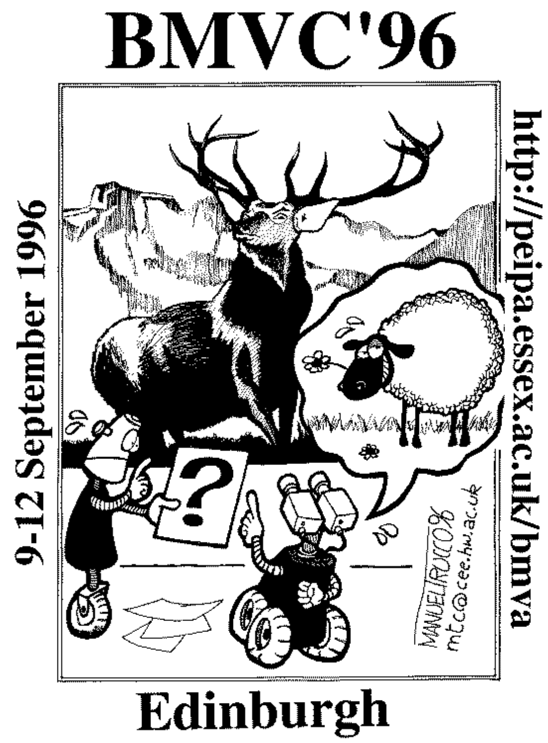
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Lecture 29: Object Recognition II: Eigenspace Methods

Contents

Introduction (1)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32



Object recognition has a long tradition. Poster of the 1996 British Machine Vision Conference illustrating its problems. Author: E. Trucco. Source: `http://www.ece.eps.hw.ac.uk/~mtc/tshirt.gif`.

M I A
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30
31 32

Introduction

Goals

- In the last lecture we have considered
  - pure 2-D object recognition methods (via moment invariants)
  - simple geometric structures (lines, circles)
  - shape-preserving transformations (translations, rotations, varying illumination)
- Now we want to focus on
  - the recognition of 3D objects using 2D images
  - more complicated geometric structures
  - transformations that allow some shape variations

M I A
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30
31 32

Examples

- face recognition
- recognition of different types of cars

Problems

- How can we represent a 3-D object by means of 2-D images?
- How can we describe shape variations?
- How can we find the most similar object in a database?

Introduction (4)

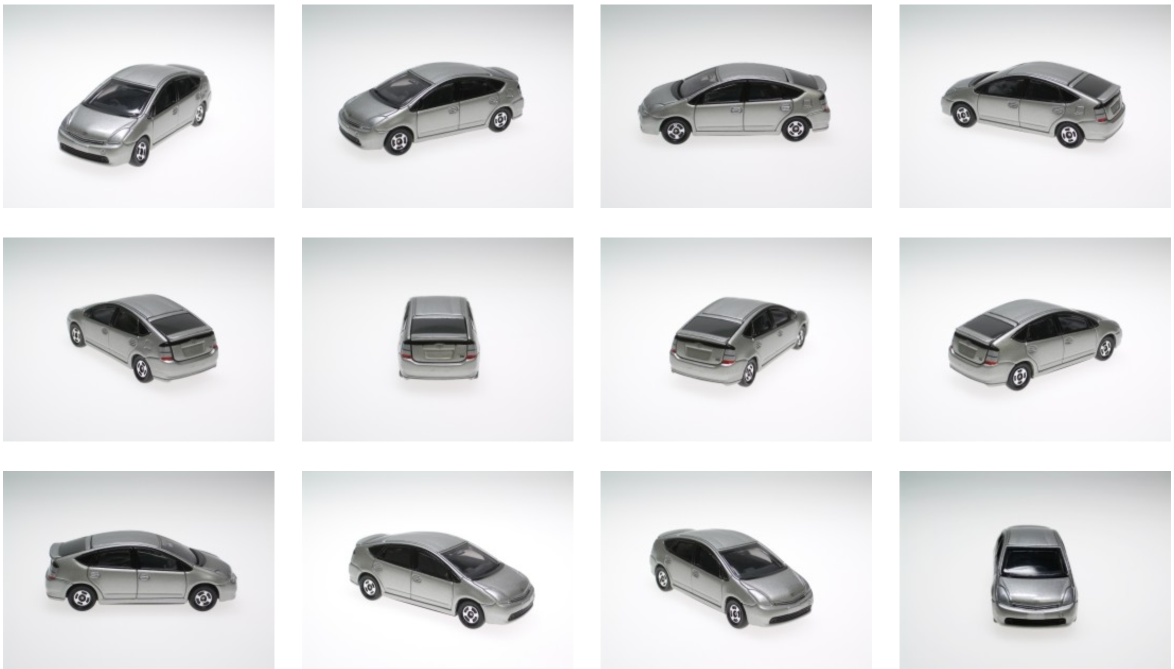
Basic Idea

- to recognise a 3-D object, use many 2-D images of it:
  - photographed under different directions and different illumination situations
- Problem:
  - requires much disk space per object
- Example:
  - Represent a single 3-D object by bytewise coded 256×256 images.
  - Using 100 directions with 10 illumination variants requires 64 MByte.
- Can one represent this large data set in a compact, non-redundant way?

To this end we consider so-called *eigenspace methods (Eigenraumverfahren).*
They are based on a principal component analysis (PCA, Hauptachsentransformation).
This gives a representation in a basis that is specifically adapted to the data set.
For specific problem classes, this can outperform non-adaptive bases such as the DCT.

Introduction (5)

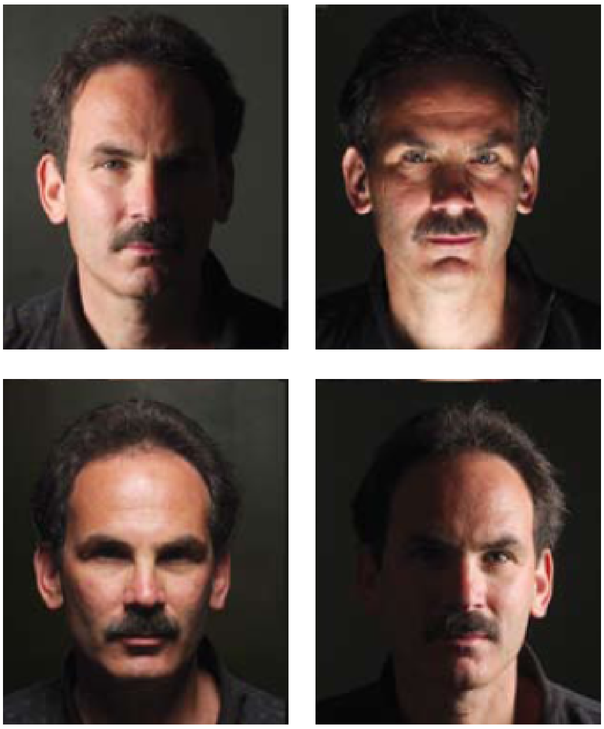
Example: Variability under Different Directions



12 images of a 3-D object being viewed from different directions. Author: S. Kiefer.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Example: Variability under Different Illumination



The same face illuminated from four different directions. Author: D. Kriegman.

Eigenspace Representation

Assumptions

- **Single Object**
  Every image depicts only a single object, and occlusions do not appear.
- **Size Normalisation**
  All images are normalised in size, e.g. by ensuring in a face data base that the image boundaries are given by the smallest rectangle that fully includes the face.
- **Grey Value Normalisation**
  The image grey values are normalised such that they have zero mean, and their summed squared intensity is 1.
  Representing an image by a vector $\boldsymbol{f} = (f_1, ..., f_N)^\top$, this means that

$$|\boldsymbol{f}|^2 \; := \; \sum_{i=1}^{N} f_i^2 \; = \; 1.$$

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Principal Theorem of Eigenspace Representations

Consider m images that are represented by vectors $\boldsymbol{f}_1,...,\boldsymbol{f}_m \in \mathbb{R}^N$.
Usually one has less images than pixels, i.e. $m \ll N$ (e.g. $m = 1000$, $N = 65536$).
Let $\bar{\boldsymbol{f}} := \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{f}_i$ denote the average image.

- Then the symmetric $N \times N$ *covariance matrix*

$$\boldsymbol{Q} \;:=\; \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{f}_i - \bar{\boldsymbol{f}})\,(\boldsymbol{f}_i - \bar{\boldsymbol{f}})^\top$$

  has at most m nonvanishing eigenvalues

$$\lambda_1 \geq ... \geq \lambda_m > 0$$

  with corresponding orthonormal eigenvectors $\boldsymbol{v}_1,...,\boldsymbol{v}_m$.

- Together with $\bar{\boldsymbol{f}}$, these m eigenvectors can represent every image $\boldsymbol{f}_i$:

$$\boldsymbol{f}_i \;=\; \bar{\boldsymbol{f}} + \sum_{j=1}^{m} a_{i,j}\,\boldsymbol{v}_j \qquad (i = 1, ..., m)$$

  with $a_{i,j} := (\boldsymbol{f}_i - \bar{\boldsymbol{f}})^\top \boldsymbol{v}_j$.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Important Properties

Interpretation of the Principal Theorem of Eigenspace Methods

- The average image $\bar{\boldsymbol{f}}$ has been subtracted, since we are mainly interested in deviations from the average shape.

- We describe m images $\boldsymbol{f}_1 - \bar{\boldsymbol{f}},...,\ \boldsymbol{f}_m - \bar{\boldsymbol{f}} \in \mathbb{R}^N$ by m eigenvectors $\boldsymbol{v}_1,...,\boldsymbol{v}_m \in \mathbb{R}^N$.
  At first glance, nothing is gained. However, this is not true:

- The covariance matrix $\boldsymbol{Q}$ resembles the structure tensor $\boldsymbol{J}$ from Lecture 13 (see next page for details):
  - Its eigenvectors $\boldsymbol{v}_1, ..., \boldsymbol{v}_m$ specify the most characteristic directions of variation. Since they are orthogonal, we have decoupled the m shape variations optimally.
  - The eigenvalue λ_i measures the amount of shape variability of the data set $\{\boldsymbol{f}_1, ..., \boldsymbol{f}_m\}$ in the direction of the eigenvector $\boldsymbol{v}_i$.

- *Hence, we have decoupled the shape variations into m orthogonal directions. Moreover, we can quantify the importance of each direction.*

Important Properties (2)

Why Does the Covariance Matrix Describe Shape Variations?

- Consider some direction that is given by a normalised vector $\boldsymbol{n} \in \mathbb{R}^N$.
  For $i = 1,...,m$, we want to measure the deviations $\boldsymbol{f}_i - \bar{\boldsymbol{f}}$ along the direction $\boldsymbol{n}$.
- These deviations can be quantified by the inner products $|\boldsymbol{n}^\top(\boldsymbol{f}_i - \bar{\boldsymbol{f}})|$ for all i.
- Thus, the average quadratic variation in direction $\boldsymbol{n}$ is measured by the energy

$$
\begin{aligned}
E(\boldsymbol{n}) &= \frac{1}{m}\sum_{i=1}^{m} \left|\boldsymbol{n}^\top(\boldsymbol{f}_i - \bar{\boldsymbol{f}})\right|^2 \\
&= \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{n}^\top (\boldsymbol{f}_i - \bar{\boldsymbol{f}})\,(\boldsymbol{f}_i - \bar{\boldsymbol{f}})^\top \boldsymbol{n} \\
&= \boldsymbol{n}^\top \left(\frac{1}{m}\sum_{i=1}^{m} (\boldsymbol{f}_i - \bar{\boldsymbol{f}})\,(\boldsymbol{f}_i - \bar{\boldsymbol{f}})^\top\right) \boldsymbol{n} \\
&= \boldsymbol{n}^\top \boldsymbol{Q}\, \boldsymbol{n}\,.
\end{aligned}
$$

Important Properties (3)

- Now consider a normalised eigenvector $\boldsymbol{v}_j$ to an eigenvalue λ_j of $\boldsymbol{Q}$.
- This gives

$$E(\boldsymbol{v}_j) = \boldsymbol{v}_j^\top \boldsymbol{Q}\, \boldsymbol{v}_j = \boldsymbol{v}_j^\top (\lambda_j\, \boldsymbol{v}_j) = \lambda_j\, \underbrace{\boldsymbol{v}_j^\top \boldsymbol{v}_j}_{1} = \lambda_j\,.$$

  Thus, λ_j quantifies the quadratic shape variation in the eigendirection $\boldsymbol{v}_j$.
- This shows that *$\boldsymbol{Q}$ is a perfect tool for describing shape variations:*
  - The variation along a direction $\boldsymbol{n}$ is given by the quadratic form $\boldsymbol{n}^\top \boldsymbol{Q}\, \boldsymbol{n}$.
  - The eigenvectors of $\boldsymbol{Q}$ specify the most characteristic directions of the variation.
  - The eigenvalues measure the average quadratic variation along the eigenvectors.

Important Properties (4)

Efficient Data Representation

- Usually, only k out of the m nonvanishing eigenvalues $\lambda_1,...,\lambda_m$ of $\boldsymbol{Q}$ are *significantly different* from zero $(k \ll m$, for instance $k = 5$ and $m = 1000)$.

- Thus, we can approximate the m images $\boldsymbol{f}_1,...,\boldsymbol{f}_m$ using $\bar{\boldsymbol{f}}$ and the k most significant eigenvectors $\boldsymbol{v}_1,...,\boldsymbol{v}_k$ of the covariance matrix:

$$\boldsymbol{f}_i \approx \bar{\boldsymbol{f}} + \sum_{j=1}^{k} \alpha_{i,j}\, \boldsymbol{v}_j \qquad (i = 1, ..., m).$$

- Since m represents the number of different directions and illumination variants, one has a much more compact representation:
  - Every convex combination of the initial images $\boldsymbol{f}_1,...,\boldsymbol{f}_m$ can be approximated very well by using only $\bar{\boldsymbol{f}}$ and $\boldsymbol{v}_1,...,\boldsymbol{v}_k$.
  - Thus, the subspace spanned by $\boldsymbol{v}_1,...,\boldsymbol{v}_k$ together with $\bar{\boldsymbol{f}}$ describes the space of learnt 2-D views of the 3-D object.
  - Since the eigenvectors are orthogonal, this representation is very compact: It contains no redundancy.

Important Properties (5)

Be Careful !

- Remember that the images $\boldsymbol{f}_i$ must describe objects that are normalised in size: Already small perturbations may have a large impact on the vector representation of the image.

- The individual images should not be too different.
  Otherwise the eigenvalues of the covariance matrix will not decrease rapidly.

- Face recognition is a good application:
  Face images are relatively similar, and it is easy to normalise them.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Computational Aspects

How Expensive is a Naive Implementation of the Method?

- Note that the covariance matrix $\boldsymbol{Q} \in \mathbb{R}^{N \times N}$ can be very large.
- Example:
  - Images of size 256×256 pixels yield $N = 65536$.
    Thus, $\boldsymbol{Q}$ has size 65536×65536.
  - Since the matrix $\boldsymbol{Q}$ is not sparse, you would not even want to store it:
    17 Gigabytes in float precision!
- Do not even think about computing all its eigenvalues and eigenvectors directly!
- Let us now study
  - how we can reduce the problem of computing the eigenvectors and eigenvalues of the large $N \times N$ matrix $\boldsymbol{Q}$ to a much smaller $m \times m$ matrix $\boldsymbol{T}$.
  - how one can compute the k largest eigenvalues and their eigenvectors of a matrix in an efficient way.

M I A

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Trick for Reducing the Complexity if $m \ll N$

- It is useful to define the $N \times m$ matrix $\boldsymbol{D} := \left(\boldsymbol{f}_1 - \bar{\boldsymbol{f}} \mid \ldots \mid \boldsymbol{f}_m - \bar{\boldsymbol{f}}\right)$.
- So far we have used the large $N \times N$ covariance matrix

$$\boldsymbol{Q} \;=\; \frac{1}{m} \sum_{i=1}^{m} (\boldsymbol{f}_i - \bar{\boldsymbol{f}})\,(\boldsymbol{f}_i - \bar{\boldsymbol{f}})^\top \;=\; \frac{1}{m} \boldsymbol{D}\boldsymbol{D}^\top.$$

- Since $m \ll N$, let us now consider the much smaller $m \times m$ matrix

$$\boldsymbol{T} \;=\; \frac{1}{m} \boldsymbol{D}^\top \boldsymbol{D}.$$

- The eigensystems of the small matrix $\boldsymbol{T}$ and the large matrix $\boldsymbol{Q}$ are connected:
  - The m eigenvalues of $\boldsymbol{T}$ are also eigenvalues of $\boldsymbol{Q}$.
    (Moreover, $\boldsymbol{T}$ contains all nonvanishing eigenvalues of $\boldsymbol{Q}$:
    The remaining $N - m$ eigenvalues of $\boldsymbol{Q}$ are zero.)
  - If $\boldsymbol{w}_i$ is an eigenvector of $\boldsymbol{T}$, then $\boldsymbol{v}_i := \boldsymbol{D}\boldsymbol{w}_i$ is an eigenvector of $\boldsymbol{Q}$.
- Note that in general $|\boldsymbol{v}_i| \neq 1$. Thus, do not forget to normalise $\boldsymbol{v}_i$.

Computational Aspects (3)

Why is this so Nice and Simple?

- Let λ_i be an eigenvalue of the $m \times m$ matrix $\boldsymbol{T} = \frac{1}{m}\boldsymbol{D}^\top\boldsymbol{D}$ with eigenvector $\boldsymbol{w}_i$:

$$\frac{1}{m}\boldsymbol{D}^\top\boldsymbol{D}\,\boldsymbol{w}_i \;=\; \lambda_i\,\boldsymbol{w}_i\,.$$

- Multiplication with $\boldsymbol{D}$ from the left gives

$$\frac{1}{m}\boldsymbol{D}\boldsymbol{D}^\top\boldsymbol{D}\,\boldsymbol{w}_i \;=\; \lambda_i\,\boldsymbol{D}\,\boldsymbol{w}_i\,.$$

- Using $\boldsymbol{Q} = \frac{1}{m}\boldsymbol{D}\boldsymbol{D}^\top$ and setting $\boldsymbol{v}_i := \boldsymbol{D}\,\boldsymbol{w}_i$ yields

$$\boldsymbol{Q}\,\boldsymbol{v}_i \;=\; \lambda_i\,\boldsymbol{v}_i\,.$$

- Hence, λ_i is eigenvalue of the $N \times N$ matrix $\boldsymbol{Q}$ with corresponding eigenvector $\boldsymbol{v}_i$.

Computational Aspects (4)

Computing the Dominant Eigenvector: The Power Iteration

Let an $m \times m$ matrix $\boldsymbol{T}$ be positive semidefinite.
Let $\lambda_1 > 0$ be its largest eigenvalue with corresponding eigenvector $\boldsymbol{w}_1$.
Then we can compute λ_1 and $\boldsymbol{w}_1$ with a very simple iterative algorithm:

- Initialise with a normalised random vector $\boldsymbol{w}_1^{(0)} \in \mathbb{R}^m$.
- For $j = 0, 1, 2, ...$ do:
  - Compute the matrix–vector product $\quad \tilde{\boldsymbol{w}}_1^{(j+1)} := \boldsymbol{T}\boldsymbol{w}_1^{(j)}$.
  - Compute its norm $\quad \lambda_1^{(j+1)} := \left|\tilde{\boldsymbol{w}}_1^{(j+1)}\right|$.
  - Normalise: $\quad \boldsymbol{w}_1^{(j+1)} := \tilde{\boldsymbol{w}}_1^{(j+1)} / \lambda_1^{(j+1)} \quad$ if $\quad \lambda_1^{(j+1)} \neq 0$.

Then with probability 1, the following convergence results hold:

- $\lambda_1^{(j)}$ converges to the dominant eigenvalue λ_1.
- $\boldsymbol{w}_1^{(j)}$ converges to the dominant eigenvector $\boldsymbol{w}_1$.

Computational Aspects (5)

Why Does This Work?

- Let $\boldsymbol{w}_1,\ldots,\boldsymbol{w}_m$ be the orthonormal eigenvectors of $\boldsymbol{T}$ to the eigenvalues $\lambda_1,\ldots,\lambda_m$. Assume that $\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_m \geq 0$.
- Represent the random initial vector $\boldsymbol{u} := \boldsymbol{w}_1^{(0)}$ in this orthonormal basis:

$$\boldsymbol{u} \;=\; \sum_{\ell=1}^{m} \alpha_\ell \, \boldsymbol{w}_\ell \, .$$

- Using $\boldsymbol{T}\boldsymbol{w}_\ell = \lambda_\ell \, \boldsymbol{w}_\ell$ we obtain

$$\boldsymbol{T}^j \boldsymbol{u} \;=\; \sum_{\ell=1}^{m} \alpha_\ell \, \lambda_\ell^j \, \boldsymbol{w}_\ell \;=\; \lambda_1^j \left(\alpha_1 \, \boldsymbol{w}_1 + \sum_{\ell=2}^{m} \alpha_\ell \left(\frac{\lambda_\ell}{\lambda_1} \right)^j \boldsymbol{w}_\ell \right).$$

- From $\left| \frac{\lambda_\ell}{\lambda_1} \right| < 1$ it follows that $\boldsymbol{T}^j \boldsymbol{u}$ approximates $\lambda_1^j \alpha_1 \boldsymbol{w}_1$ for large j.
- With probability 1 we have $\alpha_1 \neq 0$. Then $\boldsymbol{T}^j \boldsymbol{u} \to \lambda_1^j \alpha_1 \boldsymbol{w}_1$ implies:
  - $\boldsymbol{T}^{j+1} \boldsymbol{u} \;\approx\; \lambda_1 \, \boldsymbol{T}^j \boldsymbol{u}$ for large j.
  - Normalising $\boldsymbol{T}^j \boldsymbol{u}$ approximates $\boldsymbol{w}_1$.

Computational Aspects (6)

The Power Iteration for the k Most Dominant Eigenvectors

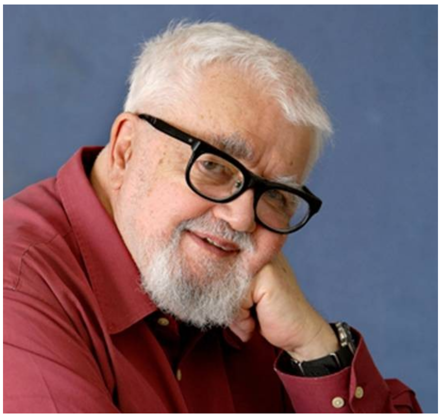
- Assume that we have already computed the $k-1$ most dominant eigenvalues $\lambda_1,\ldots,\lambda_{k-1}$ of $\boldsymbol{T}$, along with their normalised eigenvectors $\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{k-1}$.
- Consider the modified matrix

$$\boldsymbol{T}_k \;:=\; \boldsymbol{T} - \sum_{i=1}^{k-1} \lambda_i \, \boldsymbol{w}_i \, \boldsymbol{w}_i^\top .$$

- Then $\boldsymbol{T}_k$ has the dominant eigenvalue λ_k with corresponding eigenvector $\boldsymbol{w}_k$.
- Thus, λ_k and $\boldsymbol{w}_k$ can be obtained with the power method for $\boldsymbol{T}_k$ instead of $\boldsymbol{T}$.
- This extension of the power method is called the *method of deflation*.

Why Do I Have to Learn All This Mathematics?

- Because you simply have no insights how to solve problems efficiently without it.



http://www.saildart.org/jmc2012.jpg

He who refuses to do arithmetic is doomed to talk nonsense.

John McCarthy (1927–2011)
Father of Artificial Intelligence

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Application to Object Recognition

Training Phase

- Create m normalised image vectors $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_m \in \mathbb{R}^N$ of a 3-D object, using different directions and illumination conditions.
- Compute the average image $\bar{\boldsymbol{f}}$ and the covariance matrix $\boldsymbol{Q}$ (computing the smaller matrix $\boldsymbol{T}$ instead of $\boldsymbol{Q}$ is sufficient).
- Compute the k largest eigenvalues and their eigenvectors of $\boldsymbol{Q}$ (via $\boldsymbol{T}$).
- This k-dimensional subspace of the $\mathbb{R}^N$ characterises the image object.
- Typical numbers:

$$N = 256^2 = 65536, \qquad m = 1000, \qquad k = 5.$$

Thus, we identify a 3-D object that we have learnt from 1000 images with a 5-dimensional subspace in a 65536-dimensional space.
This subspace is spanned by the 5 most dominant eigenvectors *(eigenimages)* of $\boldsymbol{Q}$.
Different objects create different 5-dimensional subspaces.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Recognition Phase

- Given:
  - a data base where different objects are represented by different subspaces
  - a new image vector $\boldsymbol{g}$, for which we want to identify the corresponding object in the data base (i.e. we want to find the closest subspace to $\boldsymbol{g}$)
- Normalise the image $\boldsymbol{g}$ such that it has mean 0 and norm 1.
- Compare $\boldsymbol{g}$ with every 3-D object in the data base:
  - This is done by computing its projection to the corresponding subspace.
  - This projection to a subspace with k orthonormal vectors $\boldsymbol{v}_1,...,\boldsymbol{v}_k$ is given by

$$\sum_{i=1}^{k} (\boldsymbol{g}^\top \boldsymbol{v}_i)\, \boldsymbol{v}_i.$$

  - The distance between $\boldsymbol{g}$ and this subspace is given by

$$\left| \boldsymbol{g} - \sum_{i=1}^{k} (\boldsymbol{g}^\top \boldsymbol{v}_i)\, \boldsymbol{v}_i \right|^2.$$

- The closest subspace gives the most similar object in the data base.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Application to Shape Variation

Example

- In medical imaging, organs can have very complicated shapes.
- One would like to describe them with a few parameters only.

Shape Representation via Contours

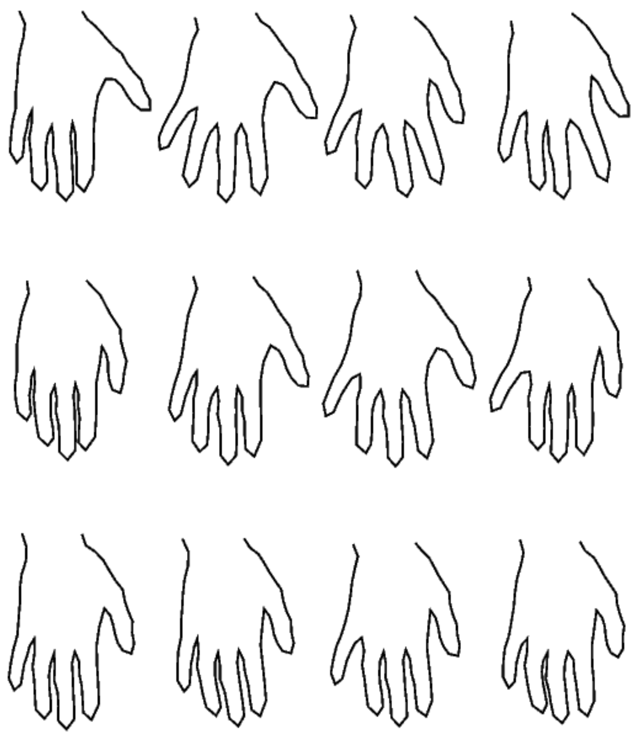
- Shapes are characterised by their contours.
- Contours can be described e.g. by their *Fourier descriptors:*
  - interpret contour with N points (x_j, y_j) as N complex numbers $z_j = x_j + iy_j$
  - 1-D Fourier transform $(\hat{z}_0, ..., \hat{z}_{N-1})$ gives Fourier descriptors
  - convenient for contour operations such as
    – smoothing by low-pass filtering
    – more compact contour representations with only a few low frequencies
- This vector of coefficients can also be used for an eigenspace representation.
  It is smaller and more appropriate than a vector with all image pixels.

Application to Shape Variation (2)

Analysing and Visualising Shape Variation

- One learns the shape in eigenspace.
  Often a small number of significant eigenvectors is sufficient, e.g. $k = 4$.
- The corresponding eigenvalues λ_i measure the shape variation along the eigenvectors $\boldsymbol{v}_i$.
- Often one considers $\bar{\boldsymbol{f}} + \alpha\, \boldsymbol{v}_i$ and varies the weight α in the interval $[-2\lambda_i,\, 2\lambda_i]$.
  This creates a good visual impression of the influence of the eigenvector $\boldsymbol{v}_i$.
- This shape variation along the eigenvectors gives the so-called *modes.*
- Such an *active shape model* can be used e.g. within flexible, knowledge-based segmentation methods.

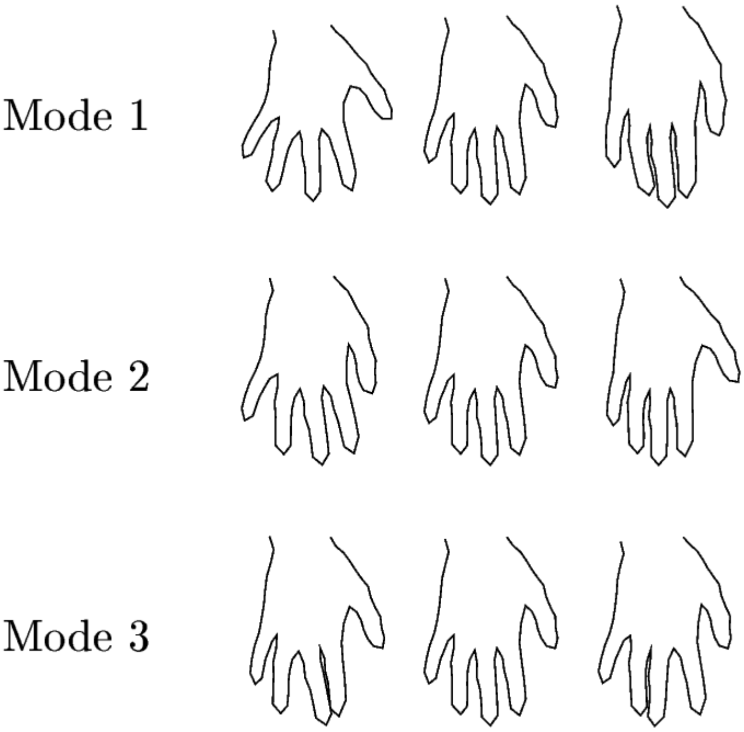
Application to Shape Variation (3)



Training set for characterising the shape variation of hands. Authors: T. F. Cootes, C. J. Taylor.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Application to Shape Variation (4)



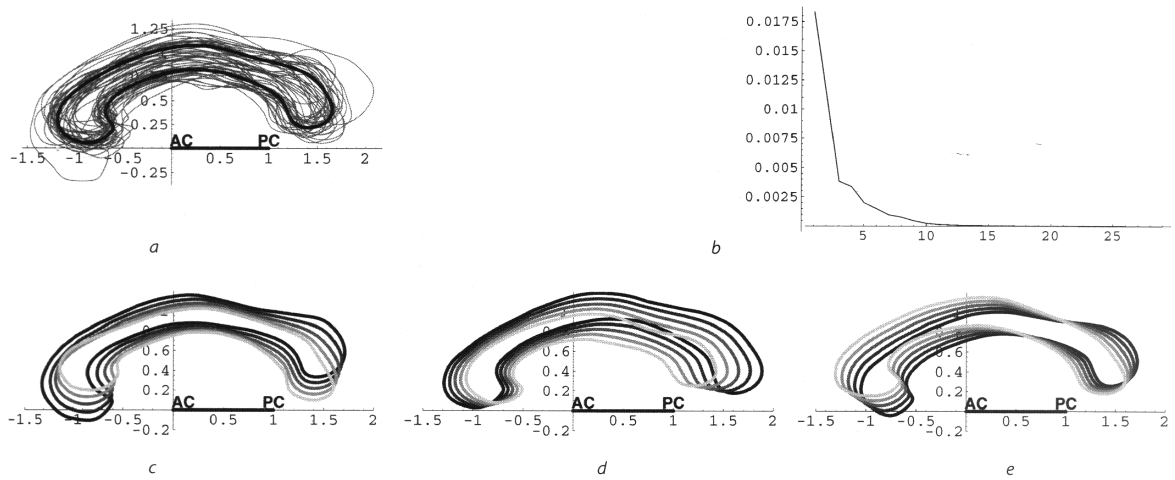
Shape variation of hands. The first three modes are depicted, i.e. variations along the eigenvectors with the three largest eigenvalues are considered. Authors: T. F. Cootes, C. J. Taylor.

Application to Shape Variation (5)

M I
A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

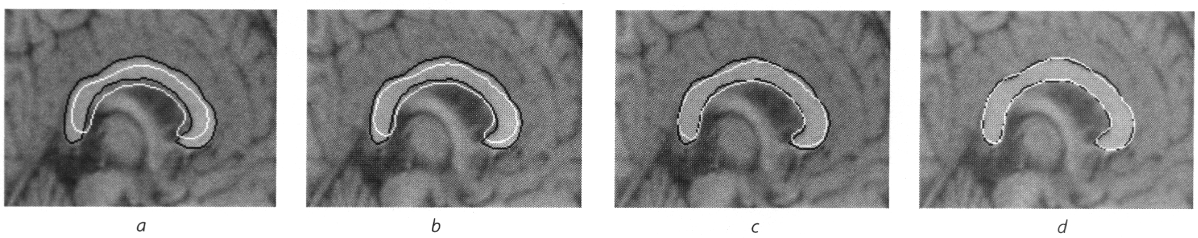
Shape variation in face recognition. The first two modes (± 3 standard deviations). Authors: T. F. Cootes, C. J. Taylor.

Application to Shape Variation (6)



Shape variation in a medical application. (a) Training set (corpus callosum). (b) Magnitude of the eigenvalues. (c),(d),(e) The first three modes. Authors: G. Székely, G. Gerig.

Application to Shape Variation (7)



From left to right: Adaptation of the trained shape to the object by adding 1, 2, or 3 eigenvectors to the average shape. Knowing the average shape and the three most significant eigenvectors from a medical data base, only three parameters are required to describe a given shape almost perfectly. Authors: G. Székely, G. Gerig.

Summary

- Eigenspace methods consider images of a 3-D object as vectors $\boldsymbol{f}_1,\ldots,\boldsymbol{f}_m$.
- They perform a principal component analysis (PCA) of their covariance matrix

$$\boldsymbol{Q} = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{f}_i - \bar{\boldsymbol{f}})(\boldsymbol{f}_i - \bar{\boldsymbol{f}})^\top.$$

- In the training phase, objects are represented by low-dimensional subspaces. They are spanned by the k most dominant eigenvectors of the covariance matrix.
- In the recognition phase, one searches for the subspace that describes the new image best.
- flexible tool for a compact representation, even for highly complicated shapes
- useful e.g. in face recognition and knowledge-based medical image analysis
- requires normalisation of the input images

References

- E. Trucco, A. Verri: *Introductory Techniques for 3-D Computer Vision*. Prentice–Hall, Upper Saddle River, 1998.
  (This lecture is based on Section 10.4.)
- T. F. Cootes, C. J. Taylor: *Statistical Models of Appearance for Computer Vision*. Technical Report, Imaging Science and Biomedical Engineering, University of Manchester, England, 2004 (`www.face-rec.org/algorithms/AAM/app_models.pdf`).
  (survey by two of the pioneers of that area)
- M. Turk, A. Pentland: Eigenfaces for recognition. *Journal of Computational Neuroscience*, Vol. 3, No. 1, 71–86, 1991.
  (classical paper on PCA-based face recognition)
- L. Eldén: *Matrix Methods in Data Mining and Pattern Recognition*. SIAM, Philadelphia, 2007.
  (book on numerical methods)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32