Scene Interpretation using Description Logics

Towards Scene Interpretation

garbage collection
+
mail delivery in Hamburg

unusual breakfast

(Buster Keaton: The Navigator)

Contents

- What is scene interpretation?
- Representational requirements
- Inferencing for scene interpretation
- Preference measure
- Experiments
- Conclusions

Characteristics of Scene Interpretation

- Representing and recognizing structures consisting of several spatially and temporally related components (e.g. object configurations, situations, occurrences, episodes)
- Exploiting high-level knowledge and reasoning for scene prediction
- Understanding purposeful behaviour (e.g. obstacle avoidance, grasping and moving objects, behaviour in street traffic)
- Mapping between quantitative and qualitative descriptions
- Natural-language communication about scenes
- Learning high-level concepts from experience

Some Application Scenarios for High-level Scene Interpretation

- **street traffic observations (long history)**
- **cameras monitoring parking lots, railway platforms, supermarkets, nuclear power plants, ...**
- **video archiving and retrieval**
- **soccer commentator**
- **smart room cameras**
- **autonomous robot applications**
  (e.g. robot watchmen, playmate for children)

State of the Art

- **Computer Vision has been preoccupied with single-object recognition**
  - feature-based classification
  - 2D - 3D reconstruction
  - object categorization
- **Probabilistic approaches for multi-object scene analysis**
  - Hidden-Markov Models
  - Bayesian Network Models
  - Learning
- **EU funding of "Cognitive Vision"**

Cognitive computer vision is concerned with integration and control of vision systems using explicit but not necessarily symbolic models of context, situation and goal-directed behaviour. Cognitive vision implies functionalities for knowledge representation, learning, reasoning about events & structures, recognition and categorization, and goal specification, all of which are concerned with the semantics of the relationship between the visual agent and its environment.

Cognitive Vision Projects 5th Framework

ACTIPRET: Interpreting and Understanding Activities of Expert Operators for Teaching and Education

CAVIAR: Context Aware Vision Using Image-Based Active Recognition

COGVIS: Cognitive Vision Systems

COGVISYS: Cognitive Vision Systems

DETECT: Real Time Detection of Motion Picture Content in Live Broadcasts

ECVISION: European Research Network for Cognitive AI-enabled Computer Vision Systems

LAVA: Learning for adaptable visual assistants

VAMPIRE: Visual Active Memory Processes and Interactive REtrieval

VISATEC: Vision-based Integrated Systems Adaptive to Task and Environment with Cognitive abilities

Cognition Projects 6th Framework

COSPAL: Cognitive Systems using Perception-Action Learning

Design and architecture of Artificial Cognitive Systems (ACS)

COSY: Cognitive Systems for Cognitive Assistants

GNOSYS: Conceptual architecture for Cognitive Agents

MACS: Multisensory Autonomous Cognitive Systems Interaction with Dynamic Environments for Perceiving and Using Affordances

MindRaces: From Reactive to Anticipatory Cognitive Embodied Systems

ROBOT-CUP: Robotic Open-architecture Technology for Cognition, Understanding and Behaviours

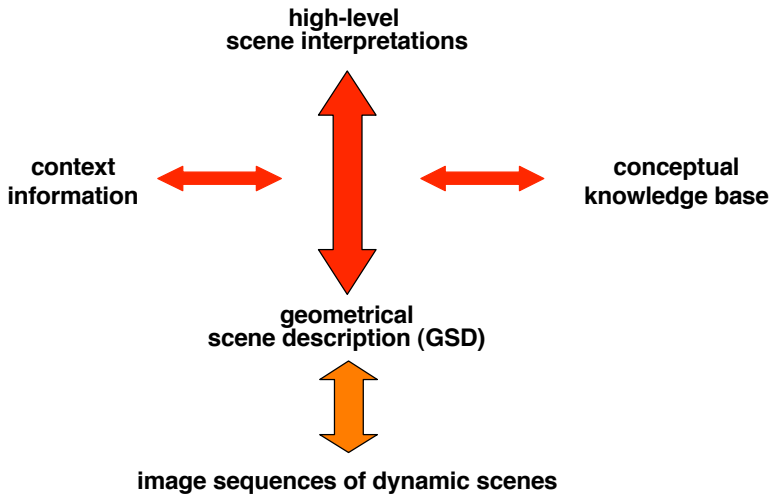
SPARK: Spatial-temporal patterns for action-oriented perception in roving robots

http://www.cordis.lu/ist/directorate_e/cognition/projects.htm

Contents

- What is scene interpretation?
- Representational requirements
- Inferencing for scene interpretation
- Preference measure
- Experiments
- Conclusions

Structure of Scene Interpretation System



Why Consider Description Logics?

- **Scene interpretation is a knowledge-heavy task**
- **Large knowledge bases need well-founded semantics**
- **Desirable to use standard inferencing procedures**

- **DLs provide expressive object-oriented knowledge representation**
- **DLs are well understood**
- **There exist efficient DL systems with various inference procedures**

- **Long-standing research at CSL, Hamburg University**
  - Expressive DLs, RACER (Haarslev, Möller, Wessel)
  - DLs for spatial reasoning (Haarslev, Möller, Wessel)
  - DLs for scene interpretation (Möller, Neumann, Schröder)
  - DLs for case-based help-desk support (Kamp)

What is a Scene Interpretation?

Intuitively:

A scene interpretation is a scene description in terms of instantiated scene models consistent with evidence, context information and world knowledge.

evidence
context information } **ABox**
scene interpretation

scene models
world knowledge } **TBox**

Historical Scene Models

Badler 75:

Relational structures ("scene graphs") for simple traffic scenes using spatial and directional adverbials

Tsotsos 79:

Relational structures for left-ventricular heart motion using is-a, part-of and similarity relations

Neumann 86:

Hierarchical relational structures for traffic scenes based on natural language verbs

```
(OVERTAKE OBJ1 OBJ2 T1 T2)  <=>
        (MOVE OBJ1 T1 T2)
        (MOVE OBJ2 T1 T2)
        (BEHIND OBJ1 OBJ2 T1 T3)
        (BESIDE OBJ1 OBJ2 T3 T4)
        (BEFORE OBJ1 OBJ2 T4 T2)
        (APPROACH OBJ1 OBJ2 T1 T3)
        (DIS-APPROACH OBJ1 OBJ2 T4 T2)
```

Perceptual Primitives

Perceptual primitives are geometrical and photometrical attributes which can be immediately determined from a GSD.

For <u>object configurations</u>:

- **objects provide reference features in terms of**
  - **locations (center of gravity, corners, surface markings, etc.)**
  - **lines (edges, surface markings, axes of minimal inertia, etc.)**
  - **orientations (inate, motion, viewer)**
- **perceptual primitives are measurements between reference features:**
  - **distance**
  - **angle**
  - **temporal derivatives thereof**

Qualitative Primitives

Qualitative primitives are predicates over perceptual primitives constant over some time interval.

- **qualitatively constant values**
  e.g. constant orientation, constant distance
- **values within a certain range**
  e.g. topological relations, degrees of nearness, typical speeds
- **values smaller or larger than a threshold**
  e.g. increase of distance, slowing down

Basic Representational Requirements

- **object oriented**
- **n-ary relations**
- **taxonomies**
- **partonomies**
- **spatial and temporal relations**
- **qualitative predicates**

Representing N-ary as Binary Relations

Reification:

(BETWEEN A B C) ⇒
(INSTANCE BETW1 BETWEEN)
(BETWEEN-ARG1 BETW1 A)
(BETWEEN-ARG2 BETW1 B)
(BETWEEN-ARG3 BETW1 C)

(OVERTAKE VEH1 VEH2 23 46) ⇒
(INSTANCE OT1 OVERTAKE)
(OVERTAKER OT1 VEH1)
(OVERTAKEE OT1 VEH2)
(TBEG OT1 23)
(TEND OT1 42)

Table-laying Senario in CogVis

Stationary cameras observe living room scene and recognize meaningful occurrences, e.g. placing a cover onto the table.

Occurrence Model for Placing a Cover

Composite occurrences are expressed in terms of simpler models

name:	**place-cover**
parents:	**:is-a agent-activity**
parts:	**pc-tt :is-a table-top**
	pc-tp1 :is-a transport with (tp-obj :is-a plate)
	pc-tp2 :is-a transport with (tp-obj :is-a saucer)
	pc-tp3 :is-a transport with (tp-obj :is-a cup)
	pc-cv :is-a cover
time marks:	**pc-tb, pc-te :is-a timepoint**
constraints:	**pc-tp1.tp-ob = pc-cv.cv-pl**
	pc-tp2.tp-ob = pc-cv.cv-sc
	pc-tp3.tp-ob = pc-cv.cv-cp
	...
	pc-tp3.tp-te $\geq$ pc-tp2.tp-te
	pc-tb $\leq$ pc-tp3.tb
	pc-te $\geq$ pc-cv.cv-tb

Scene Objects, Physical Bodies and Views

2D views obtained by cameras are components of scene objects and related to the corresponding 3D physical bodies by constraints

name:	**plate**
parents:	**:is-a scene-object**
parts:	**pl-body :is-a body with pl-body-preds**
	pl-view :is-a view with pl-view-preds
constraints:	**(constraints between pl-body-preds and pl-view-preds)**

Intended Actions

Intentions may be modelled as invisible components of an intended action

name:	intended-place-cover
parents:	:is-a intended-action
parts:	ipc-pc :is-a place-cover
	ipc-ag :is-a agent
	ipc-cv :is-a cover
constraints:	ipc-ag.desire = ipc-cv
	(and other constraints)

RACER Concept Language

C concept term
CN concept name
R role term
RN role name

C -> *CN*
top
bottom
(not *C*)
(and *C1 ... Cn*)
(or *C1 ... Cn*)
(some *R C*)
(all *R C*)
(at-least *n R*)
(at-most *n R*)
(exactly *n R*)
(at-least *n R C*)
(at-most *n R C*)
(exactly *n R C*)
CDC

concept definition

(equivalent *CN C*)

concept axioms

(implies *CN C*)
(implies *C1 C2*)
(equivalent *C1 C2*)
(disjoint *C1 ... Cn*)

roles

R -> *RN*
(inv *RN*)

concrete-domain concepts
AN *attribute name*

CDC -> (a *AN*)
(an *AN*)
(no *AN*)
(min *AN integer*)
(max *AN integer*)
(> *aexpr aexpr*)
(>= *aexpr aexpr*)
(< *aexpr aexpr*)
(<= *aexpr aexpr*)
(= *aexpr aexpr*)

aexpr -> *AN*
real
(+ *aexpr1 aexpr1**)
aexpr1

aexpr1 -> *real*
AN
(* *real AN*)

DL Concept for a Cover

```
(equivalent cover
        (and configuration
            (exactly 1 cv-pl plate)
            (exactly 1 cv-sc (and saucer (some near plate)))
            (exactly 1 cv-cp (and cup (some on saucer)))
            (subset cv-pl (compose cv-sc near))
            (subset cv-sc (compose cv-cp on))))
```

- **parts are expressed as qualified fillers of specific roles**
  e.g. cv-pl, cv-sc, cv-scp
- **sameness (or distinctness) of parts and properties of parts are expresses by the subset construct**
- **spatial constraints are modelled as primitive predicates**
  e.g. near, on

Simplified DL Concept for Placing a Cover

```
(equivalent place-cover
    (and agent-activity
        (exactly 1 pc-tp1 (and transport (some tp-obj plate)))
        (exactly 1 pc-tp2 (and transport
                               (some tp-obj saucer)
                               (some before (and transport (some tp-obj cup))))
        (exactly 1 pc-tp3 (and transport (some tp-obj cup)))
        (subset pc-tp3 (compose pc-tp2 before))))
```

Severe disadvantage of purely symbolic spatial and temporal constraints:

Pairwise constraints must be computed bottom-up by low-level vision procedures irrespective of high-level concepts!

⇨ **Express spatial and temporal constraints as predicates over concrete-domain elements**

Concrete Domain Concepts in RACER

```
CDC →   (a AN) (an AN)
        (no AN)
        (min AN integer)
        (max AN integer)
        (equal AN integer)
        (> aexpr aexpr)
        (>= aexpr aexpr)
        (< aexpr aexpr)
        (<= aexpr aexpr)
        (= aexpr aexpr)
aexpr → AN
        real
        (+ aexpr1 aexpr1*)
        aexpr1
aexpr1 →AN
        real
        (* real AN)
```

Example:
Quantitative constraints on the size of an object

(and (min size 13) (max size 20))

integer-valued attribute "size" receives values from low-level vision

Quantitative Spatial and Temporal Constraints

```
(equivalent place-cover
      (and agent-activity
            (exactly 1 pc-tp1 (and transport (some tp-obj plate))
            (exactly 1 pc-tp2 (and transport (some tp-obj saucer))
            (exactly 1 pc-tp3 (and transport (some tp-obj cup))
            (<= pc-tp2 o tp-end pc-tp3 o tp-end)
            (= pc-beg (minim pc-tp1 o tp-beg pc-tp2 o tp-beg pc-tp3 o tp-beg))
            (= pc-end (maxim pc-tp1 o tp-end pc-tp2 o tp-end pc-tp3 o tp-end))
            (<= (- pc-end pc-beg) max-duration))))
```

- **Equality and inequality as concrete domain predicates**
- **Specific constraints for each concept**
- **Incremental constraint computation required for prediction!**
  Example: (and (= cv-sc o sc-loc cv-cp o cp-loc))
  Known saucer position restricts expected cup positions

General Structure for Aggregate Definitions

```
(equivalent  <concept-name>
             (and  <parent-concept1> ... <parent-conceptN>
                   (<number-restriction1>  <role-name1>  <part-concept1>)
                   . . .
                   (<number-restrictionK>  <role-nameK>  <part-conceptK>)
                   <constraints between parts>))
```

Summary of DL constructs required for aggregates: ALCF(D)

=> aggregates can in principle be represented in RACER, however, not all syntax features are currently available

Contents

Scene Interpretation as Model Construction

Construct a mapping of
- **constant symbols of the KR language into scene elements D**
- **predicate symbols of the KR language into predicate functions over D**

such that all predicates are true.

Operational semantics of low-level vision provide mapping into primitive constant and predicate symbols.

Finite model construction (Reiter & Mackworth, 87):

Domain closure and unique name assumption => problem can be expressed in Propositional Calculus and solved as a constraint satisfaction problem (CSP)

Partial model construction (Schröder 99):

- model may be incomplete, but must be extendable to a complete model
- disjunctions must be resolved

Practical Requirements for Partial Logical Models

- **Task-dependent scope and abstraction level**
  - **no need for checking all predicates**
    e.g. propositions outside a space and time frame may be uninteresting
  - **no need for maximal specialization**
    e.g. geometrical shape of "thing" suffices for obstacle avoidance
- **Partial model may not have consistent completion**
  - **uncertain propositions due to inherent ambiguity**
  - **predictions may be falsified**
- **Real-world agents need single "best" scene interpretation**
  - **uncertainty rating for propositions**
  - **preference measure for scene interpretations**

Logical model property provides only loose frame for possible scene interpretations

Stepwise Construction of Partial Models

Four kinds of interpretation steps for constructing interpretations consistent with evidence:

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

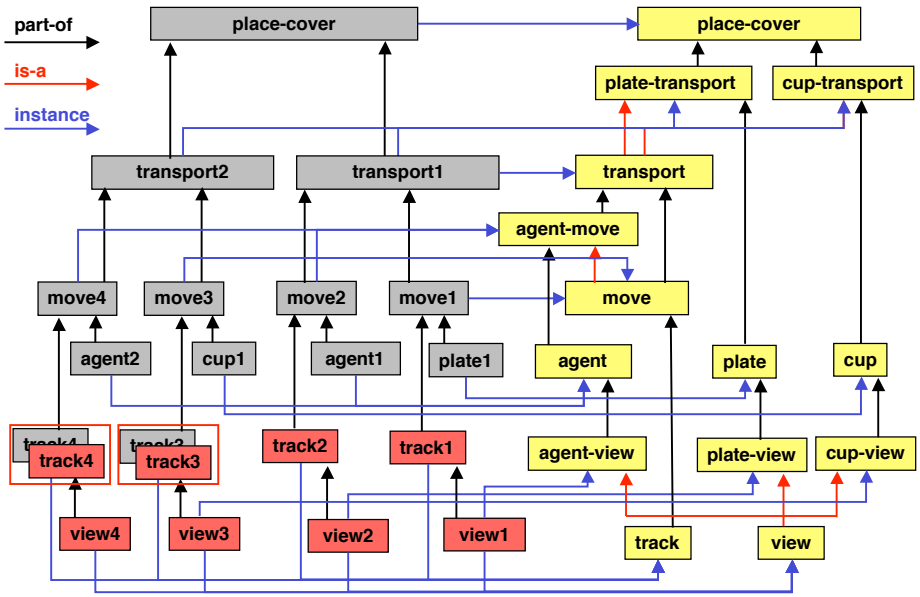
Instance specialization
Refinements along specialization hierarchy or in terms of aggregate parts

Instance expansion
Instantiating parts of an instantiated aggregate

Instance merging
Merging identical instances constructed by different interpretation steps

Repertoire of interpretation steps allow flexible interpretation strategies
e.g. mixed bottom-up and top-down, context-dependent, task-oriented

Example for Stepwise Interpretation



DL Reasoning Services

ABox consistency checking is at the heart of all reasoning services

Model construction is the method of choice for many DL reasoners

- **Concept satisfiability**
- **Concept subsumption**
- **Concept disjointness**
- **Concept classification**
- **TBox coherence**
- **ABox consistency w.r.t. a TBox**
- **Instance checking**
- **Most-specific atomic concepts of which an individual is an instance**
- **Instances of a concept**
- **Role fillers for a specified individual**
- **Pairs of individuals related by a specified role**
- **Conjunctive queries**

DL Reasoning Support for Scene Interpretation

- **Maintaining a coherent knowledge base**

  Scene interpretation may require extensive common-sense knowledge, intuitive knowledge representation is doomed

- **Maintaining consistent scene interpretations**

  A consistent ABox is a (partial) model and hence formally a (partial) scene interpretation => ABox consistency checking ensures consistent scene interpretations

ABox realization (computing most specific concepts for individuals) cannot be used in general:

- **scene interpretations cannot be deduced**
- **high-level individuals must be hypothesized before consistency check**

DL Support for Interpretation Steps

Aggregate instantiation
Determine aggregates for which an individual is a role filler
⇒ RACER query language

Instance specialization
Retrieve all specializations of a given concept
⇒ use specialization hierarchy

Instance expansion
Instantiate parts of an aggregate instance
⇒ easy service by looking up the aggregate definition

Instance merging
Determine whether it is consistent to unify two individual descriptions
=> unification by recursive specialization can be supported

Important missing service:
Preference measure for choosing "promising" alternatives

Contents

Preferred Interpretation Steps

- **Logical framework may provide infinitely many partial models**
  e.g. involving objects outside the field of view
- **Wrong choices among alternative interpretation steps may cause severe backtracking**
  e.g. wrong part-whole reasoning

Probabilistic approach based on scene statistics:

Select interpretation steps which construct the most likely interpretation given evidence

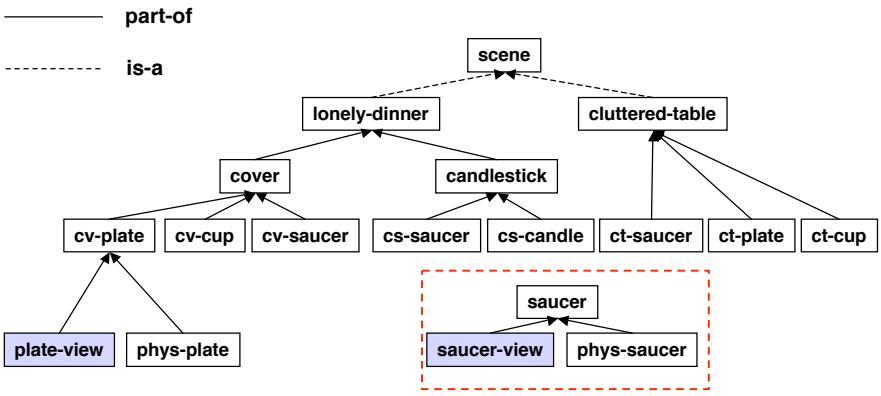
Probability distributions for

- **concept specializations**
  e.g. dinner-for-one vs. dinner-for-two
- **choices among individuals**
  e.g. choices of colours
- **discrete domain quantities**
  e.g. locations and time points

⇒ **multivariate distributions instead of constraints**

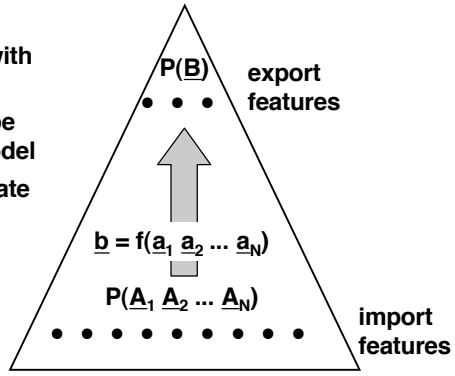
Example for Probabilistic Interpretation Decisions

For which role is the saucer a filler?



Integrating Bayesian Networks with DL Aggregates

- Each aggregate is associated with a Bayes Net fragment
- An operational Bayes Net can be constructed for each partial model
- Abstraction property of aggregate fragments ensures efficient probability computations



Example: Aggregate "cover"

JPD P(A1 A2 ... AN) for cover parts locations is mapped into JPD P(B) for cover bounding-box location

Contents

- What is scene interpretation?
- Representational requirements
- Inferencing for scene interpretation
- Preference measure
- Experiments
- Conclusions

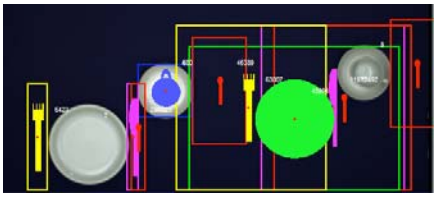
Configuration Technology for Scene Interpretation

- **Structure-based configuration is formally model construction**
  configuration = constructing an aggregate based on component definitions and customer requirements
- **CSL has developed the configuration tool KONWERK**
  - expressive object description language
  - powerful constraint system
  - flexible control structure
- **Minor changes to pose scene interpretation as a configuration problem**
  scene concepts => component definitions
  evidence => customer requirements
- **Interface to low-level vision system**
  incremental input of evidence

Examples of Table-setting Scenes

Experimental Results

natural views = evidence
coloured shapes = hypotheses
boxes = expected locations



Snapshot illustrates intermediate state of interpretation after 89 interpretation steps:

- **hypotheses based on partial evidence**
- **predictions about future actions and locations**
- **high-level disambiguation of low-level classification**
- **influence of context**

Conclusions

- **Representational requirements of scene interpretation can be met by a DL system of type ALCF(D)**
  - feature chains
  - same-as construct
  - concrete domain of integers for spatial and temporal constraints
  - operational systems not yet fully available
- **Restrictions of interpretation space**
  - task-oriented interpretations
  - incremental constraint evaluation
  - probabilistic preference measure
  - integration of Bayesian Networks and DLs

B. Neumann & R. Möller
On Scene Interpretation with Description Logics
FBI-B-257/04, Fachbereich Informatik, Universität Hamburg, 2004
To be published in Cognitive Vision Systems, H.-H. Nagel and H. Christensen, eds., Springer