Computer Vision - Lecture 9

Subspace Representations for Recognition

25.11.2009

Bastian Leibe
RWTH Aachen
http://www.mmp.rwth-aachen.de/

leibe@umic.rwth-aachen.de
Course Outline

• Image Processing Basics
• Segmentation & Grouping
• Recognition
  ➢ Global Representations
  ➢ Subspace representations
• Object Categorization I
  ➢ Sliding Window based Object Detection
• Local Features & Matching
• Object Categorization II
  ➢ Part based Approaches
• 3D Reconstruction
• Motion and Tracking
Recap: Appearance-Based Recognition

- Basic assumption
  - Objects can be represented by a set of images ("appearances").
  - For recognition, it is sufficient to just compare the 2D appearances.
  - No 3D model is needed.

⇒ Fundamental paradigm shift in the 90’s
Recap: Recognition Using Histograms

- Histogram comparison

Test image

Known objects
Recap: Comparison Measures

- Vector space interpretation
  - Euclidean distance
  - Mahalanobis distance

- Statistical motivation
  - Chi-square
  - Bhattacharyya

- Information-theoretic motivation
  - Kullback-Leibler divergence, Jeffreys divergence

- Histogram motivation
  - Histogram intersection

- Ground distance
  - Earth Movers Distance (EMD)
Recap: Recognition Using Histograms

- **Simple algorithm**
  1. Build a set of histograms $H=\{h_i\}$ for each known object
     - More exactly, for each view of each object
  2. Build a histogram $h_t$ for the test image.
  3. Compare $h_t$ to each $h_i \in H$
     - Using a suitable comparison measure
  4. Select the object with the best matching score
     - Or reject the test image if no object is similar enough.

“Nearest-Neighbor” strategy

B. Leibe
Recap: Histogram Backprojection

- „Where in the image are the colors we’re looking for?“
  - Query: object with histogram $M$
  - Given: image with histogram $I$

- Compute the „ratio histogram“: $R_i = \min\left(\frac{M_i}{I_i}, 1\right)$
  - $R$ reveals how important an object color is, relative to the current image.
  - Project value back into the image (i.e. replace the image values by the values of $R$ that they index).
  - Convolve result image with a circular mask to find the object.
Recap: Multidimensional Representations

- Combination of several descriptors
  - Each descriptor is applied to the whole image.
  - Corresponding pixel values are combined into one feature vector.
  - Feature vectors are collected in multidimensional histogram.

\[
\begin{array}{c}
D_x \\
D_y \\
Lap
\end{array}
\]
Recap: Bayesian Recognition Algorithm

1. Build up histograms \( p(m_k | o_n) \) for each training object.
2. Sample the test image to obtain \( m_k, k \in K \).
   - Only small number of local samples necessary.
3. Compute the probabilities for each training object.

\[
\begin{align*}
  m_i & \quad \Rightarrow \quad p(o_n | m_i) \\
  m_j & \quad \Rightarrow \quad p(o_n | m_j) \\
  \vdots
\end{align*}
\]

\[
p(o_n | \text{Image}) = \frac{\prod_k p(m_k | o_n)p(o_n)}{\sum_i \prod_k p(m_k | o_i)p(o_i)}
\]

4. Select the object with the highest probability
   - Or reject the test image if no object accumulates sufficient probability.
Recap: Colored Derivatives

- Generalization: derivatives along
  - Y axis → intensity differences
  - C₁ axis → red-green differences
  - C₂ axis → blue-yellow differences

- Application:
  - Brand identification in video

[Hall & Crowley, 2000]
You’re Now Ready for First Applications…

- Histogram based recognition
- Line detection
- Circle detection
- Binary Segmentation
- Moment descriptors
- Skin color detection

Image Source: http://www.flickr.com/photos/angelsk/2806412807/
Demo Competition

- Design your own Computer Vision demo!
  - Based on the techniques from the lecture...
  - Topic is up to you - it should be fun!
  - Teams of up to 3 students
  - Demo day after the end of the semester
    - Will send around a poll for a suitable date...
    - Participation is optional (but it will be fun!)
    - Demos will count for up to 30 extra exercise points
    - (Small) prizes for best teams

If you have questions, we’ll be happy to give advice...
Topics of This Lecture

- **Subspace Methods for Recognition**
  - Motivation

- **Principal Component Analysis (PCA)**
  - Derivation
  - Object recognition with PCA
  - Eigenimages/Eigenfaces
  - Limitations

- **Fisher’s Linear Discriminant Analysis (FLD/LDA)**
  - Derivation
  - Fisherfaces for recognition
Recap: Appearance-Based Recognition

- Basic assumption
  - Objects can be represented by a set of images ("appearances").
  - For recognition, it is sufficient to just compare the 2D appearances.
  - No 3D model is needed.

$\Rightarrow$ Represent objects by sets of global descriptors
Recap: Recognition Using Global Features

- E.g. histogram comparison
Representations for Recognition

- More generally, we want to obtain representations that are well-suited for
  - Recognizing a certain class of objects
  - Identifying individuals from that class (identification)

- How can we arrive at such a representation?

- Approach 1:
  - Come up with a brilliant idea and tweak it until it works.

- Can we do this more systematically?
Example: The Space of All Face Images

- When viewed as vectors of pixel values, face images are extremely high-dimensional.
  - 100x100 image = 10,000 dimensions
- However, relatively few 10,000-dimensional vectors correspond to valid face images.
- We want to effectively model the subspace of face images.
The Space of All Face Images

- We want to construct a low-dimensional linear subspace that best explains the variation in the set of face images.
Subspace Methods

- Images represented as points in a high-dim. vector space
- Valid images populate only a small fraction of the space
- Characterize subspace spanned by images

Image set $\rightarrow$ Basis images $\rightarrow$ Representation coefficients
Subspace Methods

- Today’s topics: PCA, FLD

Slide credit: Ales Leonardis
Topics of This Lecture

- Subspace Methods for Recognition
  - Motivation

- Principal Component Analysis (PCA)
  - Derivation
  - Object recognition with PCA
  - Eigenimages/Eigenfaces
  - Limitations

- Fisher’s Linear Discriminant Analysis (FLD/LDA)
  - Derivation
  - Fisherfaces for recognition
Principal Component Analysis

- Given: $N$ data points $x_1, \ldots, x_N$ in $\mathbb{R}^d$
- We want to find a new set of features that are linear combinations of original ones:
  
  $$u(x_i) = u^\top(x_i - \mu)$$

($\mu$: mean of data points)

- What unit vector $u$ in $\mathbb{R}^d$ captures the most variance of the data?
Principal Component Analysis

- Direction that maximizes the variance of the projected data:

\[
\text{var}(u) = \frac{1}{N} \sum_{i=1}^{N} u^T (x_i - \mu)(u^T (x_i - \mu))^T
\]

Projection of data point

\[
= \frac{1}{N} u^T \left[ \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \right] u
\]

Covariance matrix of data

\[
= \frac{1}{N} u^T \Sigma u
\]

- The direction that maximizes the variance is the eigenvector associated with the largest eigenvalue of \( \Sigma \).
Remember: Fitting a Gaussian

- Mean and covariance matrix of data define a Gaussian model
Interpretation of PCA

- Compute eigenvectors of covariance $\Sigma$.
- Eigenvectors: main directions
- Eigenvalue: variance along eigenvector

$\mathbf{u}_1, \mathbf{u}_2$

- Result: coordinate transform to best represent the variance of the data
Interpretation of PCA

- Now, suppose we want to represent the data using just a single dimension.
  - I.e., project it onto a single axis
  - What would be the best choice for this axis?
Interpretation of PCA

- Now, suppose we want to represent the data using just a single dimension.
  - i.e., project it onto a single axis
  - What would be the best choice for this axis?

- The first eigenvector gives us the best reconstruction.
  - Direction that retains most of the variance of the data.
Properties of PCA

- It can be shown that the mean square error between $x_i$ and its reconstruction using only $m$ principle eigenvectors is given by the expression:

$$\sum_{j=1}^{N} \lambda_j - \sum_{j=1}^{m} \lambda_j = \sum_{j=m+1}^{N} \lambda_j$$

- Interpretation
  - PCA minimizes reconstruction error
  - PCA maximizes variance of projection
  - Finds a more “natural” coordinate system for the sample data.

Cumulative influence of eigenvectors

90% of variance
Projection and Reconstruction

- An $n$-pixel image $x \in \mathbb{R}^n$ can be projected to a low-dimensional feature space $y \in \mathbb{R}^m$ by
  $$ y = Ux $$

- From $y \in \mathbb{R}^m$, the reconstruction of the point is $U^Ty$

- The error of the reconstruction is
  $$ \| x - U^T U x \| $$
Example: Object Representation
Principal Component Analysis

Get a compact representation by keeping only the first \( k \) eigenvectors!
Object Detection by Distance TO Eigenspace

- Scan a window $\omega$ over the image and classify the window as object or non-object as follows:
  - Project window to subspace and reconstruct as earlier.
  - Compute the distance between $\omega$ and the reconstruction (reprojection error).
  - Local minima of distance over all image locations $\Rightarrow$ object locations
  - Repeat at different scales
  - Possibly normalize window intensity such that $|\omega|=1$. 

Slide credit: Peter Belhumeur
Eigenfaces: Key Idea

- Assume that most face images lie on a low-dimensional subspace determined by the first $k$ ($k < d$) directions of maximum variance.
- Use PCA to determine the vectors $u_1, \ldots, u_k$ that span that subspace:

$$x \approx \mu + w_1u_1 + w_2u_2 + \ldots + w_ku_k$$

- Represent each face using its “face space” coordinates $(w_1, \ldots, w_k)$
- Perform nearest-neighbor recognition in “face space”

Eigenfaces Example

- Training images $x_1, \ldots, x_N$
Eigenfaces Example

Top eigenvectors: $u_1, \ldots, u_k$

Mean: $\mu$

Slide credit: Svetlana Lazebnik
Eigenface Example 2 (Better Alignment)
Eigenfaces Example

- Face $x$ in “face space” coordinates:

$$x \to \begin{bmatrix} u_1^T(x - \mu), \ldots, u_k^T(x - \mu) \end{bmatrix} = \begin{bmatrix} w_1, \ldots, w_k \end{bmatrix}$$
Eigenfaces Example

- Face $x$ in “face space” coordinates:

$$x \rightarrow [u_1^T(x - \mu), \ldots, u_k^T(x - \mu)]$$

$$= w_1, \ldots, w_k$$

- Reconstruction:

$$x = \mu + w_1u_1 + w_2u_2 + w_3u_3 + w_4u_4 + \ldots$$

Slide credit: Svetlana Lazebnik
Recognition with Eigenspaces

• Process labeled training images:
  - Find mean $\mu$ and covariance matrix $\Sigma$
  - Find $k$ principal components (eigenvectors of $\Sigma$) $u_1, \ldots, u_k$
  - Project each training image $x_i$ onto subspace spanned by principal components:
    $$(w_{i1}, \ldots, w_{ik}) = (u_1^T(x_i - \mu), \ldots, u_k^T(x_i - \mu))$$

• Given novel image $x$:
  - Project onto subspace:
    $$(w_1, \ldots, w_k) = (u_1^T(x - \mu), \ldots, u_k^T(x - \mu))$$
  - Optional: check reconstruction error $x - \hat{x}$ to determine whether image is really a face
  - Classify as closest training face in $k$-dimensional subspace
Obj. Identification by Distance IN Eigenspace

- Objects are represented as coordinates in an $n$-dim. eigenspace.
- Example:
  - 3D space with points representing individual objects or a manifold representing parametric eigenspace (e.g., orientation, pose, illumination).

- Estimate parameters by finding the NN in the eigenspace.
Parametric Eigenspace

- Object identification / pose estimation
  - Find nearest neighbor in eigenspace [Murase & Nayar, IJCV’95]
Applications: Recognition, Pose Estimation

H. Murase and S. Nayar, Visual learning and recognition of 3-d objects from appearance, IJCV 1995
Applications: Visual Inspection


B. Leibe
Important Footnote

• Don’t really implement PCA this way!
  - Why?

1. How big is $\Sigma$?
   - $n \times n$, where $n$ is the number of pixels in an image!
   - However, we only have $m$ training examples, typically $m \ll n$.
   $\Rightarrow$ $\Sigma$ will at most have rank $m$!

2. You only need the first $k$ eigenvectors
Singular Value Decomposition (SVD)

- Any $m \times n$ matrix $A$ may be factored such that
  \[ A = U \Sigma V^T \]
  
  \[ [m \times n] = [m \times m][m \times n][n \times n] \]

- **$U$: $m \times m$, orthogonal matrix**
  - Columns of $U$ are the eigenvectors of $AA^T$

- **$V$: $n \times n$, orthogonal matrix**
  - Columns are the eigenvectors of $A^TA$

- **$\Sigma$: $m \times n$, diagonal with non-negative entries ($\sigma_1, \sigma_2, \ldots, \sigma_s$)** with $s = \min(m,n)$ are called the singular values.
  - Singular values are the square roots of the eigenvalues of both $AA^T$ and $A^TA$. **Columns of $U$ are corresponding eigenvectors!**
  - Result of SVD algorithm: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$
SVD Properties

• Matlab: \([u \ s \ v] = \text{svd}(A)\)
  - where \(A = u^*s*v'\)

• \(r = \text{rank}(A)\)
  - Number of non-zero singular values

• \(U, V\) give us orthonormal bases for the subspaces of \(A\)
  - first \(r\) columns of \(U\): column space of \(A\)
  - last \(m-r\) columns of \(U\): left nullspace of \(A\)
  - first \(r\) columns of \(V\): row space of \(A\)
  - last \(n-r\) columns of \(V\): nullspace of \(A\)

• For \(d \leq r\), the first \(d\) columns of \(U\) provide the best \(d\)-dimensional basis for columns of \(A\) in least-squares sense
Performing PCA with SVD

• Singular values of $A$ are the square roots of eigenvalues of both $AA^T$ and $A^TA$.
  - Columns of $U$ are the corresponding eigenvectors.

• And
  $$
  \sum_{i=1}^{n} a_i a_i^T = [a_1 \ldots a_n][a_1 \ldots a_n]^T = AA^T
  $$

• Covariance matrix
  $$
  \Sigma = \frac{1}{n} \sum_{i=1}^{n} (\bar{x}_i - \bar{\mu})(\bar{x}_i - \bar{\mu})^T
  $$

• So, ignoring the factor $1/n$, subtract mean image $\mu$ from each input image, create data matrix, and perform (thin) SVD on the data matrix.
Thin SVD

- Any $m$ by $n$ matrix $A$ may be factored such that
  $$A = U \Sigma V^T$$

- If $m > n$, then one can view $\Sigma$ as:
  $$\begin{bmatrix}
  \Sigma' \\
  0 
  \end{bmatrix}$$

- Where $\Sigma' = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$ with $s = \min(m, n)$, and lower matrix is $(n-m \text{ by } m)$ of zeros.

- Alternatively, you can write:
  $$A = U' \Sigma' V^T$$

- In Matlab, thin SVD is: $[U \ S \ V] = \text{svds}(A, k)$

This is what you should use!

Slide credit: Peter Belhumeur
Limitations

- Global appearance method: not robust to misalignment, background variation

- Easy fix (with considerable manual overhead)
  - Need to align the training examples
Limitations

- PCA assumes that the data has a Gaussian distribution (mean $\mu$, covariance matrix $\Sigma$)

The shape of this dataset is not well described by its principal components

Slide credit: Svetlana Lazebnik
Limitations

- The direction of maximum variance is not always good for classification
Topics of This Lecture

- Subspace Methods for Recognition
  - Motivation
- Principal Component Analysis (PCA)
  - Derivation
  - Object recognition with PCA
  - Eigenimages/Eigenfaces
  - Limitations
- Fisher’s Linear Discriminant Analysis (FLD/LDA)
  - Derivation
  - Fisherfaces for recognition

B. Leibe
Restrictions of PCA

- PCA minimizes projection error

- PCA is „unsupervised“, no information on classes is used
- Discriminating information might be lost
Fischer’s Linear Discriminant Analysis (FLD)

- FLD is an enhancement to PCA
  - Constructs a discriminant subspace that minimizes the scatter between images of the same class and maximizes the scatter between different class images
  - Also sometimes called LDA...

Slide adapted from Peter Belhumeur

B. Leibe
Mean Images

- Let $X_1, X_2, \ldots, X_k$ be the classes in the database and let each class $X_i$, $i = 1, 2, \ldots, k$ have $N_i$ images $x_j$, $j = 1, 2, \ldots, k$.

- We compute the mean image $\mu_i$ of each class $X_i$ as:
  $$\mu_i = \frac{1}{k} \sum_{j=1}^{N_i} x_j$$

- Now, the mean image $\mu$ of all the classes in the database can be calculated as:
  $$\mu = \frac{1}{C} \sum_{i=1}^{k} \mu_i$$
Scatter Matrices

• We calculate the **within-class** scatter matrix as:

\[
S_W = \sum_{i=1}^{k} \sum_{x_j \in X_i} (x_j - \mu_i)(x_j - \mu_i)^T
\]

• We calculate the **between-class** scatter matrix as:

\[
S_B = \sum_{i=1}^{k} N_i (\mu_i - \mu)(\mu_i - \mu)^T
\]
Visualization

Good separation

$S_B$

$S_W = S_1 + S_2$

$S_1$

$S_2$

Slide credit: Ales Leonardis
Fisher’s Linear Discriminant Analysis (FLD)

- Maximize distance between classes
- Minimize distance within a class

Criterion: \[ J(w) = \frac{w^T S_B w}{w^T S_W w} \]

- The optimal solution for \( w \) can be obtained as:
  \[ w \propto S_W^{-1}(\mu_2 - \mu_1) \]

- Classification function:
  \[ y(x) = w^T x + w_0 \]
  \[ \begin{align*}
  y(x) &< 0 & \text{Class 1} \\
  y(x) &\geq 0 & \text{Class 2}
  \end{align*} \]
Multiple Discriminant Analysis

- Generalization to $K$ classes

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|}$$

where

$$W = [w_1, \ldots, w_K] \quad \mu = \frac{1}{N} \sum_{n=1}^{N} x_n = \frac{1}{N} \sum_{k=1}^{K} N_k \mu_k$$

$$S_B = \sum_{k=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^T$$

$$S_W = \sum_{k=1}^{K} \sum_{n \in C_k} (x_n - \mu_k)(x_n - \mu_k)^T$$

B. Leibe
Maximizing $J(W)$

- Generalized eigenvalue problem
  
  $$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|}$$

  - The columns of the optimal $W$ are the eigenvectors corresponding to the largest eigenvectors of
    
    $$S_B w_i = \lambda_i S_W w_i$$

  - Defining $v = S_B^{\frac{1}{2}} w$, we get
    
    $$S_B^{\frac{1}{2}} S_W^{-1} S_B^{\frac{1}{2}} v = \lambda v$$

    which is a regular eigenvalue problem.
    
    $\Rightarrow$ Solve to get eigenvectors of $v$, then from that of $w$.

- For the K-class case we obtain (at most) $K-1$ projections.
  - (i.e. eigenvectors corresponding to non-zero eigenvalues.)
Face Recognition Difficulty: Lighting

- The same person with the same facial expression, and seen from the same viewpoint, can appear dramatically different when light sources illuminate the face from different directions.
Application: Fisherfaces

- **Idea:**
  - Using Fisher’s linear discriminant to find class-specific linear projections that compensate for lighting/facial expression.

- **Singularity problem**
  - The within-class scatter is always singular for face recognition, since #training images << #pixels
  - This problem is overcome by applying PCA first

\[ W_{opt}^T = W_{fld}^T U_{pca}^T \]

where

\[ U_{pca} = \arg\max_U |U^T S_T U|, \quad S_T = S_B + S_W \]

\[ W_{fld} = \arg\max_W \frac{|W^T U_{pca}^T S_B U_{pca} W|}{|W^T U_{pca}^T S_W U_{pca} W|} \]

---

Slide credit: Peter Belhumeur

B. Leibe

[Belhumeur et.al. 1997]
Fisherfaces: Experiments

- Variation in lighting
Fisherfaces: Experiments

Subsets 1 to 5 of face images are displayed, with each subset showing variations in facial expressions and lighting conditions.
Fisherfaces: Experimental Results

Error Rate (%)

Subset 1  Subset 2  Subset 3

Lighting Direction Subset

- Eigenface (10)
- Eigenface (10) w/o first 3
- Correlation
- Linear Subspace
- Fisherface

Slide credit: Peter Belhumeur

B. Leibe

[Belhumeur et al. 1997]
Fisherfaces: Experiments

- Variation in facial expression, eye wear, lighting
Fisherfaces: Experimental Results

![Graph showing error rates for different lighting direction subsets and methods.](image)

Slide credit: Peter Belhumeur

B. Leibe

[Belhumeur et.al. 1997]
Example Application: Fisherfaces

- Visual discrimination task
  - Training data:
    - $C_1$: Subjects with glasses
    - $C_2$: Subjects without glasses
  - Test:
    - glasses? 

Take each image as a vector of pixel values and apply FLD...

Image source: Yale Face Database
Fisherfaces: Interpretability

- Example Fisherface for recognition “Glasses/NoGlasses”
References and Further Reading

- Background information on PCA/FLD can be found in Chapter 22.3 of
  D. Forsyth, J. Ponce,
  *Computer Vision - A Modern Approach.*
  Prentice Hall, 2003

- Important Papers (available on webpage)
  - M. Turk, A. Pentland
eigenfaces for Recognition
  - P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman
  Eigenfaces vs. Fisherfaces: Recognition Using Class Specific