Face Recognition Using Eigenfaces

Mathew A. Turk and Alex P. Pentland

Presenter: Polina Golland

Problem Definition

- Recognize faces vs. non-faces
- Recognize faces of a particular person vs. faces of other people

- Do it fast

Basic Idea

- PCA on face images
  - Face images lie in a low dimensional space
  - Images of the same person are close to each other
  - Images of different people are farther away

Recognition

- Given a new image, project onto face space
  - If the residual is too high, it’s not a face
  - If the projection is close to one of the “prototypes”, assign it to that class
  - Otherwise, it’s a new face

Illustration of Face Space

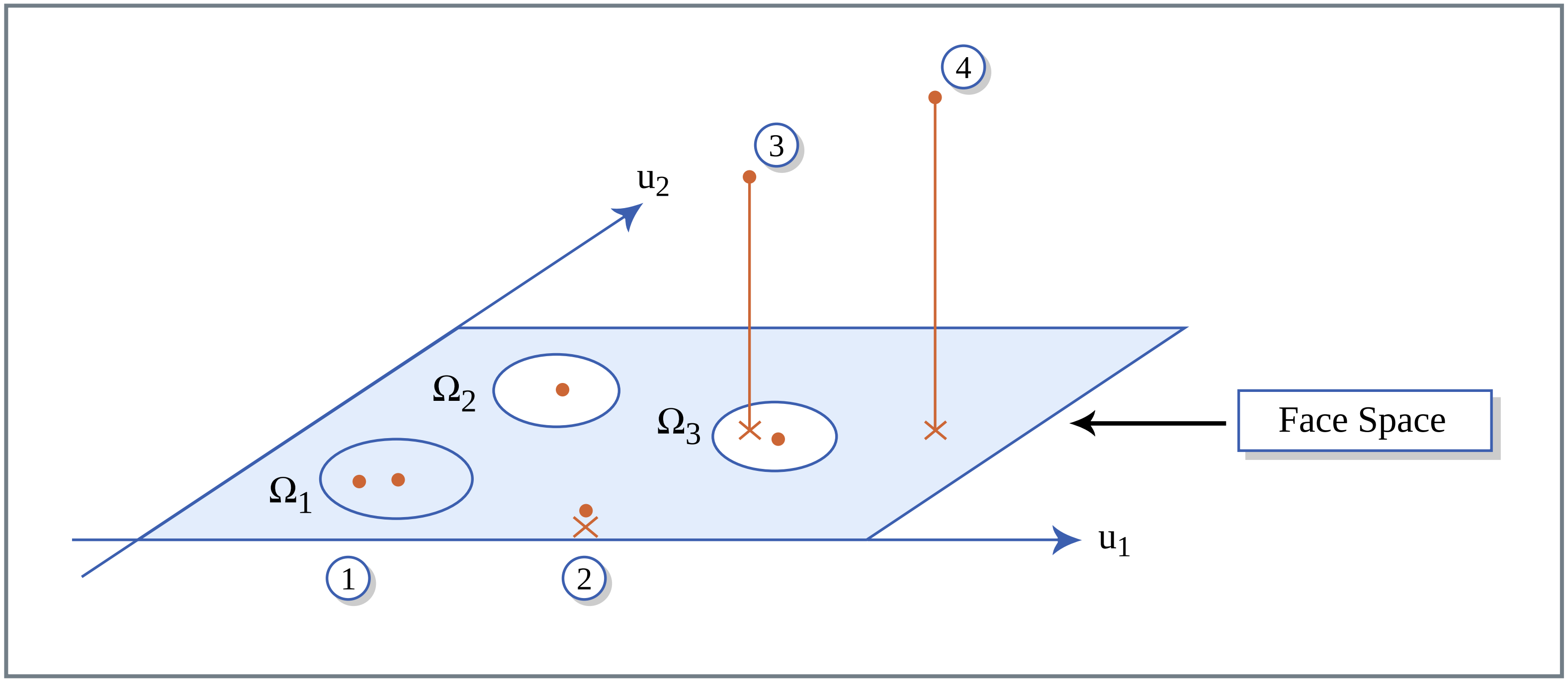

Figure by MIT OCW.

Training

- Input: *M* input images in a vector form $\mathbf{\Gamma}_i$

- Mean face $\mathbf{\Psi} = \mathrm{M}^{-1} \sum \mathbf{\Gamma}_i$

- Zero-mean data $\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{\Psi}$
- Zero-mean data matrix $A = [\mathbf{\Phi}_1\ \mathbf{\Phi}_2 \ldots \mathbf{\Phi}_M]$

- Covariance matrix $C = \mathrm{M}^{-1} \sum \mathbf{\Phi}_i \mathbf{\Phi}_i^{\mathrm{T}} = AA^{\mathrm{T}}$

Training cont'd

- Eigenfaces $\mathbf{u}_k$ are the eigenvectors of C

- Keep a small number (M') of eigenfaces
  - In the experiments, $M = 16$, $M' = 7$

- Resulting model:
  - Mean face $\boldsymbol{\Psi}$
  - M' eigenfaces $\mathbf{u}_k$ ($k = 1,\ldots,M'$)

Detection

- Given a new image $\mathbf{\Gamma}$, compute its weights

$$\omega_k = \mathbf{u}_k^{\mathrm{T}}(\mathbf{\Gamma} - \mathbf{\Psi})$$

- The face is represented by the weight vector

$$\mathbf{\Omega} = [\omega_1\ \omega_2\ \ldots\ \omega_{M'}]^{\mathrm{T}}$$

- If $\left\| \mathbf{\Gamma} - \mathbf{\Psi} - \sum \omega_k \mathbf{u}_k \right\| < \theta_{\varepsilon}$, this is a face image.

- If $\left\| \mathbf{\Omega} - \mathbf{\Omega}_n \right\| < \theta_{\delta}$, this is a face from class *n*.

Illustration of Face Space

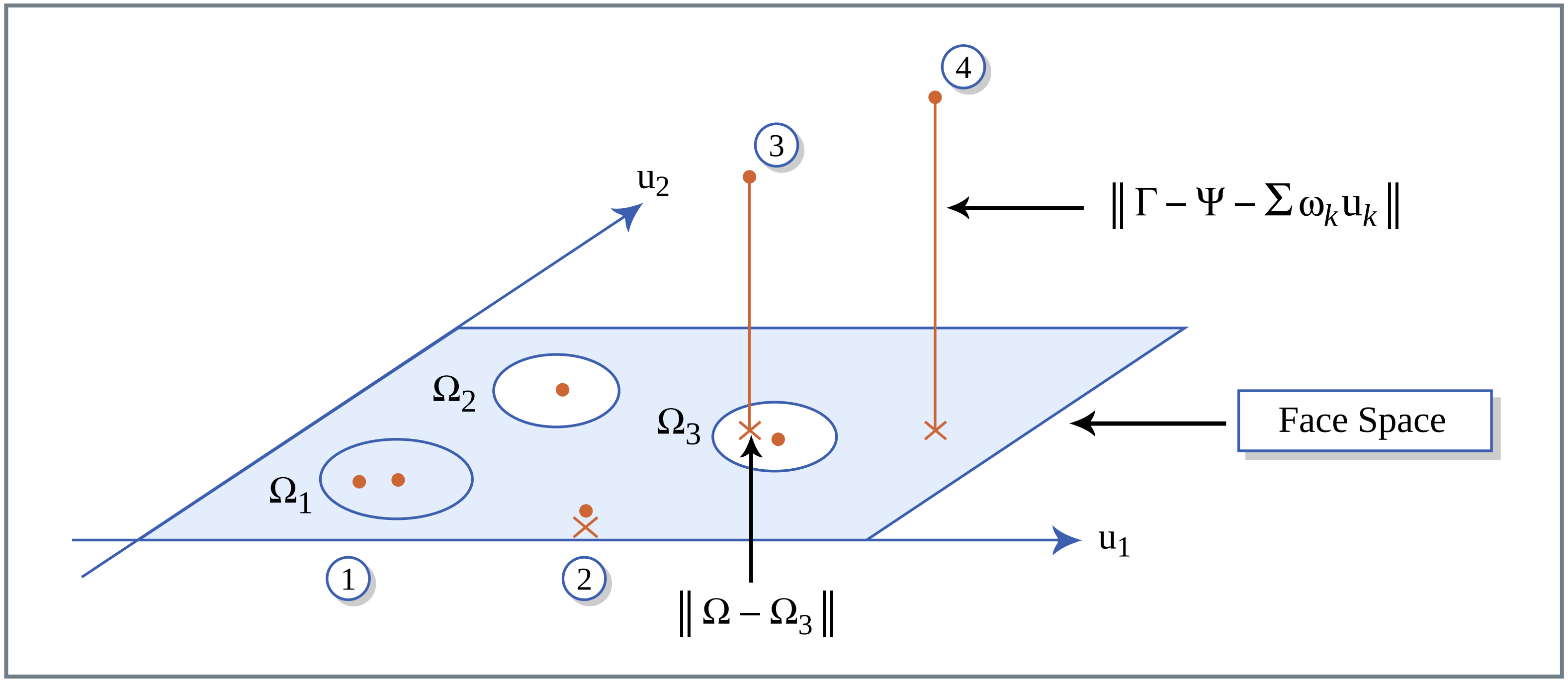

Figure by MIT OCW.

Examples of projections

Image removed due to copyright considerations. Please see Figure 3 in:

M.E. Leventon, E.L. Grimson, and O. Faugeras, "Statistical Shape Influence in Geodesic Active Contours," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 316-323, 2000.

Notes

- Use Euclidean distance
  - not Mahalanobis distance
- Not clear how the classes Ω_k are constructed
  - here used just one image per person
- Free parameters:
  - number of eigenfaces M' and thresholds
  - in the experiments, $M' = 7$, and the thresholds varied
- Eigenfaces look like faces

Eigenfaces

Image removed due to copyright considerations. Please see Figure 2 in:

M.E. Leventon, E.L. Grimson, and O. Faugeras, "Statistical Shape Influence in Geodesic Active Contours," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 316-323, 2000.

“Faceness” Map

- For each window location, evaluate its distance from the face space
- Slow, faster algorithms exist

Image removed due to copyright considerations. Please see Figure 3 in:

M.E. Leventon, E.L. Grimson, and O. Faugeras, "Statistical Shape Influence in Geodesic Active Contours," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 316-323, 2000.

Experimental Results

- Image Database
  - Controlled experiment

- Real-time face recognition
  - Unconstrained recognition, closer to the real-world

Image Database Results

- Image Database
  - 16 people
  - 3 illuminations, 3 angles, 3 head sizes

- Use one image per person to train
  - $M = 16$, $M' = 7$
  - all at the same illumination, angle, size.

- Test on all other images, forcing recognition
  - 96% across illuminations
  - 85% across orientations
  - 64% across sizes

Image Database Results cont'd

- Vary the threshold (ROC, precision/recall)
  - 100% accuracy, check for "unknown" labels
    - » 19% across illuminations
    - » 39% across orientations
    - » 60% across sizes
  - 20% "unknown", check the accuracy
    - » 100% across illuminations
    - » 94% across orientations
    - » 74% across sizes

Real-time face recognition

- Motion detection, plus head localization: small blob on top of a bigger blob
- Run the recognizer on the head window

- 2-3 times/second (long time ago, so might actually be really fast on a modern computer)
- No accuracy/precision reported

Notes on Performance

- More sensitive to head size than other factors
  - Normalization, windowing
  - Related to the “faceness” map
- Should be sensitive to orientation, as parts of the face get occluded
  - Maybe the projection is helping it
- What happens when we vary all factors at once?

Summary

- Faces have a lot of structure, let's try to capture it automatically.
- Prior work focused on crafted templates
  - This paper learns the structure through PCA
- Strong assumptions on the structure
  - Face images can be approximated by a linear low dimensional space
  - They cluster in that space w.r.t. the person
- Fairly naïve classification

Since Then

- More explicit models of the object shape
  - Active Shape/Appearance Models

- Non-linear subspace/manifold modeling
  - Mixture of Gaussians
  - Kernel PCA
  - Locally Linear Embedding, Isomaps
  - non-negative decomposition, etc.

- Classification
  - Fisher faces (Fisher Linear Discriminant)
  - Other classification methods