Example for Scene Interpretation

Garbage collection in Hamburg (1 frame of a sequence)

We want to recognize parts, activities, intentions, spatial & temporal relations

high-level vision means understanding every-day occurrences
Some Application Scenarios for Scene Interpretation

- Understanding street traffic (long history)
- Cameras monitoring parking lots, railway platforms, supermarkets, nuclear power plants, ...
- Video archiving and retrieval
- Soccer game analysis
- Smart room cameras
- Monitoring elderly people in nursing homes
- Autonomous robot applications (e.g. robot watchmen, robot waiters, assistance for elderly)
English paraphrase of automatically generated description:
The scene contains four moving objects: three cars and a pedestrian.
A VW drives from the Alte-Post to the front of the FBI. It stops.
Another VW drives towards Dammtor. It turns off Schlueterstrasse. It drives on Bieberstrasse towards Grindelhof.
A BMW drives towards Hallerplatz. While doing so, it overtakes the VW which has stopped, before Bieberstrasse. The BMW stops in front of the traffic lights.
The pedestrian walks towards Dammtor. While doing so, he crosses Schlueterstrasse in front of the FBI.
Representations and Processes in Knowledge-based Systems

Characteristics of ideal knowledge-based systems:

- Problems are specified by background and task knowledge using a declarative knowledge representation language
- Problems are solved using standard inference procedures

Knowledge representation formalisms must support representations and processes (inferences)!
Basic Knowledge Representation Requirements

- accident
  - traffic-accident
    - traffic-accident-4711
      - driver: Max-Meier
        - vehicle: HH-PK-479
      - location: Siemersplatz
      - date: 13.2.03
    - constraint between parts
  - is-a = "is specialisation of"
  - instance = "is instance of"

- probability distributions
  - binary relations linking to parts of a reified n-ary relation

(traffic-accident 4711 Max-Meier Siemersplatz 13.2.03 HH-PK 479)
Rule Systems

Syntax of a rule in OPS5:

```plaintext
<rule>::= [P <rule-name> <antecedent> --> <consequent>]
<antecedent>::= {<condition>}
(condition) ::= <pattern> | - <pattern>
<pattern> ::= [object] {^attribute <value>}
<consequent> ::= {<action>}
<action> ::= [MAKE object] {^attribute <value>} | [MODIFY <pattern-number> {^attribute <value>}]
[REMOVE <pattern-number>] | [WRITE {<value>}]
```

Example: "If there are 2 circular regions (disks) close to each other and with equal size, make them a wheel pair"

```
[P find-wheel-pair [disk ^location <x1> ^size <y>]
   [disk ^location |<x2> - <x1>| < 10 ^size <y>] --> ... ]
```

- depth-first search
- limited expressiveness for constraints
Recognition by Relational Matching

Principle:

- construct relational model(s) for object class(es)
- construct relational image description
- compute morphism (best partial match) between image and model(s)
Aggregate Structure

Basic structure of a frame-based representation of an aggregate concept:

- **aggregate name** contains a symbolic ID
- **parent concepts** contains IDs of taxonomical parents
- **external properties** provide a description of the aggregate as a whole
- **reference to parts** links to subunits of which an aggregate is composed
- **constraints** specify which relations must hold between the parts

Partonomical structure:

```
 parent
 aggregate

 part
 aggregate

 ... 

 part
 aggregate
```
Primitive Occurrences

A primitive occurrence is a symbolic entity involving one or more evidence objects for which a qualitative predicate is true over a time interval.

Primitive occurrences provide the raw material for the interpretation of time-varying scenes.

- object A moves straight ahead
- object B turns
- distance between objects A and B gets smaller
- object A nearby object B

In a natural scene, one may observe many time-dependent perceptual primitives and determine many primitive occurrences. Hence it may be useful to compute primitive occurrences on demand (attention driven).
Stepwise Construction of Scene Interpretations

Given taxonomical and compositional concept hierarchies, there are five kinds of interpretation steps for constructing interpretations consistent with evidence:

- **Evidence matching**
  Assignment of evidence to object view classes or verification of view hypotheses.

- **Aggregate instantiation**
  Inferring an aggregate from (not necessarily all) parts

- **Aggregate expansion**
  Instantiating parts of an instantiated aggregate

- **Instance specialization**
  Refinements along specialization hierarchy or in terms of aggregate parts

- **Instance merging**
  Merging identical instances constructed by different interpretation steps

**Repertoire of interpretation steps allows flexible interpretation strategies**
e.g. mixed bottom-up and top-down, context-dependent, task-oriented
Basic Constraint Consistency Algorithm

Given:
- Variables $V_1, V_2, \ldots, V_N$, each with an associated domain $\text{dom}(V_i)$
- Constraint relations on various subsets of variables determine acceptable combinations of these variables.

Consistency Algorithm:
A  Of each domain, prune values which are ruled out by any of the constraints. => domain consistency
B  Of each domain, prune values for which there are no corresponding values in each of the constraint relations. Repeat until no more values can be pruned. => arc consistency
C  If one domain is empty there is no solution. If each domain has a single value, the values are a unique solution.
D  If some domains have more than one value, the values may or may not be a solution. By repeatedly splitting a domain and solving the reduced constraint problem, all solutions can be obtained. => global consistency
Constraint Propagation in Convex Time-point Algebra

Variables: time variables $T_i$
Domain of a variable: range of integers $[t_{imin} \ldots t_{imax}]$
Constraints: inequalities with offset $T_i + c_{ik} \leq T_k$

Graphical representation:

• Domains may always be represented by min- and max-values ("convexity property").
• An increase of a min-value affects only time variables connected in edge direction.
• A decrease of a max-value affects only time variables connected against edge direction.
• In a cycle-free constraint net with $N$ variables, any change of a domain can be propagated in at most $N(N-1)$ steps.
Extending Discrete Time-point Algebra to 2D-Space

Use linear inequalities independently in two spatial dimensions. (Bounding boxes must be parallel to reference system.)

Example:

plate.x-end ≤ saucer.x-beg + 10
plate.x-end ≥ saucer.x-beg + 8
plate.y-end ≤ saucer.y-beg + 5
plate.y-end ≥ saucer.y-beg + 3
plate.x-beg ≥ table.x-beg
plate.x-end ≤ table.x-end
plate.y-beg ≤ table.y-beg + 5
plate.y-beg ≥ table.y-beg

Pairwise constraints can be combined to (quantitative) interval constraints:

plate.x-end in saucer.x-beg + [8 10]
plate.y-end in saucer.y-beg + [3 5]
plate.x-beg in table.x-beg + [0 inf]
plate.x-end in table.x-end + [-inf 0]
plate.y-beg in table.y-beg + [0 5]

Plate.x-end [8 10] in equivalent! saucer.x-beg
Plate.x-end ≤ saucer.x-beg + 10
Plate.x-end ≥ saucer.x-beg + 8
Plate.y-end ≤ saucer.y-beg + 5
Plate.y-end ≥ saucer.y-beg + 3
Plate.x-beg ≥ table.x-beg
Plate.x-end ≤ table.x-end
Plate.y-beg ≤ table.y-beg + 5
Plate.y-beg ≥ table.y-beg
So what is a Scene Interpretation?

Intuitively:
A scene interpretation is a scene description in terms of instantiated concepts consistent with evidence and context information.

not all concepts are important
not all evidence is important

constructed model

real world

animated slide!
Description Logics for Knowledge Representation

DLs are a family of knowledge-representation formalisms

- object-centered, roles and features (binary relations)
- necessary vs. sufficient attributes
- inference services
  - subsumption check
  - consistency check
  - classification
  - abstraction
  - default reasoning
  - spatial and temporal reasoning
- guaranteed correctness, completeness, decidability and complexity properties
- highly optimized implementations (e.g. RACER)
Table-Top Scene Description

TBox (excerpt):

(implies plate dish)
(implies saucer dish)
(implies cup dish)
(implies napkin cloth)
(equivalent cover
  (and configuration
    (exactly 1 has-part plate)
    (exactly 1 has-part (and saucer (some near plate)))
    (exactly 1 has-part (and cup (some on saucer)))

ABox (excerpt):

(instance plate1 plate)
(instance saucer1 saucer)
(instance saucer2 saucer)
(instance cup1 cup)
(instance cup2 cup)
(instance napkin1 napkin)
(instance cover1 cover)
(related saucer1 plate1 near)
((related cup1 saucer1 on)
(related napkin1 plate1 on)
Probabilistic Scene Modelling

• Random variables assigned to scene components:
  - events, occurrences
  - objects
  - properties
  - reified relations

• Probabilistic dependencies via joint distributions (JPDs)

Example: Probabilistic description of well-set table

<table>
<thead>
<tr>
<th>fork</th>
<th>knife</th>
<th>plate</th>
<th>saucer</th>
<th>cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>f.x1</td>
<td>k.x1</td>
<td>p.x1</td>
<td>s.x1</td>
<td>c.x1</td>
</tr>
<tr>
<td>f.y1</td>
<td>k.y1</td>
<td>p.y1</td>
<td>s.y1</td>
<td>c.y1</td>
</tr>
<tr>
<td>f.x2</td>
<td>k.x2</td>
<td>p.x2</td>
<td>s.x2</td>
<td>c.x2</td>
</tr>
<tr>
<td>f.y2</td>
<td>k.y2</td>
<td>p.y2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>f.α</td>
<td>k.α</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

JPD with 22 variables
Probabilistic scene model is viewed as a generator of possible scenes. Probability that model $m$ has generated evidence $e$:

$$\text{Rank}(m, e) = p(m) \ P^{(m)}(V = e) \ P_{\text{clutter}}$$

How can $P_{\text{scene}}$ be determined from aggregate models?
Causality Graph

Conditional dependencies (causality relations) of random variables define partial order.

Representation as a directed acyclic graph (DAG):

\[
P(X_1, X_2, X_3, \ldots, X_8) = \\
P(X_1 \mid X_2, X_3, X_4) \cdot P(X_2) \cdot P(X_3 \mid X_4, X_5) \cdot P(X_4 \mid X_6) \cdot P(X_5 \mid X_6) \cdot P(X_6 \mid X_7, X_8) \cdot P(X_7) \cdot P(X_8)
\]

For any DAG, we obtain the JPD as follows:

\[
P(X_1, \ldots, X_N) = \prod_i P(X_i \mid \text{Pa}(X_i))
\]
Bayesian Compositional Hierarchies

\[ P(V) \text{ JPD of all variables} \]

Let a leaf aggregate be described by
- external properties \( Y \)
- internal properties \( Z \)

All other variables are called \( X \).

\[ P(V) = P(X \ Y \ Z) \]

\[ P(X \ Y \ Z) = P(Z \mid X \ Y) P(X \ Y) \]

\[ P(X \ Y \ Z) = P(Z \mid Y) P(X \ Y) \]

\[ P(X \ Y \ Z) = P(Z \mid Y) P(X \ Y) \]

By recursive application:
\[ P(V) = P(Y_1) \prod_{i=1..N} P(Z_i \mid Y_i) \]

\( i=1..N (Y_i, Z_i): \text{ all aggregates} \)
Representation of the BCH as a Bayesian Network

A node of the Bayesian Net contains all parts of an aggregate. The conditional dependence on a single part cannot be represented.
Use of BCH for Estimation

- Enter begin or end of events
- Propagate change throughout BCH
- Estimate non-instantiated temporal variables

obtain dynamic priors (context-dependent)

begin of Pumping-Operation at T = 26
Paths through a HMM

Given a sequence of N observations, we want to find the most probable sequence of states which may have led to the observations.

Extension of trellis representation

- arc weights leading into states $X^{(n)}$: transition probabilities $p_{ij}$
- node weights of states $X^{(n)}$: observation likelihoods $q_{jm}$ for given observations $Y^{(n)} = b_{mn}$
- product of initial probability and node and arc probabilities along path: probability of observations and states

Example:

$$W = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.1 & 0.0 & 0.9 \\ 0.4 & 0.6 & 0.0 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.2 & 0.8 \end{bmatrix}, \quad \pi = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix}$$

Given a sequence of N observations, we want to find the most probable sequence of states which may have led to the observations.

The probability of observations along path are

$$P(Y^{(1)}=b_{m_1}, \ldots, Y^{(N)}=b_{m_N}, X^{(1)}=a_{k_1}, \ldots, X^{(N)}=a_{k_N}) = 0.6 \cdot 0.2 \cdot 0.2 \cdot 0.4 \cdot 0.1 \cdot 0.8 \cdot 0.5 \cdot 0.8$$
What Kind of Bayes Net is a HMM?

Bayes Net structure:

Finding most probable paths:

Evaluating likelihood of model:
SCENIOR System Structure

- OWL + SWRL Conceptual Knowledge Base
- Probabilistic Aggregate Descriptions
- Converter
  - Jess Conceptual Knowledge Base
  - Hypotheses Graphs
  - Temporal Constraint Net Templates
  - Probabilistic Aggregate Models
- Interpreters
  - Jess Rule Engine
  - Java Constraint Solver
  - BCH Inference Engine
- Primitive Events
- Interpretations
Recursive Markov Localization

\[ b_t(s_t) = p(s_t | o_0, a_0, o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t, m) \]

Bayes

\[ = \alpha_t p(o_t | o_0, a_0, \ldots, o_{t-1}, a_{t-1}, s_t, m) p(s_t | o_0, a_0, \ldots, o_{t-1}, a_{t-1}, m) \]

Markov

\[ = \alpha_t p(o_t | s_t, m) p(s_t | o_0, a_0, \ldots, o_{t-1}, a_{t-1}, m) \]

Total Prob.

\[ = \alpha_t p(o_t | s_t, m) \int p(s_t | o_0, a_0, \ldots, o_{t-1}, a_{t-1}, s_t, m) p(s_{t-1} | o_0, a_0, \ldots, o_{t-1}, a_{t-1}, m) ds_{t-1} \]

Markov

\[ = \alpha_t p(o_t | s_t, m) \int p(s_t | a_{t-1}, s_{t-1}, m) p(s_{t-1} | o_0, a_0, \ldots, o_{t-1}, a_{t-1}, m) ds_{t-1} \]

\[ = \alpha_t p(o_t | s_t, m) \int p(s_t | a_{t-1}, s_{t-1}, m) b_{t-1}(s_{t-1}) ds_{t-1} \]

\( \alpha_t \) is normalizing factor

\[ b_t(s_t) = \alpha_t p(o_t | s_t, m) \int p(s_t | a_{t-1}, s_{t-1}, m) b_{t-1}(s_{t-1}) ds_{t-1} \]

\( p(o_t | s_t, m) \) probabilistic perceptual model - often time-invariant: \( p(o | s, m) \)

\( p(s_t | a_{t-1}, s_{t-1}, m) \) probabilistic motion model - often time-invariant: \( p(s' | a, s, m) \)

\[ \text{must be specified for a specific robot and a specific environment} \]
Structure of Probabilistic Localization

\[ b_t(s_t) = \alpha_t \, p(o_t | s_t, m) \int p(s_t | a_{t-1}, s_{t-1}, m) \, b_{t-1}(s_{t-1}) \, ds_{t-1} \]
Auf Wiedersehen!