Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 26 (2008) 53–66
www.elsevier.com/locate/imavis

Conceptual representations between video signals and natural language descriptions

M. Arens *, R. Gerber, H.-H. Nagel

Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik der Universität Karlsruhe (TH), 76128 Karlsruhe, Germany

Received 13 July 2004; received in revised form 17 June 2005; accepted 21 July 2005

Abstract

An artificial cognitive vision system associates video signals with conceptual descriptions of the depicted time-varying scene. This linkage is mediated by knowledge representation formalisms. An experimental implementation of such an approach yielded initial results for the conceptual description of videos recorded at innercity traffic scenes, see [M. Haag, H.-H. Nagel, Incremental recognition of traffic situations from video image sequences, Image and Vision Computing 18 (2) (2000) 137–153]. Accumulating experience with this system approach and its extension for the generation of natural language texts from videos caused us to redesign the overall computer vision system as well as the knowledge representation formalisms utilised within that system.

Keywords: Cognitive vision; Knowledge representation

1. Introduction

An experienced driver anticipates impending movements of most pedestrians, bicyclists, and vehicles within his field of view. Anticipatory reactions occur so 'instinctively', i.e. fast and smoothly that prior conscious deliberations appear implausible. In case the driver will be asked why he reacted the way he did, however, a terse or more verbose argument will be formulated to explain the driver's reactions.

Based on the assumption that such a scenario appears acceptable to the majority of people reading this contribution, we infer that at least five levels of representation seem to be involved: (i) a representation of the *geometry* of spatiotemporal developments in the road traffic scene, comprising both a 2D-one in the image plane and a 3D-one relating to the depicted scene, (ii) a representation of driving maneuvers closely coupled to particular traffic situations, (iii) a *conceptual* representation of visible bodies, their attributes, and their elementary movements, (iv) generic conceptual representations of spatiotemporal body configurations and their expected temporal developments, and (v) one or more versions of a natural language representation of developments centred around the current point in time. The two types of conceptual representation mentioned last, i.e. (iv) and (v)—comprise the 'elementary' ones mentioned before (i.e. (i)–(iii)) as building blocks, but usually extend to much larger spatial and temporal scales than the more elementary ones.

This sketch will not be defended dogmatically. It should allow the reader, though, to roughly position the topic of this contribution, namely a study of a particular conceptual representation for behaviour, a *Situation Graph Tree* (*SGT*). Traffic scenarios will be used to concretise the discussion. If our arguments convince, it should be easy to imagine other application domains.

Earlier investigations regarding the definition and use of SGTs for the representation and recognition of vehicle behaviour have been documented in [20]. In the meantime, the accumulation of experience unfolded our research into several branches which will be treated at different levels of detail in this contribution:

* Corresponding author.
E-mail address: michel_arens@web.de (M. Arens).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.07.026

(1) The SGTs presented in [20] constituted the result of an exploratory investigation into the use of this formalism. It appears to be time now to formulate our accumulated experience into rules which should facilitate its extension to new application domains as well as the use of this formalism by others.
(2) Tools have been developed and made generally available under GNU Public License in order to ease the initial formulation of SGTs and their subsequent extension or adaptation, see [1].
(3) The generic representation of vehicle behaviour in road traffic scenes provided by an SGT will be instantiated on the basis of tracking results obtained by the signal- and geometry-related modules of an overall system. The instantiation of an SGT in turn provides the input for modules which convert this information into natural language descriptions of the developments recorded by a video.

Section 3 summarises the status of investigations reported in [20] and thereby provides a starting point for the subsequent discussion in Section 4 which offers a more systematic basis for the formulation of SGTs and the vision system itself. Results obtained with the reformulated and extended approach are reported in Section 5.

2. Related work

The following survey emphasises the representation and usage of conceptual knowledge—especially about agent behaviour—within machine vision systems. Video sequences—as most technically produced data streams—exhibit an inherent degree of uncertainty due to noise. In addition, conceptual descriptions to be extracted from this input data often use vague concepts because they are designed to communicate with a human user and, therefore, have to rely on vague concepts used by humans. Both types of uncertainty can be handled in different ways. An initial and frequently encountered step to overcome noise consists in abstracting from numerical data to symbolic or conceptual representations. Several groups use probabilistic representation formalisms and justify this usage by the uncertainty inherent to video sequences, as treated, e.g. in [6,12,13,29]. Explicit formalisation of uncertainty and vagueness is possible by applying a fuzzy extension of predicate logics, for example as reported in [18,20,37]. The conceptual description of video sequences has to link quantitative data to qualitative concepts. This linkage has to make use of some kind of background knowledge. Different approaches make use of different sources of such background knowledge. Several publications are concerned with learning the structure of the input data and relations therein as in the case of those reported in [9,23,32,35] (see, e.g. [11] for an overview). Some approaches prefer the explicit modelling of background knowledge, as described in [10,36,39–41]. Other groups combine the advantages of learning approaches and of explicit modelling—see, e.g. [5–8,14–16,21,24–27,34].

The background knowledge employed for conceptual video description has to be formalised such that it is accessible by the computer vision system. In cases where the background knowledge is learned from the input video data, formalisms are used which naturally link to numerical data. These formalisms include *Hidden Markov Models* (*HMM*) (see, e.g. [6,27], in which these models are combined with a grammar-based formalism to incorporate a-priori knowledge, and [9,35], where the learned models are also used for generation purposes). The HMM-formalism is in some cases modified to fit certain peculiarities of video sequences (*Variable Length Markov Models* (*VLMM*), see [15,16], and *Coupled Hidden Markov Models* (*CHMM*), see [9,35]). Another family of formalisms which also directly link to numerical input data are *Bayesian Networks* (*BNs*). This formalism is used in several varieties by different groups (see [22–26]; *Recurrent-BNs*, see [29]). Approaches which emphasise the explicit modelling of background knowledge are more concerned with easy ways to input and maintain high-level knowledge. These approaches use several—often hierarchical—forms of graph-structures (*Visual Networks*, see [24–26]; *Scenarios*, see [36,39–41]). Most of these are textually entered by an operator and pre-compiled into representations usable during video analysis.

There are several different scopes of application domains discussed for artificial vision systems: some groups are interested in more surveillance-like systems. This leads to systems which can detect a few simple concepts in video streams [32]. Others—surveillance-oriented systems, too—can recognise again only a few, but more complex concepts (see, e.g. [14,22,34,39]). Some groups envisage a more complete description of a time-varying scene depicted in a video in contrast to the surveillance systems. This approach leads to systems which can recognise many complex concepts (see, e.g. [26]).

The above-mentioned range and complexity of system approaches directly affect the runtime performance of the systems surveyed here: surveillance-oriented systems often aim at on-line, real-time processing of video streams—see [9,12,13,22,23,32,39], where the aim of (near-)real-time performance of the system leads to a combination of bottom-up with top-down approaches. Other approaches, which aim at a complete conceptual description of videos, result in runtimes which still allow only off-line processing of video streams (see [20] or [26]). The same background knowledge exploited by computer vision is rarely used, too, for the *generation* of synthetic videos. Only few publications envisage such knowledge representations: [4] (see also [38]) used their background knowledge representation to analyse video sequences and re-create aspects of the conceptualised scene, where re-creation serves the purpose of improving the conceptual description created from video signals. The authors of [39] use their representation formalisms to create synthetic behaviours, which can be used to

feed or control their vision system in order to check the system's completeness. Alternatively, some authors use synthetic agents to create training data for their vision system (see, e.g. [35]).

Our system aims at an encompassing *conceptual* description of video sequences (see, too [26]). This ultimately (and hopefully) culminates in the generation of encompassing *natural language textual descriptions* of a scene observed by a computer vision system. On the processing path from video signals towards the desired descriptions, we explicitly model each part of the background knowledge used (compare, e.g. [39]), in contrast to the learning approaches cited above. This explicit modelling increases the ability to track down errors or shortcomings of the present system and thus allows for improvements of the overall descriptive capability. We use *Situation Graph Trees* (SGTs) as representation formalism for behavioural knowledge. These graph-like structures allow us to easily incorporate and extend the knowledge needed for describing temporally extended behaviours observable in a scene. Moreover, these structures allow to exploit behavioural background knowledge for descriptive and generative usage, as reported in [3] (compare, too [38,39]). A fuzzy extension of predicate logic provides an underlying inference mechanism which facilitates a well-defined and analysable knowledge base.

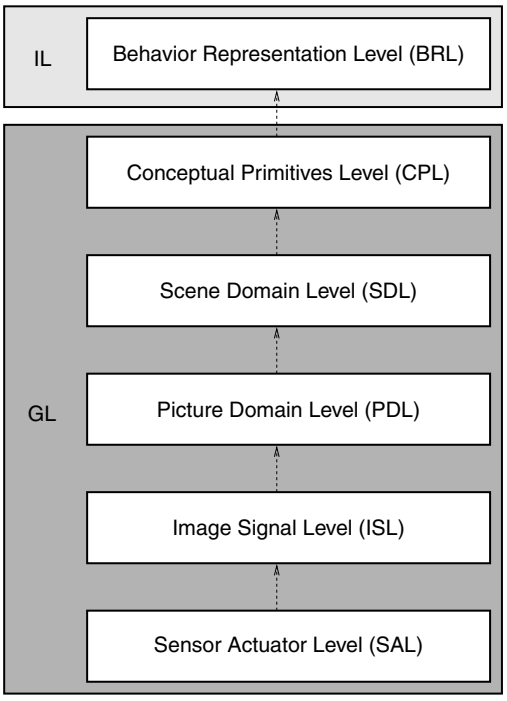


Fig. 1. Sketch of the architecture of the overall computer vision system discussed in [20].

3. Incremental recognition of traffic situations from video image sequences

In [20], the authors reported experimental results on the algorithmic generation of conceptual descriptions from video sequences. Innercity traffic had been chosen as the experimentation domain, for two reasons: on the one hand, this domain exhibits enough complexity—with regard to agent interaction and temporal extent of observable behaviours—to facilitate an intensive test of their approach. On the other hand, the task to track the depicted agents, i.e. the vehicles driving on the streets, and to describe their behaviour in this domain is not too complicated because the agents are rigid bodies[1] and they are *supposed* to follow the rules set for vehicular road traffic.

The system architecture outlined in [20] (roughly) consists of two sub-systems, namely the *geometric layer* (GL, comprising SAL, ISL, PDL, SDL, and CPL, see Fig. 1) and the *inference layer* (IL, consisting of only BRL, compare Fig. 1). The geometric layer is concerned with updating the geometrical scene description at each (half-)frame time point. This includes the initialisation and tracking of moving vehicles. The state estimates for each depicted (and recognised) agent comprise numerical values for location, orientation, speed, and rotation rate. The geometric layer does not stop there, however, but associates *fuzzy* conceptual attributes with these numerical values. These attributes include, for example, concepts like *slowly*, *normal*, or *fast* to conceptually describe at what speed a particular vehicle is driving, but also concepts relating agents to certain locations in the scene, to other agents, or to certain points or areas on the lane. Altogether, there are 13 basic relations and more than 38 possible conceptual attribute values (compare, too [18]). The basic conceptual scene description derived in the geometric layer is passed on to the inference layer. This layer is based on a *Fuzzy Metric Temporal Horn Logic* (FMTHL) which has been introduced by Schäfer [37]. It constitutes an extension of first-order predicate (Horn) logic by explicitly representing time, metrics on time, and fuzzy measures. The conceptual scene description generated by the geometric layer is imported as logic facts into the inference layer. Here, these basic concepts are—conceptually and temporally—aggregated into more complex concepts describing not only the isolated state of a single agent, but also its relation to others and its development in time. These more complex concepts are organised into generic representations of *situations* (compare [30]). In the inference layer, so-called *Situation Graph Trees* (*SGTs*) represent the knowledge about which basic concepts at which step in the evolving scene should be aggregated into which situation concept. An example SGT (taken from [20]) describing the behaviour of vehicles at innercity intersections is depicted in Fig. 2.

A situation scheme generically combines a certain physical state of an agent—expressed in logic predicates compatible with the facts imported from the geometric

[1] This additional assumption of rigid bodies normally holds for cars. Trucks with trailers, in contrast, need not to satisfy this assumption. Other publications (see [31]) reported results on how to cope with such (slightly) more articulated bodies.

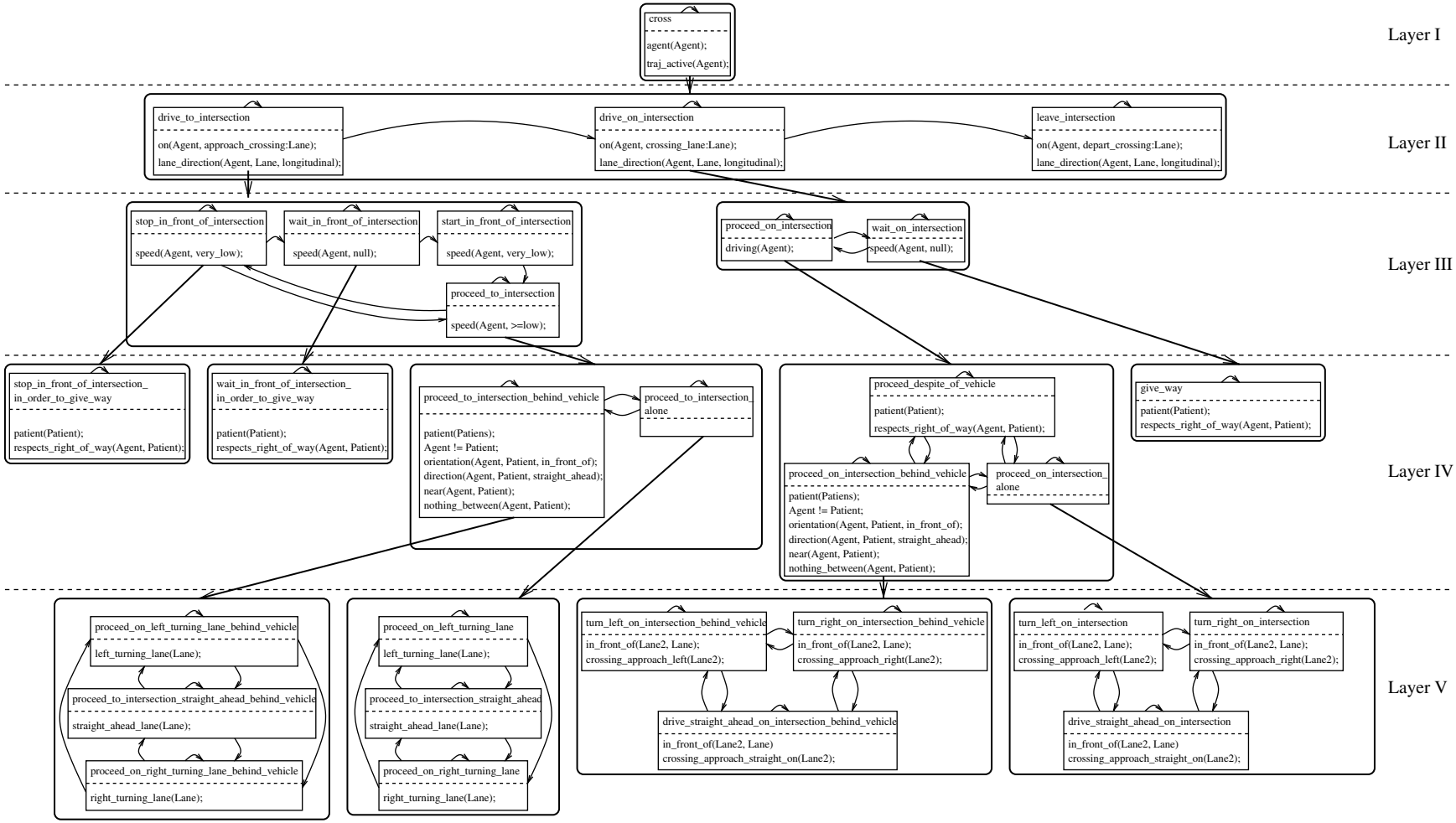


Fig. 2. Original situation graph tree from [20] for crossing innercity intersections. Further explanations in the text.

layer—with actions the agent or the observing system is expected to carry out in that state. In Fig. 2, situation schemes are depicted as rectangles comprising a situation scheme name above a dashed line and the state description expressed as logic facts below that dashed line (action descriptions have been omitted due to space limitations). Temporal successor relations between situation schemes are expressed in the form of *prediction edges* connecting one (the present) situation scheme with another (putatively next) scheme. These edges are shown as thin arrows in Fig. 2. Situation schemes together with prediction edges build *situation graphs*, which are always directed and may be cyclic. Such graphs are indicated as thick, rounded rectangles in Fig. 2. A situation scheme can also be connected to a complete situation graph by so-called *specialisation edges*, depicted as thick arrows in Fig. 2. These edges represent the knowledge that a particular situation scheme can be detailed temporally or conceptually by a complete sequence of other situation schemes. This sequence, in turn, is represented as one path in the graph to which the specialisation edge points. The specialisation edges lead to tree-like structures, the *Situation Graph Trees* (SGTs). The root graph of such an SGT represents the most general description of an agent behaviour (compare Fig. 2, 'Layer I'), while leaf graphs—i.e. graphs containing only situation schemes which are not connected to any other graph by a specialisation edge—describe an agent behaviour in the most detailed way accessible by this particular SGT (compare Fig. 2, e.g. 'Layer V'). As such, SGTs describe in which situation an agent presently is, what actions are associated to that situation, and what situations might follow in the future. Thus, SGTs represent *potential behaviours* of agents.

The situation analysis, i.e. the process of finding a valid situation for each recognised agent at each time point, searches for a valid path through a given SGT. This is effected by first searching for an instantiable situation scheme within the root graph of the SGT. If such a situation is found, probably existing specialising situation graphs of that situation are investigated for another, more detailed situation which is instantiable, too. In such a way, the most detailed situation scheme instantiable for the agent is considered to be the proper situation description for that agent and the present time step. In the next time step, only those situations are investigated, which are connected to the most recent situation scheme by a prediction edge. Again, if such a situation is found and could be instantiated, the algorithm tries to specialise that situation scheme *as deeply as possible* along specialisation edges. If at one point in time, no prediction edge leads from the most recent scheme to another probable successor scheme, or such an edge is present, but the instantiation of the probable successor scheme is not possible, then the algorithm is allowed to generalise again. This means, if the most recent situation is contained inside a situation graph which specialises another, more general scheme, then perhaps that more general scheme possesses possible successor situations which should be investigated, too. If such a generalisation fails, too, the complete traversal is considered to have failed and an appropriate error message is produced by the algorithm.

SGTs are specified for the inference layer of the overall computer vision system in the description language SIT + + developed, too, by Schäfer [37]. This representation is then precompiled into a logic program of FMTHL, as described in detail in [20,37]. The resulting FMTHL-program contains the SGT on one hand and combines it with the SGT-traversal described above. The execution of such a program results in a behavioural description of each recognised agent in terms of situations. In [20], several different image sequences recorded at innercity intersections have been characterised by high-level conceptual descriptions based on this approach. One of the examples described there is also depicted in Fig. 3.

4. Changes due to modified requirements

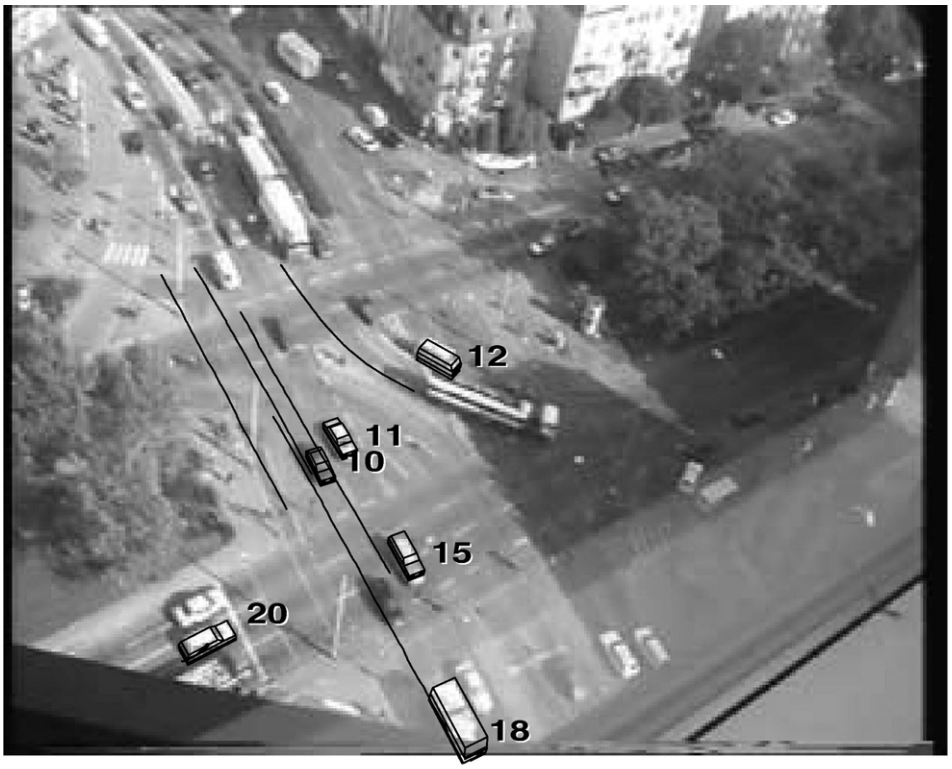
Despite the results obtained with the approach presented in [20], there still remained several limitations: due to the time prototypic implementation, situation schemes could only possess, e.g. a single specialising situation graph. This lead to SGTs which had to combine several aspects of detailing in one and the same subtree.[2] We substantially redesigned the overall vision system with the goal not only to overcome these more technical limitations, but also due to our experience accumulated during experiments with the conceptual description of videos in general and with SGTs in particular. This experience led to the desire to quickly create, inspect, and change SGTs. In addition, the challenge to generate a natural language—instead of (only) a conceptual—description of videos (see [17,19]) resulted in new requirements for the overall system. The purpose of this section is to motivate and present these changes.

Previously, SGTs were written in SIT + +, the description language developed by Schäfer [37], which is sufficient for small SGTs. However, more complex SGTs quickly raise the efforts necessary to maintain and extend their textual description in SIT + +. In addition, it appeared that SGTs as graphical structures would be easier to understand and modify in a graphical rather than textual form. These considerations culminated in the development and implementation of SGTEDITOR [1]. With this tool,[3] it is now possible to graphically create, inspect, and modify SGTs. An SGT built with SGTEDITOR can be saved in the SIT + + format, just as SIT + +-files containing SGTs can be loaded and processed using SGTEDITOR.

It turned out that the ability to graphically create SGTs—among other things—also affected the way SG were *designed*

[2] In [20], the detailing of situations has been treated with respect to the lane on which an agent drives and with respect to the presence or absence of an additional (leading) object (compare, too Fig. 2).

[3] SGTEDITOR has been made available publically under GNU Public License. It can be downloaded from http://cogvisys.iaks.uni-karlsruhe.de/Vid-Text



start	end	situation
105	106	cross(**object_20**)
107	157	proceed_to_intersection_alone(**object_20**,fobj_2)
158	164	stop_in_front_of_intersection_in_order_to_give_way(**object_20**,fobj_2,object_18)
165	171	wait_in_front_of_intersection_in_order_to_give_way(**object_20**,fobj_2,object_18)
172	178	start_in_front_of_intersection(**object_20**,fobj_2)
179	185	wait_in_front_of_intersection_in_order_to_give_way(**object_20**,fobj_2,object_18)
186	220	wait_in_front_of_intersection(**object_20**,fobj_2)
221	225	start_in_front_of_intersection(**object_20**,fobj_2)
226	297	wait_in_front_of_intersection(**object_20**,fobj_2)
298	299	start_in_front_of_intersection(**object_20**,fobj_2)
300	374	wait_in_front_of_intersection(**object_20**,fobj_2)
375	377	start_in_front_of_intersection(**object_20**,fobj_2)
378	478	wait_in_front_of_intersection(**object_20**,fobj_2)
479	482	start_in_front_of_intersection(**object_20**,fobj_2)
483	516	wait_in_front_of_intersection(**object_20**,fobj_2)
517	529	start_in_front_of_intersection(**object_20**,fobj_2)
530	616	wait_in_front_of_intersection(**object_20**,fobj_2)
105	106	cross(**object_11**)
107	138	proceed_to_intersection_straight_ahead_behind_vehicle(**object_11**,object_15,fobj_14)
139	404	drive_straight_ahead_behind_vehicle(**object_11**,fobj_78,object_15)
405	481	leave_intersection(**object_11**,fobj_79)
105	106	cross(**object_12**)
107	442	turn_left_on_intersection(**object_12**,fobj_81)
443	543	leave_intersection(**object_12**,fobj_65)
544	721	leave_intersection(**object_12**,fobj_63)
105	106	cross(**object_10**)
107	151	proceed_to_intersection_straight_ahead(**object_10**,fobj_13)
152	373	drive_straight_ahead(**object_10**,fobj_76)
374	434	leave_intersection(**object_10**,fobj_77)

Fig. 3. *Top*: one frame of the Nibelungen-Platz image sequence with overload tracking results for objects 10, 11, 12, 15, 18, and 20 at time point no. 300. *Bottom*: Sequence of situation nodes visited during tracking. *fobj_xx* is an individual constant referring to a particular lane (taken from [20]).

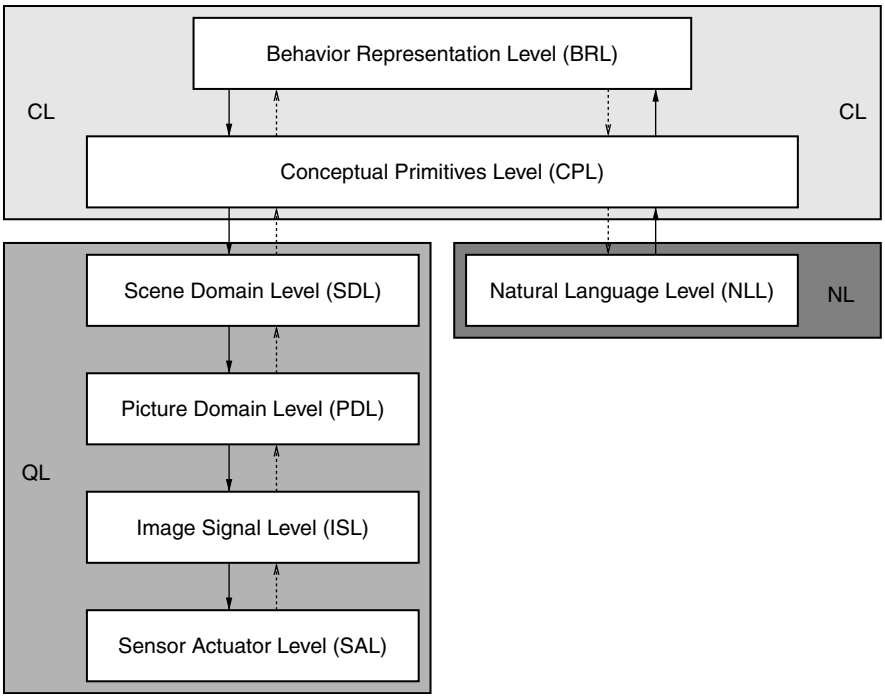


Fig. 4. New layer-architecture of a *cognitive* vision system (adapted from [3]).

henceforth. For example, in the SGT used in [20] (compare Fig. 2), the situation `proceed_to_intersection` had been connected to a graph containing the schemes `proceed_to_intersection_behind_vehicle` and `proceed_to_intersection_alone`. These two schemes were connected by prediction edges to each other, though none is the temporal successor of the other in the sense that this is the normal behaviour of vehicles at an intersection and, therefore, should be modelled in this manner. In fact, the meaning of prediction edges had been shifted thereby from a temporal successor relation towards a logical *XOR*-relation. By restricting the meaning of prediction edges to that of temporal successor relations in this behavioural sense, the graph mentioned above naturally broke into two different graphs both specialising `proceed_to_intersection`. Once the restriction of only one specialisation of a situation scheme had been removed, we realised that specialisation edges in the original SGTs have two different meanings: *terminologic specialisation* and *temporal decomposition* (see, too [2]).

Terminologic specialisations turned out to connect situation schemes to graphs containing only a single scheme: this conforms with the common understanding of specialisation as *further confinement*. If the parent scheme (from which the specialisation edge starts) can be instantiated, then the specialising scheme (contained in the specialising graph) can only be instantiated if the predicates contained in its state description can be verified in addition to those in the parent scheme. A terminologic specialisation thus adds constraints to the state description to be verified for an agent.

Temporal decompositions in turn connected situation schemes to graphs containing more than one scheme. These schemes were most often simply connected in the form of linear sequences. Only on a few occasions, graphs representing a temporal decomposition contained cycles due to prediction edges. This fact might be explained by the observation that in the domain of vehicle behaviour at intersections, repetitive behaviour of single agents occurs in only very few cases. Only sub-behaviours like *stop-and-go*, for example, can—and probably have to—be modelled in situation graphs containing cycles.

What, then, is the connection between SGTEDITOR and the observations concerning the structure of SGTs reported above? Not surprisingly, the desire to graphically edit SGTs, which requires algorithms to graphically layout those graphs, resulted in the goal to keep the SGTs structurally as simple as possible without losing their representational power. The simplifications most often applied to existing SGTs, however, coincided with the semantic restrictions presented in the preceding paragraphs, i.e. the separation of detailing situation schemes into separate graphs and the restriction of prediction edges to temporal successor relations between schemes.

The desire to generate natural language texts from the conceptual descriptions obtained, e.g. in [20], also caused modifications and extensions to the overall vision system as described there. First, an additional sub-system had to be introduced which transforms conceptual descriptions obtained by an SGT-traversal into natural language text. Secondly, the separation of the overall system into quantitative, conceptual, and natural language parts should be explicated in order to facilitate tracking down shortcomings or errors. The structure of the new overall system can be visualised by a layered configuration of transforma-

Table 1
Types of knowledge which are provided to the quantitative layers (QL), the conceptual layers (CL), and the natural language layers (NL) (adapted from [20])

Knowledge	Representation	Purpose	Layer
Camera model	Matrices, vectors	Inference from image features to scene descriptions	QL
Vehicle model	Polyhedral models	Update step in object tracking	QL
Motion model	Difference equations as approximations for the differential equations of object motion	Prediction step in object tracking	QL
Illumination model	Equations describing the orientation of the incoming sun light	Taking shadows in the update step of object tracking into account	QL
Lane model	Labeled polygons	Segment optical flow fields for initialisation	QL
	Concept terminology	Inference from estimated vehicle positions to the semantic of the corresponding lane	CL
3D Scene model	Polyhedral models	Consider occlusions and shadows cast by static scene components during vehicle tracking	QL/CL
Situation model	FMTHL formulation	Associate situations, infer intentions	CL
Language model	Grammars, lexical	Transform situation analysis results into natural anguage sentences	NL

tion processes as depicted in Fig. 4. Unfilled rectangles in this figure represent single representations of the information conveyed from image sequences to conceptual descriptions. Arrows between these rectangles depict transformation processes. The single representations are subsumed into more encompassing sub-systems. The lower-left sub-system (middle-grey in Fig. 4) represents system parts which only treat numerical data. It will be referred to as *quantitative layers* (QL) in the following. The upper part (light-grey) deals with symbolic, conceptual information and will be referred to henceforth as *conceptual layers* (CL). The lower-right system part (dark-grey) finally converts conceptual representations of information into natural language texts (*natural language layers*, NL).

Each transformation step within the vision system uses different types of background knowledge to different extents (compare Table 1). The quantitative layers (QL, comprising SAL, ISL, PDL, and SDL, see Fig. 4) use a camera model to transform image domain cues into scene domain cues. It utilises several polyhedric models for vehicles, the lane structure, and other static scene components. In addition, illumination models can be applied to estimate shadows cast by static scene components and vehicles in order to improve the tracking performance [33]. The conceptual layers (CL, comprising CPL and BRL, compare Fig. 4) of our system are completely based on the *Fuzzy Metric Temporal Horn Logic* (FMTHL) introduced by Schäfer [37]. The conceptual layers use background knowledge, too, though in a different representational form. Some of the knowledge has already been introduced into the quantitative layers, though in numerical form. Now, the conceptual representation of the same lane model and static scene objects enables the *conceptual layers* to relate tracking results to certain areas or locations in the scene (see Table 1). Moreover, terminological knowledge associates these results obtained in the quantitative layers with conceptual primitives (as described in [18]). These conceptual primitives comprise notions of orientation, speed, acceleration, vicinity, and so on. Based on these primitives, more complex concepts can be introduced by defining appropriate facts and rules in FMTHL.

The last sub-system depicted in Fig. 4 (NL) is concerned with the transformation of conceptual scene descriptions into natural language texts as reported in [17]. This transformation is based on the *Discourse Representation Theory* described in [28]. Again, a clear interface between conceptual and natural language layers is ensured by passing only the results of an SGT-traversal to the natural language layers. Additional knowledge concerning, e.g. the syntax of natural language texts is again explicitly modelled and only used in this latter sub-system.

The quantitative layers of the new system architecture are comparable to the geometric layer reported in [20]. Similarly, the conceptual layers of the new system correspond to the inference layer reported there (see Fig. 1). There exists one important difference, however, in the interface between the two system parts then and now: in the original geometric layer, not only numerical state descriptions of agents were derived, but also—due to historical reasons—representations for basic concepts concerning speed, orientation, etc. In the present system, only state descriptions are generated by the quantitative layers and are subsequently imported into the conceptual layers. The derivation of basic and complex concepts is completely managed in the conceptual layers based on logic rules and facts. This approach thus has a much cleaner interface between quantitative and qualitative system components. In addition to this advantage, the terminological knowledge has to be explicitly modelled in the form of logic rules, which are easier to examine and modify than procedures buried deep in some part of the quantitative vision system. Finally, terminological knowledge in the form of logic rules naturally fits into the overall system of SGT-traversal based on logic programs: now, basic and complex concepts are only computed if they are needed during the traversal.

5. Results

The results to be presented in the following can be separated into two categories: first, a new SGT will be discussed which also represents the behaviour of vehicles at intersections, but follows the design rules presented in Section 4. Secondly, SGT-traversal results obtained with this new SGT will be illustrated for one example image sequence also evaluated in [20] in order to facilitate comparisons.

5.1. A new SGT for intersections

Fig. 5 shows the SGT created to represent the behaviour of vehicles at an intersection following the design rules formulated in the previous section. This figure only serves as an overview. Details can be seen in Figs. 6–8. The crossing of an intersection[4] can be temporally decomposed into three situations: driving to the intersection, giving way at the intersection, and leaving the intersection (see Fig. 6). This had also been represented in the old SGT (compare Fig. 2), though situation scheme identifiers differed from those used in the SGT presented here. Fig. 7 shows the further detailing of the situation scheme `driving_to_intersection`. Notice that the complete sub-graph below this situation scheme contains only terminologic specialisations, easily identifiable as situation graphs containing only one situation scheme. Fig. 8 in turn contains the sub-graph detailing `sit_giving_way`. In addition to several terminological specialisations, this graph contains one temporal decomposition, namely the graph detailing `sit_driving_near_occupied_crossing_lanes`. The graph detailing this scheme contains three schemes, constituting two alternative (temporal) sequences of situations a vehicle could instantiate while driving near the actual crossing lane: it could be moving (`sit_moving_towards_crossing_lanes`) and then accelerating (`sit_accelerating_towards_crossing_lanes`), because no other vehicles have to be given way to in the moment the agent reaches the crossing lane. Alternatively, the vehicle could again be driving near the crossing lane, but would have to stop (`standing_while_giving_way_to`) in order to give way to a crossing vehicle. Then, it could be accelerating towards the crossing lane (`sit_accelerating_towards_crossing_lanes`).
All the figures of the new SGT presented here—and the SGT itself—have been created with SGTEDITOR. This tool offers functionalities to hide or re-show sub-graphs in order to generate customised visualisations of parts of an SGT, which also eases the editing of SGTs in total. For details on SGTEDITOR, see [1]. Some of the details depicted in

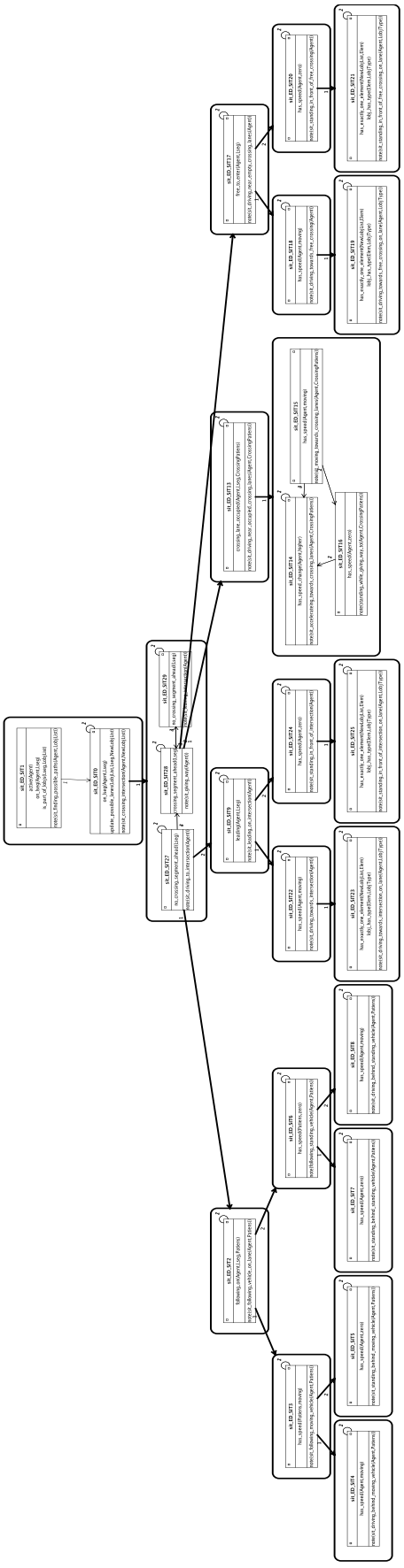


Fig. 5. Overview picture of new SGT describing the behaviour of vehicles at intersections. See Figs. 6-8 for detail views.

[4] See the situation scheme **sit_ED_SIT0** in Fig. 6; the argument **`sit_crossing_intersection`**—which will be output to the user as a side-effect of instantiating this note-predicate—conveys the intended semantics of this node to the user. In the sequel, we will just mention the intended semantics instead of the situation-scheme identifier.

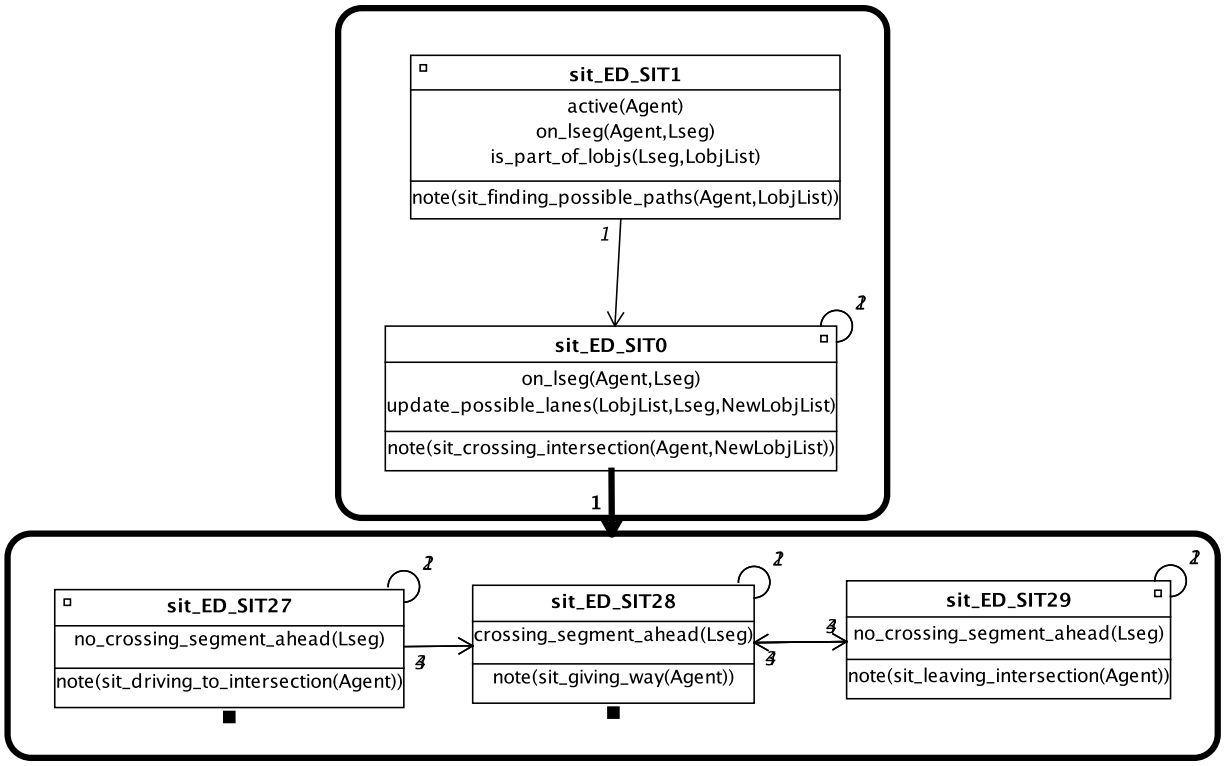


Fig. 6. Two top-most levels of new SGT. Rectangles show situation schemes with identifiers (e.g. **sit_ED_SIT1**), state predicates (e.g. **on_iseg(Agent, Lseg)**), and action predicates (e.g. **note(sit_finding_possible_paths(Agent, LobjList))**) separated by lines. While the state predicates of a situation scheme have to be satisfied during traversal to instantiate this scheme, the action predicates of a scheme are executed by the traversal algorithm whenever the situation scheme has been instantiated. Here, the **note**-predicates cause the traversal-algorithm to print out situation-dependent messages which then serve as input for the generation of natural language text. Thin arrows indicate prediction edges, while small circles in the upper right corner of situation schemes show prediction edges from and to a single scheme. Bold, rounded rectangles enclose situation (sub-)graphs. Bold arrows stand for specialisation edges. Small filled rectangles below situation schemes denote sub-graphs not yet shown in this figure.

Figs. 5–8 have not been explained here because they only concern SGT-traversal and not the design rules discussed here. For details on SGT-traversal, see [3,37].

5.2. Traversal results obtained with the new SGT

The new architecture of the overall computer vision system together with the tools for generation and maintenance of representations for behavioural knowledge in the form of SGTs would lose its justification if we lost the ability to obtain results comparable to those presented in [20]. Due to space limitations, we cannot show results on all image sequences examined there. We will, however, demonstrate the descriptive ability of the present system on one example image sequence, summarised in Fig. 9. As can be seen there, we evaluated the behaviour of the same agents as shown in Fig. 3. For **object_20**, we could improve the conceptual description due to the fact that conceptual speed descriptions are now computed within the conceptual layers of the system. In the previous approach, these concepts were derived by procedures contained in the geometric layer. Due to noise in the associated measurement process, this eventually led to an alternation of the concepts `start_in_front_of_intersection` and `wait_in_front_of_intersection` (compare Fig. 3). The behaviour is now more correctly described as the sequence of the two concepts `driving_towards_intersection_on_lane` and `standing_in_front_of_intersection_on_lane` (see Fig. 9). The results for **object_11** can be compared directly, though syntactically different concepts were used in the new SGT. The previous results described **object_12** as `turning_...` and `leaving_...` In the new results, the description also provides these two concepts (`driving_towards_free_crossing_on_lane` and `leaving_intersection`), but then switches back to `driving_towards_free_crossing_on_lane`, followed by the final `leaving_intersection`. This description is created due to the fact that **object_12** has to cross two different crossing lanes, as can be seen in the example frame shown in the upper part of Fig. 9. This structure of the actual intersection observed in the example image sequence had not been modelled in the previous approach, nor could it be described with the SGT employed there. The new results for **object_10** are again similar to those reported in [20] (compare Fig. 3). The transformation of such a conceptual representation into a natural language text has been treated already—see, e.g. [17].

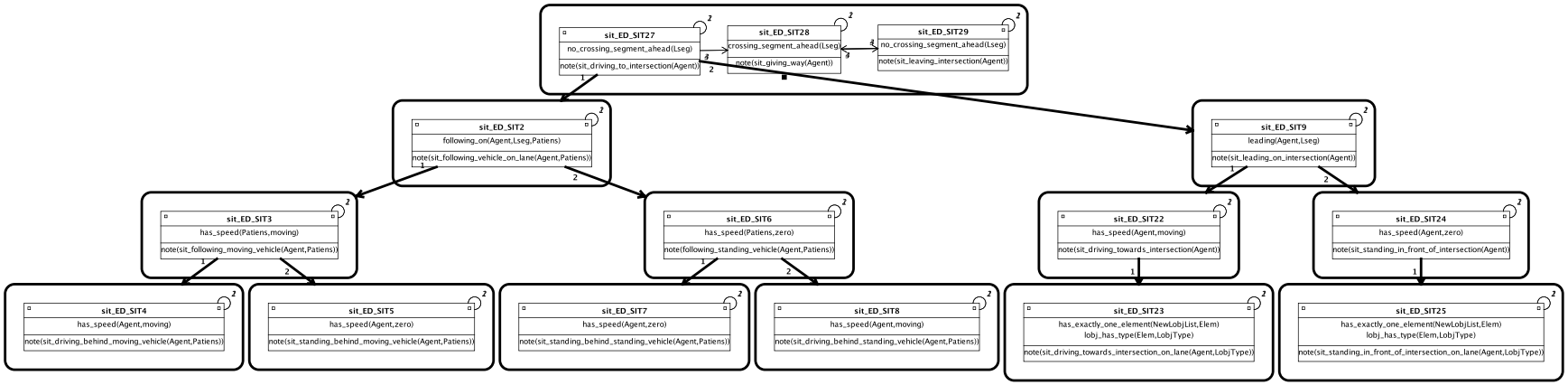


Fig. 7. Second level of new SGT illustrating specialisations of `sit_driving_to_intersection` (**sit_ED_SIT27** in Fig. 6).

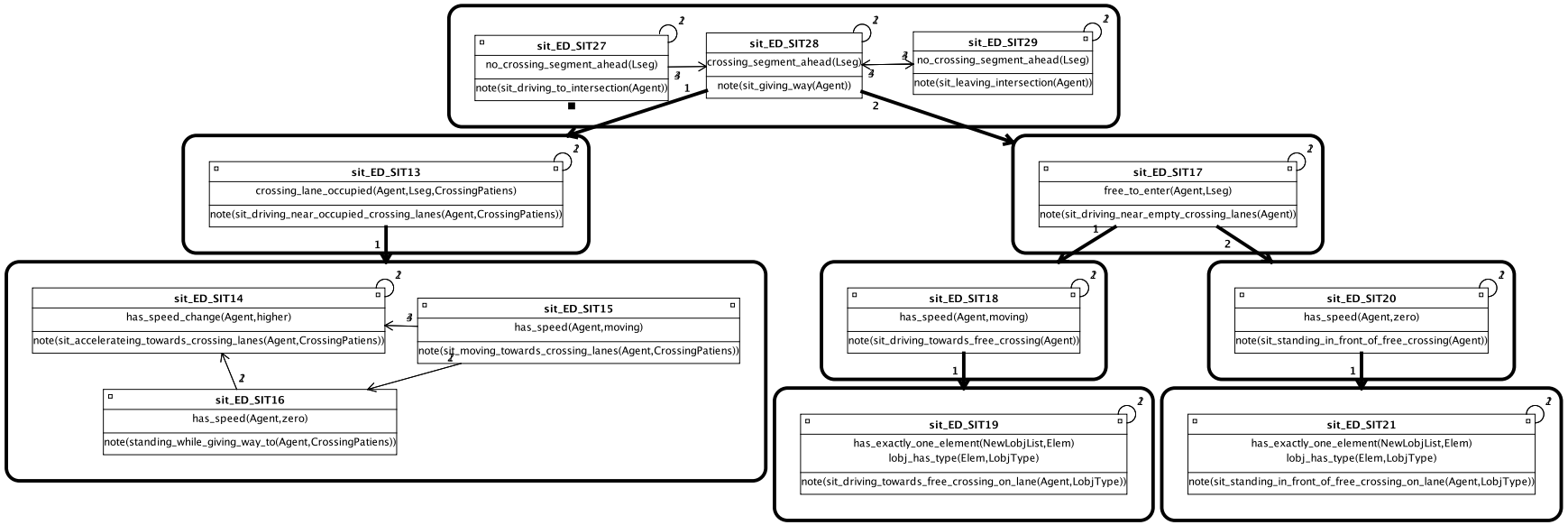
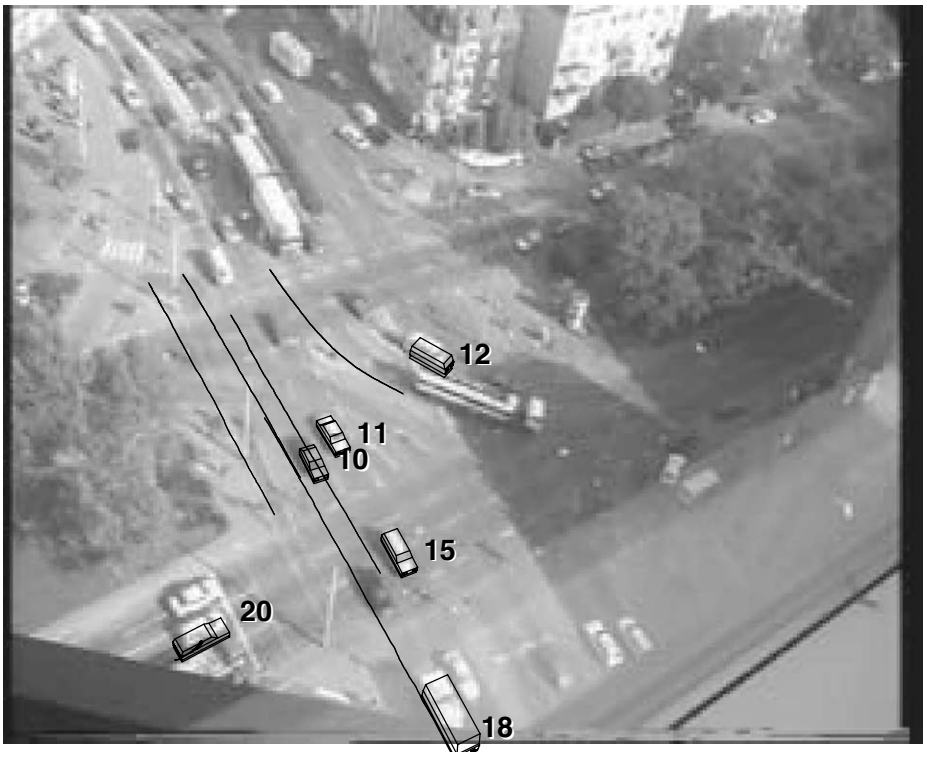


Fig. 8. Second level of new SGT illustrating specialisations of `sit_giving_way` (**sit_ED_SIT28** in Fig. 6).



start	end	situation
106	107	sit_Þnding_possible_paths(**object_20**,[lobj_8])
108	158	sit_driving_towards_intersection_on_lane(**object_20**,straight_ahead_lane)
159	658	sit_standing_in_front_of_intersection_on_lane(**object_20**,straight_ahead_lane)
106	107	sit_Þnding_possible_paths(**object_11**,[lobj_1])
108	108	sit_driving_behind_moving_vehicle(**object_11**,object_15)
109	358	sit_driving_towards_free_crossingon_lane(**object_11**,straigh_tahead_lane)
359	458	sit_leaving_intersection(**object_11**)
106	107	sit_Þnding_possible_paths(**object_12**,[lobj_7])
108	108	sit_driving_towards_intersection_on_lane(**object_12**,leftt_urning_lane)
109	208	sit_driving_towards_free_crossing_on_lane(**object_12**,left_turning_lane)
209	308	sit_leaving_intersection(**object_12**)
309	408	sit_driving_towards_free_crossing_on_lane(**object_12**,left_turning_lane)
409	658	sit_leaving_intersection(**object_12**)
106	107	sit_Þnding_possible_paths(**object_10**,[lobj_0])
108	108	sit_driving_towards_intersectionon_lane(**object_10,**straight_ahead_lane)
109	358	sit_driving_towards_free_crossing_on_lane(**object_10**,straight_ahead_lane)
359	408	sit_leaving_intersection(**object_10**)

Fig. 9. *Top*: Same image frame as in Fig. 3. *Bottom*: Sequence of situation nodes visited during traversal of new SGT. *lobj_x* is an individual constant referring to a particular lane.

6. Conclusion

Algorithmic approaches have become feasible which transform a video into a natural language description of developments in the recorded scene. Any such approach eventually has to map intermediate results extracted from a video into concepts which in turn can be translated algorithmically into natural language expressions. This transition requires links between signal and geometric representations of the depicted scene on the one hand and conceptual representations of the same developments on the other hand. Such links can be constructed either by supervised learning of suitable system-internal representations or by asking a system designer to engineer the required representations. Obviously, one can argue that engineered knowledge bases might reflect more the presuppositions

of the designer rather than the details of the reality in the depicted scene.

Given the numerous research problems encountered along the path to construct and evaluate such a system, it appears at least plausible during an exploratory phase to attempt to minimise unexpected difficulties by relying on engineered knowledge bases. This leaves the option open to eventually turn to learning approaches once the boundary conditions for their usage have become clearer. Research reported in this contribution reflects this line of argumentation. All modules required for an experimental system have been designed, implemented, tested, and tuned to the point where their interaction could be explored. This in turn enabled us to determine the weakest links and to attempt to remove them. The overall system gradually gained a robustness which opened the road to study details of the knowledge representation required to detect and describe vehicle behaviour at road intersections.

As a result of such studies, the originally conceived representation for vehicle behaviour has been reworked completely. The transition from the geometric to a conceptual representation has been separated from geometric tracking phases and moved entirely to the conceptual layers, even at the price that certain facts about the geometry of the depicted scene now have to be represented both at a geometrical and a conceptual level. This design decision yielded much clearer boundary conditions for the design of behavioural representations with the consequence that the advantage offered by appropriate design tools became more obvious. Already the considerations about how to conceive such design tools in detail forced us to scrutinise previous ideas about the representation of behaviours by Situation Graph Trees (SGTs). The admission of alternative specialisations and a clean separation between specialisation and (temporal) decomposition operations resulted in more satisfying SGTs. The exploitation of these redesigned SGTs led to originally unexpected improvements of instantiated conceptual representations and natural language text generated from them. An example has been presented which illustrates the ideas and the advances realised thereby. It is hoped that these results stimulate others to use the tools which have been made generally available in order to explore their suitability in different application domains. Initial attempts along this line in collaboration with other laboratories are encouraging.

Admittedly, progress along the path outlined here is slow and tedious. The reader may decide for himself, though, whether the difference to alternative approaches may be worth the efforts.

Acknowledgements

Partial support of this research by the European Union through the project 'Cognitive Vision Systems (CogViSys)' (**IST-2000-29404**) is gratefully acknowledged, similarly earlier partial support by the Deutsche Forschungsgemeinschaft. The authors acknowledge helpful comments on an earlier version of this contribution by anonymous reviewers.

References

[1] M. Arens, SGTEditor reference manual. Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik der Universität Karlsruhe (TH), April 2003, Available from: http://cogvisys.iaks.uni-karlsruhe.de/Vid-Text/sgt_editor/

[2] M. Arens, Repräsentation und Nutzung von Verhaltenswissen in der Bildfolgenauswertung. Dissertation, Fakultät für Informatik, Universität Karlsruhe (TH). Juli 2004: Dissertationen zur Künstlichen Intelligenz (DISKI) 287; Akademische Verlagsgesellschaft Aka GmbH, Berlin, 2005, (in German).

[3] M. Arens, H.-H. Nagel, Behavioral knowledge representation for the understanding and creation of video sequences, in: A. Günter, R. Kruse, B. Neumann (Eds.), Proceedings of the 26th German Conference on Artificial Intelligence (KI-2003), 15–18 September 2003, Hamburg, Germany; Lecture Notes in Artificial Intelligence, vol. 2821, Springer, Berlin, 2003, pp. 149–163.

[4] A. Blocher, J.R.J. Schirra, Optional deep case filling and focus control with mental images: ANTLIMA-KOREF, in: C.S. Mellish (Ed.), Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95), 20–25 August, Montréal, Canada 1995, pp. 417–423.

[5] A.F. Bobick, Video annotation: computers watching video, in: S.Z. Li, D.P. Mital, E.K. Teoh, H. Wang (Eds.), Proceedings of the Second Asian Conference on Computer Vision (ACCV'95), 5–8 December 1995, Singapore, Lecture Notes in Computer Science, vol. 1035, Springer, Berlin, 1996, pp. 23–31.

[6] A.F. Bobick, Y.A. Ivanov, Action recognition using probabilistic parsing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-98), 23–25 June 1998, Santa Barbara, CA, IEEE Press, New York, 1998, pp. 196–202.

[7] A.F. Bobick, S.S. Intille, J.W. Davis, F. Baird, L.W. Campbell, Y.A. Ivanov, C.S. Pinhanez, A. Schütte, A. Wilson, The kidsroom: a perceptually based interactive and immersive story environment, Presence Teleoperators and Virtual Environments 8 (4) (1999) 369–393.

[8] A.F. Bobick, S.S. Intille, J.W. Davis, F. Baird, C.S. Pinhanez, L.W. Campbell, Y.A. Ivanov, A. Schütte, A. Wilson, Perceptual user interfaces: the kidsroom, Communications of the ACM 43 (3) (2000) 60–61.

[9] M. Brand, N. Oliver, A. Pentland, Coupled hidden markov models for complex action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-97), 17–19 June 1997, San Juan, Puerto Rico, IEEE Press, New York, 1997, pp. 994–999.

[10] F. Brémond, M. Thonnat, Issues of representing context illustrated by video-surveillance applications, International Journal of Human–Computer Studies 48 (3) (1998) 375–391.

[11] H. Buxton, Learning and understanding dynamic scene activity: a review, Image and Vision Computing 21 (1) (2003) 125–136.

[12] H. Buxton, R.J. Howarth, Behavioural descriptions from image sequences, in: Proceedings of the Workshop on Integration of Natural and Vision Processing Language, August 1994, AAAI Press, New York, 1994, pp. 231–239.

[13] H. Buxton, S. Gong, Visual surveillance in a dynamic and uncertain world, Artificial Intelligence 78 (1–2) (1995) 431–459.

[14] J. Fernyhough, A.G. Cohn, D.C. Hogg, Constructing qualitative event models automatically from video input, Image and Vision Computing 18 (2) (2000) 81–104.

[15] A. Galata, N. Johnson, D.C. Hogg:, Learning variable length markov models of behaviour, Computer Vision and Image Understanding 81 (3) (2001) 113–398.

[16] A. Galata, A.G. Cohn, D.R. Magee, D.C. Hogg, Modelling interaction using learnt qualitative spatio-temporal relations and variable

length markov models, in: F. van Harmelen (Ed.), Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-2002), 21–26 July 2002, Lyon, France, IOS Press, Amsterdam, 2002, pp. 741–745.

[17] R. Gerber, Natürlichsprachliche Beschreibung von Straßenverkehrsszenen durch Bildfolgenauswertung. Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Karlsruhe, January 2000, (in German). Electronically Available from: http://www.ubka.uni-karlsruhe.de/cgi-bin/psview?document = 2000/informatik/8

[18] R. Gerber, H.-H. Nagel, Occurrence extraction from image sequences of road traffic scenes, in: L. van Gool, B. Schiele (Eds.), Proceedings of the Workshop on Cognitive Vision, 19–20 September 2002, ETH Zurich, Switzerland, pp. 1–8, Available from: http://www.vision.ethz.ch/cogvis02/finalpapers/gerber.pdf

[19] R. Gerber, H.-H. Nagel, H. Schreiber, Deriving textual descriptions of road traffic queues from video sequences, in: F. van Harmelen (Ed.), Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-2002), 21–26 July 2002, Lyon, France, IOS Press, Amsterdam, 2002, pp. 736–740.

[20] M. Haag, H. Nagel, Incremental recognition of traffic situations from video image sequences, Image and Vision Computing 18 (2) (2000) 137–153.

[21] S.M. Hazarika, A.G. Cohn, Abducing qualitative spatio-temporal histories from partial observations, in: D. Fensel, F. Giunchiglia, D.L. McGuinness, M.-A. Williams (Eds.), Proceedings of the eigth International Conference on Knowledge Representation and Reasoning, 22–25 April 2002, Toulouse, France, Morgan Kaufmann, San Mateo, CA, 2002, pp. 14–25.

[22] R.J. Howarth, Interpreting a dynamic and uncertain world: task-based control, Artificial Intelligence 100 (1–2) (1998) 5–85.

[23] R.J. Howarth, H. Buxton, Conceptual descriptions from monitoring and watching image sequences, Image and Vision Computing 18 (2) (2000) 105–135.

[24] S.S. Intille, A.F. Bobick, Visual recognition of multi-agent action using binary temporal relations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-99), 23–25 June 1999, Ft Collins, CO. IEEE Press, New York, 1999, pp. 1056–1062.

[25] S.S. Intille, A.F. Bobick, A framework for recognizing multi-agent action from visual evidence, in: Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99), 18–22 July 1999, Orlando, FL, MIT Press, Cambridge, MA, 1999, pp. 518–525.

[26] S.S. Intille, A.F. Bobick, Recognizing planned, multiperson action, Computer Vision and Image Understanding 81 (3) (2001) 414–445.

[27] N. Johnson, D.C. Hogg, The acquisition and use of interaction behaviour models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-98), 23–25 June 1998, Santa Barbara, CA, IEEE Press, New York, 1998, pp. 866–871.

[28] H. Kamp, U. Reyle, From Discourse to Logic, Kluwer, Dordrecht The Netherlands, 1993.

[29] N. Moënne-Loccoz, F. Brémond, M. Thonnat, Recurrent bayesian network for the recognition of human behaviors from video, in: J.L. Crowley, J.H. Piater, M. Vincze, L. Paletta (Eds.), Proceedings of the third International Conference on Computer Vision Systems (ICVS-2003), 1–3 April 2003, Graz, Austria, Lecture Notes in Computer Science, vol. 2626, Springer, Berlin, 2003, pp. 68–77.

[30] H.-H. Nagel, From images sequences towards conceptual descriptions, Image and Vision Computing 6 (2) (1988) 59–74.

[31] H.-H. Nagel, T. Schwarz, H. Leuck, M. Haag, Tracking turning trucks with trailers, in: S. Maybank, T. Tan (Eds.), Proceedings of the IEEE Workshop on Visual Surveillance, 2 January, Bombay, India 1998, pp. 65–72.

[32] N. Oliver, A. Pentland, Graphical models for driver behavior recognition in a smartcar, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV-2000), Detroit, MI, USA, IEEE Press, New York, 3–5 October 2000, pp. 7–12.

[33] A. Ottlik, H.-H. Nagel, On consistent discrimination between directed and diffuse outdoor illumination, in: Proceedings of the 25th DAGM-Symposium (DAGM-2003), 10–12 September 2003, Magdeburg, Germany, Lecture Notes in Computer Science, vol. 2781, Springer, Berlin, 2003, pp. 418–425.

[34] C.S. Pinhanez, A.F. Bobick, Approximate world models: incorporating qualitative and linguistic information into vision systems, in: Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96), 4–8 August 1996, Portland, OR, MIT Press, Cambridge, MA, 1996, pp. 1116–1123.

[35] B. Rosario, N. Oliver, A. Pentland, A synthetic agent system for bayesian modeling human interactions, in: Proceedings of the third Annual Conference on Autonomous Agents (AGENTS-99), 1–5 May 1999, Seattle, WA, ACM Press, New York , 1999, pp. 342–343.

[36] N. Rota, M. Thonnat, Activity recognition from video sequences using declarative models, in: W. Horn (Ed.), Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000), 20–25 August 2000, Berlin, Germany, IOS Press, Amsterdam, 2000, pp. 673–677.

[37] K.H. Schäfer, Unscharfe zeitlogische modellierung von situationen und handlungen in der bildfolgenauswertung und robotik. dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Karlsruhe, Juli 1996; Dissertationen zur Künstlichen Intelligenz (DISKI), vol. 135, infix-Verlag Sankt Augustin, 1996, (in German).

[38] J.R.J. Schirra, Bildbeschreibung als Verbindung von visuellem und sprachlichem Raum. Dissertation, Fakultät für Informatik der Universität des Saarlandes, Saarbrücken, April 1994, Dissertationen zur Künstlichen Intelligenz (DISK1) 71; infix-Verlag: Sankt Augustin, 1994, (in German).

[39] V.T. Vu, F. Brémond, M. Thonnat, Human behaviour visualisation and simulation for automatic video understanding, in: Proceedings of the 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG-2002), Plzen-Bory, Czech Republic, 2002, pp. 485–492.

[40] V.T. Vu, F. Brémond, M. Thonnat, Temporal constraints for video interpretation, in: T. Walsh (Ed.): Proceedings of the ECAI-2002 Workshop on Modelling and Solving Problems with Constraints, Lyon, France, 23 July 2002, Available from: http://www-users.cs.york.ac.uk/~tw/ecai02/

[41] V.T. Vu, F. Brémond, M. Thonnat, Automatic video interpretation: a recognition algorithm for temporal scenarios based on pre-compiled scenario models, in: J.L. Crowley, J.H. Piater, M. Vincze, L. Paletta (Eds.), Proceedings of the third International Conference on Computer Vision Systems (ICVS-2003), 1–3 April 2003, Graz, Austria, Lecture Notes in Computer Science, vol. 2626, Springer, Berlin, 2003, pp. 523–533.