Announcements

- Assignment 4: Due date extended
  - Electronic submission: Sunday, June 11, 6:00 PM.
  - Hardcopy: Monday, June 12, 6:00 PM
- Read: Trucco & Verri, Chapter 10 on recognition
- Final Exam: Thursday, 6/15, 8:00AM

Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.

Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional
- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability (perspective, orthographic projection)
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial
- Clutter in background -> false positives

A Rough Recognition Spectrum

Appearance-Based Recognition (Eigenface, Fisherface) – Local Features + Spatial Relations
Shape Contexts – Aspect Graphs
Geometric Invariants – 3-D Model-Based Recognition
Image Abstractions/Volumetric Primitives – Function
Increasing Generality

Sketch of a Pattern Recognition Architecture

Image (window) – Feature Extraction
Feature Vector – Classification
Object Identity
Example: Face Detection
- Scan window over image.
- Classify window as either:
  - Face
  - Non-face

Sketch of a Pattern Recognition Architecture

Image as a Feature Vector
- Consider an n-pixel image (window) to be a point in an n-dimensional space, \( x \in \mathbb{R}^n \).
- Each pixel value is a coordinate of \( x \).

Simplest Recognition Scheme
- \( R \) is an image.
- \( c(R, I) \) is Euclidean distance.

Dimensionality Reduction: Linear Projection
- An n-pixel image \( x \in \mathbb{R}^n \) can be projected to a low-dimensional feature space \( y \in \mathbb{R}^m \) by
  \[ y = Wx \]
  where \( W \) is an \( m \) by \( n \) matrix.
- Recognition is performed using nearest neighbor in \( \mathbb{R}^m \).
- How do we choose a good \( W \)?
Eigenfaces: Principal Component Analysis (PCA)

Assume we have a set of $n$ feature vectors $x_i$ ($i = 1, \ldots, n$) in $\mathbb{R}^d$. Write

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i, \quad \Sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T$$

The unit eigenvectors of $\Sigma$ — which we write as $v_1, v_2, \ldots, v_k$, where the order is given by the size of the eigenvalue and $v_1$ has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{v_1, \ldots, v_k\}$ gives the $k$-dimensional set of linear features that preserves the most variance.

Algorithm 22.5: Principal component analysis identifies a collection of linear features that are independent, and capture as much variance as possible from a dataset.

Some details: Use Singular value decomposition, “trick” described in appendix of text to compute basis when $n<<d$.

Eigenfaces

- **Modeling**
  1. Given a collection of $n$ labeled training images,
  2. Compute mean image and covariance matrix.
  3. Compute $k$ Eigenvectors (note that these are images) of covariance matrix corresponding to $k$ largest Eigenvalues. (Or perform using SVD!!)
  4. Project the training images to the $k$-dimensional Eigenspace.

- **Recognition**
  1. Given a test image, project to Eigenspace.
  2. Perform classification to the projected training images.

And important footnote: Don’t really implement PCA this way!

Why?
1. How big is $\Sigma$?
   - $n$ by $n$ where $n$ is the number of pixels in an image!!
2. You only need the first $k$ Eigenvectors
3. [However HW 4, the images are small enough that the direct Eigenspace approach will work.]
**Singular Value Decomposition**

- **Any** $m \times n$ matrix $A$ may be factored such that $A = U \Sigma V^T$
  -  $U$: $m \times m$, 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 eigenvalues of both $A^TA$ and $AA^T$.

- **Result of SVD algorithm:** $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$

**Performing PCA with SVD**

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

- **Covariance matrix is:**
  $$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T$$

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

**Thin SVD**

- **Any** $m \times n$ matrix $A$ may be factored such that $A = U \Sigma V^T$
  -  $U$: $m \times m$, orthogonal matrix
  -  Columns of $U$ are the eigenvectors of $AA^T$
  -  $V$: $n \times n$, orthogonal matrix
  -  Columns are the eigenvectors of $A^TA$

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

**Fisherfaces: Class specific linear projection**

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

  -  Where $W$ is an $n \times m$ matrix.
  -  Recognition is performed using nearest neighbor in $\mathbb{R}^m$.
  -  How do we choose a good $W$?

**SVD Properties**

- In Matlab $[U \ S \ V] = svd(A)$, and you can verify that $A = U^S V^T$.
  -  $r = \text{rank}(A) = \# \text{ of non-zero singular values}$.
  -  $U$, $V$ give us orthonormal bases for the subspaces of $A$:
    -  $1 \times r$ columns of $U$: Column space of $A$
    -  $1 \times (m-r)$ columns of $U$: Left nullspace of $A$
    -  $1 \times r$ columns of $V$: Row space of $A$
    -  $1 \times (n-r)$ columns of $V$: Nullspace of $A$
  -  For $d \leq r$, the first $d$ column of $U$ provide the best $d$-dimensional basis for columns of $A$ in least squares sense.

**Alternative projections**
PCA & Fisher’s Linear Discriminant

- Between-class scatter
  \[ S_W = \sum_{i=1}^{c} |x_i - \mu_i| (x_i - \mu_i)^T \]
- Within-class scatter
  \[ S_B = \sum_{i=1}^{c} n_i (x_i - \mu_i) (x_i - \mu_i)^T \]
- Total scatter
  \[ S_T = S_W + S_B \]

Where
- \( c \) is the number of classes
- \( \mu_i \) is the mean of class \( \chi_i \)
- \( |x_i| \) is number of samples of \( \chi_i \).

- If the data points are projected by \( y = Wx \) and scatter of points \( x_i \) is \( S \), then the scatter of the projected points \( y_i \) is \( WSW \).

- If \( S_W \) is rank \( N-c \), project training set to subspace spanned by first \( N-c \) principal components of the training set.
- Apply FLD to \( N-c \) dimensional subspace yielding \( c-1 \) dimensional feature space.

- Fisher’s Linear Discriminant projects away the within-class variation (lighting, expressions) found in training set.
- Fisher’s Linear Discriminant preserves the separability of the classes.

Fisherfaces

\[ W = W_{FLD}W_{PCA} \]
\[ W_{FLD} = \arg \max_{W} \frac{W^T S_B W}{W^T S_W W} \]
\[ W_{PCA} = \arg \max_{W} W^T S_W W \]

\[ W_{FLD} = \arg \max_{W} \frac{W^T W_{PCA} S_B W_{PCA} W}{W^T W_{PCA} S_W W_{PCA} W} \]

- PCA (Eigenfaces)
  \[ W_{FLD} = \arg \max_{W} \frac{W^T S_B W}{W^T S_W W} \]

Maximizes ratio of projected between-class to projected within-class scatter

PCA vs. FLD

- PCA (Eigenfaces) maximizes projected total scatter
- Fisher’s Linear Discriminant maximizes ratio of projected between-class to projected within-class scatter

Using Matlab:
- The \( w_i \) are orthonormal
- There are at most \( c-1 \) non-zero generalized Eigenvalues, so \( m \leq c-1 \)
- Can be computed with \( \text{eig} \) function in Matlab

Computing the Fisher Projection Matrix

\[ W_{FLD} = \arg \max_{W} \frac{W^T S_B W}{W^T S_W W} \]

where \( \{w_i \mid i = 1, 2, \ldots, m\} \) is the set of generalized eigenvectors of \( S_B \) and \( S_W \) corresponding to the \( m \) largest generalized eigenvalues \( \{\lambda_i \mid i = 1, 2, \ldots, m\} \), i.e.,

\[ S_B w_i = \lambda_i S_W w_i, \quad i = 1, 2, \ldots, m \]
Harvard Face Database

- 10 individuals
- 66 images per person
- Train on 6 images at 15°
- Test on remaining images

Recognition Results: Lighting Extrapolation

Appearance manifold approach

- For every object
  1. Sample the set of viewing conditions
  2. Crop & scale images to standard size
  3. Use as feature vector
- Apply a PCA over all the images
- Keep the dominant PCs
- Set of views for one object is represented as a manifold in the projected space
- Recognition: What is nearest manifold for a given test image?

Variability:
- Camera position
- Illumination
- Internal parameters
- Within-class variations

An example: input images

An example: basis images
An example: surfaces of first 3 coefficients

Parameterized Eigenspace

Recognition

Appearance-based vision for robot control

[ Nayar, Nene, Murase 1994 ]

Limitations of these approaches

- Object must be segmented from background (How would one do this in non-trivial situations?)
- Occlusion?
- The variability (dimension) in images is large, so is sampling feasible?
- How can one generalize to classes of objects?

Appearance-Based Vision: Lessons

Strengths

- Posing the recognition metric in the image space rather than a derived representation is more powerful than expected.
- Modeling objects from many images is not unreasonable given hardware developments.
- The data (images) may provide a better representations than abstractions for many tasks.
Appearance-Based Vision: Lessons

Weaknesses

• Segmentation or object detection is still an issue.
• To train the method, objects have to be observed under a wide range of conditions (e.g. pose, lighting, shape deformation).
• Limited power to extrapolate or generalize (abstract) to novel conditions.

Final Exam

• Closed book
• One cheat sheet
  – Single piece of paper, handwritten, no photocopying, no physical cut & paste.
• What to study
  – Basically material presented in class, and supporting material from text
  – If it was in text, but NEVER mentioned in class, it is very unlikely to be on the exam
• Question style:
  – Short answer
  – Some longer problems to be worked out.

Further Studies

• CSE166: Image Processing
• AI (CSE150,151)
• CSE159: Projects in Computer Vision