# Object Recognition



## Object Recognition in Living Creatures

- Most important aspect of visual perception
- Least understood
- Young children can recognize large variety of objects
  - Child can generalize from a few examples of dogs to many dogs under a variety of visual conditions
- Insects such as bees use visual recognition for navigation and finding its home, identifying flower shapes



## Goals of Object Recognition

- Goal is to retrieve information that is not apparent in the images we perceive.
- The name of things is one piece of information
- Animals recognize without words. Important information may be whether to ignore, eat, flee, etc.
- A robot could connect the objects it sees to the information it knows about them, and also connect new information about objects to what it already knows about them.



## Object Recognition with Computers

- Recognition of common objects is way beyond capability of artificial systems proposed so far
- How do we program a computer to capture the essence of a dog, a house or a tree?



## Object Recognition Issues

- What makes object recognition difficult?
- Are there different types of object recognition?
- How can different views of an object including views that we never saw be identified as representing a single object?



## Multiple Mechanisms

- Characteristic shape
  - Faces, printed character
- Color pattern, texture
  - Tiger, giraffe, skin of toad
- Branching patterns: trees in winter
- Various material types
  - Montain terrain (rocks), lake scenery (reflections)
- Location relative to other objects
  - Door knob, even if it is shaped like a duck head
- Characteristic motion: fly in a room

## Other Methods

- Expectations, prior knowledge
  - White thing on desk in the dark has to be sheet of paper
- Reasoning
  - Thing has to be a fence because it surrounds a field

7

## Multiple Facets of Recognition

- Visual object recognition is not a single mechanism
- Diversity of approaches used in computer vision should parallel the diversity of paths leading to object recognition by humans, using different sources of observations

8

## Shape

- Most common objects can be recognized in isolation, without use of context or expectations
- Without use of color, texture, motion
  - Dancing pink elephant with stripes in Dumbo
- Recognition from shape may be most common and important aspect

9

## Why is Recognition Difficult

- Is more computational power a solution?
- Assume a large and efficient memory system
  - Store a sufficient number of different views
  - Does the image corresponds to something we have seen in the past?
    - Compare image with all views in memory
  - But image comparison is not enough to solve the problem because of large variations between images of single object

10

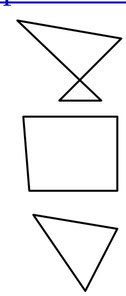
## Large Scale Memory

- Large scale memory *is* important
  - Pigeons can learn to sort a set of 320 slides of natural scenes in 2 arbitrary categories, remember it after 2 years
  - Fly can remember visual patterns
    - Direct comparison

11

## Problems with Direct Comparison

- Space of all possible views of all objects is very large
  - Change in viewing direction produces large differences in appearance
- Image not similar enough to the one seen in the past
- Background is different and there are occlusions
- Deformation: human body, scissors
- Illumination: human faces



12

## Problems with Direct Comparison

- For faces, difference due to viewing conditions may be much larger than differences between individuals
  - Using distance between faces based on pixel differences, machine recognition is poor
  - For humans, recognition is highly accurate and variations of illuminations are not noticed.

13

## Three Classes of Recognition Methods

- **Alignment methods**
- **Invariant properties methods**
- **Parts decompositions methods**

Taxonomy of ideas, not of recognition systems

- Systems may combine methods from the 3 classes

14

## Examples for 3 Classes

- Alignment methods
  - Using points: triangles (cf. class notes for Object Pose)
  - For rounded objects with smooth contours
    - Cups, toy cars, teddy bears (R. Nelson)
- Invariant properties methods
  - Color indexing (Swain)
  - Salient points (Swain)
  - Geometric hashing (Landam and Wolfson)
- Parts decompositions methods
  - Body Plans (Forsyth and Fleck)

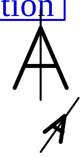
15

## Alignment Approach

- For each model, set of allowable transformations
- Compensate for transformations separating viewed object and stored model
- Search for model and transformation to maximize a measure of fit between object and model
- Transformations are explicitly applied to stored model

16

## Simplified Character Recognition



- Given input character, alignment phase
  - "Undo" shift, scale and rotation transformations
  - Undo shift with center of mass
  - Undo scale using area of convex hull
  - For orientation, horizontal symmetry (A), vertical symmetry (B), direction of vertical straight lines (F), horizontal straight lines (Z)
- When pose has been compensated for, check alignment of model and image
  - Some parts may be given more weight: tail of Q distinguishes from O

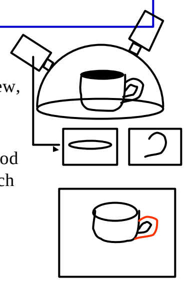
17

## 3D Image-Model Alignment

- Given:
  - A 3-D object modeled as a collection of points
  - Image of a scene suspected to include an instance of the object, segmented into feature points
- Goal
  - **Hypothesize** the pose of the object in the scene by matching (collections of) $n$ model points against $n$ feature points, enabling us to solve for the rigid body transformation from the object to world coordinate systems, and
  - **Verify** that hypothesis by projecting the remainder of the model into the image and matching (check if projection is *aligned* with image)

18

## Image Alignment for Smooth Objects



- Use 100 views of objects as models
- Extract contour fragments from each view, and store them *along with camera pose*
- In images, detect contour fragments, match them to contours in database. Good matches increase score of object to which contour belongs, if pose is consistent.
- Select objects with highest score
- Verify by projecting objects in image
  - Good recognition results with hundreds of possible objects



## Invariant Properties and Feature Vectors

- Properties that are common to many views
  - Colors, color contiguities
  - Compactness measure for cells seen on microscope
  - Moments (inertia of shape computed wrt. axes or points)
- Define a number of such measures
  - "Features" = measurements
  - Measurements that change a lot with view are not very useful; should lie within a restricted range
  - Invariant measures should be easy to measure



## Examples

- Geometric features
  - Elongation, perimeter length, shape moments
  - OK for flat un-occluded parts only



## Example of Invariants Method: Color Indexing

- Also called backprojection algorithm
- Swain and Ballard,1990
- Use color information instead of pure spatial information

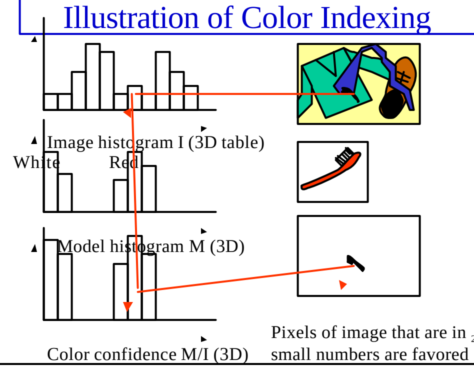


## Color Indexing Technique

- Let M be color histogram of model
- Let I be the color histogram of whole image (3D)
- Build M/I: Each bin of color *i* is replaced by the ratio $M_i / I_i$ : for color I, pixel count in model divided by pixel count in image
  - Confidence value: How much color I is characteristic of model
  - If bin $I_i$ has a lot more pixels than $M_i$, low confidence value: most don't come from the modeled object
- Replace each pixel of color *i* by its confidence value
- Smooth confidence image
- Expected locations of model should appear as peaks in confidence image



## Illustration of Color Indexing



## Extensions of Color Indexing

- In Color Indexing, we measure 3 color components at every pixel, then build a histogram
- We can collect a more complex feature vector at every pixel
  - Apply masks to measure color gradients in 2 orthogonal directions
  - Apply mask to measure Laplacian
    - This defines components of a local feature vector
- Construct histograms of feature vector for image and model
  - More dimensions than color histograms
- Locate object from cluster of pixels with high confidence value as in color indexing



## Example 2: Salient Point Method

- Find most salient point of model
  - For every pixel, define a high-dimensional feature vector
  - For every pixel, find the distance of its feature vector to all the others.
  - Keep as salient point the pixel with the largest distance to the others
- Locating a model in image:
  - For every image pixel, find feature vector
  - Calculate distance from feature vector of every pixel to salient point of model
  - Select pixel with minimum distance to salient point of model as candidate point corresponding to salient point
- This is a "focus of attention" mechanism. A more complete recognition method can be used in the region around the detected salient point.



## Example 3: Geometric Hashing

- Uses *affine* projection model
  - Flat objects "far" from camera
  - Objects may be at an angle with respect to camera optical axis



## Special Homography: Affine Transformation

$$P_w = RP_o + T$$

$$[X_w, Y_w, Z_w] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$X_w = r_{11}X_0 + r_{12}Y_0 + r_{13}Z_0 + t_x$$

And the image coordinates of $(X_w, Y_w, Z_w)$ are

$$x = fX_w / Z_w = f\,\frac{r_{11}X_0 + r_{12}Y_0 + r_{13}Z_0 + t_x}{r_{31}X_0 + r_{32}Y_0 + r_{33}Z_0 + t_z}$$



## Special Homography: Affine Transformation

- $P$ is "far" from the camera. Then in the denominator of these expressions, $t_z$ dominates. So we rewrite them as:

$$x = [f\,\underbrace{r_{11}/t_z}_{}]X_0 + [\underbrace{f\,r_{12}/t_z}_{b}]Y_0 + \underbrace{t_x/t_z}_{t_1}$$

$$y = [\underbrace{f\,r_{21}/t_z}_{c}]X_0 + [\underbrace{f\,r_{22}/t_z}_{d}]Y_0 + \underbrace{t_y/t_z}_{t_2}$$

$Z_0 = 0$ (planar object in plane $OX_0Y_0$)

- This is an affine transformation $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_1 \\ c & d & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ 1 \end{bmatrix}$



## Properties of Affine Transformation

- With non projective coordinates, mapping from point M to point M' is

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \Rightarrow \begin{array}{l} \mathbf{M'} = \mathbf{AM} + \mathbf{T} \\ \mathbf{M_0'} = \mathbf{AM_0} + \mathbf{T} \end{array}$$

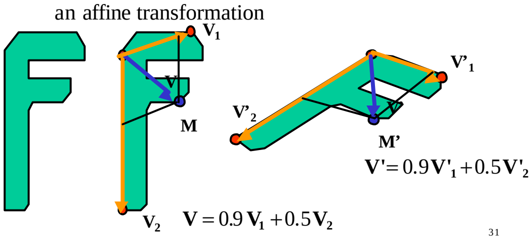
- Mapping from vector $\mathbf{M_0M}$ to $\mathbf{M'_0M'}$ is

$$\mathbf{M'_0\,M'} = \mathbf{A\ M_0M}$$

- $\mathbf{V} = a_1\mathbf{V_1} + a_2\mathbf{V_2} \Rightarrow \mathbf{AV} = a_1\mathbf{AV_1} + a_2\mathbf{AV_2} \Rightarrow \mathbf{V'} = a_1\mathbf{V'_1} + a_2\mathbf{V'_2}$
- Therefore, components $a_1$ and $a_2$ of a point M are invariant in an affine transformation

## Invariants in Geometric Hashing

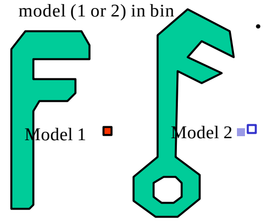
- $\mathbf{V}_1$ and $\mathbf{V}_2$ define a *basis*
- Components $(a_1, a_2)$ of a point M are invariant in an affine transformation



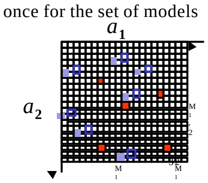



## Building a Table from Models

- Coordinate pairs are "signatures" or "keys" of models
  - We use these invariants to detect models
- For each model
  - For each basis (3 points), consider each feature point, find 2 coordinates. They locate a bin in a table. Store index of model (1 or 2) in bin




- Expensive (order $m^4$) but done only once for the set of models




## Using the Table for Recognition

- Pick 3 feature points from the image to define a basis.
- Compute coordinate pairs of all remaining image feature points with respect to that basis.
- Use these coordinates to access bins in the table
  - In a bin, we may find the index of model $M_i$ - if the corresponding 3 points in model $M_i$ were used as basis, and the corresponding point in the model was considered when building the table
- Repeat for all plausible triples of feature points
- Keep track of scores of each model $M_i$ encountered
- Models that obtain high scores are recorded as possible detections



## Plus and Minus of Invariants

- Plus: no storing of a set of views
- Minus: no ideal set of measurements we can apply to all objects. No universal features independent of viewing position and depending only on nature of object
  - What simple invariances would distinguish a fox from a dog?



## Parts and Structural Descriptions

- Many objects seem to contain natural parts
  - Face contains eyes, nose, mouth
  - These can be recognized on their own
  - Then recognition of object can use identified parts



## Part Decomposition Assumptions

- Each object can be decomposed into a small set of generic components
  - Generic: all objects can be described as different combinations of same components
  - Stable decomposition: decomposition is preserved across views of object
- Parts can be classified independently from whole object

## From Parts to Structure

- Two main approaches
  - Repeat decomposition process:
    - Certain parts are decomposed into simpler parts
  - Identify low-level parts, then group them to form higher-level parts

37

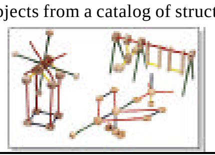
## Recognition Process

- Describe objects in terms of constituent parts
- Locate parts
- Classify them into different types of generic components
- Check relationships between parts
- Select objects for which structure matches detected relationships best

38

## Advantages

- Parts are simpler to detect than whole object, vary less with change of view
- Variability of object views is due to variability of structure, and structure can be detected by connectivity between parts
  - If we can recognize Tinkertoy elements, then we can recognize objects from a catalog of structures



39

## Relations between Parts

- The *relations between parts* are the invariants
  - Letter A:
    - 3 line segments
    - 2 line segments meet at vertex
- Invariances are expressed in terms of relations between two or more parts
  - Above, to the left of, longer than, containing, …

40

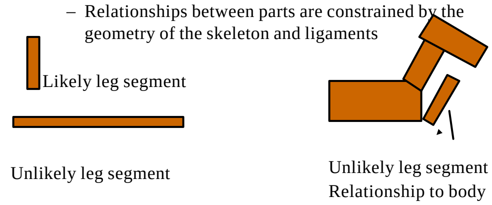
## 2D and 3D Relations

- For 2D applications, distances and angles
- For 3D applications, “connected together”, “larger than”, “inside of” remain invariant over a wide range of viewing positions
- This allows to distinguish between configurations of similar parts in different arrangements
  - Fundamental to human visual system
  - Pigeons recognize successfully people, trees, pigeons, letters, but don’t make distinction between figure and scrambled version: recognition from local parts, not structure

41

## Example of Structural Approach: Recognizing Horses using Body Plans

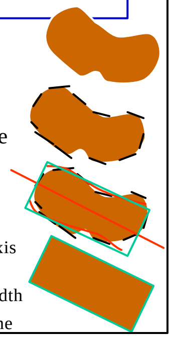
- Animals can be viewed as an assembly of nearly cylindrical parts (seen as rectangles in images)
  - Proportions of individual parts are constrained
  - Relationships between parts are constrained by the geometry of the skeleton and ligaments




42

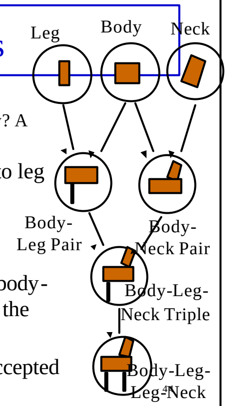
## Detecting Body Segments

- Identify regions that could be *hide* (horse skin)
  - Color and texture properties
- Inside skin regions, apply Canny edge detector
- Find coarse oriented rectangles
  - Find ribbons with edges that are symmetrical with respect to a straight axis using a Hough transform
  - Rectangle width is average of ribbon width
  - Rectangle length is length of ribbon spine



## Body Plans

- One classifier for each body part
  - Is this segment possibly a leg? A body? A neck?
- One classifier for connecting body to leg
  - Does this leg have the right angle and proportion with respect to this body
- Classifier for body-neck
- Classifier for body-leg-neck, using body-neck and body-leg inputs that share the same body
- Classifier for body-leg-leg-neck. Accepted groups are recognized as horses




## Classifier Training

- Body segments are defined by a vector with components
  - Centroid x and y, rectangle width and height, angle
- Support Vector Machine (SVM) classifiers are used
- Training images from CD “Arabian horses” of Corel photo library

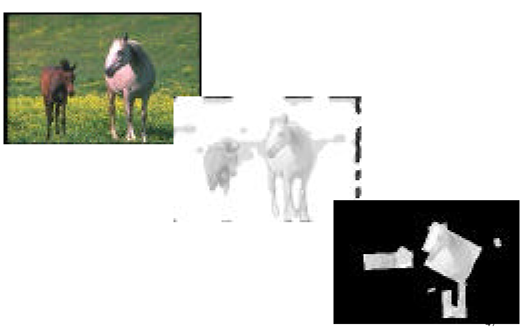
45

## Evaluation

- Rectangular body segment robust to perspective foreshortening
- Hierarchical classification is much more efficient than attempting to classify every grouping of 4 body segments
- Results are not too good:
  - Image collection containing 100 images with horses and 1000 images without horses
  - Horse recognition system would return 15 horse images and 7 non-horse images.

46

## Experiments



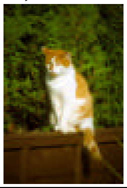
## Problems with Part Decomposition

- Decomposition falls sort of characterizing object specifically enough
  - Dog and cat have similar parts
  - Differentiation is possible if we check detailed shape at particular locations (such as the snout)

48

## Other Problems

- Many objects do not decompose naturally into a union of clearly distinct parts
  - What is a decomposition of a shoe
- Finding parts such as limbs, torso reliably is very difficult
- Useful, but insufficient



49

## Which Approach is Best?

- Invariants, parts description, alignment?
- No single best scheme is appropriate for all cases
- Recognition system must exploit the regularities of given domain
- In humans, several agents using different techniques work in parallel. If one agent succeeds, we are not aware of those that failed

50

## References


- High Level Vision: Object Recognition and Visual Cognition, Shimon Ullman, MIT Press, 1996.
- M.J. Swain and D.H. Ballard. Indexing via Color Histogram. Proc. ICCV, pp. 390-393, 1990.
- F. Ennesser and G. Medioni. Finding Waldo, or Focus of Attention using Local Color Information. PAMI 17, 8, 1995.
- M.J. Swain, C.H. Frankel and M. Lu. View-Based Techniques for Searching for Objects and Textures (Salient Points). http://people.cs.uchicago.edu/~swain/pubs
- D.A. Forsyth and M.M. Fleck. Body Plans. Proc. CVPR 1997. http://www.cs.berkeley.edu/~daf/book3chaps.html


51