Eigenface for Face Recognition

Presenter: Trần Đức Minh

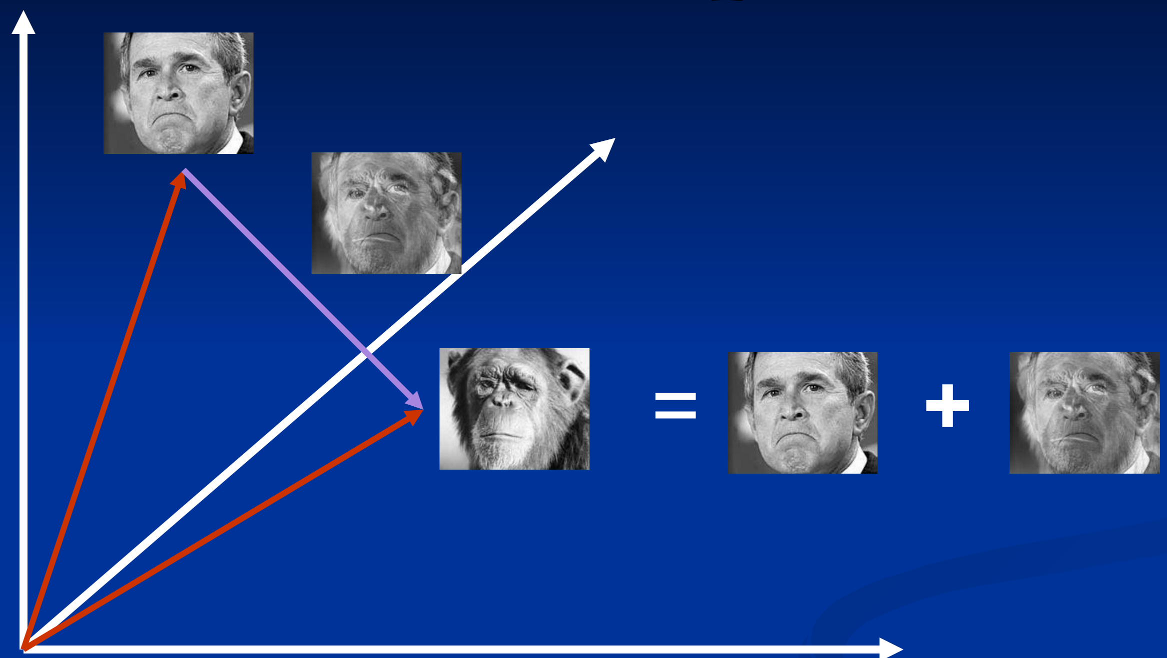
Outline

- Overview
- Eigenfaces for Recognition
- Conclusion

Overview

- Face Representation
  - **Template-based approaches**
  - **Feature-based approaches**
  - **Appearance-based approaches**
- Face Detection
  - **Utilization of elliptical shape of human head (applicable only to front views ☹) [5]**
  - **Manipulation of images in “face space” [1]**
- Face Identification
  - **Performance affected by scale, pose, illumination, facial expression, and disguise, etc.**

Face space



- An image is a point in a high dimensional space
  - An N x M image is a point in R^{NM}
  - We can define vectors in this space as we did in the 2D case

Outline

- Overview
- Eigenfaces for Recognition
- Conclusion

Eigenfaces Approach

- In the language of information theory……
  - **Efficient encoding followed by comparing one face encoding with a database of models encoded similarly**

Eigenfaces Approach *(Contd.)*

- In mathematical terms…….
  - **Find the principal components of the face distribution, or the eigenvectors of the covariance matrix of the set of face images, called eigenfaces**
  - **Eigenfaces are a set of features that characterize the variation between face images**
  - **Each training face image can be represented in terms of a linear combination of the eigenfaces, so can the new input image**
  - **Compare the feature weights of the new input image with those of the known individuals**

Example for eigenface

Eigenfaces look somewhat like generic faces.

Major Steps

1. Initialization: acquire the training set of face images and calculate the eigenfaces, which define the face space
2. Given an image to be recognized, calculate a set of weights of the *M* eigenfaces by projecting it onto each of the eigenfaces
3. Determine if the image is a face at all by checking if the image is sufficiently close to the face space
4. If it is a face, classify the weight pattern as either a known person of as unknown
5. (Optional) If the same unknown face is seen several times, update the eigenfaces / weight patterns, calculate its characteristic weight pattern and incorporate into the known faces

Calculating Eigenfaces

- **Set of training images** $\Gamma_1, \Gamma_2, \ldots\ldots, \Gamma_M$ **(** Γ_n **is a column vector of size** $N^2 \times 1$**)**

Average face of the training set: $$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

Each training image differs from the average face by: $$\Phi_n = \Gamma_n - \Psi$$

- **A total number of** N^2 **pairs of eigenvectors** μ_n **and eigenvalues** λ_n **of the covariance matrix**

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T$$ **(*C*:** $N^2 \times N^2$ **matrix) Eq. (1)**

where $A = [\Phi_1, \Phi_2, \ldots\ldots, \Phi_M]$ **(*A*:** $N^2 \times M$ **matrix)**

Computationally Intractable☹!

Calculating Eigenfaces

- **For Computational Feasibility**

Only *M* - 1 eigenvectors are meaningful ($M < N^2$ **)**

Eigenvectors v_n **and associated eigenvalues** λ_n **of** $L = A^T A$ **:**

$$A^T A v_n = \lambda_n v_n =>$$

$$A A^T A v_n = \lambda_n A v_n$$ **Eq. (2)**

Therefore, $A v_n$ **are the eigenvalues of** $C = A A^T$, λ_n **are the associated eigenvalues**

$$\mu_n = \sum_{k=1}^{M} v_{nk} \Phi_k = A v_n$$ **Eq. (3)**

The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images

Using Eigenfaces for Identification

- **Construction of Known Individuals' Face Classes**

-- Images of known individuals are projected onto "face space" by a simple operation $\omega_{ik} = \mu_k^T(\Gamma_i - \Psi)$, where i=1, 2, ……, M represents the *i*th individual, and *k* =1, 2, ……, *M'* represents the weight coefficient of eigenvector μ_k. The pattern vector of the *i*th individual

$$\Omega_i = [\omega_{i1}, \omega_{i2}, \ldots\ldots, \omega_{iM'}]$$

-- If an individual has more than one image, take the average of the pattern vectors of this person

Using Eigenfaces for Identification (Contd.)

- **Given a new image**

  -- Project Γ onto face space, and get its pattern vector $\Omega = [\omega_1, \omega_2, \ldots\ldots, \omega_{M'}]$

  -- Determine whether Γ is a face image: $\varepsilon^2 = \sum_{n=1}^{M'} (\omega_n - \lambda_n)^2$ **Eq. (4)**

  If ε < a predefined threshold θ, it is a face image; otherwise, not

  -- Classify Γ either as a known individual or as unknown:

  $$\varepsilon_k^2 = \left\| (\Omega - \Omega_k)^2 \right\|$$ **Eq. (5)**

  If $\min\{\varepsilon_k, k = 1, 2, \ldots\ldots, M'\} = \varepsilon_{k'}$ **< a predefined threshold θ_ε, Γ is identified as the *k'*th face class; otherwise, it is identified as unknown**

Outline

- Overview
- Eigenfaces for Recognition
- Conclusion

Advantages

- Ease of implementation
- No knowledge of geometry or specific feature of the face required
- Little preprocessing work

Limitations

- **Sensitive to head scale**
- **Applicable only to front view**
- **Good performance only under controlled background (not including natural scenes)**

Reference

1. "Eigenfaces for recognition", M. Turk and A. Pentland, *Journal of Cognitive Neuroscience, vol.3, No.1, 1991*
2. "Face recognition using eigenfaces", M. Turk and A. Pentland, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 586-591, 1991
3. Lindsay. I. Smith. A tutorial on principal components analysis, February 2002
4. *Ilker Atalay M.Sc Thesis: Face Recognition Using Eigenfaces Istanbul Technical University – January 1996*
5. Dimitri Pissarenko, Eigenface-based facial recognition, Feb 06, 2003

Demo

Thank you