Recognition
(Part 3)

Introduction to Computer Vision
CSE 152
Lecture 16

Announcements
• Homework 4 is due today, 11:59 PM
• Reading:
  – Chapter 15: Learning to Classify
  – Chapter 16: Classifying Images
  – Chapter 17: Detecting Objects in Images

A Rough Recognition Spectrum

Appearance-Based Recognition
(Eigenface, Fisherface)

Shape
Contexts

Geometric
Invariants

Local Features + Spatial Relations

Image Abstractions/
Volumetric Primitives

Aspect
Graphs

3-D Model-Based Recognition

Function

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

Recognition by Hypothesize and Test

• General idea
  – Hypothesize object identity and pose
  – Recover camera parameters
  – Render object using camera parameters
  – Compare to image
• Issues
  – Where do the hypotheses come from?
  – How do we compare to image (verification)?

• 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

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)
  – Recover camera parameters (e.g., calibrated camera rotation and translation) and verify
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 (similar to a Hough transform)
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 three intersecting lines
  - Affine coordinates

Geometric hashing

- Vote on identity and correspondence using invariants
  - Take hypotheses with large enough votes
- Building a table:
  - Take all triplets of points on model image to be base points $P_1$, $P_2$, and $P_3$
  - Take every fourth point and compute $\mu_{ka}$ and $\mu_{kb}$
  - Fill up a table, indexed by $\mu_{ka}$ and $\mu_{kb}$, with
    - The base points and fourth point that yielded $\mu_{ka}$ and $\mu_{kb}$
    - The object identity

Algorithm 18.3: Geometric hashing: voting on identity and point labels

For all groups of three image points $T(j)$
For every other image point $p$
  Compute the $\mu$'s from $p$ and $T(j)$
  Obtain the table entry at those values
  if there is one, it will label the three points in $T(j)$
  with the name of the object
  and the names of those particular points.
  Cluster these labels
  if there are enough labels, backproject and verify
end
end
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
- Texture
  - For example, 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

Figures by kind permission of Eric Grimson; further information can be obtained from his web site http://www.ai.mit.edu/people/welg/welg.html.
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 without clutter, detect corners, and compute and store feature descriptors
• 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

\[ P(\text{patch of type } i \text{ appears in image}/\text{pattern is present}) = p_i \]

\[ P(\text{patch of type } i \text{ is pattern in present}) = \mu_i \]

• Assume

\[ p_i = \mu \text{ if the pattern can produce this patch and 0 otherwise} \]
\[ \mu_i = \lambda < \mu \text{ for all } i. \]

• Likelihood of image given pattern

that \( n_i \) patches come from that pattern and \( n_0 \) patches come from noise, is

\[ P(\text{interpretation}/\text{pattern}) = \lambda^r \mu^{n_r - r} \]

Employ spatial relations

Finding faces using relations

• Strategy:
  – Face is eyes, nose, mouth, etc. with appropriate relations between them
  – Build a specialized detector for each of these (template matching) and look for groups with the right internal structure
  – Once a face is detected, there is little uncertainty about where the other parts could be

Finding faces using relations

• Strategy: compare

\[ P(\text{face at } \mathbf{X}_e = x_1, \mathbf{X}_n = x_2, \mathbf{X}_m = x_3, \mathbf{X}_b = x_4, \text{all other responses}) \]

\[ P(\text{non face}; \mathbf{X}_e = x_1, \mathbf{X}_n = x_2, \mathbf{X}_m = x_3, \mathbf{X}_b = x_4, \text{all other responses}) \]

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

Next Lecture

• Color
• Reading:
  – Chapter 3: Color