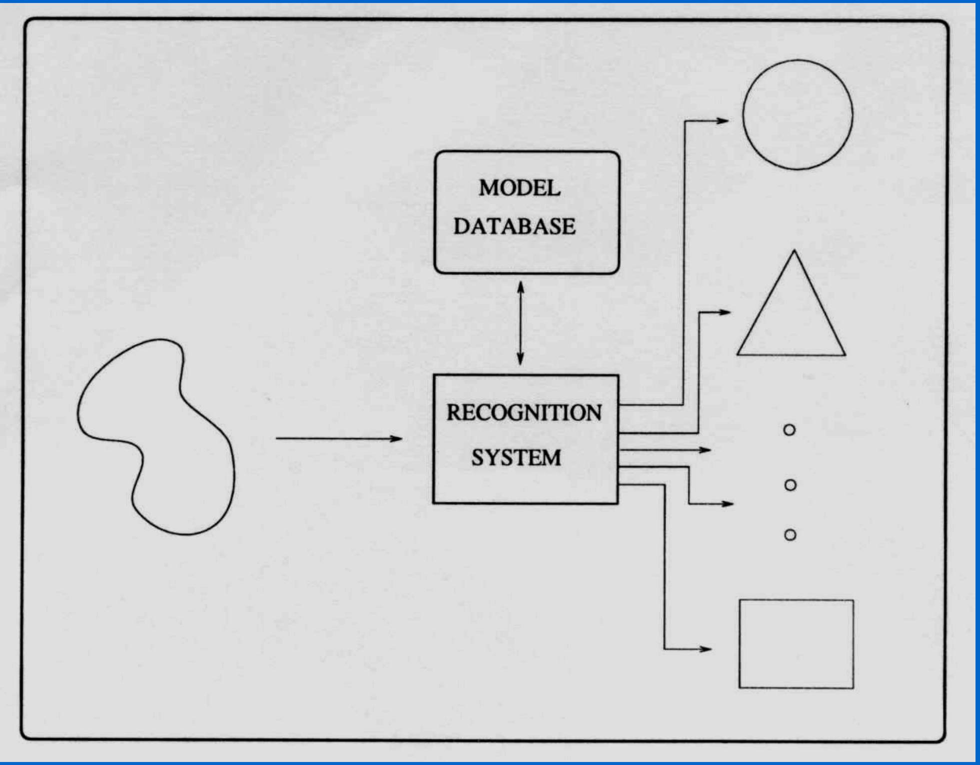
Object Recognition Method

Object Recognition

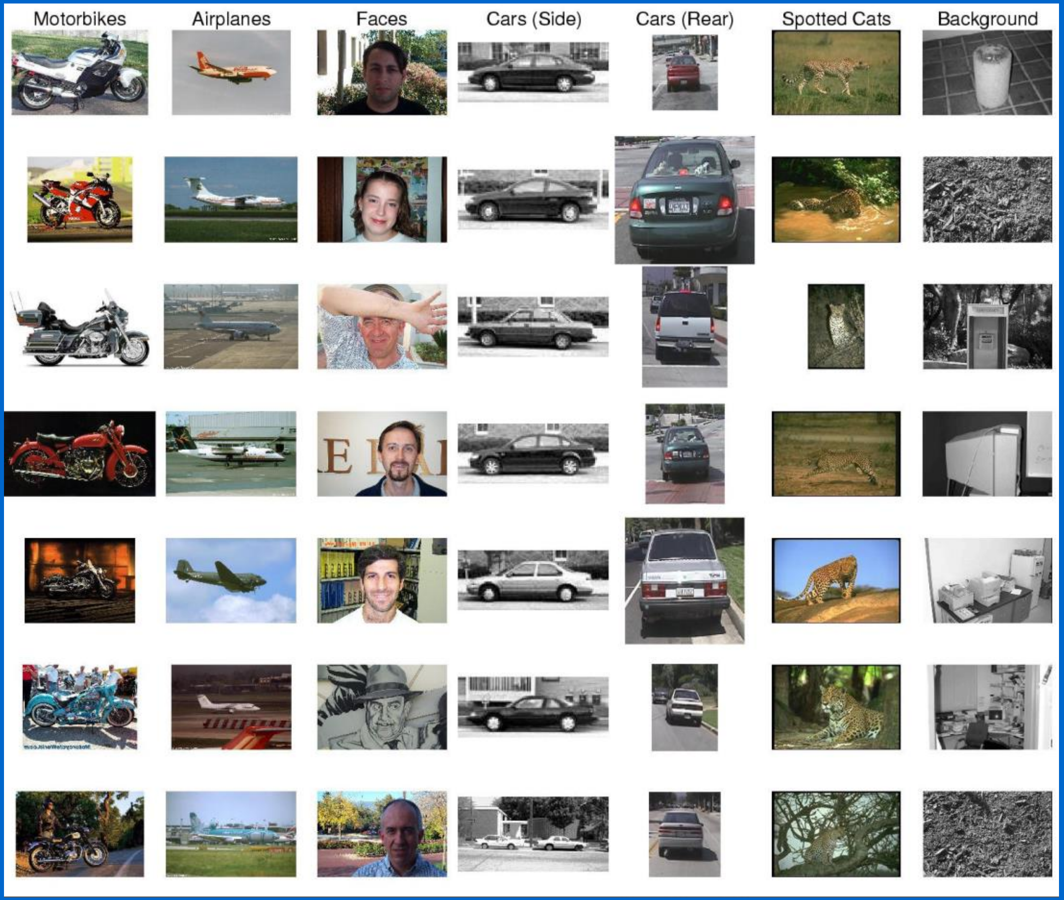
- Model-based Object Recognition
- Generic Object Recognition

Model-Based Object Recognition

- Recognition relies upon the existence of a set of predefined objects.



Generic Object Recognition (or Object Categorization)



Main Steps

- **Preprocessing**
  - A model database is built by establishing associations between features and models.
- **Recognition**
  - Scene features are used to retrieve appropriate associations stored in the model database.

Challenges

- The appearance of an object can have a large range of variation due to:
  - viewpoint changes
  - shape changes (e.g., non-rigid objects)
  - photometric effects
  - scene clutter
- Different views of the same object can give rise to widely different images!

Requirements

- Invariance
  - Geometric transformations (translation, rotation, scale)
    - Caused by viewpoint changes due to camera/object motion
- Robustness
  - Noise (i.e., sensor noise)
  - Detection errors (e.g., edge or corner detection)
  - Illumination/Shadows
  - Partial occlusion (i.e., self and from other objects)
  - Intrinsic shape distortions (i.e., non-rigid objects)

Performance Criteria

(1) Scope

– What kind of objects can be recognized and in what kind of scenes ?

(2) Robustness

– Does the method tolerate reasonable amounts of noise and occlusion in the scene ?
– Does it degrade gracefully as those tolerances are exceeded ?

Performance Criteria (cont'd)

(3) Efficiency

– How much time and memory are required to search the solution space ?

(4) Accuracy

– Correct recognition
– False positives (wrong recognitions)
– False negatives (missed recognitions)

Approaches Differ According To:

- Restrictions on the form of the objects
  - 2D or 3D objects
  - Simple vs complex objects
  - Rigid vs deforming objects
- Representation schemes
  - Object-centered
  - Viewer-centered

Approaches Differ According To: (cont'd)

- Matching scheme
  - Geometry-based (i.e., shape)
  - Appearance-based (e.g., eigenfaces)

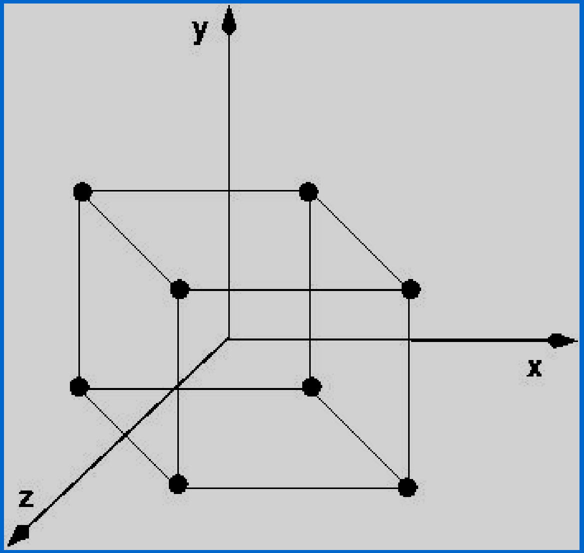
- Type of features
  - Local
  - Global

Approaches Differ According To: (cont'd)

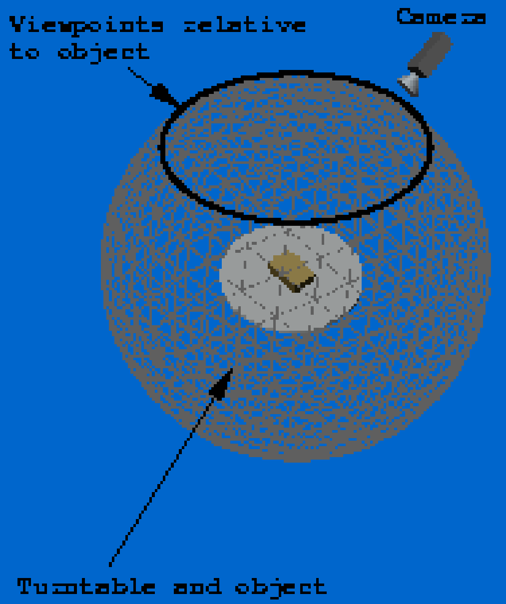
- Image formation model
  - Perspective projection
  - Orthographic projection + scale
  - Affine transformation (e.g., for planar objects)

Representation Schemes

(1) Object-centered



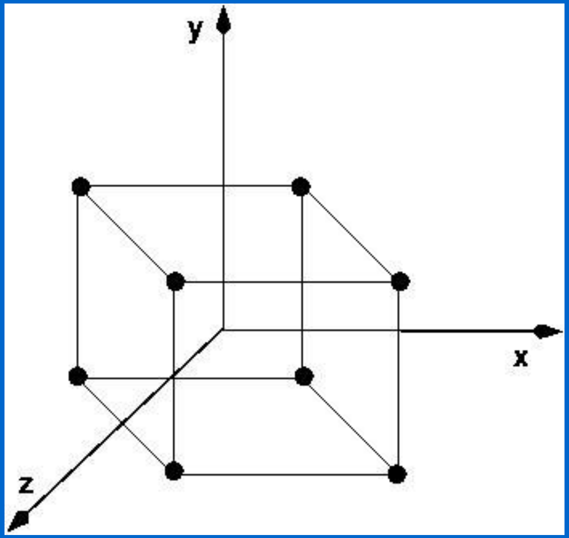
(2) Viewer-centered



R. Chin and C. Dyer, "Model-based recognition in robot vision," **Computing Surveys**, vol. 18, no. 1, pp. 67–108, 1986.

Object-centered Representation

- A 3D model of the object is available.



Advantage: every view of the object is available.

Disadvantage: might not be easy to build.

Object-centered Representation (cont'd)

- Two different matching approaches:

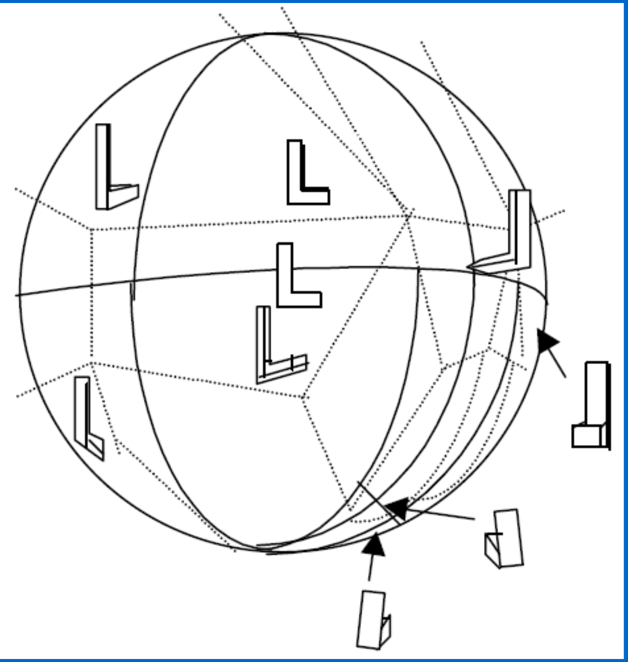
(1) (3D/3D): reconstruct object from the scene and match it with the models.

(2) (3D/2D): **back-project** candidate model onto the scene and match it with the objects in the scene.

i.e., requires camera calibration

Viewer-centered Representation

- Objects are described by a set of characteristic views or aspects



<u>Advantages:</u>

- Easier to build compared to object-centered.
- Matching is easier since it involves 2D information.

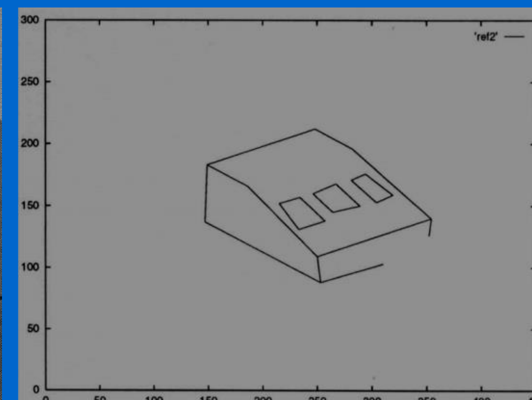
<u>Disadvantages:</u>

- Requires a large number of views.

Matching Schemes

(1) Geometry-based

- Employ geometric features



(2) Appearance-based

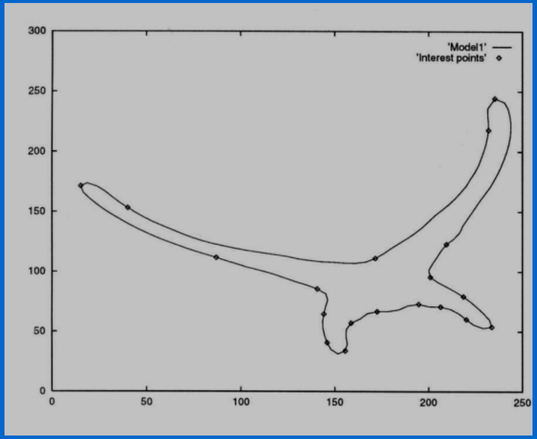
- Represent objects from many possible viewpoints and illumination directions using dimensionality reduction.

Geometry-based Recognition

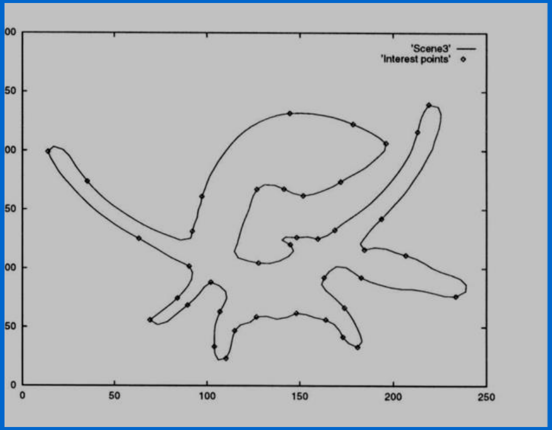
(1) Identify a group of features from an unknown scene which approximately match a group of features from a model object (i.e., **correspondences**).

(2) Recover the **geometric transformation** that the model object has undergone and look for additional matches.

model



scene

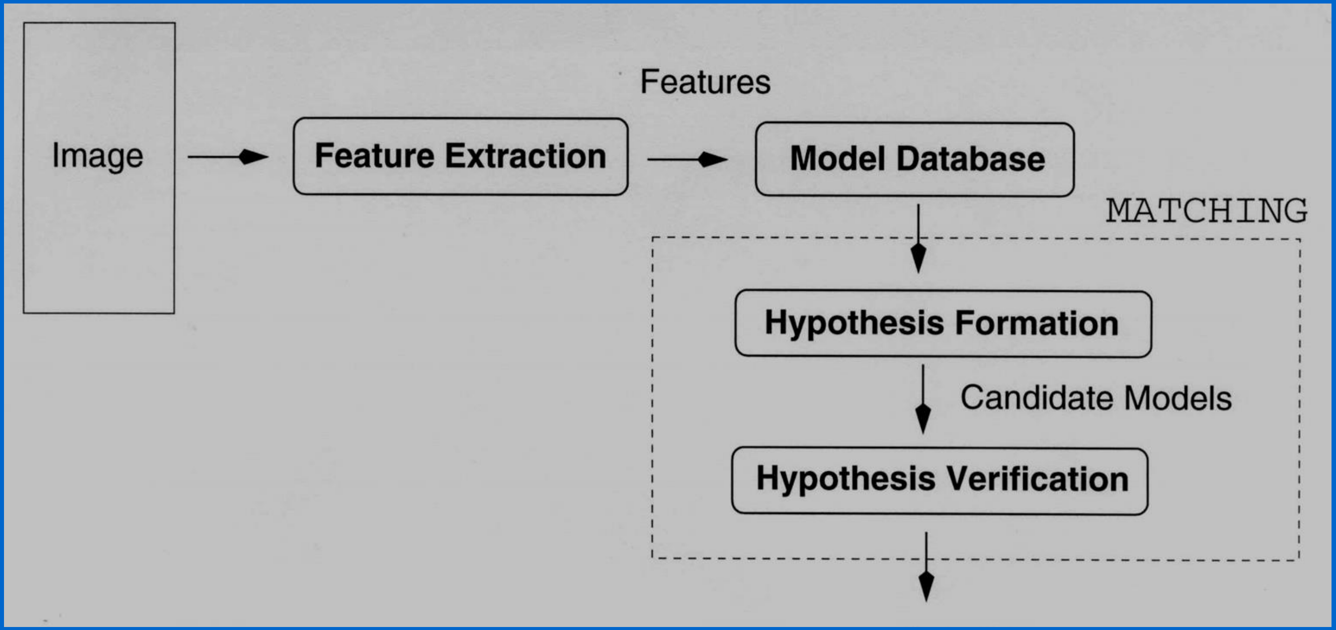


2D Transformation Spaces

- Rigid transformations (3 parameters)
- Similarity transformations (4 parameters)
- Affine transformations (6 parameters)

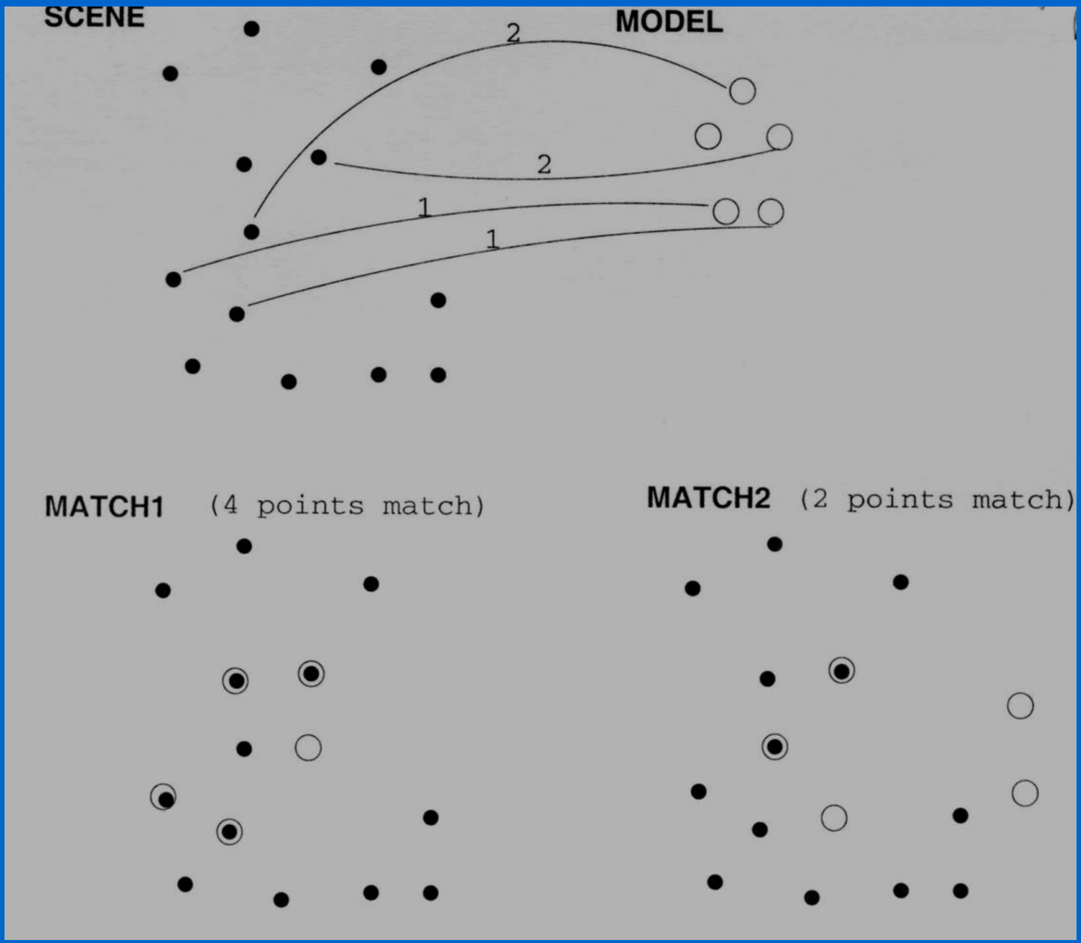
Matching – Main Steps

- **Hypothesis generation:** the identities of one or more models are hypothesized.
- **Hypothesis verification:** tests are performed to check if a given hypothesis is correct or not.



Hypothesis Verification - Example

two hypotheses:



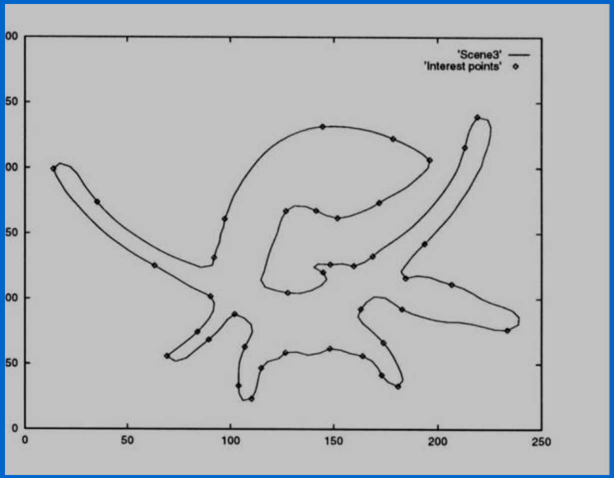
Features might correspond to:

(1) curvature extrema or zero-crossings along the boundary of an object .

(2) interest points.

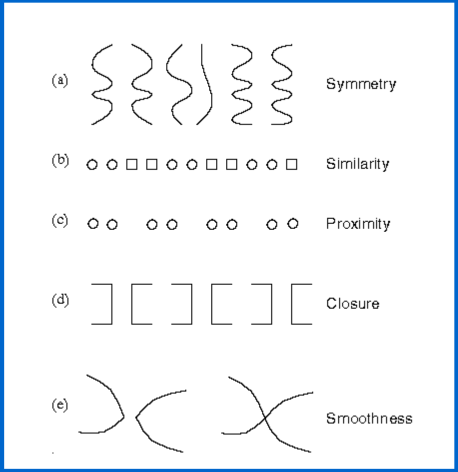
Hypothesis Generation

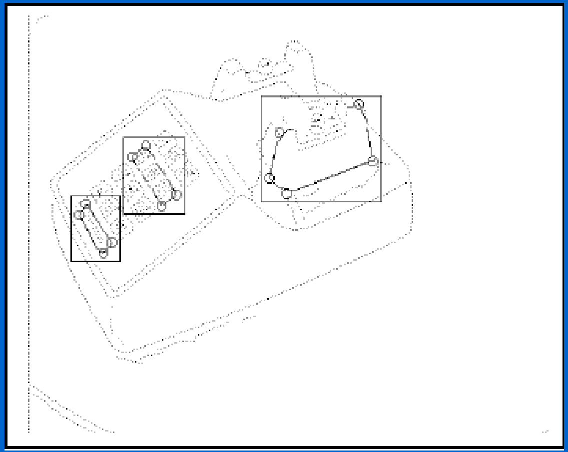
- How to **choose** the scene groups?
  - Do we need to consider every possible group?
  - How to find groups of features that are likely to belong to the same object?
  - **“Grouping”** schemes might be helpful.
  - **“Local descriptors”** might be helpful.



Grouping

- Employs non-accidental properties based on perceptual organization rules to extract groups of features that are likely to come from the same object.
  - e.g., orientation, co-linearity, parallelism, proximity, convexity

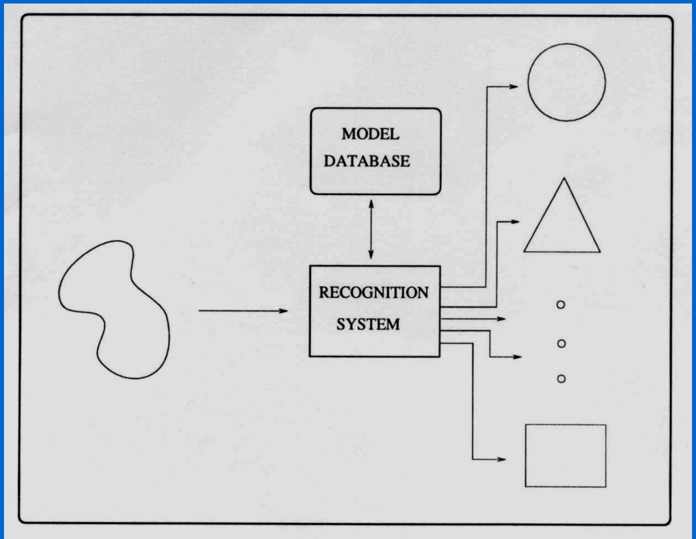




D. Jacobs, Robust and Efficient Detection of Convex Groups, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. 18, no. 1, pp. 23-37, 1996.

Hypothesis Generation (cont'd)

- How to **organize** and **search** the model database?
  - Do we need to search the whole database of models?
  - How should we organize the model database to allow for fast and efficient storage and retrieval?
  - **"Indexing"** schemes are helpful.



Object Recognition using SIFT features

1. Match individual SIFT features from an image to a database of SIFT features from known objects (i.e., find nearest neighbors).

2. Find <u>**clusters**</u> of SIFT features belonging to a single object (hypothesis generation).

Verification

- Back-project model on the scene and look for additional matches.
- Discard outliers (i.e., incorrect matches) by imposing stricter matching constraints (e.g., half error).
- Find additional matches by refine the transformation computed (i.e., iterative **affine refinements**).

Verification (cont'd)

- Evaluate **probability** that match is correct.
  - Use Bayesian (probabilistic) model, to estimate the probability that a model is present based on the actual number of matching features.
  - Bayesian model takes into account:
    - Object size in image
    - Textured regions
    - Model feature count in database
    - Accuracy of fit

Lowe, D.G. 2001. Local feature view clustering for 3D object recognition.**IEEE Conference on Computer Vision and Pattern Recognition**, Kauai, Hawaii, pp. 682–688.

Planar recognition

- Training images (models)

Planar recognition

- Reliably recognized at a rotation of 60° away from the camera.
- Affine fit approximates perspective projection.
- Only 3 points are needed for recognition.

3D object recognition

- Training images

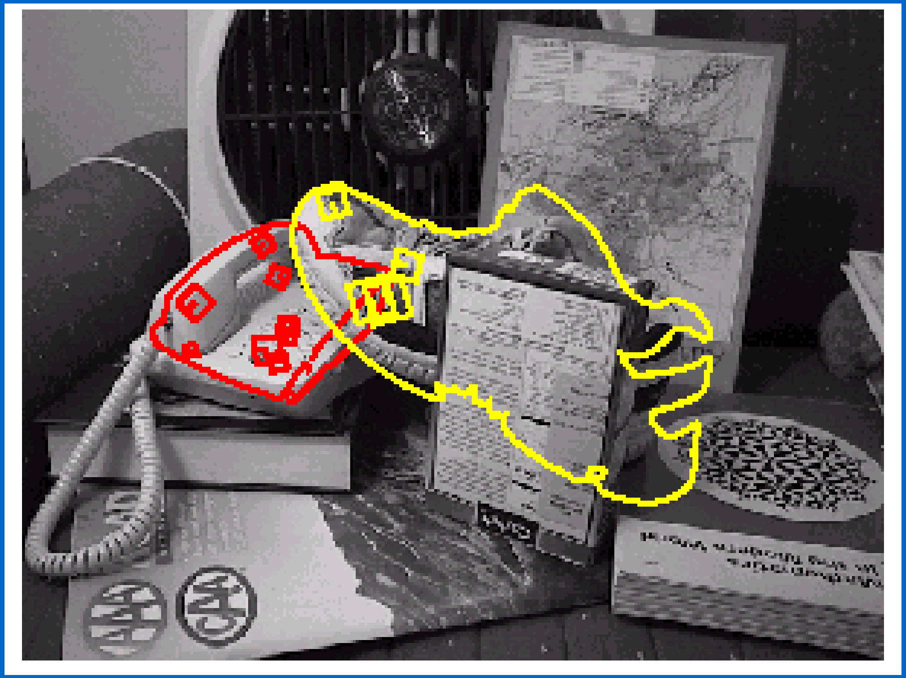
3D object recognition

- Only 3 keys are needed for recognition; extra keys provide robustness.

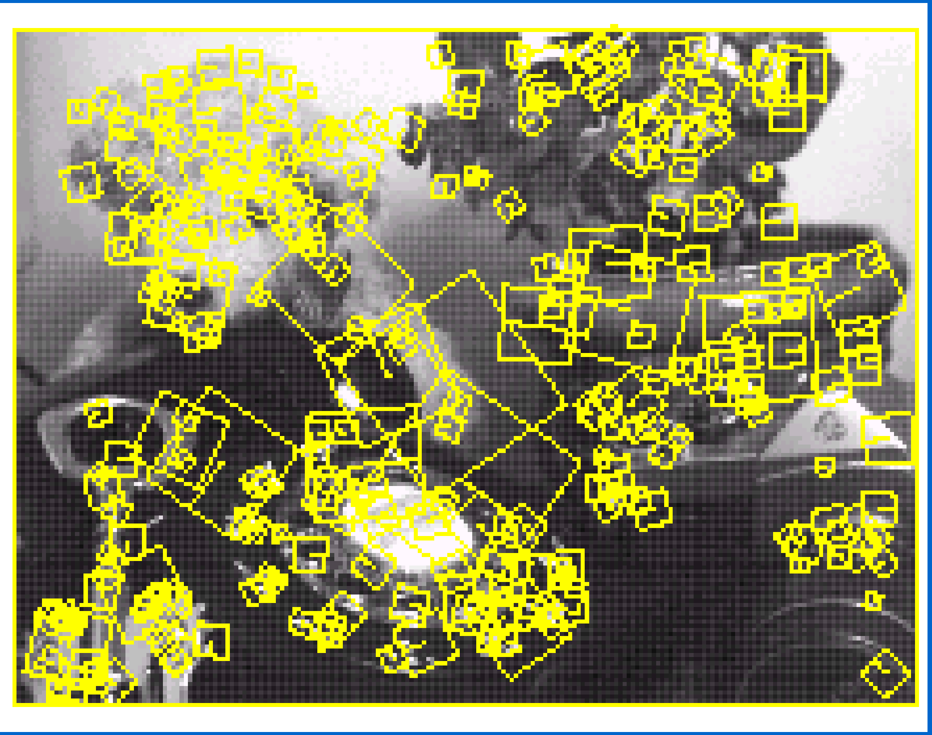
Recognition under occlusion

What is Occlusion in Computer Vision?

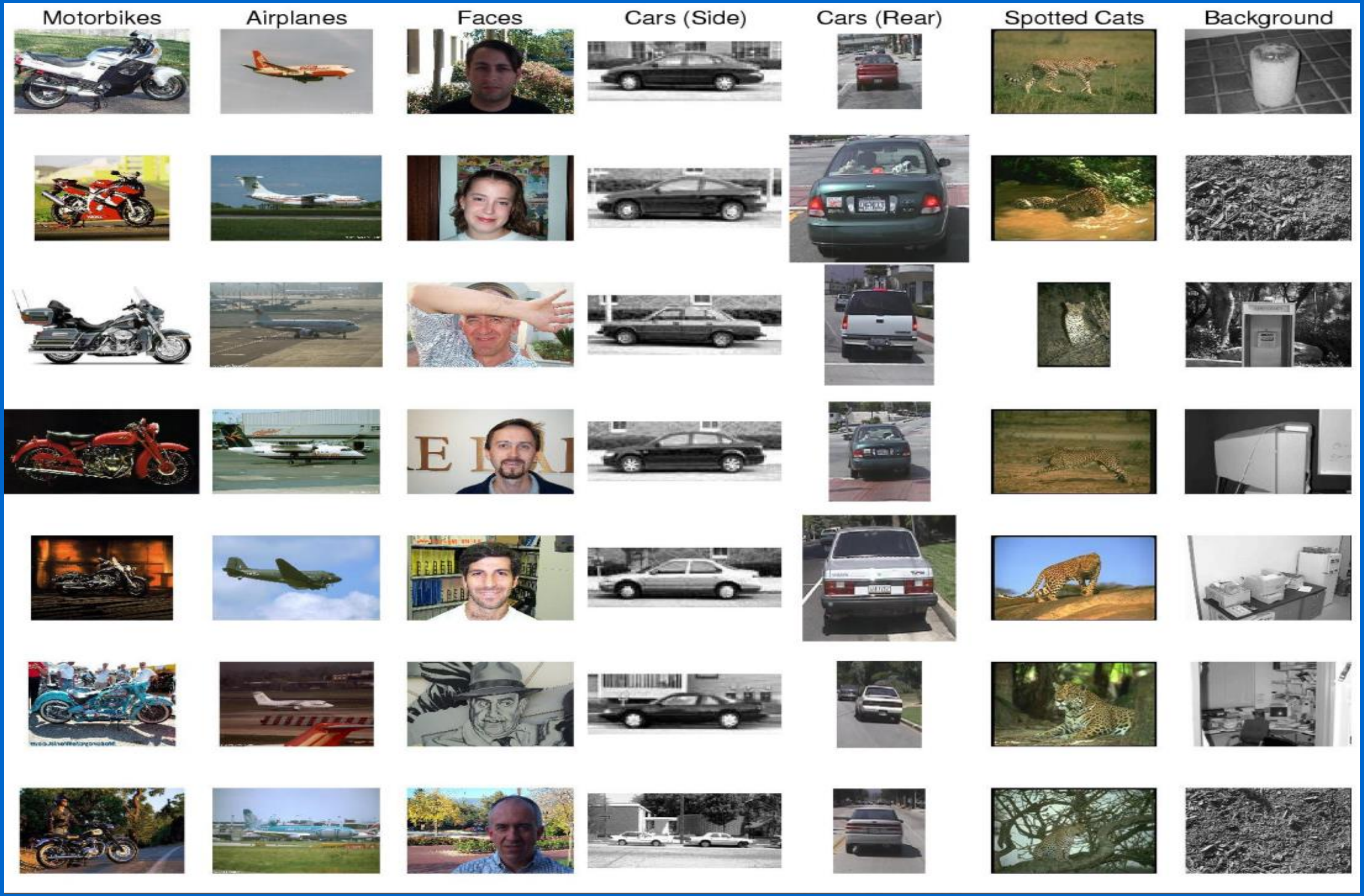
Occlusion techniques in computer vision **block a portion of an image** during **training time**.



Illumination invariance



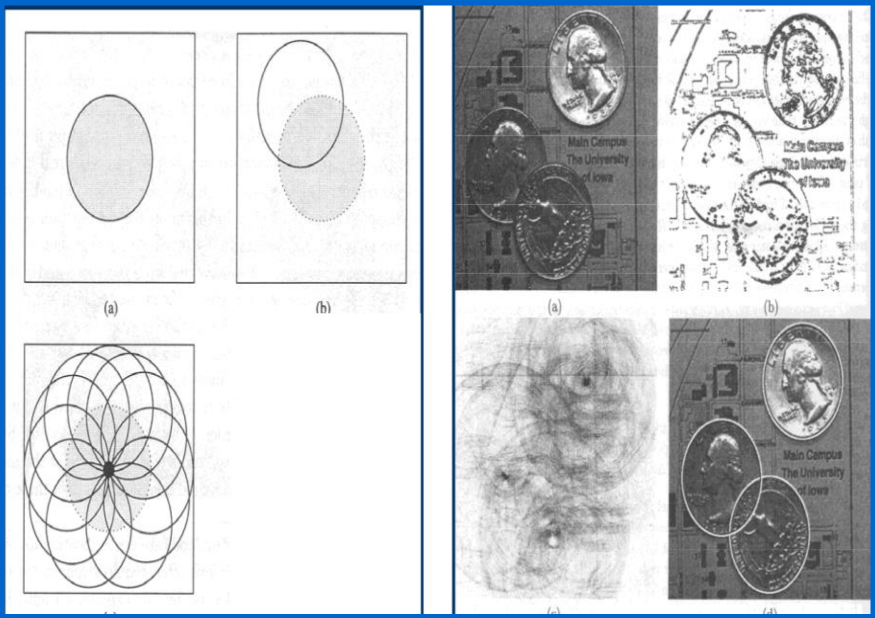
Object Categorization



Hough transforms Techniques in Object recognition

History

- The Hough transform was first proposed by **Paul Hough in 1962** as a method for detecting lines in images. It was later extended to detect other shapes like circles and ellipses.
- The technique has since been widely used in image processing applications, particularly in the areas of computer vision and pattern recognition.



- Hough Transform is a computer vision technique to detect shapes like lines and circles in an image.
- It converts these shapes into mathematical representations in a parameter space, making it easier to identify them even if they're broken or obscured.

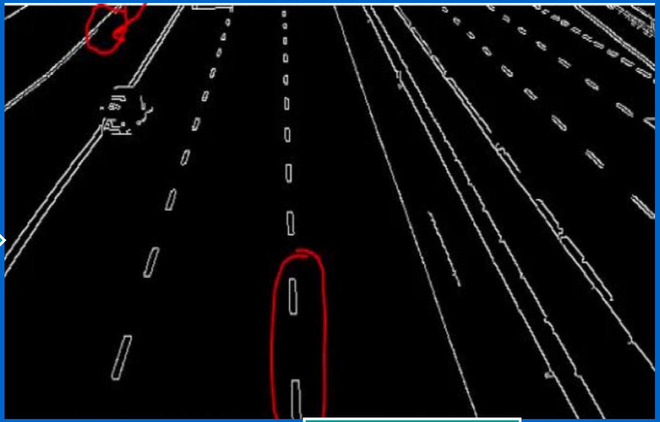
Hough transform Steps

- **Preprocess the image**: Before applying the Hough transform, it is recommended to preprocess the image to reduce noise, enhance edges, and improve contrast. Common preprocessing techniques include filtering, thresholding, and edge detection.
- **Choose the appropriate variant**: It is important to choose the appropriate variant of the Hough transform for the specific application.
- **Select parameters carefully**: It is important to select these parameters carefully to achieve the desired level of accuracy and efficiency such as the threshold value, the minimum line length, and the maximum gap between line segments
- **Use multiple scales**: If the shape being detected varies in size, it may be useful to use a multi-scale Hough transform that detects shapes at different scales.
- **Combine with other techniques**: The Hough transform can be combined with other image processing techniques, such as template matching, machine learning, or feature extraction, to improve the accuracy and efficiency of shape detection.
- **Validate the results**: The Hough transform can sometimes produce false positives or false negatives, especially in noisy or complex images.

Red circles show that the line is breaking there.

Original Image of the lane

after applying edge detection technique



After using Hough Transform