Appearance-Based Object Recognition
– Subspace Methods

CMPE 264: Image Analysis and Computer Vision
Hai Tao
Appearance-based methods

- Store images of 3D objects as their representation. Object recognition is then converted into the following problem
  - Given an image containing an object to identify and a database of object models, each one formed by a set of images showing the object under a large number of viewpoints and illumination conditions, find the set containing the image which is the most similar to the input image
  - The size of the database is large
    
    \[(\text{#objects}) \times (\text{#viewpoint}) \times (\text{#illumination conditions}) \times (\text{#deformations/expressions}) \ldots\]
  - Direct image comparison is computational expensive
    
    \[c = I_1 \circ I_2 = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} I_1(i, j)I_2(i, j)\]

- The last problem can be solved using the subspace (Eigenspace) method
Appearance-based methods

Figure 10.5  A simple database exemplifying appearance-based object representation. Only the viewpoint, not the illumination, was changed to obtain the views shown.
Principal component analysis (Karhunen-Loeve Transform)

- Exploit the correlation between components of data vectors so that data can be represented more efficiently.
- Project the high-dimensional data onto a lower dimensional space and seek the best projection in a least-squares sense.
PCA (K-L Transform)

We first consider a single vector $\mathbf{x}_0$ that best represents a set of $d$-dimensional sample $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. We seek $\mathbf{x}_0$ that minimizes the following squared error

$$\sum_{i=1}^{n} \| \mathbf{x}_i - \mathbf{x}_0 \|^2$$

The solution can be easily derived as

$$\mathbf{x}_0 = \mathbf{m} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$$

This is the best zero-dimensional representation of the data set.
PCA (K-L Transform)

- The data set can be better represented by projecting the data onto an $M$-dimensional subspace spanned by $M$ orthonormal vectors $e_1, e_2, ..., e_M$.
- We seek this subspace so that the squared-error is minimized. In other words, the following function is minimized with respect to a set of orthonormal vectors $e_1, e_2, ..., e_M$

$$\sum_{i=1}^{n} \left\| x_i - \bar{m} - \sum_{k=1}^{M} a_i^k e_k \right\|^2$$

subject to $e_p e_q = \delta_{pq}$

It can be observed that $a_i^k = (x_i - \bar{m})^T e_k$, therefore, the following error needs to be minimized w.r.t. $e_1, e_2, ..., e_M$

$$\sum_{i=1}^{n} \left\| x_i - \bar{m} - \sum_{k=1}^{M} e_k e_k^T (x_i - \bar{m}) \right\|^2$$

subject to $e_p e_q = \delta_{pq}$
PCA (K-L Transform) - derivation

\[
\sum_{i=1}^{n} \left| \mathbf{x}_i - \mathbf{m} - \sum_{k=1}^{M} \mathbf{e}_k (\mathbf{x}_i - \mathbf{m})^T \mathbf{e}_k \right|^2 \\
= \sum_{i=1}^{n} \left( (\mathbf{x}_i - \mathbf{m})^T - \sum_{k=1}^{M} \mathbf{e}_k^T (\mathbf{x}_i - \mathbf{m}) \mathbf{e}_k^T \right) \left( (\mathbf{x}_i - \mathbf{m}) - \sum_{k=1}^{M} \mathbf{e}_k (\mathbf{x}_i - \mathbf{m})^T \mathbf{e}_k \right) \\
= \sum_{i=1}^{n} \left( (\mathbf{x}_i - \mathbf{m})^T (\mathbf{x}_i - \mathbf{m}) - 2 \sum_{k=1}^{M} \mathbf{e}_k^T (\mathbf{x}_i - \mathbf{m}) \mathbf{e}_k^T (\mathbf{x}_i - \mathbf{m}) + \sum_{i=1}^{M} \mathbf{e}_k^T (\mathbf{x}_i - \mathbf{m}) (\mathbf{x}_i - \mathbf{m})^T \mathbf{e}_k \right) \\
= \sum_{i=1}^{n} \left( (\mathbf{x}_i - \mathbf{m})^T (\mathbf{x}_i - \mathbf{m}) - \sum_{k=1}^{M} \mathbf{e}_k^T (\mathbf{x}_i - \mathbf{m}) (\mathbf{x}_i - \mathbf{m})^T \mathbf{e}_k \right) \\
= \sum_{i=1}^{n} \left( (\mathbf{x}_i - \mathbf{m})^T (\mathbf{x}_i - \mathbf{m}) \right) - \sum_{i=1}^{n} \left( \sum_{k=1}^{M} \mathbf{e}_k^T (\mathbf{x}_i - \mathbf{m}) (\mathbf{x}_i - \mathbf{m})^T \mathbf{e}_k \right) \\
= \alpha - \sum_{k=1}^{M} \mathbf{e}_k^T \sum_{i=1}^{n} \left( (\mathbf{x}_i - \mathbf{m}) (\mathbf{x}_i - \mathbf{m})^T \right) \mathbf{e}_k \\
= \alpha - \sum_{k=1}^{M} \mathbf{e}_k^T \mathbf{C} \mathbf{e}_k \\
\text{where} \quad \mathbf{C} = \sum_{i=1}^{n} \left( (\mathbf{x}_i - \mathbf{m}) (\mathbf{x}_i - \mathbf{m})^T \right) \text{is the covariance matrix of the data set}

\]

As the result, we want to find \( \mathbf{e}_1, \ldots, \mathbf{e}_M \) so that \( \sum_{k=1}^{M} \mathbf{e}_k^T \mathbf{C} \mathbf{e}_k \) is maximized. If \( M = 1 \), \( \mathbf{e}_1 \) is the eigenvector corresponding to the largest eigenvalue of \( \mathbf{C} \). Otherwise, \( \mathbf{e}_1, \ldots, \mathbf{e}_M \) should span the same space spanned by the \( M \) eigenvectors corresponding to the \( M \) largest eigenvalues.
A data set $x_1, x_2, ..., x_n$ can be best represented in a least-squares sense using the data mean and a subspace spanned by the $M$ eigenvectors corresponding to the $M$ largest eigenvalues. The projection of data $x_i$ onto this subspace can be written as

$$m + \sum_{k=1}^{M} e_k (x_i^T e_k)$$
The Eigenface representation

- In order to exploit the correlation between pixels in face images, these images need to be aligned so that the same pattern of spatial image intensity variations occur in all images.

- Each image is represented as a vector that consists of all the pixels in the image. For an image with 100x100 pixels, it is represented as a 10,000-dimensional vector.

- PCA is performed over these vectors to extract the subspace spanned by $M$ (30~100) eigenvectors (in this case, they are also called eigenfaces).

- The original face vectors are projected into this low dimensional space. Each face image is now represented by $M$ coordinates in the subspace.

- The eigenface space is usually learned using a large database of face images.
Some face images in the FERET database
The Eigenface representation

Courtesy of Matthew Turk and Alex Pentland
Face recognition based on the eigenface representation

- Each face image can be mapped onto a point in the $M$-dimensional eigenface space.
- The similarity between two faces is measured as the distance between the corresponding data points in the eigenface space.
- To find the best match in a face database, the eigenface representation of an input image is first obtained. Then it is compared with all the eigenface representation of faces in the database. The closest one is the match.

Difficulty
- Sensitive to pose variations
- Sensitive to illumination changes
- Sensitive to other appearance changes
A trick in eigenface computation

To compute the eigenvector of the huge covariance matrix (\(N^2\) by \(N^2\), where \(N^2\) is the number of pixels in an image) based on image vectors involves the SVD of a huge matrix.

This problem can be solved by observing that instead of the covariance matrix

\[
C = \sum_{i=1}^{n} (x_i - \mathbf{m})(x_i - \mathbf{m})^T = [x_1 - \mathbf{m}, \ldots, x_n - \mathbf{m}] [x_1 - \mathbf{m}, \ldots, x_n - \mathbf{m}]^T = \mathbf{A} \mathbf{A}^T
\]

if we construct the following matrix

\[
C' = \mathbf{A}^T \mathbf{A}
\]

then the eigenvector of \(C'\) satisfies

\[
\mathbf{A}^T \mathbf{A} \mathbf{V}_i = \lambda_i \mathbf{V}_i
\]

Multiply both sides by \(\mathbf{A}\), we obtain

\[
(\mathbf{A} \mathbf{A}^T) \mathbf{A} \mathbf{V}_i = \lambda_i \mathbf{A} \mathbf{V}_i
\]

This means \(\mathbf{A} \mathbf{V}_i\) is the eigenvector of the original covariance matrix \(C\). Since \(C'\) is in general much smaller (\(M\) by \(M\), where \(M\) is the number of face images) than \(C\), this is an easier way for computing eigenfaces.
Homework

- Prove that the covariance matrix $C$ and the matrix $C'$ have the same set of non-zero eigenvalues.