvectors).

Step 6.3: compute the  $M$  best eigenvectors of  $AA^T$ :  $u_i = A v_i$

**(important:** normalize  $u_i$  such that  $\|u_i\| = 1$ )

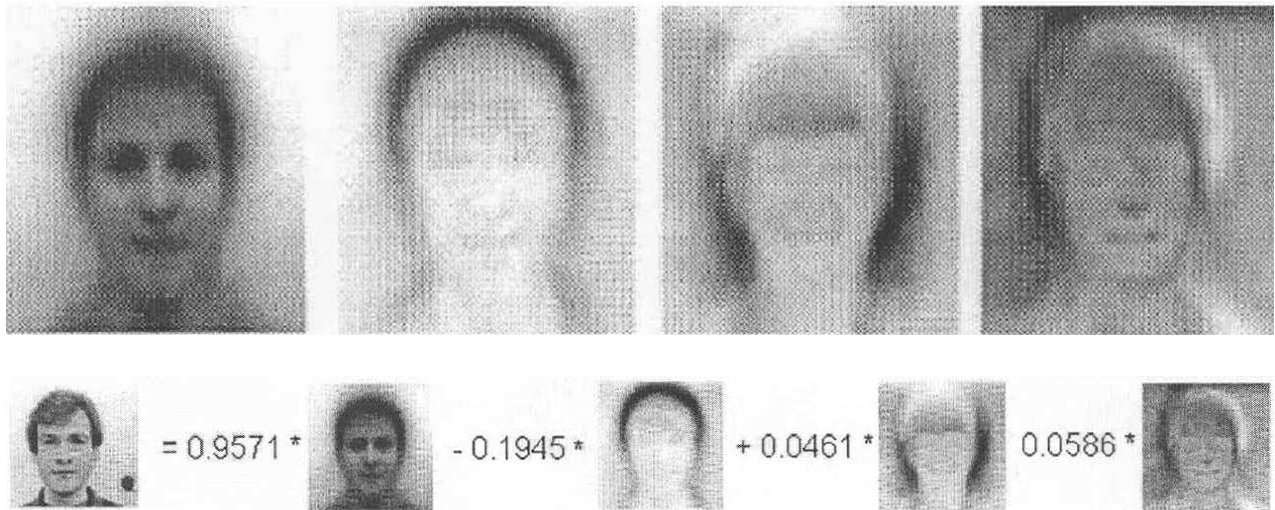
Step 7: keep only  $K$  eigenvectors (corresponding to the  $K$  largest eigenvalues)

## Representing faces onto this basis

- Each face (minus the mean)  $\Phi_i$  in the training set can be represented as a linear combination of the best  $K$  eigenvectors:

$$\hat{\Phi}_i - \text{mean} = \sum_{j=1}^K w_j u_j, \quad (w_j = u_j^T \Phi_i)$$

(we call the  $u_j$ 's *eigenfaces*)



- Each normalized training face  $\Phi_i$  is represented in this basis by a vector:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \dots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \dots, M$$

## Face Recognition Using Eigenfaces

- Given an unknown face image  $\Gamma$  (centered and of the same size like the training faces) follow these steps:

Step 1: normalize  $\Gamma$ :  $\Phi = \Gamma - \Psi$

Step 2: project on the eigenspace

$$\hat{\Phi} = \sum_{i=1}^K w_i u_i \quad (w_i = u_i^T \Phi)$$

Step 3: represent  $\Phi$  as:  $\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_K \end{bmatrix}$

Step 4: find  $e_r = \min_l \|\Omega - \Omega^l\|$

Step 5: if  $e_r < T_r$ , then  $\Gamma$  is recognized as face  $l$  from the training set.

- The distance  $e_r$  is called distance within the face space (difs)

Comment: we can use the common Euclidean distance to compute  $e_r$ , however, it has been reported that the *Mahalanobis distance* performs better:

$$\|\Omega - \Omega^k\| = \sum_{i=1}^K \frac{1}{\lambda_i} (w_i - w_i^k)^2$$

(variations along all axes are treated as equally significant)

## Face Detection Using Eigenfaces

- Given an unknown image  $\Gamma$

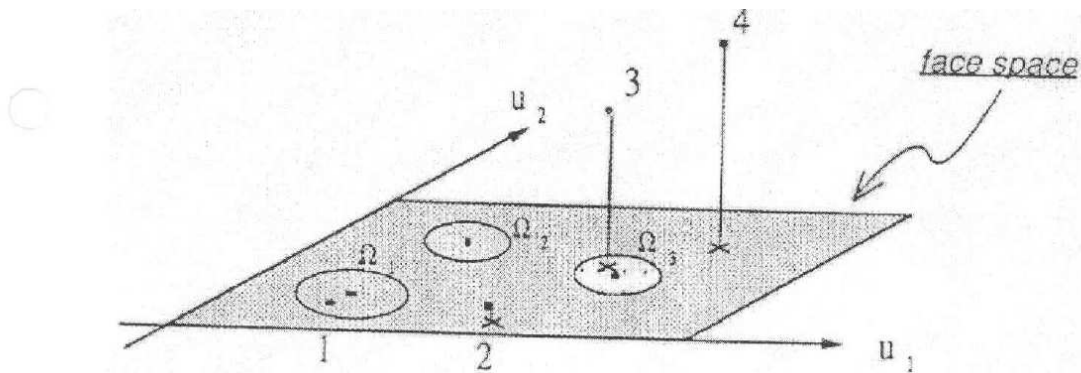
Step 1: compute  $\Phi = \Gamma - \Psi$

Step 2: compute  $\hat{\Phi} = \sum_{i=1}^K w_i u_i$  ( $w_i = u_i^T \Phi$ )

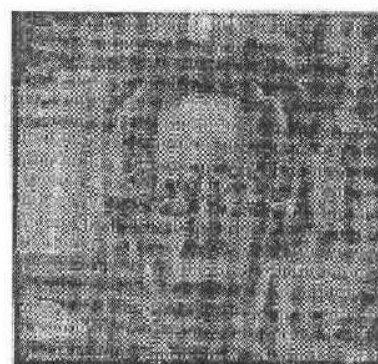
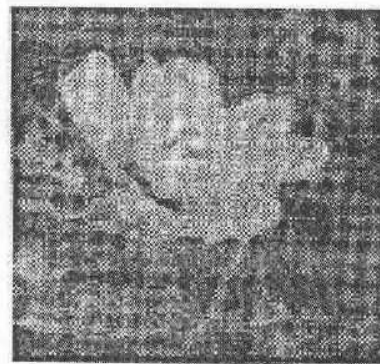
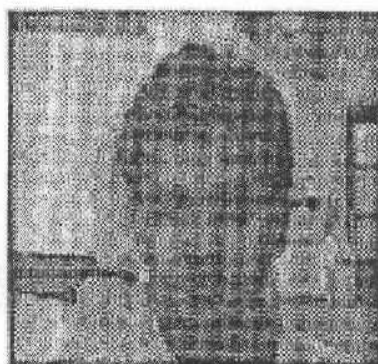
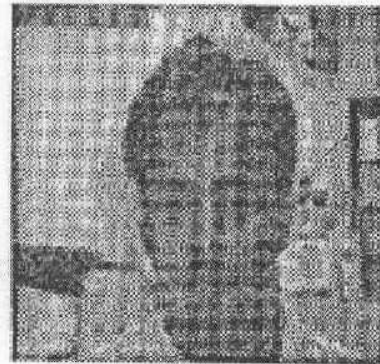
Step 3: compute  $e_d = \|\Phi - \hat{\Phi}\|$

Step 4: if  $e_d < T_d$ , then  $\Gamma$  is a face.

- The distance  $e_d$  is called distance from face space (dffb)



- Reconstruction of faces and non-faces

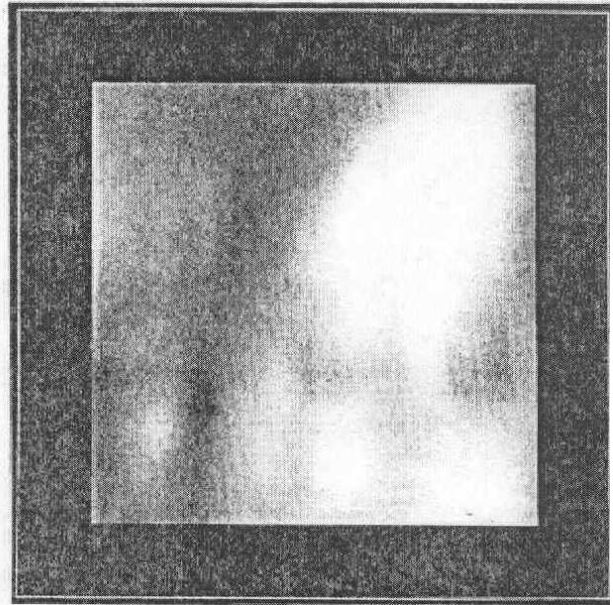
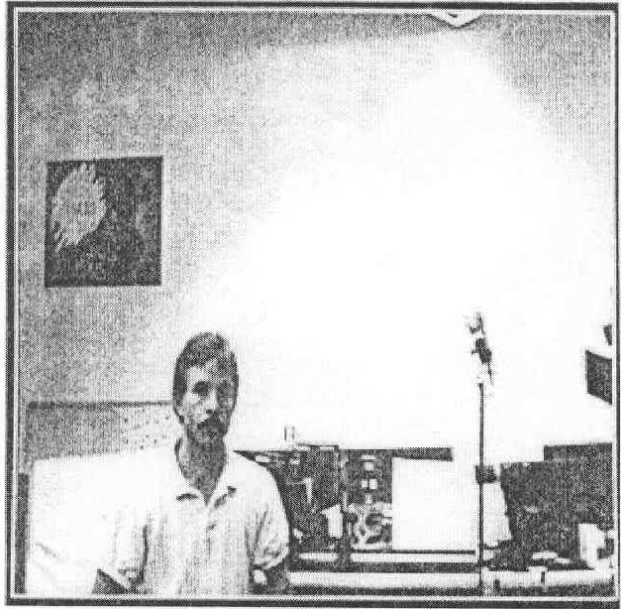


## • Time requirements

- About 400 msec (Lisp, Sun4, 128x128 images)

## • Applications

- Face detection, tracking, and recognition



## • Problems

- Background (deemphasize 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 the head size)
  - \* multiscale 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 more difficult to handle



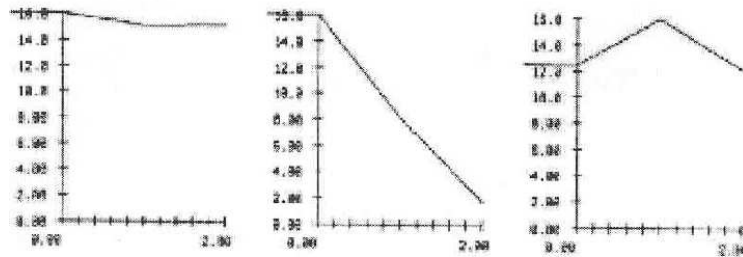
## • Experiments

- 16 subjects, 3 orientations, 3 sizes
- 3 lighting conditions, 6 resolutions (512x512 ... 16x16)
- Total number of images: 2,592



## Experiment 1

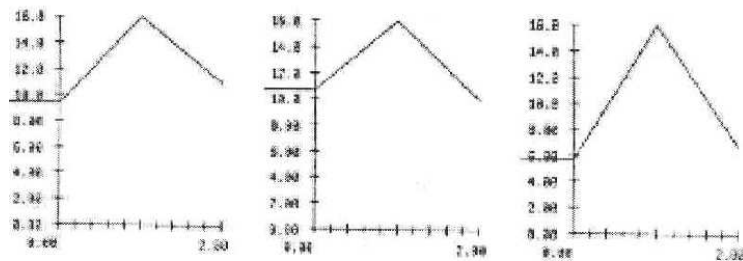
- \* Used various sets of 16 images for training
- \* One image/person, taken under the same conditions
- \* Eigenfaces were computed offline (7 eigenfaces were used)
- \* Classify the rest images as one of the 16 individuals
- \* No rejections (i.e., no threshold for *difs*)



(a)

(b)

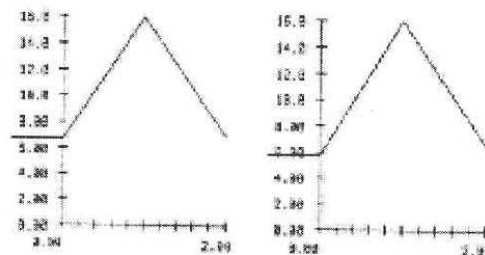
(c)



(d)

(e)

(f)



- Performed a large number of experiments and averaged the results:

96% correct averaged over light variation  
85% correct averaged over orientation variation  
64% correct averaged over size variation

## Experiment 2

- They considered rejections (i.e., by thresholding *difs*)
- There is a tradeoff between correct recognition and rejections.
- Adjusting the threshold to achieve 100% recognition accuracy resulted in:
  - \* 19% rejections while varying lighting
  - \* 39% rejections while varying orientation
  - \* 60% rejections while varying size

## Experiment 3

- Reconstruction using partial information

