High-Level Computer Vision

- Detection of classes of objects (faces, motorbikes, trees, cheetahs) in images
- Recognition of specific objects such as George Bush or machine part #45732
- Classification of images or parts of images for medical or scientific applications
- Recognition of events in surveillance videos
- Measurement of distances for robotics
High-level vision uses techniques from AI.

| Graph-Matching: A*, Constraint Satisfaction, Branch and Bound Search, Simulated Annealing |
| Learning Methodologies: Decision Trees, Neural Nets, SVMs, EM Classifier |
| Probabilistic Reasoning, Belief Propagation, Graphical Models |
Graph Matching for Object Recognition

- For each specific object, we have a geometric model.
- The geometric model leads to a symbolic model in terms of image features and their spatial relationships.
- An image is represented by all of its features and their spatial relationships.
- This leads to a graph matching problem.
Model-based Recognition as Graph Matching

- Let $U$ = the set of model features.
- Let $R$ be a relation expressing their spatial relationships.
- Let $L$ = the set of image features.
- Let $S$ be a relation expressing their spatial relationships.
- The ideal solution would be a subgraph isomorphism $f: U \rightarrow L$ satisfying
  
  - if $(u_1, u_2, ..., u_n) \in R$, then $(f(u_1), f(u_2), ..., f(u_n)) \in S$
House Example

2D model

2D image

\[ P = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11}\}. \]

\[ L = \{S_a, S_b, S_c, S_d, S_e, S_f, S_g, S_h, S_i, S_j, S_k, S_l, S_m\}. \]

\[ R_P = \{(S_1, S_2), (S_1, S_5), (S_1, S_6), (S_2, S_3), (S_2, S_4), (S_3, S_4), (S_3, S_9), (S_4, S_5), (S_4, S_7), (S_4, S_{11}), (S_5, S_6), (S_5, S_7), (S_5, S_{11}), (S_6, S_8), (S_6, S_{11}), (S_7, S_9), (S_7, S_{10}), (S_7, S_{11}), (S_8, S_{10}), (S_8, S_{11}), (S_9, S_{10})\}. \]

\[ R_L = \{(S_a, S_b), (S_a, S_j), (S_a, S_n), (S_b, S_c), (S_b, S_d), (S_b, S_n), (S_c, S_d), (S_d, S_e), (S_d, S_f), (S_d, S_g), (S_e, S_f), (S_e, S_g), (S_f, S_l), (S_f, S_m), (S_g, S_h), (S_g, S_i), (S_g, S_n), (S_h, S_i), (S_h, S_k), (S_h, S_l), (S_h, S_n), (S_i, S_j), (S_i, S_k), (S_i, S_n), (S_j, S_k), (S_k, S_l), (S_l, S_m)\}. \]

\[ f(S_1) = S_j \quad f(S_4) = S_n \quad f(S_7) = S_g \quad f(S_{10}) = S_f \]

\[ f(S_2) = S_a \quad f(S_5) = S_i \quad f(S_8) = S_l \quad f(S_{11}) = S_h \]

\[ f(S_3) = S_b \quad f(S_6) = S_k \quad f(S_9) = S_d \]
But this is too simplistic

- The model specifies all the features of the object that may appear in the image.
- Some of them don’t appear at all, due to occlusion or failures at low or mid level.
- Some of them are broken and not recognized.
- Some of them are distorted.
- Relationships don’t all hold.
**TRIBORS**: view class matching of polyhedral objects

- A **view-class** is a typical 2D view of a 3D object.
- 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

The first version of this program used depth-limited A* search.
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 \quad \text{share an arc} \quad \text{coaxial arc cluster}
RIO Features

- ellipses
- coaxials
- coaxials-multi
- parallel lines close and far
- junctions L V Y Z U_{10}
- triples
RIO Relationships

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

How are 1, 2, and 3 related?

What other features and relationships can you find?

MODEL-VIEW

RELATIONS:
- a: encloses
- b: coaxial

FEATURES:
- 1: coaxials-multi
- 2: ellipse
- 3: parallel lines
Graph and 2-Graph Representations

1 coaxials-multi
   
1 encloses
   2 encloses
   3 encloses

2 ellipse

3 parallel lines

coaxial

e encloses

e encloses
e encloses
c encloses

e

e

e

c
Relational Indexing for Recognition

Preprocessing (off-line) Phase

for each model view $M_i$ in the database

- **encode** each 2-graph of $M_i$ to produce an index
- store $M_i$ 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 $M_i$ in the associated bin

3. Select the $M_i$s with high votes as possible hypotheses

4. Verify or disprove via alignment, using the 3D meshes
The Voting Process

1. **Share an arc**
2. **Hash function**
3. **2-graph**
4. **List of Models**
5. **Retrieve list of models**
6. **Vote for each model**

<table>
<thead>
<tr>
<th>M1</th>
<th>M5</th>
<th>M23</th>
<th>M81</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>+1</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
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.
Use of classifiers is big in computer vision today.

- 2 Examples:
  - Rowley’s Face Detection using neural nets
  - Our 3D object classification using SVMs
Object Detection: Rowley’s Face Finder

1. convert to gray scale
2. normalize for lighting
3. histogram equalization
4. apply neural net(s) trained on 16K images

What data is fed to the classifier?

32 x 32 windows in a pyramid structure
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 \( \alpha \) and \( \beta \) for \( c \).
  2. increment \( S(\alpha, \beta) \)
Spin Image Matching
ala Sal Ruiz
Sal Ruiz’s Classifier Approach

1. Numeric Signatures
2. Components
3. Symbolic Signatures
4. Recognition and Classification of Deformable Shapes

Architecture of Classifiers
Numeric Signatures: Spin Images

Spin images for point $P$:

- Rich set of surface shape descriptors.
- Their spatial scale can be modified to include local and non-local surface features.
- Representation is robust to scene clutter and occlusions.
How To Extract Shape Class Components?

Select Seed Points

Compute Numeric Signatures

Region Growing Algorithm

Component Detector

Grown components around seeds

Training Set
Component Extraction Example

Selected 8 seed points by hand

Grow one region at the time (get one detector per component)

Region Growing

Labeled Surface Mesh

Detected components on a training sample
How To Combine Component Information?

Extracted components on test samples

Note: Numeric signatures are invariant to mirror symmetry; our approach preserves such an invariance.
Symbolic Signature

Labeled Surface Mesh

Encode Geometric Configuration

Symbolic Signature at P

Matrix storing component labels
Symbolic Signatures Are Robust To Deformations

Relative position of components is stable across deformations: experimental evidence
Proposed Architecture (Classification Example)

Verify spatial configuration of the components

Identify Components

Identify Symbolic Signatures

Labeled Mesh

Class Label

Surface Mesh

Two classification stages

-1 (Abnormal)
At Classification Time (1)

Surface Mesh

Bank of Component Detectors

Assigns Component Labels

Multi-way classifier

Identify Components

Labeled Surface Mesh
At Classification Time (2)

- Labeled Surface Mesh
- Symbolic Signatures
- Bank of Symbolic Detectors
- Assigns Symbolic Labels

Symbolic pattern for components 1,2,4
Symbolic pattern for components 5,6,8

Two detectors
Architecture Implementation

- ALL our classifiers are (off-the-shelf) ν-Support Vector Machines (ν-SVMs) (Schölkopf et al., 2000 and 2001).

- Component (and symbolic signature) detectors are one-class classifiers.

- Component label assignment: performed with a multi-way classifier that uses pairwise classification scheme.

- Gaussian kernel.
Shape Classes
Task 1: Recognizing Single Objects (2)

- Snowman: 93%.
- Rabbit: 92%.
- Dog: 89%.
- Cat: 85.5%.
- Cow: 92%.
- Bear: 94%.
- Horse: 92.7%.

- Human head: 97.7%.
- Human face: 76%.

Recognition rates (true positives)
(No clutter, no occlusion, complete models)
### Task 2-3: Recognition in Complex Scenes (2)

<table>
<thead>
<tr>
<th>Shape Class</th>
<th>True Positives</th>
<th>False Positives</th>
<th>True Positives</th>
<th>False Positives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snowmen</td>
<td>91%</td>
<td>31%</td>
<td>87.5%</td>
<td>28%</td>
</tr>
<tr>
<td>Rabbit</td>
<td>90.2%</td>
<td>27.6%</td>
<td>84.3%</td>
<td>24%</td>
</tr>
<tr>
<td>Dog</td>
<td>89.6%</td>
<td>34.6%</td>
<td>88.12%</td>
<td>22.1%</td>
</tr>
</tbody>
</table>

**Task 2**

**Task 3**
Task 2-3: Recognition in Complex Scenes (3)