Deriving Textual Descriptions of Road Traffic Queues from Video Sequences

Ralf Gerber and **Hans-Hellmut Nagel** and **Heiko Schreiber** [1]

Abstract. Based on *geometric* results obtained by an algorithmic video (sequence) evaluation, a *generic conceptual* representation will be instantiated into a representation of the *specific* temporal developments within the recorded scene. Such an instantiated conceptual representation will in turn provide the input for a text generation subsystem.

This contribution outlines a system implementation which relies on a fuzzy metric-temporal (Horn) logic to realize the system-internal generic representation. The instantiation step is realized as the search for a suitable interpretation of the corresponding set of logic formulae where the geometric results obtained by image sequence evaluation provide the set of individuals. The practicality of this approach is demonstrated by abstraction processes which aggregate suitably selected vehicles from recorded road traffic scenes into a *vehicle queue.*

1 INTRODUCTION

A surveillance system based on computer vision usually characterises individual – or at most small subsequences of – image frames by their assignment to one from a small set of admitted categories, for example `{no_alert, alert, congestion, imminent_danger, molestation_of_others, vandalism}`. This corresponds to a pattern recognition process relying on a few discriminating 'features' derived from *local* spatiotemporal intensity or color variations in the recorded video data. Such an approach can be perfectly justified if a system has to be developed under severe price and runtime constraints. In case finely differentiated phenomena have to be distinguished, however, their detection and correct categorisation becomes more involved. The same observation can be made if the evaluation of a very local spatiotemporal environment no longer suffices. A single-step category assignment then has to be replaced by an increasingly multi-step analysis process.

From a slightly more detached point of view, the aforementioned categorisation can be looked at as a link between the video input signal and a conceptual description of temporal developments in the recorded scene. Alternatives and generalisations then come to (some) mind(s) almost effortlessly. The extraction of a 'feature vector' from images will be generalised to a *computer vision subsystem* and the categories to a *natural language textual description.* The process which mediates between the video signal and the natural language terms representing different categories will evolve into the manipulation of a full-fledged *system-internal conceptual representation* of the depicted scene and of temporal developments captured by the video recording. Different research avenues open themselves in such a context. It thus is no surprise that these become increasingly populated in recent times, not the least due to the fact that the continuously improving computing capacity of workstations facilitates the necessary experiments:

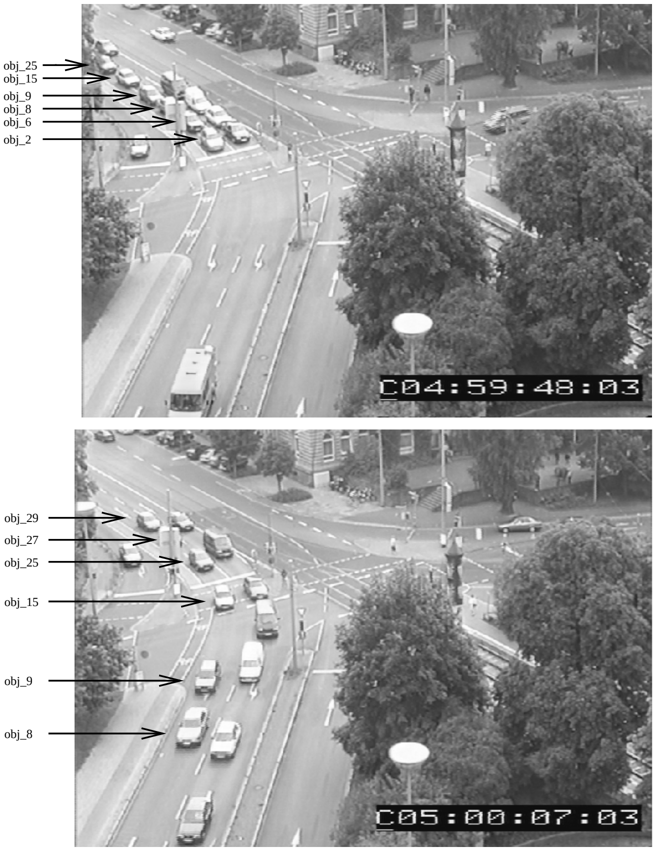

Figure 1. Two representative image frames (top: #400; bottom: #875) from a road traffic intersection sequence which comprises a total of 2320 frames. The frame rate of this sequence is equivalent to a sampling rate of 50 frames per second, i. e. the entire image sequence covers slightly more than 45 seconds of road traffic.

- Significant efforts are required to create robust and fast computer vision (sub-)systems. These have to extract evidence for those aspects which are most important for a subsequent conceptual treatment of temporal developments in the recorded scene.
- Once relevant 'cues' can be extracted with sufficient reliability,

[1] Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik, Universität Karlsruhe (TH), Postfach 6980, 76128 Karlsruhe, Germany

it appears tempting for some researchers to devise a – preferably unsupervised – *learning approach* which promises to extract the desired 'concepts' automatically from a learning sample of videos.

- An alternative approach attempts to *design* a system-internal representation.
- In both cases the question arises how the spatio-temporal developments within the recorded scene should be *represented* at a conceptual level. The gamut of representations to be considered at least reaches from a variety of ad-hoc approaches via grammars and various kinds of automata to different variants of formal logic.
- This problem is aggravated by the necessity to take uncertainties with quite different backgrounds into account: *stochastic properties* of the input signal, *artefacts* of the computer vision subsystem due to simplifications and numerical approximations (not to talk about undetected implementation errors; after all, computer vision subsystems with desired capabilities are non-trivial), and in particular the *vagueness* of natural language concepts involved.
- The necessity to treat time and uncertainties in an appropriate manner raises questions regarding the competence and performance of algorithmic approaches explored in this context. The more ambitious a system approach, the more relevant such questions will become.
- In general, people will not be satisfied by debugging a system-internal conceptual representation of temporal variations within a scene recorded by a video camera. After an initial relief to be able to inspect comments consisting of a single word concept together with the actual or a representative image frame, the desire begins to grow to have a more 'natural language like' textual description of relevant developments in the recorded scene. This then adds *computational linguistics* to the set of disciplines which have to be mastered in order to realize the original aspirations.

Getting all of this to work together requires an effort which necessitates to think about system architecture and system engineering questions. It is much easier, however, to judge the adequacy of a textual description – i. e. to assess the *system* performance – than to decide in isolation whether a particular approach or intermediate result is 'good enough'.

2 RELEVANT LITERATURE

'Motion Understanding' constitutes a prominent part of any approach to describe temporal variations in a scene. A vivid recollection of early attempts in this area can be found in [20, Section 3.1].

The goal to describe the 'story conveyed by an image sequence' in the form of a natural language text has been discussed already for almost a quarter of a century (see, e. g., [14]). Due to the difficulties to extract a reliable geometric description from real-world video sequences, however, most early publications relied at least on interactively extracted or corrected image evaluation results as input for a text generation approach – see, e.g., [1]. As an alternative, completely synthesized results provided a starting point to test a text generation approach. Vehicle trajectories extracted from *real* innercity road traffic videos were associated with a large number of motion verbs by finite state automata based on fuzzy predicates [12].

More recent publications like [2] or [8] studied the transformation of video sequences into abstract conceptual descriptions against an AI background. Of particular interest in our context are AI-oriented investigations concerning the aggregation of several moving agents into groups [10]. A more recently published, broad discussion of related literature provides the background for a system approach tested for a simplified assembly scenario [3].

Already in the mid-eighties, Neumann has studied the use of formal logic for a system-internal representation of temporal developments in a discourse domain [15]. Gradually, approaches based on formal logic as a means for system-internal representation and manipulation of conceptual descriptions began to be applied to geometric results extracted by computer vision subsystems from real video sequences, see [7]. The increasing importance of formal logic for a conceptual analysis of image sequences is also reflected by contributions to a workshop in connection with IJCAI-1999 [18] and in a special issue of Image and Vision Computing on 'Conceptualizing Images' (edited by Buxton and Mukerjee), in particular [9] and [6]. A similar line of reasoning has been emphasized in [17] discussing additional references.

Alternatives for algorithmic text generation have been discussed recently in [13]: these approaches do not start from image evaluation results and are heavily biased towards real-time performance (not relevant in our case). Bayesian Nets have been used to classify the behavior of a single agent in parking lot video sequences as well as situations comprising two agents in order to insert appropriate references into sentence templates [16]. Transformations of geometric results obtained from extended real-world image sequences into a natural language text have been reported (e. g., [4]), albeit without treating the aggregation of several vehicles into a *queue*.

3 SYSTEM STRUCTURE

Due to space limitations, the extraction of geometric tracking results from image sequences by a computer vision subsystem will not be treated here. Details can be found, for example, in recent publications cited in the preceding section and in [5]. We proceed on the basis that a kind of 'Geometric Scene Description (GSD)' [15] is available.

3.1 Representation formalism

In our case, such a GSD is provided by 'facts' obtained by interpretation of a set of formulae given in a Fuzzy Metric-Temporal Horn Logic (FMTHL) [19]. The interpretation process for these elementary or conceptual 'primitives' takes place while the results of the computer vision subsystem are *imported* into the subsystem for the conceptual representation of the behavior of single vehicles and – the topic of this contribution – of groups of vehicles, in particular vehicle queues. An FMTHL inference engine exploits these 'imported' facts in order to search for an interpretation of FMTHL-formulae which constitute a generic conceptual representation of the behavior of certain kinds of vehicle groups. With other words, an instantiated GSD coded as FMTHL facts provides the basis to infer higher abstractions by an interpretation of corresponding schemes coded, too, as a set of FMTHL formulae.

All FMTHL rules follow the usual conventions in that logical variables begin with capital letters, whereas identifiers for logical constant, function, and predicate symbols start with lower case letters. The character pair ':-' denotes the (re-)implication operator, a conjunction (disjunction) between two predicates is denoted by a comma (semicolon), respectively. The *time* argument of functions and predicates as well as the fuzzy '*degree-of-validity*' are *handled automatically* by the FMTHL inference system. The operator 'always' indicates the premise that the subsequent expression – which is enclosed in parentheses – is valid with a fuzzy degree-of-validity equal to 1 for a temporal interval extending from minus infinity to plus infinity. Results of the computer vision subsystem are imported into the conceptual representation subsystem by a (hybrid, i. e. *import*-related) predicate `trajectory(Agent,X,Y,T,V,W)` which is assumed to have a degree-of-validity equal to 1 for each time point associated with the time interval corresponding to a frame number, in our case 20 msec. The variable `Agent` denotes a particular vehicle and will be set to an identifier assigned automatically to a vehicle during its detection and initialisation phase. `X` and `Y` denote the road plane coordinates of the vehicle reference point, `T` the vehicle orientation relative to the `X`-coordinate axis of the world coordinate system in the road plane, `V` the vehicle's speed, and `W` its steering angle[2].

3.2 Linking geometric and conceptual representations of an intersection

We first need to represent all *relevant* lanes of the scene within the field of view of the recording camera in a manner accessible to the FMTHL inference engine. This conceptual representation of the lane structure is derived from the geometrical model prepared for the model-based computer vision subsystem. In addition to the geometry, this intersection representation indicates the hierarchy within the lane structure and other relations between its different elements.

An intersection is conceived to comprise an *access* area, a *crossing* area proper, and an *exit* area. Each of these constitutes a lane *element*. Lane elements are concatenated to form a lane – see, e. g., Figure 2. The tuple `S` of lane elements `E` forming a lane is internally represented as a list which is manipulated using rules according to the FMTHL format:

```
always (lane(LaneElement,Lane) :-
   access_area(LaneElement) ,
   merge_into(LaneElement2,LaneElement) ,
   crossing_area(LaneElement2) ,
   merge_into(LaneElement3,LaneElement2) ,
   exit_area(LaneElement3) ,
   aggregate_lane_elements(LaneElement,LaneElement2,
                               LaneElement3,Lane)).
always (aggregate_lane_elements(LE1,LE2,LE3,Lane) :-
   tuple_add(LE1,[],Lane1) ,
   tuple_add(LE2,Lane1,Lane2) ,
   tuple_add(LE3,Lane2,Lane)).

always member(E,[E|_]).
always (member(E,[_|S]) :- member(E,S)).
always (tuple_add(E,S,S) :- member(E,S)).
always (tuple_add(E,S,[E|S]) :- not member(E,S)).
```

3.3 Terminology

As a preparatory step for text generation from geometric tracking results, we must provide the natural language concepts to be used.

3.3.1 *Selection and aggregation of vehicle groups*

Out of the set of all vehicles which have been detected and tracked by the computer vision subsystem and for which results have been 'imported' into the conceptual representation subsystem, we select a subset denoted as vehicle '*group*'. This selection is already implemented as an FMTHL inference based on some aggregation criterion, for example the temporal interval within which vehicles enter or leave the field of view of the recording camera, their type or size, or the requirement that the members of the selected subset can all be found along certain lanes in the scene.

For each 'imported' (see above, Section 3.1) vehicle, it is recursively tested whether it is located on one of the lane elements constituting the lane under consideration. The predicate `get_group` collects a tuple of vehicles *on* the specified lane.

```
always (cluster_feature(Agent,Lane) :-
   trajectory(Agent,X,Y,T,V,W) , on(Agent,Lane)).
always (on(Agent,[E|S]) :- on(Agent,E) ; on(Agent,S)).
always (aggregate_vehicle_group(Lane, VehicleSet,
                                             Amount) :-
   get_group(X, cluster_feature(X, Lane),
                                   VehicleSet, Amount)).
always (get_group(Vars, Constraint, List, Amount) :-
   call({ findall List Vars Constraint }) ,
   card(List, Amount)).

always card([], 0.0).
always (card([H|T), N1) :- card(T,N) , N1 is N + 1.0).
```

3.3.2 *Differentiating vehicle groups into subcategories*

Vehicle groups on the selected lane are assigned to one of the subcategories `single_vehicle`, `vehicle_pair`, or `vehicle_queue` according to how many vehicles such a group comprises:

```
always (group_type(Amount,Type) :-
                                get_group(Amount,Type)).

always (get_group(Amount,free_lane) :- Amount = 0).
always (get_group(Amount,single_vehicle) :- Amount = 1).
always (get_group(Amount,vehicle_pair) :- Amount = 2).
always (get_group(Amount,vehicle_queue) :- Amount > 2).
```

So far, it did not appear necessary to require an additional criterion in order to select, e. g., a vehicle *queue*. It is perfectly conceivable, however, to introduce a supplementary condition such that, e. g., the vehicles within a queue are not too far apart.

3.4 Describing the *internal state* of a queue

Once a queue has been defined, it can be attributed with properties relating to its *internal state*. This characterises the way a queue has been formed by individual vehicles, for example according to the following rule for the determination of the last vehicle in a queue (which forms the head of the *list* comprising all vehicles in the queue):

```
always (last_vehicle_of_queue([H|T],vehicle_queue) :-
   note(be(H,the_last_vehicle_of_the_queue))).
```

If this rule is applied, a sideeffect of the `note`-predicate will output a *fact* – for use by the subsequent text generation subsystem, see Section 3.5 – indicating which vehicle is the last one of the queue in question (NB the distinction between the concept of a queue and a list as its system-internal representation). The *head* of a queue will be the last vehicle in this list, i. e. its tail element:

```
always head([H|T],H).

always (head_of_the_queue(VehicleSet, vehicle_queue) :-
   card(List, Num) , Num = 1 , head(VehicleSet,Head) ,
   note(be(Head,the_head_of_the_queue))).

always (head_of_the_queue([H|T],vehicle_queue) :-
   head_of_the_queue(T,vehicle_queue)).
```

Additional predicates determine how many vehicles have joined the queue or which vehicles have left the queue, thereby providing supplementary information about the internal state of this queue, for example

```
always (new_vehicle_joined(VehicleSet,
                VehicleSetBefore, Type) :-
      additional_vehicle(VehicleSet, VehicleSetBefore,
                                            Vehicle) ,
      not V = empty , note(enter(V,Type))).

always (additional_vehicle([], VSB, V) :-
      fill(empty,V)).
always fill(V,V).
always (additional_vehicle([H|T],VSB,H) :-
      not member(H,VSB)).
always (additional_vehicle([H|T],VSB,V) :-
      additional_vehicle(T,VSB,V)).
```

[2] Some of the results presented in the sequel have been obtained with an older version of the vehicle detection and tracking – i. e. the computer vision – subsystem which originally used the rotational velocity about an axis normal to both the road plane and the vehicle ground plane.

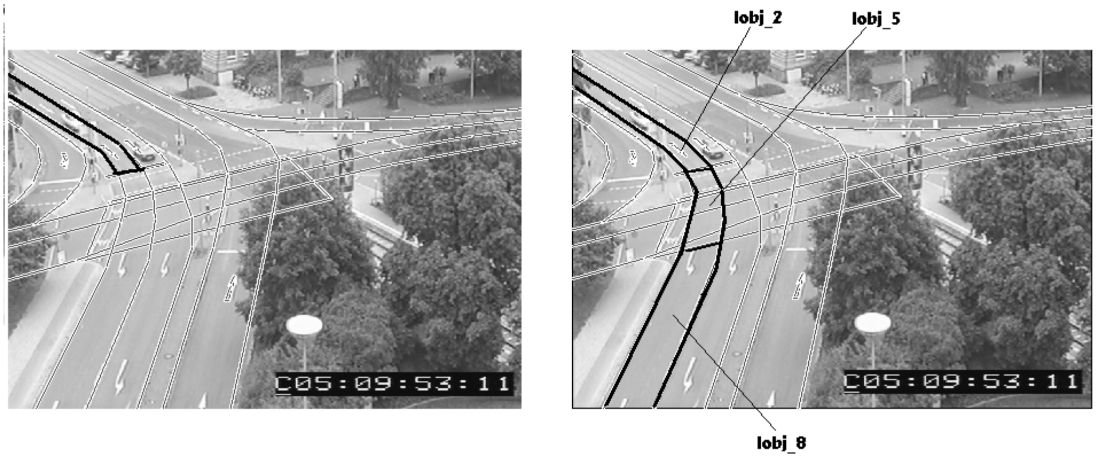

Figure 2. The left panel shows an image frame from the sequence illustrated in Figure 1, overlaid by the lane model for this intersection. One lane segment of the 'incoming' lane entering the field of view from the top left corner is marked by heavy boundary lines. The right panel exhibits the same image, but this time the 'intersection crossing' segment and the 'outgoing' segment of this same lane are marked by heavy boundary lines, too. In addition, the lane identifiers for the three segments of this lane are given.

3.5 Text generation

note predicates used in the rules mentioned above generate time-dependent *facts* – for more abstract concepts than the elementary, imported ones – of the form

interval of validity **!** *verbphrase(subject, object)*

as a side effect. These facts are conceived as a (more abstract) conceptual description of developments in the scene. The descriptions in this form are treated as a text in a *formal* language which is transformed into a *Discourse Representation Structure (DRS)* according to [11]. It offers a logic-based linguistically oriented representation of the temporal development within the recorded scene. The transformation of a DRS into a natural language text – i. e. the 'inversion' of the computational process which generates a DRS starting from a natural language text – constitutes a problem of computational linguistics which has not yet been solved *in general*. In order to test our approach within the entire systems context, we created a simple tool to generate a text from the DRS resulting from the precedingly described steps in analogy to [4].

4 RESULTS

The approach discussed in the preceding sections has been tested using amongst others the image sequence illustrated in Figure 1. The computer vision subsystem assigned automatically a unique object identifier to each vehicle which had been initialized for tracking. Figure 1 shows identifiers assigned to those vehicles which queue up on the leftmost through-lane, i. e. the one marked in Figure 2 by heavy boundary lines. The conceptual representation instantiated for these vehicles is given in the form of a list of time-dependent predicates (see Section 3.5). The two numbers separated by a colon at the start of each line indicate the *time* (i. e. frame number) *interval* where the predicate following after the exclamation mark has been instantiated. The treatment of motion verbs like 'enter' corresponds to that for a single agent [4].

```
   1 :   80 ! be_free(fobj_2).
  81 :   81 ! enter(obj_2, the_lane).
  81 :   81 ! drive_on(obj_2, fobj_2).
  81 :  219 ! be(obj_2, the_last_vehicle_of_the_queue).
  81 : 1654 ! be(obj_2, the_head_of_the_queue).
 220 :  220 ! enter(obj_6, the_lane).
 220 :  220 ! form(the_vehicles, vehicle_pair).
 220 :  349 ! be(obj_6, the_last_vehicle_of_the_queue).
 350 :  350 ! enter(obj_8, the_lane).
 350 :  350 ! form(the_vehicles, queue).
 350 :  479 ! be(obj_8, the_last_vehicle_of_the_queue).
 480 :  480 ! enter(obj_9, the_lane).
 480 :  560 ! be(obj_9, the_last_vehicle_of_the_queue).
 561 :  561 ! enter(obj_12, the_lane).
 561 :  575 ! be(obj_12, the_last_vehicle_of_the_queue).
 576 :  576 ! leave(obj_12, the_queue).
 576 :  709 ! be(obj_9, the_last_vehicle_of_the_queue).
 710 :  710 ! enter(obj_15, the_lane).
 710 : 1449 ! be(obj_15, the_last_vehicle_of_the_queue).
1361 : 1361 ! leave(obj_8, the_queue).
1450 : 1450 ! enter(obj_25, the_lane).
1450 : 1569 ! be(obj_25, the_last_vehicle_of_the_queue).
1570 : 1570 ! enter(obj_27,the_lane).
1570 : 1749 ! be(obj_27, the_last_vehicle_of_the_queue).
1655 : 1655 ! leave(obj_2, the_queue).
1655 : 1725 ! be(obj_6, the_head_of_the_queue).
1726 : 1726 ! leave(obj_6, the_queue).
1726 : 1912 ! be(obj_9, the_head_of_the_queue).
1750 : 1750 ! enter(obj_29, the_lane).
1750 : 2266 ! be(obj_29, the_last_vehicle_of_the_queue).
1913 : 1913 ! leave(obj_9, the_queue).
1913 : 2012 ! be(obj_15, the_head_of_the_queue).
2013 : 2013 ! leave(obj_15, the_queue).
2013 : 2099 ! be(obj_25, the_head_of_the_queue).
2100 : 2100 ! leave(obj_25, the_lane).
2100 : 2100 ! form(the_remaining_vehicles, vehicle_pair).
2100 : 2196 ! be(obj_27, the_head_of_the_queue).
2197 : 2197 ! leave(obj_27, the_lane).
2197 : 2266 ! be(obj_29, the_head_of_the_queue).
2267 : 2320 ! be_free(fobj_2).
```

Based on this system-internal conceptual representation in the form of a conjunction of instantiated metric-temporal predicates, the following text has been derived, see Figure 3. Similar results have been obtained for other lanes at this intersection and for vehicle groups recorded at other intersections.

5 CONCLUSIONS

The entire system has been designed deliberately on the basis of reliable methodological approaches: 3D-model-based tracking of vehicles, a fuzzy metric-temporal extension of first order predicate logic as a formalism for conceptual representations, and Discourse Representation Theory to establish a link between the system-internal conceptual representation and the natural language text to be generated. We tried to avoid placing one heuristic on top of another, even if this still precludes real-time experiments. The import of results obtained

“**Obj_2** entered the lane. Later **obj_6** entered the lane. The vehicles formed a pair.
Later **obj_8** entered the lane. In the meantime the vehicles formed a queue. **Obj_8** was the last vehicle of the queue. **Obj_2** was the head of the queue.
In the meantime **obj_9** entered the lane. It was the last vehicle of the queue.
In the meantime **obj_12** entered the lane. It was the last vehicle of the queue.
It left the queue. In the meantime **obj_9** was the last vehicle of the queue.
In the meantime **obj_15** entered the lane. It was the last vehicle of the queue.
In the meantime **obj_8** left the queue.
In the meantime **obj_25** entered the lane. It was the last vehicle of the queue.
In the meantime **obj_27** entered the lane. It was the last vehicle of the queue.
In the meantime **obj_2** left the queue.
In the meantime **obj_6** was the head of the queue. It left the queue.
In the meantime **obj_9** was the head of the queue.
In the meantime **obj_29** entered the lane. It was the last vehicle of the queue.
In the meantime **obj_9** left the queue.
In the meantime **obj_15** was the head of the queue. It left the queue.
In the meantime **obj_25** was the head of the queue. The remaining vehicles formed a pair. **Obj_25** left the lane.
Later **obj_27** left the lane. In the meantime **obj_29** remained as single vehicle.”

Figure 3. Output text generated from the internal conceptual representation in Section 4 for the vehicle queues illustrated in Figure 1.

by a *3D model-based* computer vision subsystem, e. g., obviates the need for the inference process to compensate for special cases due to working in the 2D picture domain. We thus attempt to facilitate a *systematic* search for the currently weakest link in the entire processing chain.

Obviously, the final text output leaves much to be improved regarding style. This observation allows, however, to illustrate some of the lessons offered by a systems approach. The monotonous reference to `object_nn` could be replaced by adjectives (indicating, e. g., color) *provided the computer vision system is able to determine the corresponding vehicle properties robustly*. Even more involved abstractions will be introduced, for example merging and splitting of vehicle queues. The inclusion of representations for additional behavior of vehicles and other forms of vehicle aggregations should be possible without modifications of the methodological approach. Experiments in these directions will include runs with different video input sequences in order to study the robustness of the approach reported here. The interesting point will be at which stage of such explorations it will become advisable or necessary to modify or extend the approach. We are aware, too, that we still use some ‘sharp’ rather than fuzzy *spatial* predicates, for example `on`.

Experience has shown, however, that *seemingly simple* improvements may have consequences all along the processing chain and thus need considerable efforts until they operate reliably. The possibility to investigate such questions in the context of a basically operational system is likely to provide feedback for more theoretical studies of representational formalisms and properties of related inference mechanisms.

ACKNOWLEDGEMENTS

Partial support of this research by the European Union (Project `CogViSys`, IST-2000-29404) is gratefully acknowledged.

REFERENCES

[1] E. Andre, G. Herzog, and Th. Rist: On the Simultaneous Interpretation of Real World Image Sequences and Their Natural Language Description: the System SOCCER. In Y. Kodratoff (Ed.), Proc. ECAI-88, 1-5 August 1988, Munich, Germany, pp. 449–454.

[2] H. Buxton and S. Gong: Visual Surveillance in a Dynamic and Uncertain World. *Artificial Intelligence* **78** (1995) 431–459.

[3] A. Chella, M. Frixione, and S. Gaglio: Understanding dynamic scenes. Artificial Intelligence **123**:1–2 (2000) 89-132.

[4] R. Gerber and H.-H. Nagel: (Mis-?)Using DRT for Generation of Natural Language Text from Image Sequences. In *Proc. ECCV’98*, 2-6 June 1998, Freiburg/Germany; H. Burkhardt and B. Neumann (Eds.), LNCS 1407 (Vol. II), pp. 255–270.

[5] M. Haag and H.-H. Nagel: Combination of Edge Element and Optical Flow Estimates for 3D-Model-Based Vehicle Tracking in Traffic Image Sequences. International Journal of Computer Vision **35**:3 (1999) 295-319.

[6] M. Haag and H.-H. Nagel: Incremental Recognition of Traffic Situations from Video Sequences. Image and Vision Computing **18**:2 (2000) 137-153.

[7] M. Haag, W. Theilmann, K. Schäfer, and H.-H. Nagel: Integration of Image Sequence Evaluation and Fuzzy Metric Temporal Logic Programming. In Proc. 21st Annual German Conference on Artificial Intelligence, Freiburg/Germany, 9–12 September 1997; G. Brewka, Ch. Habel, and B. Nebel (Eds.), LNAI 1303, pp. 301–312.

[8] R.J. Howarth: Interpreting a Dynamic and Uncertain World: Taks-Based Control. *Artificial Intelligence* **100** (1998) 5–85.

[9] R.J. Howarth and H. Buxton: Conceptual Descriptions from Monitoring and Watching Image Sequences. Image and Vision Computing **18**:2 (2000) 105-135.

[10] St. Intille and A. Bobick: Visual Recognition of Multi-Agent Action Using Binary Temporal Relations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR’99)*, 23–25 June 1999, Fort Collins, Colorado, Vol. 1, pp. 56–62.

[11] H. Kamp and U. Reyle: *From Discourse to Logic.* Kluwer Academic Publishers Dordrecht Boston London 1993.

[12] H. Kollnig, H.-H. Nagel, and M. Otte: Association of Motion Verbs with Vehicle Movements Extracted from Dense Optical Flow Fields. In Proc. ECCV ’94, 2-6 May 1994, Stockholm, Sweden. J.O. Eklundh (Ed.), LNCS 801, pp. 338-347.

[13] S.W. McRoy, S. Channarukui, and S.S. Ali: Creating Natural Language Output for Real-Time Applications. *intelligence* **12**:2 (2001) 21-34.

[14] H.-H. Nagel: From Image Sequences towards Conceptual Descriptions. *Image and Vision Computing* **6**:2 (1988) 59–74.

[15] B. Neumann: Natural Language Description of Time-Varying Scenes. In D. Waltz (Ed.), Semantic Structures - Advances in Natural Language Processing, Lawrence Erlbaum Assoc., Hillsdale/NJ and London/UK 1989, pp. 167–206.

[16] P. Remagnino, T. Tan, and K. Baker: Agent Oriented Annotation in Model Based Visual Surveillance. Proc. Sixth ICCV, 4-7 January 1998, Bombay, India, pp. 857-862.

[17] N.A. Rota and M. Thonnat: Activity Recognition from Video Sequences Using Declarative Models. In Proc. ECAI-2000, 20-25 August 2000, Berlin, Germany, pp. 673-677.

[18] G. Sagerer and S. Wachsmuth (Eds.): Integration of Speech and Image Understanding. Proc. IEEE Workshop, 21 September 1999, Corfu, Greece; IEEE Computer Society, Los Alamitos, CA 2000.

[19] K. Schäfer: *Unscharfe zeitlogische Modellierung von Situationen und Handlungen in Bildfolgenauswertung und Robotik.* Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Juli 1996. Erschienen in ‘Dissertationen zur Künstlichen Intelligenz (DISKI)’ **135**, infix-Verlag Sankt Augustin 1996 (in German).

[20] J.K. Tsotsos: Motion Understanding: Task-Directed Attention and Representations that Link Perception with Action. *International Journal of Computer Vision* **45**:3 (2001) 265-280.