Recognition (Part 3)

Introduction to Computer Vision

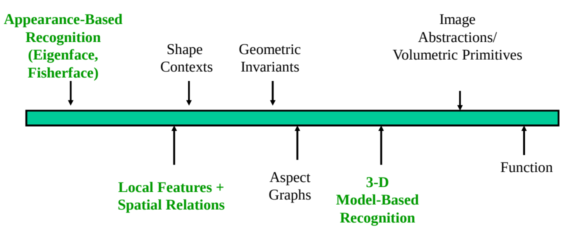
CSE 152

Lecture 16

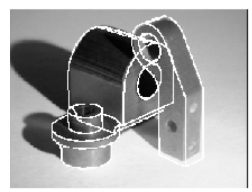
Announcements

- 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



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

```
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
    end
  end
end
```


Example

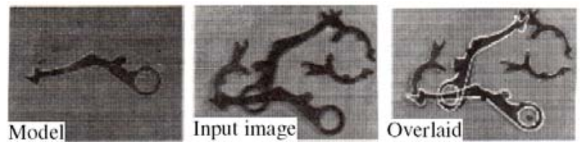


Figure from "Object recognition using alignment," D.P. Huttenlocher and S. Ullman, Proc. Int. Conf. Computer Vision, 1986, copyright IEEE, 1986

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)


```
For all objects O
  For all object frame groups F(O)
    For all image frame groups F(I)
      For all correspondences C between
        elements of F(I) and elements
        of F(O)

        Use F(I), F(O) and C to infer object pose P(O)

        Add a vote to O's pose space at the bucket
        corresponding to P(O).
      end
    end
  end
end
For all objects O
  For all elements P(O) of O's pose space that have
    enough votes

    Use the P(O) and the
    camera model estimate to render the object

    If the rendering conforms to the image,
    the object is present
  end
end
```

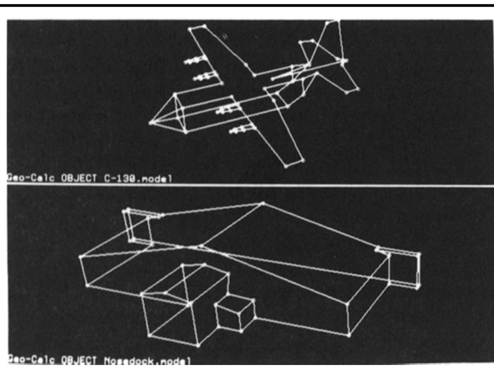



Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

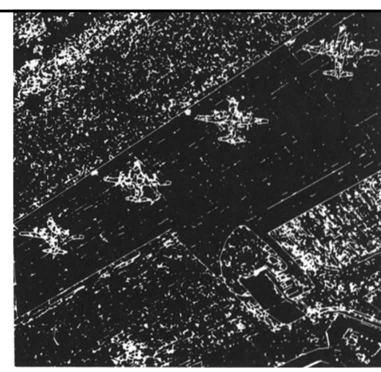


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

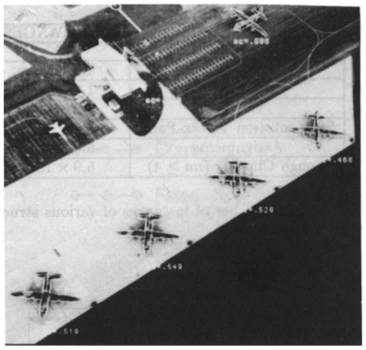


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

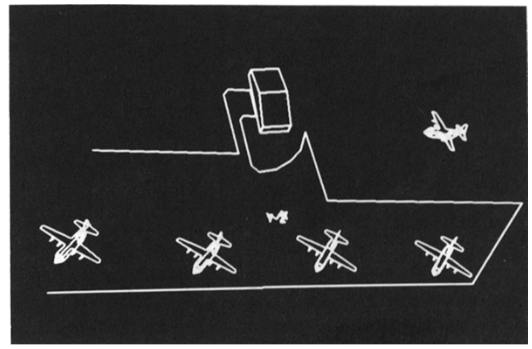


Figure from "The evolution and testing of a model-based object recognition system", J.L. Mundy and A. Heller, Proc. Int. Conf. Computer Vision, 1990 copyright 1990 IEEE

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

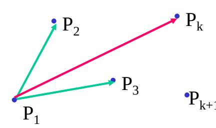
Invariance

- 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..3) on the object
  - Any other point on the object can be written as

$$P_k = P_1 + \underline{\mu_{ka}}(P_2 - P_1) + \underline{\mu_{kb}}(P_3 - P_1)$$

- Now image points are obtained by multiplying by a planar affine transformation, so

$$\begin{aligned} p_k &= AP_k \\ &= A\left(P_1 + \mu_{ka}(P_2 - P_1) + \mu_{kb}(P_3 - P_1)\right) \\ &= p_1 + \mu_{ka}(p_2 - p_1) + \mu_{kb}(p_3 - p_1) \end{aligned}$$



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 μ_{ka} and μ_{kb}
  - Fill up a table, indexed by μ_{ka} and μ_{kb}, with
    - The base points and fourth point that yielded μ_{ka} and μ_{kb}
    - The object identity

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

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

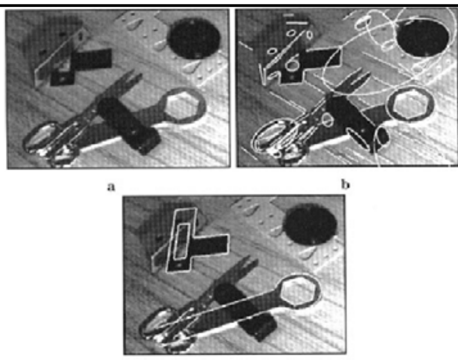
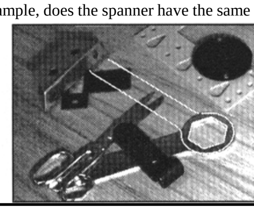


Figure from “Efficient model library access by projectively invariant indexing functions,” by C.A. Rothwell et al., Proc. Computer Vision and Pattern Recognition, 1992, copyright 1992, IEEE

Verification

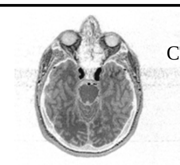
- Edge score
  - Are there image edges near predicted object edges?
  - Very unreliable; in texture, answer is usually yes
- Oriented edge score
  - Are there image edges near predicted object edges with the right orientation?
  - Better, but still hard to do well
- Texture
  - For example, does the spanner have the same texture as the wood?



Application: Surgery

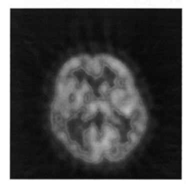
- To minimize damage by operation planning
- To reduce number of operations by planning surgery
- To remove only affected tissue
- Problem
  - Ensure that the model with the operations planned on it and the information about the affected tissue lines up with the patient
  - Display model information supervised on view of patient
  - **Big Issue**: coordinate alignment, as above

MRI



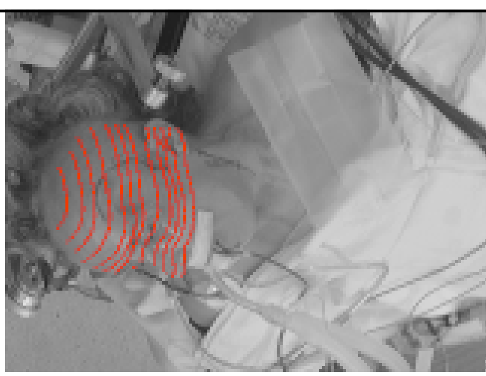
CTI

NMI

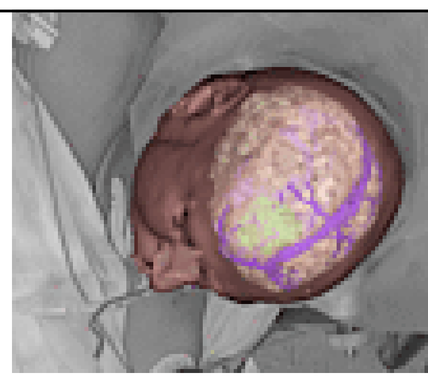


USI

Reprinted from Image and Vision Computing, v. 13, N. Ayache, “Medical computer vision, virtual reality and robotics”, Page 296, copyright, (1995), with permission from Elsevier Science



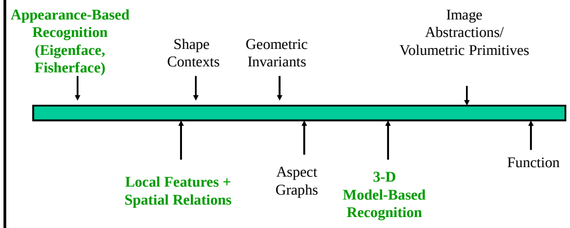
Figures by kind permission of Eric Grimson; further information can be obtained from his web site http://www.ai.mit.edu/people/welg/welg.html.



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.

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.

A Rough Recognition Spectrum



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

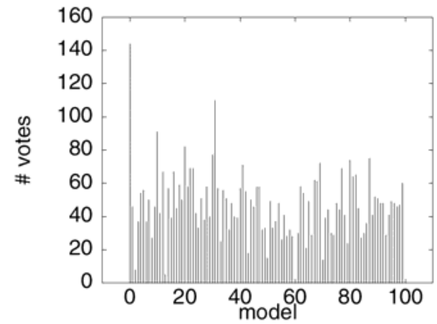


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

Probabilistic interpretation

- Write

$$P\{\text{patch of type } i \text{ appears in image} | j\text{'th pattern is present}\} = p_{ij}$$

$$P\{\text{patch of type } i | \text{no pattern is present}\} = p_{ix}$$

- Assume

$p_{ij} = \mu$ if the pattern can produce this patch and 0 otherwise
$p_{ix} = \lambda < \mu$ for all i.

- Likelihood of image given pattern

that n_p patches came from that pattern and $n_i - n_p$ patches come from noise, is

$$P(\text{interpretation}|\text{pattern}) = \lambda^{n_p}\mu^{(n_i - n_p)}$$

Employ spatial relations

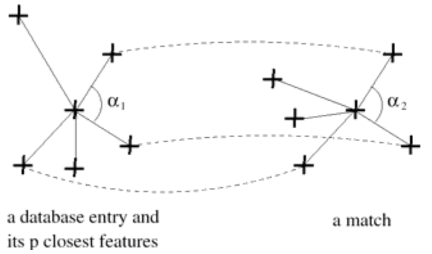
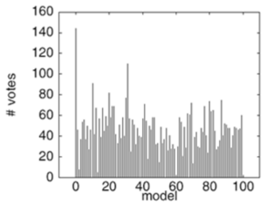


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE



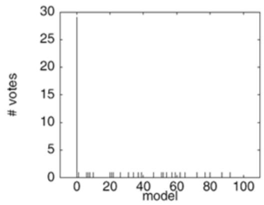


Figure from "Local grayvalue invariants for image retrieval," by C. Schmid and R. Mohr, IEEE Trans. Pattern Analysis and Machine Intelligence, 1997 copyright 1997, IEEE

Example

Training examples

Test image

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

$$P(\text{one face at } \boldsymbol{F} | \boldsymbol{X}_{\text{le}} = \boldsymbol{x}_1, \boldsymbol{X}_{\text{re}} = \boldsymbol{x}_2, \boldsymbol{X}_{\text{m}} = \boldsymbol{x}_3, \boldsymbol{X}_{\text{n}} = \boldsymbol{x}_4, \text{all other responses})$$

with

$$P(\text{no face} | \boldsymbol{X}_{\text{le}} = \boldsymbol{x}_1, \boldsymbol{X}_{\text{re}} = \boldsymbol{x}_2, \boldsymbol{X}_{\text{m}} = \boldsymbol{x}_3, \boldsymbol{X}_{\text{n}} = \boldsymbol{x}_4, \text{all other responses})$$

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

Figure from, "Finding faces in cluttered scenes using random labelled graph matching," by Leung, T. ;Burl, M and Perona, P., Proc. Int. Conf. on Computer Vision, 1995 copyright 1995, IEEE

Figure from, "Finding faces in cluttered scenes using random labelled graph matching," by Leung, T. ;Burl, M and Perona, P., Proc. Int. Conf. on Computer Vision, 1995 copyright 1995, IEEE

Next Lecture

- Color
- Reading:
  - Chapter 3: Color