Description Logic for Vision-Based Intersection Understanding

Britta Hummel
Institut für Mess- und Regelungstechnik
Universität Karlsruhe (TH)
D-76131 Karlsruhe, Germany
Email: hummel@mrt.uka.de

Werner Thiemann
BlueAGE
www.blue-age.de
D-76137 Karlsruhe
Email: thiemann@blue-age.de

Irina Lulcheva
Institut für Mess- und Regelungstechnik
Universität Karlsruhe (TH)
D-76131 Karlsruhe, Germany
Email: lulcheva@mrt.uka.de

***Abstract*— Road recognition from video sequences has been solved robustly only for small, often simplified subsets of possible road configurations. A massive augmentation of the amount of prior knowledge may pave the way towards a generation of estimators of more general applicability. This contribution introduces Description Logic extended by rules as a promising knowledge representation formalism for scene understanding.**

A Description Logic knowledge base for arbitrary road and intersection geometries and configurations is set up. Logically stated geometric constraints and road building regulations constrain the hypothesis space. Sensor data from an in-vehicle vision sensor and from a digital map provide evidence for a particular intersection. Partial observability and different abstraction layers of the input data are naturally handled.

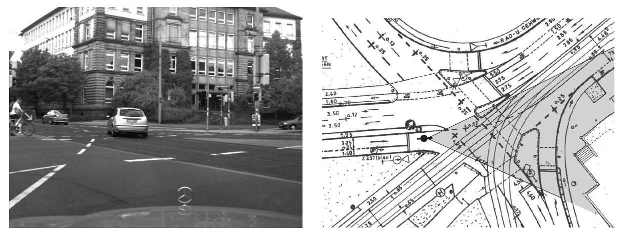
Deductive inference services – namely classification, entailment, satisfiability and consistency – are then used to narrow down the intersection hypothesis space based on the evidence and the background knowledge, and to retrieve intersection information relevant to a user, i.e. a human or a driver assistance system. The paper concludes with an outlook towards non-deductive inference, namely model construction, and probabilistic inference.

I. Motivation

Building on the term Image Understanding ([17]), we define *Intersection Understanding* as the subtask of interpreting an image of a road intersection that enables (at least) the generation of a human-readable, qualitative scene description and an autonomous navigation through the intersection according to traffic rules. The vast majority of current road or intersection recognition systems solely deal with geometric reconstruction. Moreover, the algorithms are restricted to highly specialized domains, e.g. highways. The rare works on intersections focus on one particular, non-complex type of intersection ([12]).

Typically, these methods first extract contour and/or region based cues (edges, their aggregation to lane markings, road texture, ...) from images of an onboard vision sensor. Based on these cues a generic road geometry model of low dimensionality[1] is instantiated. An additionally available model of the vehicle dynamics can be used for tracking and smoothing the parameter estimates over time (cf. [13] and [15] for an overview).

[1]The still popular clothoid representation is often approximated by a second or third order polynomial.



(a) (b)

Fig. 1. **Inner city intersection** 1(a) image taken by an onboard camera with 50° opening angle, 1(b) map from land surveying office.

Despite more than 20 years of research these approaches have failed in proving their scalability from niche solutions to more general applicability. One possible explanation is that an ill-posed estimation problem would then arise. This argument is based on the following observations about inner-city intersections (cf. Fig. 1):

- The abundance of existing intersection geometries necessitates a high-dimensional parameter space.
- A large part of the intersection does not enter the field of view of a standard onboard camera during traversal.
- Dense traffic and inner-city infrastructure lead to a massive amount of occlusion of relevant image clues.
- Frequently omitted markings on the intersection lead to a lack of image cues.
- The presence of an abundance of unmodelled objects feed as noise into the estimation process.
- The image features are of inferior quality, due to an – on average – worse road quality, more variations in marking shape and more rapidly changing lighting conditions.

In brief, a reduced and noisy amount of features contrasts with the necessity of a high dimensional parameter space.

The latter problem can only be adressed by appropriately constraining a high-dimensional parameter space. Such constraints can be derived from general domain knowledge and from specific information about a particular intersection. Due to the complexity of the domain – thus for reasons of understandability, maintainability and extendability – an explicit formulation of this considerable amount of constraints is preferable over an implicit hard-wiring in source code.

A. Knowledge Representation Formalisms

We introduce *Description Logic* (DL, [1]) as a knowledge representation formalism for intersection understanding. Description Logic is a 2-variable fragment of First Order Logic. It provides several advantages compared to other formalisms:

Most DLs are decidable, which means that sound, complete, and terminating algorithms exist. This is a clear advantage over theorem provers for full first-order logic or Horn clauses with function symbols (e.g. PROLOG). The more recently added expressivity on so-called concrete domains (like the natural or real numbers), allows for a more natural integration of quantitative constraints than earlier logic formalisms.

DL axioms are similar to human language which –after a training period– allows for understandable and thus maintainable knowledge bases. Due to their nevertheless rigid formal framework, the chance of semantic ambiguities is reduced with respect to human-to-machine and machine-to-machine communication. The integration of several DL knowledge bases (desirable domains for road and intersection understanding include marking types, traffic signs, traffic participants, . . .) is a common task, understood better for DL (e.g. [3]) than for maybe any other representation formalism.

In contrast to purely geometric lane recognition algorithms, information of different type and abstraction layer can be fused within one coherent framework, as will be shown in this contribution by fusing digital map and video data.

Whereas *Bayesian Belief Nets* (BBN, [19]), the most prominent representation from the probabilistic world, can capture only propositional, i.e. variable-free, statements, DL provides a clear-cut separation between general knowledge („A man with a child is a father.“) and the individuals in the domain („John is a man. Emily is John's child.“). This allows for modular and thus reusable knowledge bases, as well as for more efficient coding of knowledge ([21], [4]). Some DL systems allow to formulate complex queries on the knowledge base (e.g.: „Retrieve all of John's children!“), which is impossible in purely propositional knowledge bases. Additionally, in contrast to BBN, constraints involving lots of input variables can be formulated without jeopardizing performance.

In contrast to *databases*, it can deal naturally with incomplete information due to its open world semantics[2].

Open challenges in Description logics involve the representation of spatial relations among individuals, how to deal with limited inference power due to the monotonicity requirement, and how to incorporate probabilistic information. Fortunately, each of these are active research areas and promising approaches have recently appeared. These problems will be adressed below.

B. Outline

After an introduction into Description Logic (Section 2) we develop a knowledge base for arbitrary roads and intersections (Section 3). Geometric as well as semantical properties are covered. Using input data from a commercially available digital map and from a video sensor, we deductively perform instance classification for missing information (e.g.: „Is this lane a right turn lane?“), query the knowledge base for further entailed information („Which lane is the vehicle on?“), and show how inconsistencies in the knowledge base and in the sensor data can be detected (Section 4). Current limitations of DL based reasoning and possible remedies for them are discussed in Section 5.

[2]Open World semantics denotes that if something cannot be proven to be true, then it is not automatically assumed false.

II. Description Logic

The description logic $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D})^-$ ([8]) is briefly introduced. It augments the basic logic $\mathcal{ALC}$ ([23]) with qualified number restrictions ($\mathcal{Q}$), role hierarchies ($\mathcal{H}$), inverse roles ($\mathcal{I}$), transitive roles ($_{R^+}$), and concrete domains ($\mathcal{D}^-$). $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D})^-$ is supported by the DL reasoning system RACER ([7]), which was used for implementing the subsequently described knowledge base. RACER also supports rules and all of the inference services described below.

A. Concept and Role Descriptions

Atomic concepts and atomic roles are elementary descriptions, denoted by AC and AR. Complex descriptions, denoted by C and D, can be built inductively using concept and role constructors according to the following syntax:

$$\begin{aligned} \mathsf{C}, \mathsf{D} &\longrightarrow \mathsf{AC} \mid \top \mid \bot \mid \neg \mathsf{C} \mid \mathsf{C} \sqcap \mathsf{D} \mid \mathsf{C} \sqcup \mathsf{D} \mid \forall \mathsf{R.C} \mid \\ &\qquad \exists \mathsf{R.C} \mid \exists_{\leq n} \mathsf{R.C} \mid \exists_{\geq n} \mathsf{R.C} \\ \mathsf{R} &\longrightarrow \mathsf{AR} \mid \mathsf{R}^- \end{aligned} \quad (1)$$

The semantics of concept and role descriptions is defined in terms of an *interpretation* $\mathcal{I}$, which consists of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain* of $\mathcal{I}$, and an interpretation function. This function assigns to every atomic concept AC a set $AC^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role AR a binary relation $AR^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function is extended to complex descriptions according to Table I.

Concept and role constructors		
Name	**Syntax**	**Semantics**
Top	$\top$	$\Delta^{\mathcal{I}}$ for concepts, $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for roles
Bottom	$\bot$	$\emptyset$
Conjunction	$\mathsf{C} \sqcap \mathsf{D}$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Disjunction	$\mathsf{C} \sqcup \mathsf{D}$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Value restr.	$\forall \mathsf{R.C}$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b \in \Delta^{\mathcal{I}} : (a,b) \in R^{\mathcal{I}} \Rightarrow b \in C^{\mathcal{I}}\}$
Exists restr.	$\exists \mathsf{R.C}$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}} : (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
Qualified nr	$\exists_{\leq n} \mathsf{R.C}$	$\{a \in \Delta^{\mathcal{I}} \mid \|\|\{x \mid (a,x) \in R^{\mathcal{I}}, x \in C^{\mathcal{I}}\}\|\| \leq n\}$
restriction	$\exists_{\geq n} \mathsf{R.C}$	$\{a \in \Delta^{\mathcal{I}} \mid \|\|\{x \mid (a,x) \in R^{\mathcal{I}}, x \in C^{\mathcal{I}}\}\|\| \geq n\}$
Role Inverse	R^-	$\{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (b,a) \in R^{\mathcal{I}}\}$

TABLE I

SYNTAX AND SEMANTICS OF $\mathcal{ALCQHI}_{\mathcal{R}^+}$.

Qualified number restrictions are only allowed for so-called simple roles, that neither are transitive nor have any transitive subroles.

B. Knowledge Bases

A DL knowledge base is a pair $(\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a set of terminological axioms, called a *TBox*, and $\mathcal{A}$ is a set of assertional axioms, called an *ABox*. The TBox contains intensional general knowledge about the domain and is built through axioms that describe general properties of concepts. The ABox captures extensional knowledge that is specific to the individuals of the domain of discourse.

Terminological Axioms		
Name	**Syntax**	**Satisfied if**
(General) Concept inclusion	$\mathsf{C} \sqsubseteq \mathsf{D}$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Concept equality	$\mathsf{C} \equiv \mathsf{D}$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
Role inclusion	$\mathsf{R} \sqsubseteq \mathsf{S}$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$

Assertional Axioms		
Name	**Syntax**	**Satisfied if**
Concept assertion	$a : \mathsf{C}$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
Role assertion	$(a, b) : \mathsf{R}$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
Semantic Equality	$a = b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
Semantic Inequality	$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

TABLE II

$\mathcal{ALCQHI}_{\mathcal{R}^+}$ KNOWLEDGE BASE AXIOMS.

An interpretation $\mathcal{I}$ *satisfies* a TBox $\mathcal{T}$ iff, for each axiom in $\mathcal{T}$, the condition in the right column of table II are met. Such an interpretation is called a *model* of $\mathcal{T}$: $\mathcal{I} \models \mathcal{T}$.

Quantitative reasoning is supported by introducing *concrete domains*. $\mathbb{N}$and $\mathbb{R}$ are examples for concrete domains. They have been introduced in [2], where an overview of the syntax and semantics of the corresponding axioms can be found.

C. Inference Services

Various standard inference services are provided for DL TBoxes (e.g. [1]):

- **Satisfiability** A concept C is called *satisfiable* with respect to a TBox $\mathcal{T}$ iff there is a model $\mathcal{I}$ of $\mathcal{T}$ where $C^{\mathcal{I}} \neq \emptyset$.
- **Subsumption** A concept D *subsumes* a concept C with respect to a TBox $\mathcal{T}$ iff $C^{\mathcal{I}} \subseteq \mathcal{D}^{\mathcal{I}}$ holds for each model $\mathcal{I}$.
- **Equivalence** Two concepts C and D are equivalent iff $C^{\mathcal{I}} = \mathcal{D}^{\mathcal{I}}$ holds for each model $\mathcal{I}$.

These reasoning services are implemented in most DL reasoning systems. They are useful during the development of a knowledge base to test whether a TBox „makes sense".

With respect to ABoxes the following inference tasks are common:

- **Consistency** An ABox $\mathcal{A}$ is *consistent* with respect to $\mathcal{T}$ iff there exists a model of $\mathcal{T}$ that satisfies each assertion in $\mathcal{A}$.
- **Instance Checking/Entailment** An ABox assertion α is *entailed* by $\mathcal{A}$, written $\mathcal{A} \models \alpha$, iff every model of $\mathcal{A}$ also satisfies α.
- **Instance Classification/Realization Problem** Given an individual a in $\mathcal{A}$ find the most specific concept of which a is an instance, written $\mathcal{A} \models a{:}\mathsf{C}$.
- **Retrieval** Given a concept C find all individuals a such that $\mathcal{A} \models a{:}\mathsf{C}$
- **Conjunctive Query** This task is similar to retrieval, but finds tuples of individuals $(a_1, \ldots, a_n)$ among which a set of stated role and concept assertions must hold.

These tasks are common during usage of the knowledge base within an application. Only some of the currently available DL systems support ABox reasoning, and of these, only few support all of the tasks stated above.

D. Extension with Rules

So-called role chains are not supported in $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D})^-$. Role chains are compositions $\mathsf{R}_1 \circ \ldots \circ \mathsf{R}_n$ of roles. Axioms relating three or more objects are thus not possible, e.g. there is no way to state $\mathsf{hasBrother} \circ \mathsf{hasSon} \sqsubseteq \mathsf{Uncle}$, i.e. that the brother of a father is an uncle. Therefore, recently DL systems get augmented with rules ([9]). With rules, the above fact can be expressed in a first-order syntax:

$$\mathsf{hasFather}(x, y) \wedge \mathsf{hasBrother}(y, z) \rightarrow \mathsf{Uncle}(z) \qquad (2)$$

Rules are used extensively in the knowledge base developed in this contribution.

III. ROAD NETWORK KNOWLEDGE BASE

In the sequel, a knowledge base $\mathcal{KB}$ for road networks is introduced, which is formalized in Description Logic. Its TBox $\mathcal{T}$ describes general knowledge about road networks. The ABox $\mathcal{A}$ initially captures partial information about a particular road or intersection acquired with onboard vehicle sensors. It is later enhanced by new assertions obtained through ABox inference. The description of the rule base $\mathcal{R}$ is omitted here for brevity.

A. The TBox

The TBox of the road network knowledge base introduces relevant concepts of a road network – namely roads, lanes, dividers between roads, road markings and junctions – and the relations that must hold between them.

1) Taxonomy: All atomic concepts are introduced and arranged in a specialization hierarchy called a taxonomy, which is visualized as a UML diagram in Fig. 2(a). Arrows denote inclusion axioms, as e.g.

$$\mathsf{Highway} \sqsubseteq \mathsf{Road} \quad \textit{// All highways are roads.} \qquad (3)$$

To denote that individuals of the superclass have to be a member of at least one subclass, one can use a *covering axiom* instead:

// A road is either an autobahn,
// highway or an urban road. (4)

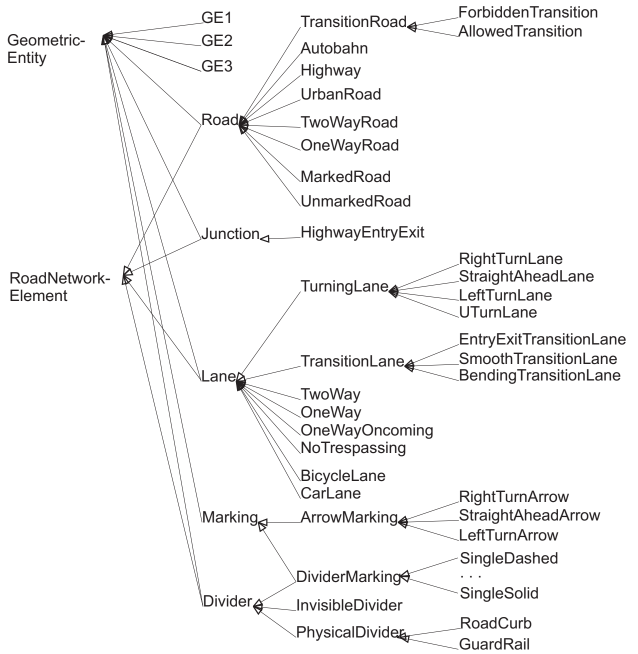
$$\mathsf{Road} \equiv \mathsf{Autobahn} \sqcup \mathsf{Highway} \sqcup \mathsf{UrbanRoad}$$

Disjointness between subclasses can optionally be stated, too:

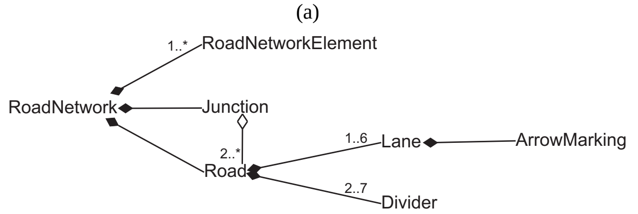
// The set of highways is disjoint from
// the set of urban roads and autobahns. (5)

$$\mathsf{Highway} \sqsubseteq \neg\mathsf{UrbanRoad} \sqcap \neg\mathsf{Autobahn}$$

2) Mapping atomic concepts to geometric primitives: Fig. 2(a) shows that nearly all RoadNetworkElements are also descendents of GeometricEntity, from which three generic types of geometric primitives – GE1, GE2 and GE3 – inherit. The three types are visualized in Fig. 3. Their free parameters are elements of the concrete domain $\mathbb{R}$. These types are pairwise disjoint but no covering axiom is used. This way, a concept can be of no type, which means that it is composed out of several geometric entities, i.e. complex shapes can be built out of the primitive ones. A more exhaustive description of a preliminary version of this geometry model can be found in [12].



(a)



(b)

Fig. 2. **RoadNetwork TBox** 2(a) Taxonomy, 2(b) partonomy. For clarity, only direct parts are visualized, although part-of is transitive.

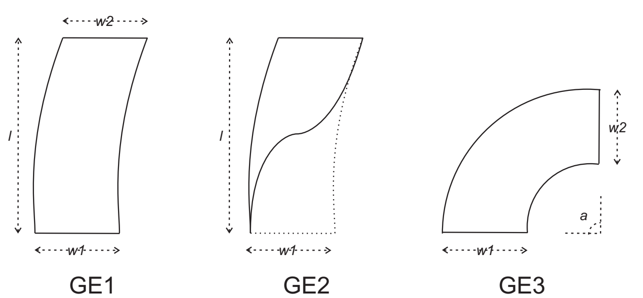
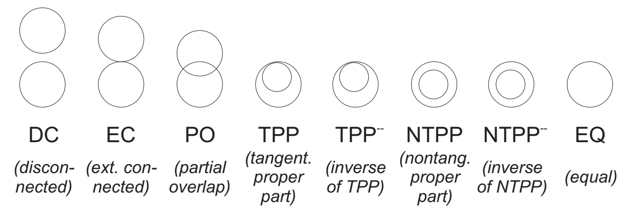
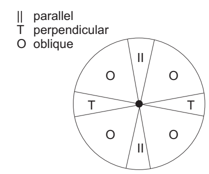


Fig. 3. **Geometric entities** The three generic shapes that are used to construct the road network.

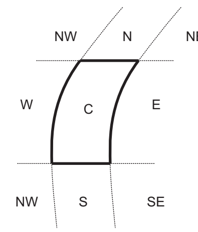
3) Spatial representation: To capture the relative spatial arrangement of scene elements, three types of spatial relations are introduced (see Fig. 4): The *degree of overlap* of two individuals is described using the common RCC8 representation ([22]). Their *relative orientation* is coarsely discretized into three intervals of the unit circle, namely parallel, perpendicular, and oblique. Their *relative position* is discretized using the eight cardinal directions. The types are introduced as roles with the possible values being their subroles. A description of the spatial arrangement of one individual with respect to another comprises three ABox role assertions, one of each type. The set of subroles of each type is *jointly exhaustive and pairwise disjoint* (JEPD). Unfortunately, this property cannot be modelled in $\mathcal{ALCQHI}_{\mathcal{R}^+}(\mathcal{D}^-)$. As a workaround, the complement of each subrole is explicitly introduced (e.g. notNTPP). Each individual pair is required to have either the subrole or its complement stated, and to have exactly one non-complement subrole of each type in total.



(a)



(b)



(c)

Fig. 4. **Spatial Relations** 4(a) degreeOfOverlap using RCC8 relations, 4(b) relativeOrientation, 4(c) relativePosition. 4(b) is defined wrt the first individual's coordinate system. For 4(c), each GeometricEntity has to supply its own „defintion of perspective" as to what it considers north etc. of it. This is depicted here for GE1.

4) Constraints: Using the concept names and the spatial roles, *constraints* are formulated on the set of possible models. Without constraints, arbitrary concept and role assertions could be asserted. The types of constraints roughly fall into two categories: *geometric constraints* and *road building regulations*. A subset of geometric constraints are part-of constraints (NTPP, TPP and their inverse) which are arranged in a *partonomy*. The partonomy is visualized in Fig. 2(b), and written in DL as follows:

// From all road network elements, only lanes
// dividers and arrows can be part of a Road.

$$\text{Road} \sqsubseteq \forall \text{NTPP}.(\text{Lane} \sqcup \text{Divider} \sqcup \text{ArrowMarking}) \quad (6)$$

Cardinality constraints can be imposed on the parts as well:

$$\text{Road} \sqsubseteq \exists_{\geq 1}\text{NTPP}.\text{Lane} \sqcap \exists_{\leq 6}\text{NTPP}.\text{Lane} \quad (7)$$

Compositions are parts that cannot exist without their wholes, denoted by a black diamond ended arrow, written as:

// A lane is part of exactly one road.

$$\text{Lane} \sqsubseteq \exists_{=1}\text{NTPP}^{-}.\text{Road} \quad (8)$$

Besides the very basic geometric constraints captured in the partonomy many more geometric constraints of often much more complex nature must hold, which have to be left out for brevity. One example is:

// An exit lane is connected longitudinally
// only to right or left turn lanes.

$$\text{ExitLane} \sqsubseteq \forall \text{isLongitudinallyConnectedTo}.(\text{RightTurnLane} \sqcup \text{LeftTurnLane}) \quad (9)$$

The role isLongitudinallyConnectedTo is asserted iff its parents EC, II and LON hold. The latter is asserted if either N or S are true (cf. Fig. 4). The role isLaterallyConnectedTo is defined analogously. From these, is[east|west|north|south]ConnectedTo are derived with the obvious semantics.

The road building regulations comprise many rules that human drivers have internalized and that are used when approaching an unknown intersection. Few examples include:

$$
\begin{aligned}
&\textit{// Only right turn lanes can be right of right turn lanes.}\\
&\mathsf{RightTurnLane} \sqsubseteq \forall\, \mathsf{hasEastNeighbor.RightTurnLane}\\
&\textit{// All autobahns and highways are one way roads.}\\
&\mathsf{Autobahn} \sqcup \mathsf{Highway} \sqsubseteq \mathsf{OneWayRoad}\\
&\textit{// A one way road is defined as a road}\\
&\textit{// which has only one way lanes.}\\
&\mathsf{OneWayRoad} \equiv \mathsf{Road} \sqcap (\forall\, \mathsf{NTPP.OneWayN} \sqcup\\
&\forall\, \mathsf{NTPP.OneWayS})\\
&\textit{// A one way road does not have a uturn lane.}\\
&\mathsf{OneWayRoad} \sqsubseteq \forall\, \mathsf{NTPP.}\neg\mathsf{UTurnLane}
\end{aligned}
\tag{10}
$$

The role hasNeighbor is asserted iff II and LAT holds, and if both individuals are of the same type. A connection EC is not necessary. The subroles are derived analgously to isConnectedTo.

The hypothesis space that emerges from this combined conceptual-geometrical intersection representation is by orders of magnitude smaller compared to a traditional, purely geometric representation, as the vast majority of hypotheses are elegantly ruled out on the conceptual level using constraints. At the same time, complex geometries are representable with sufficient accuracy, as shown exemplarily for an inner-city area in [12].

B. The ABox

A non-stationary video camera and a commercially available digital map along with positioning devices are used as input devices. All are readily availabe in our experimental vehicles ([25], [11]). However, arbitrary informative (in terms of the terminology introduced by the TBox) sensors can be used, provided they operate in a common coordinate frame. After a brief description of the data registration, the mapping from map and video data to ABox axioms is described.

1) Coordinate systems and data registration: All computations are done within a cartesian, vehicle-centered, two-dimensional coordinate system in the road plane (assuming a locally *flat earth*). A common frame is needed to enable the computation of spatial relations between objects as introduced in sec. III-A.3. Map data is transformed using the common UTM projection from geographical to cartesian – and thus length- and angle-preserving – coordinates, and then applying a map matching algorithm as described in [10]. As this version did not deliver lateral position estimates within the road, the ego lane has been manually assigned. However, lane precise estimates will become available in the future.

Video input data is transformed by using a calibrated camera wrt the vehicle coordinate system, and by additionally knowing the camera's height above ground and yaw angle.

2) Map input data: Digital maps are produced for navigation purposes only, which means that the road network topology is captured correctly, but not the geometry. Roads are represented in the form of coarsely digitized straight line segments whose start and end coordinates coincide with junctions. Junctions are merely represented by a coordinate pair ([10]). Few more attributes are available: the road class, ranging from freeway to pedestrian mall, the allowed driving directions, and few more, which are not of importance here. Current work on the map provider side involves including the number of lanes per road.

This data is used to generate geometry patches as shown in Fig. 5. At the moment, the geometry is generated according to some fixed non-logic rules. Transferring this task to DL inference is the subject of ongoing work. The rules implicitly used during generation of the above described spline patches will then be explicated, too, as logic rules like:

$$
\begin{aligned}
&\textit{// A two lane road connected to a three lane road must}\\
&\textit{// contain (cf.Fig.4(a)) some GE2-shaped lane (cf.Fig.3).}\\
&\mathsf{Road} \sqcap \exists_{=2}\mathsf{NTPP.Lane} \sqcap \exists_{=3}\mathsf{isLongitudinally\text{-}}\\
&\mathsf{ConnectedTo.Lane} \sqsubseteq \exists \mathsf{NTPP.(Lane} \sqcap \mathsf{GE2)}
\end{aligned}
\tag{11}
$$

This leaves room for uncertainty in the model, as, in this example it is not hardwired whether the GE2-shaped lane is the rightmost or the leftmost lane of that road.

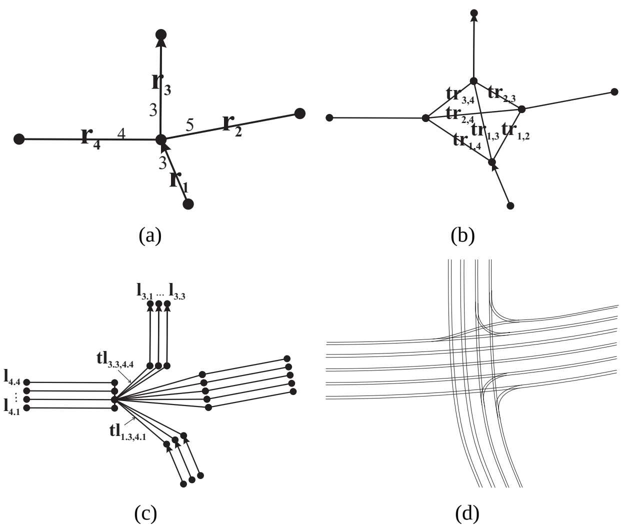


Fig. 5. **Map Preprocessing** A full-fledged geometry model as described in [12] is generated, whose parameters are fed into the ABox. 5(a) original map, including the number of lanes, 5(b) shortening of the original roads and pairwise insertion of additional roads in the middle (subsequently called transition roads), 5(c) generation of lanes from roads, and generation of transition lanes from transition roads pairwise between *all* connected lanes (for clarity only those for lane $l_{4.2}$ are visualized). In the course of reasoning most of these individuals will be classified as NoTrespassing. 5(d) geometry generation for each lane and each lane transition by assigning a geometric entity subconcept including concrete domain values (Fig. 3) to each individual. All lanes are GE1. For clarity, only few transition lanes have been visualized (several GE1, one GE2, and two GE3)).

Each generated patch l_i and d_j is then included in the ABox by stating l_i:Lane and d_j:Divider, respectively, and by adding their types of geometric entity, e.g.: l_i:GE3, and the respective concrete domain fillers.

Each road r_k is added by stating

$$
\begin{array}{ll}
r_k : \mathsf{Autobahn} & \textit{// } \in \textit{Autobahn, Highway, UrbanRoad}\\
r_k : \mathsf{TwoWayRoad} & \textit{// } \in \textit{TwoWayRoad, OneWayRoad}\\
(r_k, l) : \mathsf{NTPP} & \textit{// for each lane } l \textit{ that is part of } r_k
\end{array}
\tag{12}
$$

For the transition roads $tr_{i,k}$ an axiom $(tr_{i,k}, j)$: NTPP^{-} is added additionally, where j is a generated instance of type Junction.

To state that our map knowledge is assumed complete we add the following *local closed world assumptions*:

$$
\begin{array}{ll}
rn : \exists_{\le i}\mathsf{NTPP.Road} & \textit{// with } i = |R|\\
rn : \exists_{\le j}\mathsf{NTPP.Lane} & \textit{// with } j = |L|\\
rn : \exists_{\le k}\mathsf{NTPP.Divider} & \textit{// with } k = |D|\\
rn : \exists_{\le l}\mathsf{NTPP.Junction} & \textit{// with } l = |J|
\end{array}
\tag{13}
$$

The spatial relations of all GeometryEntity individuals with respect to each other, as introduced in Section III-A.3, which have been previously computed, are added to the DL via statements of the form $(l_{4.i}, l_{4.j}$: II) for all i, j, for example.

3) Video input data: Video-based object detections are straightforwardly integrated in the ABox with axioms like, e.g.

$$\begin{array}{ll} a_1 : \mathsf{StraightAheadArrow} & \textit{// a1 is newly introduced.} \\ d_3 : \mathsf{SingleDashedDivider} & \end{array} \quad (14)$$

and additional axioms stating the spatial relations to all other GeometricEntities, like, e.g.

$$\begin{array}{ll} (a_1, d_3) : \mathsf{DC} & \textit{// } a_1 \textit{ and } d_3 \textit{ are disconnected,} \\ (a_1, d_3) : \mathsf{II} & \textit{// parallel,} \\ (a_1, d_3) : \mathsf{E} & \textit{// and } d_3 \textit{ is east of } a_1. \end{array} , \quad (15)$$

which have to be computed (automatically in a straightforward manner) outside of the DL system. As few reasoning as possible should be done *within* an object detection algorithm. For example, don't hastily conclude Autobahn from a detected GuardRail. Instead, exclusively state the detection result of specialised video object detectors and leave the inference to the KB, where these topics are dealt with more thoroughly. In this example, a guard rail can occur on highways and even on urban roads.

It is worth mentioning that the ABox can be built incrementally online, just as usually more data about the intersection will become available during approach. Instead of recomputation of the ABox just an additional axiom has to be added.

IV. EXPERIMENTS

The knowledge base is denoted with $\mathcal{KB} = (\mathcal{T}, \mathcal{A}, \mathcal{R})$. $\mathcal{T}$ is of the form described in sec. III-A, $\mathcal{A}$ is initially empty. The description of the rule base $\mathcal{R}$ has been omitted for brevity. After a description of the features detected by the video sensor and the map, classical deduction is used to draw conclusions about the properties of the intersection. At first, the types of several lanes are deduced using instance classification. Then it is shown how the ego lane of the vehicle is inferred using entailment. Eventually it is demonstrated how both the sensor data and the domain knowledge can be tested for inconsistencies.

A. Evidence

We are approaching the intersection depicted in Fig. 1 (land surveying office ground truth map) and 5(a) (digital map). The digital map is automatically processed to generate ABox axioms as described in sec III-B.2.

The map matching correctly yields that the vehicle is on r_4:

$$\begin{array}{ll} egovehicle : \mathsf{Vehicle} & \textit{// create new vehicle} \\ (egovehicle, r_4) : \mathsf{NTPP}^- & \textit{// is inverse proper part of road } r_4 \\ (egovehicle, r_4) : \mathsf{II} & \textit{// is parallel to road } r_4 \end{array} \quad (16)$$

The object detectors described in [5] and [26] have processed an image taken one second before the one shown in Fig. 1(a) and detected the two dividers and the arrow shown in Fig. 6. The following statements are thus additionally automatically added to the ABox:

$$\begin{array}{ll} arrow_1 : \mathsf{StraightAheadArrow} & \\ divider_1 : \mathsf{RoadCurb} & \\ divider_2 : \mathsf{Marking20\text{-}50} & \\ \textit{// known spatial relations (Fig. 4)} & \\ \textit{// wrt all other geometric entities:} & \\ (arrow_1, egovehicle) : \mathsf{II} & \textit{// parallel,} \\ (arrow_1, egovehicle) : \mathsf{S} & \textit{// south of,} \\ (arrow_1, egovehicle) : \mathsf{DC} & \textit{// and disconnected.} \\ \ldots & \end{array} \quad (17)$$

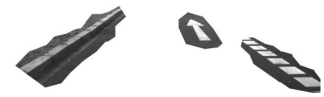


Fig. 6. **Illustration of video-based object detectors.**

B. Task 1: Determining lane types (Instance Classification)

Several assertions are immediately available after *classifying* the individuals. Classification can be seen as a special case of *entailment* where the entailed axiom is a concept assertion.

Results 1-4:

$$\begin{array}{lll} \mathcal{KB} & \models & l_{4.1} : \mathsf{OneWayN} \sqcap \mathsf{BicycleLane} \sqcap \\ & & \quad \mathsf{StraightAheadLane} \\ & & l_{4.2} : \mathsf{OneWayN} \sqcap \mathsf{CarLane} \sqcap \\ & & \quad \mathsf{StraightAheadLane} \\ & & l_{4.3} : \mathsf{OneWayS} \\ & & l_{4.4} : \mathsf{OneWayS} \end{array} \quad (18)$$

All classification results can be automatically added to the ABox. Although only the classification results for Lane individuals of road r_4 are given in Eq. 18, all ABox individuals, e.g. all dividers d_j, are classified this way.

Results were given here without proof. To provide a better understanding on how additional assertions are inferred, a proof sketch for a particular entailment query is given in the next subsection.

C. Task 2: Determining the vehicle's ego lane (Entailment)

The ego lane of our vehicle is not determined by the positioning device. However, the ego lane can be deduced from the available domain knowledge and the sensor data. Querying for assertions that are *entailed* in this knowledge base yields:

Result5:

$$\begin{array}{l} \textit{// The vehicle is on lane } l_{4.2}. \\ \mathcal{KB} \models (egovehicle, l_{4.i}) : \mathsf{NTPP}^- \text{ , iff } i = 2 \ . \end{array} \quad (19)$$

Proof sketch:

The TBox of the $\mathcal{KB}$ contains the following statements:

$$\begin{array}{l} \textit{// Marking50-20 is a divider for bicycle lanes.} \\ \mathsf{Marking50\text{-}20} \sqsubseteq \exists \mathsf{isLaterallyConnectedTo.BicycleLane} \\ \textit{// This type of arrow only occurs on car lanes.} \\ \mathsf{Arrow} \sqsubseteq \forall \mathsf{NTPP}^-.\mathsf{CarLane} \\ \textit{// Bicycle lanes are not next to each other.} \\ \mathsf{BicycleLane} \sqsubseteq \forall \mathsf{hasNeighbor.CarLane} \end{array} \quad (20)$$

These deductively lead to:

Result 5a:

$$\begin{array}{l} \textit{// The driver's lane is a car lane (fortunately :)),} \\ \textit{// and right of it, there is a bicycle lane.} \\ \mathcal{KB} \models egovehicle : \exists \mathsf{NTPP}^-.(\mathsf{CarLane} \sqcap \\ \exists \mathsf{hasEastNeighbor.BicycleLane}) \end{array} \quad (21)$$

From the third axiom in Eq. 20 we know as well:

Result 5b:

$$\begin{aligned}&\textit{// Right of the bicycle lane there can only be a car lane.}\\&\mathcal{KB} \models egovehicle : \exists \mathsf{NTPP}^{-}.(\mathsf{CarLane} \sqcap\\&\exists \mathsf{hasEastNeighbor}.(\mathsf{BicycleLane} \sqcap\\&\forall \mathsf{hasEastNeighbor}.\mathsf{CarLane}))\end{aligned} \tag{22}$$

The remaining necessary TBox axioms are described only textually for brevity: A lane with a straight ahead arrow is a straight ahead lane. Right neighbors of straight ahead lanes are only straight ahead or right turn lanes. Bicycle lanes do not occur between lanes of the same turning lane type. Therefore,

Result 5c:

$$\begin{aligned}&\textit{// Right of the bicycle lane can only be a right turn lane.}\\&\mathcal{KB} \models egovehicle : \exists \mathsf{NTPP}^{-}.(\mathsf{CarLane} \sqcap\\&\exists \mathsf{hasEastNeighbor}.(\mathsf{BicycleLane} \sqcap\\&\forall \mathsf{hasEastNeighbor}.\mathsf{RightTurnLane}))\end{aligned} \tag{23}$$

Road r_1 is a one way road towards the junction. Therefore, r_4 cannot have any right turn lane at all. Therefore,

Result 5d:

$$\begin{aligned}&\textit{// There is no more lane right of the bicycle lane.}\\&\mathcal{KB} \models egovehicle : \exists \mathsf{NTPP}^{-}.(\mathsf{CarLane} \sqcap\\&\exists \mathsf{hasEastNeighbor}.(\mathsf{BicycleLane} \sqcap\\&\neg\exists \mathsf{hasEastNeighbor}))\end{aligned} \tag{24}$$

From this, together with the closure assumption from Eq. 13 we can deductively infer that the ego vehicle is a proper part of (geometrically, concerning its projection on the road plane (see sec. III-B.1)) lane $l_{4.2}$. □

This consequence is confirmed by looking at the corresponding land surveying office map form Fig. 1(b). As entailment is a standard reasoning task this fact is obtained instantly with an appropriate reasoner.

Likewise, many further assertions about the intersection are entailed in the knowledge base. They can be viewed as inferred constraints on the intersection hypothesis space. Those hypotheses that have not been ruled out can then be tested using variants of the classical lane recognition methods described in the introduction. An initial hypothesis testing algorithm has been developed in [12].

D. Task 3: Detecting Inconsistencies (Satisfiability)

Inconsistencies in both TBox and ABox are detected straightforwardly by checking for TBox satisfiability and ABox conistency, respectively. Defining a new TBox concept by stating

$$\begin{aligned}&\mathsf{TwoWayAutobahnLane} \equiv\\&\quad\mathsf{TwoWayLane} \sqcap \exists \mathsf{NTPP}^{-}.\mathsf{Autobahn}\end{aligned} \tag{25}$$

or, alternatively, stating in the ABox, that

$$lane_1 : \mathsf{TwoWayLane} \sqcap \exists \mathsf{NTPP}^{-}.\mathsf{Autobahn} \tag{26}$$

will immediately lead to an insatisfiable TBox in the former case as TwoWayAutobahnLane will never have any instances, and to an inconsistent ABox in the latter.

V. Extensions of Classical Logic

A. Model Construction

Classical logic is confined to purely *deductive reasoning*, which poses limits to its expressivity. First, once a conclusion is sustained by a valid argument, this argument can never be invalidated, no matter which new assertions are added. This is known as *monotonicity*. This contrasts the modern understanding of vision, that hypotheses are be generated based on partially missing evidence via *jumping to conclusions* ([6]), which implies that the arrival of new information will oftentimes lead to the withdrawal of hypotheses. This process is also known as *belief revision*. Second, deduction cannot create new individuals. Transferred to scene interpretation, this amounts to delivering a complete low-level a priori segmentation of the scene. However, the past 50 years of research in computer and biological vision have shown that a purely data-driven segmentation is not feasible, as already low-level segmentation is crucially dependent on top-down input from higher processing levels ([14]).

In accordance with [18], [20], [24], we conclude that classical deductive reasoning is not sufficient in general for real-world scene interpretation. Instead, deductive and hypothetical reasoning must be combined. This way, the soundness of deduction, as demonstrated in this contribution, can be united with the more far-reaching conclusions possible in non-monotonic reasoning. *Logical model construction* can be seen as an instance of hypo-deductive reasoning. From the model construction perspective, scene interpretation amounts to incrementally constructing a (partial) logical model of the TBox axioms that is consistent with the ABox axioms ([18], [24]).

The development of a model construction algorithm that is suitable for scene interpretation is the focus of ongoing work. The model construction algorithm enhances the reasoning capabilities described in this chapter by *incrementally* hypothesizing new individuals, i.e. lane patches in this example, which have not been asserted in the ABox yet, based on the evidence and on previous hypotheses. Additionally, the *set of all models*, i.e. the set of all plausible intersection hypotheses in this example, can be constructed if desired.

In general, a model construction algorithm proceeds in close analogy to the *Tableau Calculus* algorithms that are used for satisfiability testing of TBoxes and consistency testing of ABoxes (cf. [1]). They apply a set of so-called consistency-preserving completion rules to the original ABox. The presence of $\sqcup$ and $\exists_{\leq n}$ symbols in the TBox triggers so-called non-deterministic rules which split the ABox in a depth-first-search way. The process stops when no more rules can be applied or when an obvious contradiction occurs. In the first case the generated model satisfies all TBox and all ABox axioms. The latter proves that the knowledge base is inconsistent.

Although tableau calculi are suitable to scene interpretation as new individuals are generated and as – in principle – the set of all models of the knowledge base can be generated, some modifications are necessary. Implementations of tableau calculi do not output the model but a mere yes/no answer, they are highly optimised and therefore produce rather „canonical models“, and they stop after having found one model, preferring simpler ones. In scene interpretation, a preference for a simple model makes no sense, and in many cases it is desirable to return all models instead of just one. [16] have shown an initial sketch of how a model construction could be implemented on top of the RACER reasoner, using its available inference services and query language.

B. Probabilistic Logic

Even though non-determinism is common in a DL knowledge base due to the presence $\sqcup$ and $\exists_{\leq n}$ symbols and due to the open world semantics, this is not sufficient in general to describe the uncertainty in both the domain knowledge and the sensor data. If an interpretation violates only one axiom of the KB, then by definition it cannot be a model of the KB anymore. Instead, it is desirable to ask for the „amount of entailment“ of a formula. Future work will focus on using the developed knowledge base with a probabilistic description logic reasoner (see e.g. [1], [16]) for an overview), additionally supplying TBox and ABox axioms with attached probabilities which reflect the amount of uncertainty in the respective statements.

VI. Conclusion

This contribution argues for a general paradigm shift towards a stronger acknowledgement of the role of knowledge engineering in real world high-level scene interpretation tasks. Up to date, no satisfying knowledge representation and reasoning framework for such estimation tasks exists.

We introduced Description Logic extended by rules as a knowledge representation formalism for the sensor-based understanding of complex roads and intersections. It was demonstrated how highly incomplete sensor data, coming on various abstraction layers, can be fused within one coherent and semantically sound framework. Data can be processed iteratively on arrival, not requiring a recomputation of the hypothesis space. The stated domain knowledge is extendable to other domains like traffic signs, traffic participants, etc. Deductive reasoning was used to constrain and query the intersection hypothesis space, and to detect inconsistencies in the sensor data as well as in the stated domain knowledge.

In summary, the more recent developments in logic provide a highly promising framework for developing, constraining and querying the large and complex hypothesis spaces that are typical for image understanding tasks.

Future work includes refinement of the spatial relations, and embedding the existing knowledge base and reasoning services in a model construction framework, which is capable of hypothesizing new individuals and of non-deterministic assertions, to incrementally build the set of all logical models of an intersection. Eventually, this framework will be ported to a probabilistic logic.

VII. Acknowledgement

This work was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) within the scope of the Transregional Collaborative Research Centre on Cognitive Automobiles (SFB/Tr 28).

References

[1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[2] F. Baader and P. Hanschke. A schema for integrating concrete domains into concept languages. In *In Proc. 12th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 452–457, 1991.

[3] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *J. Data Semantics*, 1:153–184, 2003.

[4] P. Domingos. What's missing in AI: The interface layer. In P. Cohen, editor, *Artificial Intelligence: The First Hundred Years*. AAAI Press, to appear.

[5] C. Duchow. A novel, signal model based approach to lane detection for use in intersection assistance. In *IEEE Intelligent Transportation Systems Conference*, 2006.

[6] R. L. Gregory. Knowledge in perception and illusion. *Philosophical Transactions of the Royal Society B (Biological Sciences)*, 352(1358):1121–1128, August 1997.

[7] V. Haarslev and R. Möller. Racer system description. In *In Proc. International Joint Conference on Automated Reasoning (IJCAR), LNAI 2083*, pages 701–705. Springer, 2001.

[8] V. Haarslev and R. Möller. Practical reasoning in racer with a concrete domain for linear inequations. In *Proceedings of the International Workshop on Description Logics (DL-2002), Toulouse, France, April 19-21*, pages 91–98, 2002.

[9] I. Horrocks and P. F. Patel-Schneider. A proposal for an OWL rules language. In *In Proc. 13th ACM International World Wide Web Conference (WWW)*, 2004.

[10] B. Hummel. *Dynamic and Mobile GIS: Investigating Changes in Space and Time*, chapter Map Matching for Vehicle Guidance. CRC Press, 2006.

[11] B. Hummel, S. Kammel, T. Dang, C. Duchow, and C. Stiller. Vision-based path-planning in unstructured environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 176–181, 2006.

[12] B. Hummel, Z. Yang, and C. Duchow. Kreuzungsverstehen – ein wissensbasierter Ansatz. *IT-Schwerpunktheft Fahrerassistenzsysteme*, 1:5–16, 2007.

[13] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing technqiues for traffic applications. *Image and Vision Computing*, 21:359–381, 2003.

[14] T. Lee, D. Mumford, R. Romero, and V. Lamme. The role of the primary visual cortex in higher level vision. *Vision Research*, 38:2429–2454, 1998.

[15] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7:1:20–37, 2006.

[16] R. Möller and B. Neumann. Ontology-based reasoning techniques for multimedia interpretation and retrieval. In Y. Kompatsiaris P. Hobson, editor, *Semantic Multimedia and Ontologies : Theory and Applications*. to appear, 2007.

[17] B. Neumann. Bildverstehen – ein Überblick. In G. Görz, editor, *Einführung in die künstliche Intelligenz*, pages 559–588. Addison-Wesley, 1993.

[18] B. Neumann and R. Möller. On scene interpretation with description logics. In H. I. Christensen and H.-H. Nagel, editors, *Cognitive Vision Systems: Sampling the Spectrum of Approaches*, volume 3948 of *Lecture Notes in Computer Science*, pages 247–278. Springer Berlin / Heidelberg, 2006.

[19] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[20] D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[21] D. Poole. Logic, knowledge representation and bayesian decision theory. In *International Conference on Computational Logic (CL)*, 2000.

[22] D. A. Randell, Z. Cui, and A. Cohn. *A Spatial Logic Based on Regions and Connection*, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.

[23] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[24] C. Schröder. *Bildinterpretation durch Modellkonstruktion: Eine Theorie zur rechnergestützten Analyse von Bildern*. DISKI 196, Infix, 1999.

[25] C. Stiller, G. Färber, and S. Kammel. Cooperative cognitive automobiles. In *Proc. Intelligent Vehicles Symposium*, 2007.

[26] Z. Yang. Fahrbahngeometriemodellierung und videobasierte Pfeilmarkierungserkennung. Master's thesis, Universität Karlsruhe, Institut für Mess- und Regelungstechnik, 2006.