Recognition III

Introduction to Computer Vision
CSE 152
Lecture 20

Announcements

• Assignment 4: Due Friday

• Final Exam: Wed, 6/8/04, 3:00-6:00, in class room

• My last office hours: Tuesday, 6/7, 2:00-3:30

• I’ll discuss briefly exam today

A Rough Recognition Spectrum

Appearance-Based Recognition (Eigenface, Fisherface)

Local Features + Spatial Relations

Shape Contexts

Geometric Invariants

Image Abstractions/ Volumetric Primitives

3-D Modell-Based Recognition

Function

Increasing Generality

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

\[
y = Wx
\]

• From \( y \in \mathbb{R}^m \), the reconstruction of the point is \( W^Ty \)

• The error of the reconstruction is:

\[
||x - W^TWx||
\]

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

\[
\mu = \frac{1}{n} \sum x_i
\]

\[
\Sigma = \frac{1}{n-1} \sum (x_i - \mu)(x_i - \mu)^T
\]

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 \( v_k \) has the largest eigenvalue — give a set of features with the following properties:

• They are independent.

• Projection onto the basis \( \{v_1, v_2, \ldots, v_k\} \) gives the k-dimensional set of linear features that preserves the most variance.

Algorithm 22.5: Principal components 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 \( n < d \)

 Singular Value Decomposition

• 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 \( A A^T \)

• \( V \) \( n \times n \), orthogonal matrix,

• columns are the eigenvectors of \( A^T A \)

• \( \Sigma \) \( m \times n \), diagonal with non-negative entries \( (\sigma_1, \sigma_2, \ldots, \sigma_k) \) with \( s = \min(m, n) \) are called the called the singular values

• Singular values are the square roots of eigenvalues of both \( A A^T \) and \( A^T A \) & Columns of \( U \) are corresponding Eigenvectors!!

• 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 $A^TA$ & Columns of $U$ are corresponding Eigenvectors
- And $\sum a_i a_i^T = [a_1, a_2, \ldots, a_i, a_{i+1}, \ldots, a_n] = AA'$
- Covariance matrix is:
  $$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T$$
- So, ignoring $1/n$ subtract mean image from each input image, create data matrix, and perform (thin) SVD on the data matrix.

PCA & Fisher’s Linear Discriminant

- **PCA (Eigenfaces)**
  $$W_{PC} = \text{arg max}_w \frac{w^TS_w w}{W^TS_w W}$$
  Maximizes projected total scatter
- **Fisher’s Linear Discriminant**
  $$W_{FLD} = \text{arg max}_w \frac{w^TS_w w}{W^TS_w W}$$
  Maximizes ratio of projected between-class to projected within-class scatter

Computing the Fisher Projection Matrix

$$W_{opt} = \text{arg max}_w \frac{|w^TS_w w|}{W^TS_w W}$$

where $\{w_i, i = 1, 2, \ldots, m\}$ is the set of generalized eigenvectors of $S_w$ and $S_b$ corresponding to the $m$ largest generalized eigenvalues $\{\lambda_i, i = 1, 2, \ldots, m\}$, i.e.

$$S_w w_i = \lambda_i S_b w_i, \quad i = 1, 2, \ldots, m$$

- The $w_i$ are orthonormal
- There are at most $c-1$ non-zero generalized Eigenvalues, so $m \ll c-1$
- Can be computed with $\text{eig}$ in Matlab

Fisherfaces

- Since $S_w$ 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.

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?
An example: input images

An example: basis images

An example: surfaces of first 3 coefficients

Parameterized Eigenspace

Recognition

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?
Bayesian Classification

Discussed on blackboard, but slides may be helpful

Basic ideas in classifiers

• Generally, we should classify as 1 if the expected loss of classifying as 1 is better than for 2
  gives

\[
\begin{align*}
1 & \text{ if } p(1|x)L(1 \rightarrow 2) > p(2|x)L(2 \rightarrow 1) \\
2 & \text{ if } p(1|x)L(1 \rightarrow 2) < p(2|x)L(2 \rightarrow 1)
\end{align*}
\]

• Crucial notion: Decision boundary – points where the loss is the same for either case

Some loss may be inevitable: the minimum risk (shaded area) is called the Bayes risk

Example: known distributions

\[
p(k|x) \propto \frac{1}{2\pi} \mid \Sigma \mid^{\frac{1}{2}} \exp \left[ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right]
\]
Classifier boils down to:
choose class that minimizes:
\[ \delta(x, \mu_k) - 2 \log \pi_k \]
where
Mahalanobis distance
because covariance is common, this simplifies to sign of a linear expression (i.e. Voronoi diagram in 2D for \( \Sigma=I \))

Finding skin

- Skin has a very small range of (intensity independent) colours, and little texture
  - Compute an intensity-independent colour measure, check if colour is in this range, check if there is little texture (median filter)
  - See this as a classifier - we can set up the tests by hand, or learn them.
  - get class conditional densities (histograms), priors from data (counting)

- Classifier:
  - if \( p(\text{skin}) > \theta \), classify as skin
  - if \( p(\text{skin}) < \theta \), classify as not skin
  - if \( p(\text{skin}) = \theta \), choose classes uniformly and at random

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.

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.
Model-Based Vision

- Given 3-D models of each object
- Detect image features (often edges, line segments, conic sections)
- Establish correspondence between model & image features
- Estimate pose
- Consistency of projected model with image.

A Rough Recognition Spectrum

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

Recognition by Hypothesize and Test

- General idea
  - Hypothesize object identity and pose
  - Recover camera parameters (widely known as backprojection)
  - Render object using camera parameters
  - Compare to image
- Simplest approach
  - Construct a correspondence for all object features to every correctly sized subset of image points
  - These are the hypotheses
  - Expensive search, which is also redundant.
- Issues
  - Where do the hypotheses come from?
  - How do we compare to image (verification)?

Pose consistency

- Correspondences between image features and model features are not independent.
- A small number of correspondences yields a camera matrix --- the others correspondences must be consistent with this.
- Strategy:
  - Generate hypotheses using small numbers of correspondences (e.g. triples of points for a calibrated perspective camera, etc., etc.)
  - Backproject and verify

For all object frame groups $O$
For all image frame groups $F$
For all correspondences $C$ between elements of $F$ and elements of $O$
Use $F$, $C$ and $O$ to infer the missing parameters in a camera model
Use the camera model estimate to render the object
If the rendering conforms to the image, the object is present

Voting on Pose

- Each model leads to many correct sets of correspondences, each of which has the same pose
  - Vote on pose, in an accumulator array
  - This is a hough transform, with all its issues.
Invariance

- Properties or measures that are independent of some group of transformation (e.g., rigid, affine, projective, etc.)
- For example, under affine transformations:
  - Collinearity
  - Parallelism
  - Intersection
  - Distance ratio along a line
  - Angle ratios of two intersecting lines
  - Affine coordinates

- There are geometric properties that are invariant to camera transformations
  - Easiest case: view a plane object in scaled orthography.
  - Assume we have three base points \( P_i \) (\( i = 1, 2, 3 \)) on the object
    - then any other point on the object can be written as
      \[
      P_k = P_1 + \mu_{k1}(P_2 - P_1) + \mu_{k2}(P_3 - P_1)
      \]

Geometric hashing

- Vote on identity and correspondence using invariants
  - Take hypotheses with large enough votes
- Building a table:
  - Take all triplets of points in on model image to be base points \( P_1, P_2, P_3 \).
  - Take every fourth point and compute \( \mu \)'s
  - Fill up a table, indexed by \( \mu \)'s, with
    - the base points and fourth point that yield those \( \mu \)'s
    - the object identity

Verification

- Edge score
  - are there image edges near predicted object edges?
  - very unreliable, in texture, answer is usually yes
- Oriented edge score
  - are there image edges near predicted object edges with the right orientation?
  - better, but still hard to do well (see next slide)
- Texture
  - e.g. does the spanner have the same texture as the wood?
Application: Surgery

- To minimize damage by operation planning
- To reduce number of operations by planning surgery
- To remove only affected tissue
- **Problem**
  - ensure that the model with the operations planned on it and the information about the affected tissue lines up with the patient
  - display model information supervised on view of patient
  - **Big Issue:** coordinate alignment, as above
Matching using Local Image features
Simple approach
- Detect corners in image (e.g. Harris corner detector).
- Represent neighborhood of corner by a feature vector produced by Gabor Filters, K-jets, affine-invariant features, etc.).
- Modeling: Given an training image of an object w/o clutter, detect corners, compute feature descriptors, store these.
- Recognition time: Given test image with possible clutter, detect corners and compute features. Find models with same feature descriptors (hashing) and vote.

Probabilistic interpretation
- Write
  \[ P(\text{patch of type } i \text{ appears in image}/\text{this pattern is present}) = p_{xi} \]
  \[ P(\text{patch of type } i \text{ is present}) = p_i \]
- Assume
  \[ p_{xi} = \mu \text{ if the pattern can produce this patch and } 0 \text{ otherwise} \]
  \[ p_x = \lambda < \mu \text{ for all } i \]
- Likelihood that \( n_i \) patches come from that pattern and \( n_x = n \), patches come from noise, is
  \[ P(\text{interpretation}/\text{pattern}) = \lambda^n p^{n(n-1)/2} \]

Employ spatial relations

Example

Figure from “Local grayvalue invariants for image retrieval,” by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE
Finding faces using relations

- Strategy:
  - Face is eyes, nose, mouth, etc. with appropriate relations between them
  - build a specialised detector for each of these (template matching) and look for groups with the right internal structure
  - Once we’ve found enough of a face, there is little uncertainty about where the other bits could be

Strategy: compare

\[
\Pr(\text{face is } X_{10} = x_1, X_{12} = x_2, X_{13} = x_3, \text{all other responses})
\]

\[
\Pr(\text{face is } X_{11} = x_1, X_{12} = x_2, X_{13} = x_3, \text{all other responses})
\]

Notice that once some facial features have been found, the position of the rest is quite strongly constrained.


Even without shading, shape reveals a lot - line drawings

Scene Interpretation

“The Swing” Fragonard, 1766

Final Exam

- Closed book
- One cheat sheet
  - Single piece of paper, handwritten, no photocopying, no physical cut & paste. -- you can start with sheet from the midterm, if you want.
- 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