Appearance-Based Recognition

Computer Vision I

CSE252A

Lecture 19

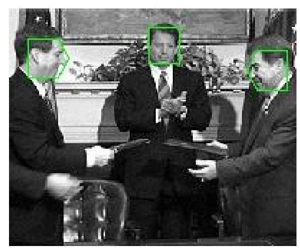
Announcements

- HW 4 Due Date extended to Friday
- Final Exam – next Friday, 3:00-6:00 here.
- TA Evaluation forms
- Some sources on Homogenous coordinates
- Introduction to Computer Graphics: A Mathematical Approach (maybe this is the title) by **Sam Buss.**
- Trucco & Verri,

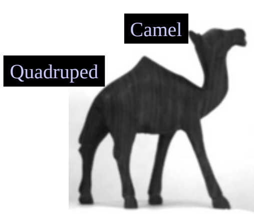
Recognition

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

Recognition



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

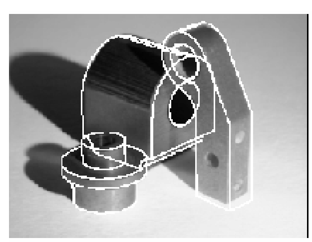


Problem:

Recognizing instances

Recognizing categories

Recognition

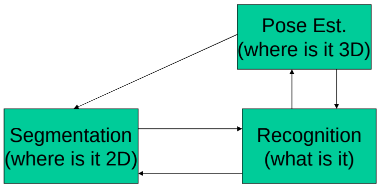


Given a database of objects and an image determine what, if any of the objects are present in the image.

Object Recognition: The Problem

Given: A database D of "known" objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object



WHAT AND WHERE!!!

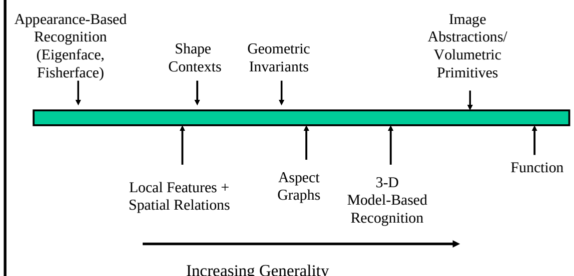
Recognition Challenges

- 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

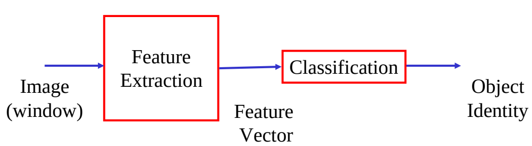
Object Recognition Issues:

- How general is the problem?
  - 2D vs. 3D
  - range of viewing conditions
  - available context
  - segmentation cues
- What sort of data is best suited to the problem?
  - Whole images
  - Local 2D features (color, texture,
  - 3D (range) data
- What information do we have in the database?
  - Collection of images?
  - 3-D models?
  - Learned representation?
  - Learned classifiers?
- How many objects are involved?
  - small: brute force search
  - large: ??

A Rough Recognition Spectrum



Sketch of a Pattern Recognition Architecture



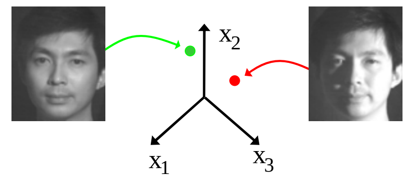
Example: Face Detection

- Scan window over image.
- Classify window as either:
  - Face
  - Non-face

Pattern Classification Summary

- Supervised vs. Unsupervised: Do we have labels?
- Supervised
  - Nearest Neighbor
  - Bayesian
    - Plug in classifier
    - Distribution-based
    - Projection Methods (Fisher's, LDA)
  - Neural Network
  - Support Vector Machine
  - Kernel methods
- Unsupervised
  - Clustering
  - Reinforcement learning

Image as a Feature Vector

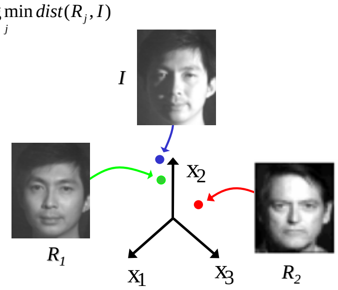


- Consider an n-pixel image to be a point in an n-dimensional space, $\mathbf{x} \in \mathbf{R}^n$.
- Each pixel value is a coordinate of **x**.

Nearest Neighbor Classifier

$\{ R_j \}$ are set of training images.

$ID = \arg\min_j dist(R_j, I)$



Comments

- Sometimes called "Template Matching"
- Variations on distance function (e.g. L_1, robust distances)
- Multiple templates per class- perhaps many training images per class.
- Expensive to compute k distances, especially when each image is big (N dimensional).
- May not generalize well to unseen examples of class.
- Some solutions:
  - Bayesian classification
  - Dimensionality reduction

Bayesian Classification

Blackboard

Basic ideas in classifiers

- Loss
  - some errors may be more expensive than others
    - e.g. a fatal disease that is easily cured by a cheap medicine with no side-effects -> false positives in diagnosis are better than false negatives
  - We discuss two class classification: L(1->2) is the loss caused by calling 1 a 2
- Total risk of using classifier s

$$R(s) = Pr\{1 \to 2 | \text{using } s\} L(1 \to 2) + Pr\{2 \to 1 | \text{using } s\} L(2 \to 1)$$

Basic ideas in classifiers

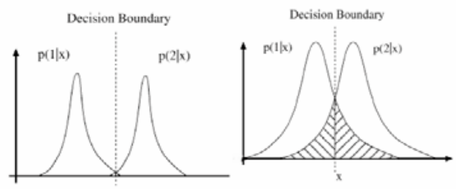
- Generally, we should classify as 1 if the expected loss of classifying as 1 is better than for 2
- gives

1 if $p(1|\boldsymbol{x})L(1 \to 2) > p(2|\boldsymbol{x})L(2 \to 1)$

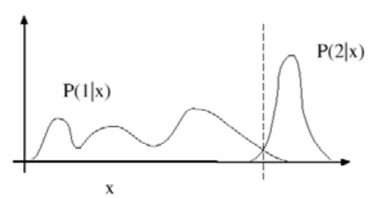
2 if $p(1|\boldsymbol{x})L(1 \to 2) < p(2|\boldsymbol{x})L(2 \to 1)$

- 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



Finding a decision boundary is not the same as modeling a conditional density.



Example: known distributions

$$p(\underline{x}|k) = \left(\frac{1}{2\pi}\right)^{-p/2} \left|\underline{\underline{\Sigma}}\right|^{-1/2} \exp\left[-\frac{1}{2}\left(\underline{x} - \underline{\mu}_k\right)^T \underline{\underline{\Sigma}}^{-1}\left(\underline{x} - \underline{\mu}_k\right)\right]$$

- Assume normal class densities, p-dimensional measurements with common (known) covariance and different (known) means
- Class priors are π_k
- Can ignore a common factor in posteriors - important; posteriors are then:

$$p(k \mid x) \propto (\pi_k)\left(\frac{1}{2\pi}\right)^{-p/2} \left|\underline{\underline{\Sigma}}\right|^{-1/2} \exp\left[-\frac{1}{2}\left(\underline{x} - \underline{\mu}_k\right)^T \underline{\underline{\Sigma}}^{-1}\left(\underline{x} - \underline{\mu}_k\right)\right]$$

- Classifier boils down to: choose class that minimizes:

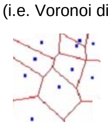
$$\delta\left(\underline{x}, \underline{\mu}_k\right)^2 - 2\log \pi_k$$

where

Mahalanobis distance

$$\delta\left(\underline{x}, \underline{\mu}_k\right) = \left[\left(\underline{x} - \underline{\mu}_k\right)^T \underline{\underline{\Sigma}}^{-1}\left(\underline{x} - \underline{\mu}_k\right)\right]^{1/2}$$

because covariance is common, this simplifies to sign of a linear expression (i.e. Voronoi diagram in 2D for Σ=I)



Plug-in classifiers

- Assume that distributions have some parametric form - now estimate the parameters from the data.
- Common:
  - assume a normal distribution with shared covariance, different means; use usual estimates
  - ditto, but different covariances; ditto
- Issue: parameter estimates that are “good” may not give optimal classifiers.

Example: Face Detection

- Scan window over image.
- Classify window as either:
  - Face
  - Non-face

Example: Finding skin Non-parametric Representation of CCD

- Skin has a very small range of (intensity independent) colors, and little texture
  - Compute an intensity-independent color measure, check if color 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 is
  - if $p(\text{skin}|\boldsymbol{x}) > \theta$, classify as skin
  - if $p(\text{skin}|\boldsymbol{x}) < \theta$, classify as not skin
  - if $p(\text{skin}|\boldsymbol{x}) = \theta$, choose classes uniformly and at random

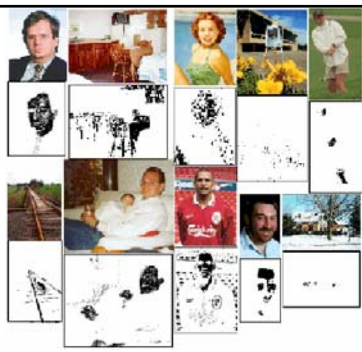


Figure from "Statistical color models with application to skin detection," M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE

Receiver **O**perating **C**urve

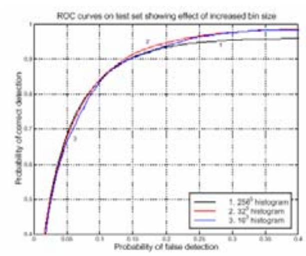


Figure from "Statistical color models with application to skin detection," M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE

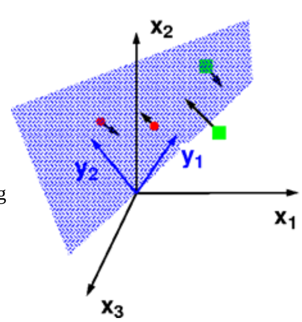
Eigenfaces: linear projection

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

$$\boldsymbol{y} = W\boldsymbol{x}$$

where W is an n by m matrix.

- Recognition is performed using nearest neighbor in $\mathbf{R}^m$.
- How do we choose a good W?



Eigenfaces: Principal Component Analysis (PCA)

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

$$\boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{x}_i$$

$$\Sigma = \frac{1}{n-1}\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

The unit eigenvectors of Σ — which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue — give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ 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 text to compute basis when n<<d