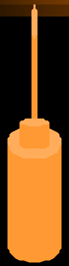
3D Models and Matching

- representations for 3D object models
- particular matching techniques
- alignment-based systems
- appearance-based systems



GC model of a screwdriver

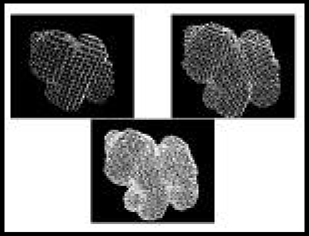
3D Models

- Many different representations have been used to model 3D objects.
- Some are very coarse, just picking up the important features.
- Others are very fine, describing the entire surface of the object.
- Usually, the recognition procedure depends very much on the type of model.

Mesh Models

Mesh models were originally for computer graphics.

With the current availability of range data, they are now used for 3D object recognition.



What types of features can we extract from meshes for matching ?

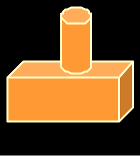
In addition to matching, they can be used for verification.

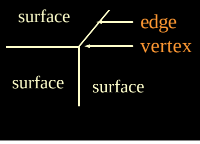
Surface-Edge-Vertex Models

SEV models are at the opposite extreme from mesh models.

They specify the (usually linear) features that would be extracted from 2D or 3D data.

They are suitable for objects with sharp edges and corners that are easily detectable and characterize the object.

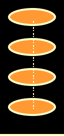




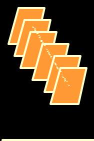
Generalized-Cylinders

A generalized cylinder is a volumetric primitive defined by:

- a space curve axis
- a cross section function



standard cylinder



rectangular cross sections

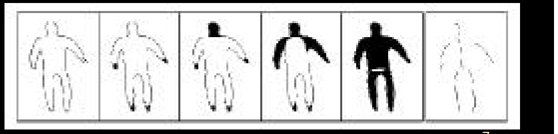
This cylinder has
- curved axis
- varying cross section

Finding GCs in Intensity Data

Generalized cylinder models have been used for several different classes of objects:

- airplanes (Brooks)
- animals (Marr and Nishihara)
- humans (Medioni)
- human anatomy (Zhenrong Qian)

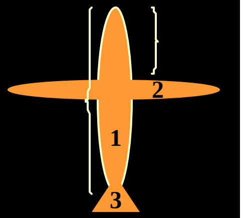
The 2D projections of standard GCs are
- ribbons
- ellipses

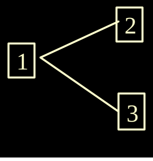


Generalized-Cylinder Models

Generalized cylinder models include:

1. a set of generalized cylinders
2. the spatial relationships among them
3. the global properties of the object





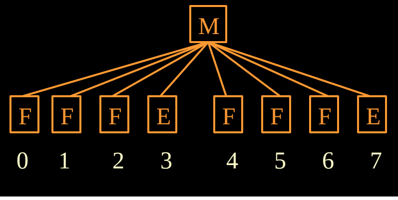
How can we describe the attributes of the cylinders and of their connections?

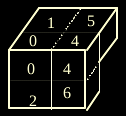
Octrees

- Octrees are 8-ary tree structures that compress a voxel representation of a 3D object.

- Kari Puli used them to represent the 3D objects during the space carving process.

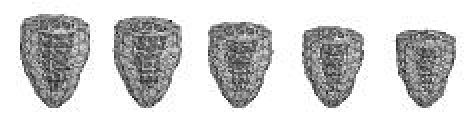
- They are sometimes used for medical object representation.





Superquadrics

- Superquadrics are parameterized equations that describe solid shapes algebraically.
- They have been used for graphics and for representing some organs of the human body, ie. the heart.



2D Deformable Models

A 2D deformable model or snake is a function that is fit to some real data, along its contours.

The fitting mimizes:

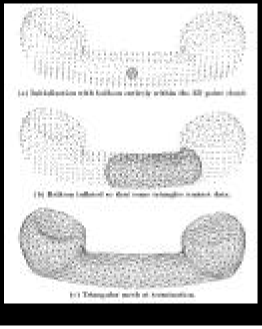
- internal energy of the contour (smoothness)

- image energy (fit to data)

- external energy (user-defined constraints)

3D Deformable Models

In 3D, the snake concept becomes a balloon that expands to fill a point cloud of 3D data.



Matching Geometric Models via Alignment

Alignment is the most common paradigm for matching 3D models to either 2D or 3D data. The steps are:

1. **hypothesize a correspondence** between a set of model points and a set of data points

2. From the correspondence **compute a transformation** from model to data

3. **Apply the transformation** to the model features to produce transformed features

4. **Compare** the transformed model features to the image features to verify or disprove the hypothesis

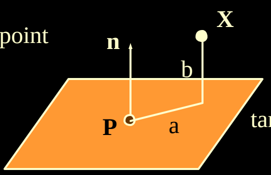
3D-3D Alignment of Mesh Models to Mesh Data

- Older Work: match 3D features such as 3D edges and junctions or surface patches
- More Recent Work: match surface signatures
  - curvature at a point
  - curvature histogram in the neighborhood of a point
  - Medioni's splashes
  - Johnson and Hebert's spin images

The Spin Image Signature

P is the selected vertex.

X is a contributing point of the mesh.



α is the perpendicular distance from X to P's surface normal.

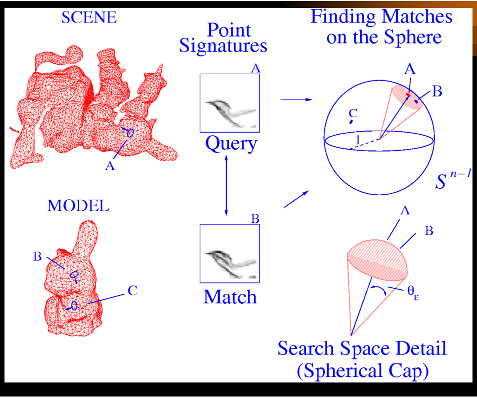
β is the signed perpendicular distance from X to P's tangent plane.

Spin Image Construction

- A spin image is constructed
  - about a specified oriented point o of the object surface
  - with respect to a set of contributing points C, which is controlled by maximum distance and angle from o.
- It is stored as an array of accumulators $S(\alpha,\beta)$ computed via:
- For each point c in C(o)

1. compute α and β for c.
2. increment S (α,β)

Spin Image Matching ala Sal Ruiz



Spin Images Object Recognition

Offline: Compute spin images of each vertex of the object model(s)

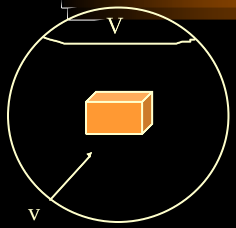
1. Compute spin images at selected points of a scene.
2. Compare scene spin images with model scene images by correlation or related method.
3. Group strong matches as in pose clustering and eliminate outliers.
4. Use the winning pose transformation to align the model to the image points and verify or disprove.

2D-3D Alignment

- single 2D images of the objects
- 3D object models
  - full 3D models, such as GC or SEV
  - view class models representing characteristic views of the objects

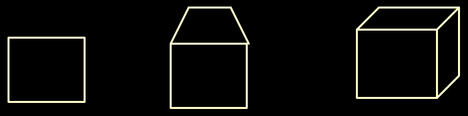
View Classes and Viewing Sphere

- The space of view points can be partitioned into a finite set of characteristic views.
- Each view class represents a set of view points that have something in common, such as:



1. same surfaces are visible
2. same line segments are visible
3. relational distance between pairs of them is small

3 View Classes of a Cube



1 surface 2 surfaces 3 surfaces

TRIBORS: view class matching of polyhedral objects

- Each object had 4-5 view classes (hand selected)
- The representation of a view class for matching included:
  - triplets of line segments visible in that class
  - the probability of detectability of each triplet determined by graphics simulation

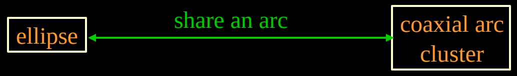
---

RIO: Relational Indexing for Object Recognition

- RIO worked with more complex parts that could have
  - planar surfaces
  - cylindrical surfaces
  - threads

Object Representation in RIO

- 3D objects are represented by a 3D mesh and set of 2D view classes.
- Each view class is represented by an attributed graph whose nodes are features and whose attributed edges are relationships.
- For purposes of indexing, attributed graphs are stored as sets of 2-graphs, graphs with 2 nodes and 2 relationships.

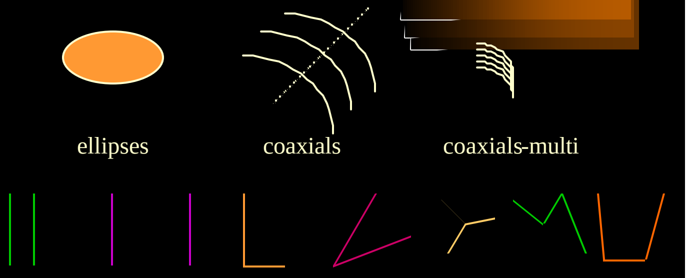


ellipse

share an arc

coaxial arc cluster

RIO Features



ellipses

coaxials

coaxials-multi

parallel lines close and far

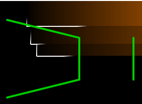
junctions

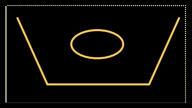
L V Y Z U

triples

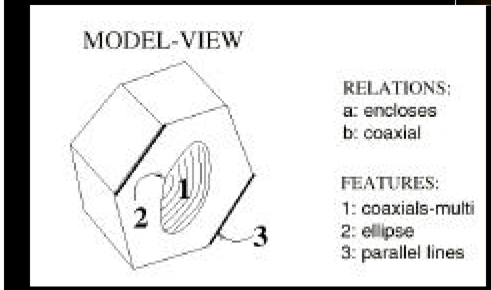
RIO Relationships

- share one arc
- share one line
- share two lines
- coaxial
- close at extremal points
- bounding box encloses / enclosed by



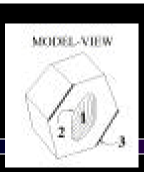


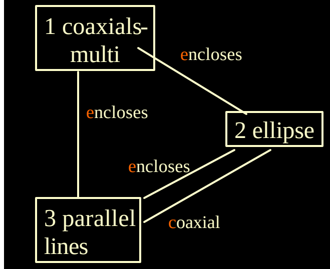
Hexnut Object

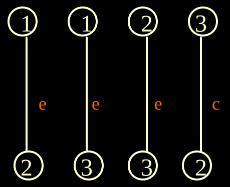


What other features and relationships can you find?

Graph and 2-Graph Representations







Relational Indexing for Recognition

Preprocessing (off-line) Phase

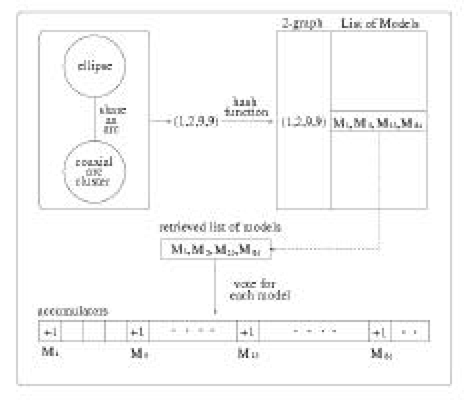
for each model view Mi in the database

- encode each 2-graph of Mi to produce an index
- store Mi and associated information in the indexed bin of a hash table H

Matching (on-line) phase

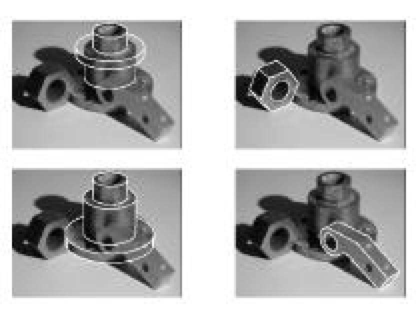
1. Construct a relational (2-graph) description D for the scene
2. For each 2-graph G of D
   - encode it, producing an index to access the hash table H
   - cast a vote for each Mi in the associated bin
3. Select the Mis with high votes as possible hypotheses
4. Verify or disprove via alignment, using the 3D meshes

The Voting Process



RIO Verifications

incorrect hypothesis



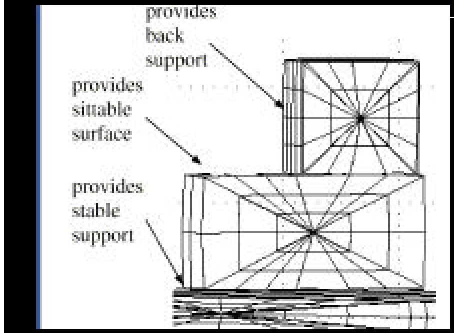
1. The matched features of the hypothesized object are used to determine its **pose**.
2. The **3D mesh** of the object is used to project all its features onto the image.
3. A **verification procedure** checks how well the object features line up with edges on the image.

Functional Models (Stark and Bowyer)

- Classes of objects are defined through their functions.
- Knowledge primitives are parameterized procedures that check for basic physical concepts such as

  - dimensions
  - orientation
  - proximity
  - stability
  - clearance
  - enclosure

Example: Chair



Functional Recognition Procedure

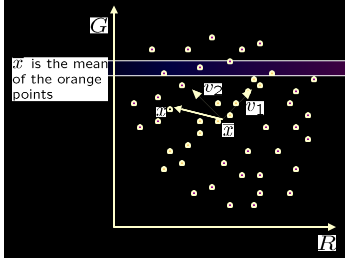
- Segment the range data into surfaces
- Use a bottom-up analysis to determine all functional properties
- From this, construct indexes that are used to rank order the possible objects and prune away the impossible ones
- Use a top-down approach to fully test for the most highly ranked categories.

What are the strengths and weaknesses of this approach?

Recognition by Appearance

- Appearance-based recognition is a competing paradigm to features and alignment.
- No features are extracted!
- Images are represented by basis functions (eigenvectors) and their coefficients.
- Matching is performed on this compressed image representation.

Linear subspaces



Consider the sum squared distance of a point **x** to all of the orange points:

$$SSD(\mathbf{v}) = \sum_{\text{orange point } \mathbf{x}} \|(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\|$$

What unit vector **v** minimizes SSD?

$$\mathbf{v}_2 = min_{\mathbf{v}}\ \{SSD(\mathbf{v})\}$$

What unit vector **v** maximizes SSD?

$$\mathbf{v}_1 = max_{\mathbf{v}}\ \{SSD(\mathbf{v})\}$$

$$\begin{aligned} SSD(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}} \cdot \mathbf{v}\| \\ &= \sum_{\mathbf{x}} \mathbf{v}^{\mathbf{T}}(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}}\mathbf{v} \\ &= \mathbf{v}^{\mathbf{T}}\left[\sum_{\mathbf{x}}(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}}\right]\mathbf{v} \\ &= \mathbf{v}^{\mathbf{T}}\mathbf{A}\mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}}(\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathbf{T}} \end{aligned}$$

Solution: $\mathbf{v}_1$ is eigenvector of **A** with *largest* eigenvalue
$\mathbf{v}_2$ is eigenvector of **A** with *smallest* eigenvalue

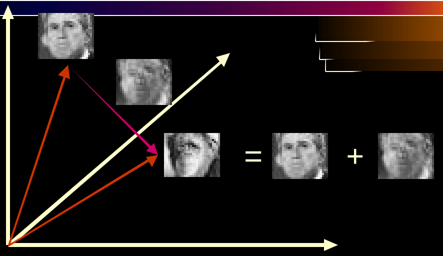
Principle component analysis

- Suppose each data point is N-dimensional
  - Same procedure applies:

$$\begin{aligned} SSD(\mathbf{v}) &= \sum_{\mathbf{x}} \|(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}} \cdot \mathbf{v}\| \\ &= \mathbf{v}^{\mathrm{T}}\mathbf{A}\mathbf{v} \quad \text{where } \mathbf{A} = \sum_{\mathbf{x}} (\mathbf{x}-\overline{\mathbf{x}})(\mathbf{x}-\overline{\mathbf{x}})^{\mathrm{T}} \end{aligned}$$

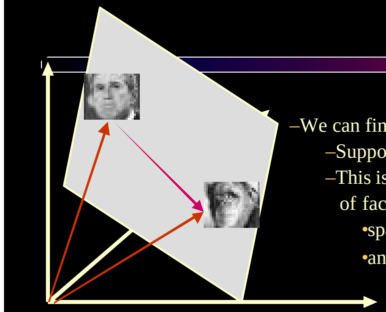
  - The eigenvectors of **A** define a new coordinate system
    - eigenvector with largest eigenvalue captures the most variation among training vectors **x**
    - eigenvector with smallest eigenvalue has least variation
  - We can compress the data by only using the top few eigenvectors
    - corresponds to choosing a “linear subspace”
      - represent points on a line, plane, or “hyper-plane”

The space of faces



- An image is a point in a high dimensional space
  - An N x M image is a point in R^{NM}
  - We can define vectors in this space

Dimensionality reduction



–We can find the best subspace using PCA
- –Suppose it is K dimensional
- –This is like fitting a “hyper-plane” to the set of faces
  - spanned by vectors $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_K$
  - any face $\mathbf{x} \approx a_1\mathbf{v}_1 + a_2\mathbf{v}_2 +, ..., + a_K\mathbf{v}_K$

The set of faces is a “subspace” of the set of images.

Turk and Pentland’s Eigenfaces: Training

- Let F1, F2,…, FM be a set of training face images. Let F be their mean and Φi = Fi – F
- Use principal components to compute the eigenvectors and eigenvalues of the covariance matrix of the Φi s
- Choose the vector u of most significant M eigenvectors to use as the basis.
- Each face is represented as a linear combination of eigenfaces

u = (u1, u2, u3, u4, u5); F27 = a1*u1 + a2*u2 + … + a5*u5

Matching

unknown
face image
I
→
convert to its
eigenface
representation
→
$\Omega = (\Omega 1, \Omega 2, \ldots, \Omega m)$

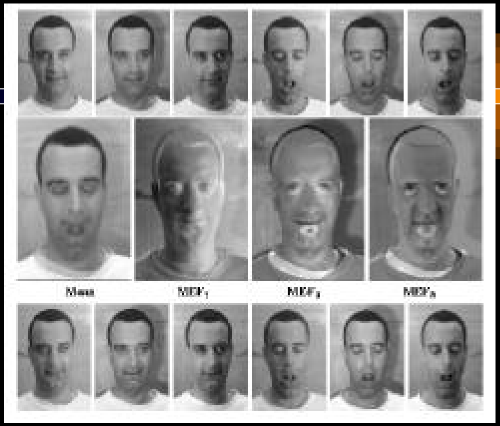
Find the face class k that minimizes

$\varepsilon k = || \Omega - \Omega k ||$

training
images

mean
image

linear
approxi-
mations



3 eigen-
images

Extension to 3D Objects

- Murase and Nayar (1994, 1995) extended this idea to 3D objects.
- The training set had multiple views of each object, on a dark background.
- The views included multiple (discrete) rotations of the object on a turntable and also multiple (discrete) illuminations.
- The system could be used first to identify the object and then to determine its (approximate) pose and illumination.

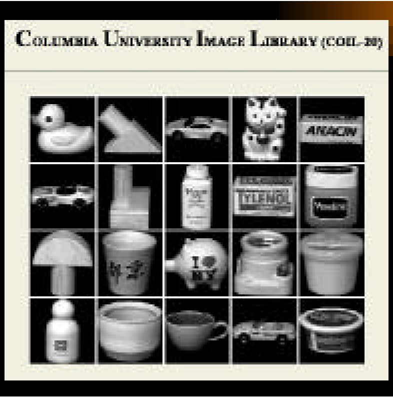
Significance of this work

- The extension to 3D objects was an important contribution.
- Instead of using brute force search, the authors observed that

  All the views of a single object, when transformed into the eigenvector space became points on a manifold in that space.

- Using this, they developed fast algorithms to find the closest object manifold to an unknown input image.
- Recognition with pose finding took less than a second.

Sample Objects
Columbia Object Recognition Database



Appearance-Based Recognition

- Training images must be representative of the instances of objects to be recognized.
- The object must be well-framed.
- Positions and sizes must be controlled.
- Dimensionality reduction is needed.
- It is still not powerful enough to handle general scenes without prior segmentation into relevant objects.-- my comment
- Hybrid systems (features plus appearance) seem worth pursuing.