Association of Motion Verbs with Vehicle Movements Extracted from Dense Optical Flow Fields

H. Kollnig[1], H.-H. Nagel[1,2], and M. Otte[1]

[1] Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik der Universität Karlsruhe (TH), Postfach 6980, D-76128 Karlsruhe, Germany
[2] Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB), Fraunhoferstr. 1, D-76131 Karlsruhe, Germany

Abstract. This contribution addresses the problem of detection and tracking of moving vehicles in image sequences from traffic scenes recorded by a stationary camera. By replacing the low level vision system component for the estimation of displacement vectors by an optical flow estimation module we are able to detect all moving vehicles in our test image sequence. By replacing the edge detector and by doubling the sampling rate we improve the model-based object tracking system significantly compared to an earlier system. The trajectories of vehicles are characterized by motion verbs and verb phrases. Results from various experiments with real world traffic scenes are presented.

1 Introduction

The quality of trajectories which are now available as an output of the system reported by [Koller *et al.* 93] gives us the opportunity to associate motion verbs with trajectory segments which are extracted from image sequences. Therefore, object movements are described not only geometrically but also conceptually.

So far, only few approaches towards the extraction of conceptual descriptions from image signals exist. A survey of literature can be found in [Nagel 88]. The NAOS system, for instance, creates a retrospective natural language description of object movements in a traffic scene [Neumann & Novak 86], but it has so far only been tested with synthetic image data. [Mohnhaupt & Neumann 90] use natural language utterances for a top-down control in traffic scene analysis. The SOCCER system simultaneously generates running reports for short sections from soccer games [André *et al.* 88], which serve as a basis for recognition of intentions and interactions of multiple agents [Retz-Schmidt 91]. [Nagel 91] proposes transition diagrams to represent admissible sequences of actions used in a system for visual road vehicle guidance and shows how more complex actions can be hierarchically formalized by means of approaches used in formal language theory and how sequences of actions can be visualized. Transition diagrams are also presented by [Herzog 92] as a means for an incremental generation of motion descriptions. He shows how motion descriptions can be constructed automatically from interval-based event representations using temporal constraint propagation techniques.

Recently, [Birnbaum *et al.* 93] report on the BUSTER system which explains why stacked block structures are not moving. However, their system is restricted to simple static scenes and 2-D image analysis. In order to develop a traffic surveillance system by means of image sequence analysis, [Toal & Buxton 92] used spatio-temporal reasoning to analyze occlusion behavior. In their approach, temporarily occluded vehicles are correctly relabeled after re-emerging rather than being treated as completely independent vehicles.

Lecture Notes in Computer Science, Vol. 801
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94

Our system with 67 motion verbs links directly to the evaluation of real world image sequences and extracts conceptual descriptions for vehicle movements in a greater variety. In contrast to the cited papers [Neumann & Novak 86; Mohnhaupt & Neumann 90; André *et al.* 88; Herzog 92; Retz-Schmidt 91] we do not use synthetic data to extract conceptual descriptions. The problems which arise by the analysis of real and noisy data are not yet covered in literature. It is one of our main intentions to extract the conceptual descriptions from trajectory data obtained by object tracking in real image sequences.

The system reported by [Koller *et al.* 93] works as follows: Starting from a token-based estimation of a displacement vector field, hypotheses for object image candidates are created. By means of an off-line calibration, these vehicle hypotheses can be backprojected into the 3-D world which results in pose estimates to initialize a Kalman-Filter. By projecting an hypothesized 3-D polyhedral vehicle model into the image plane, 2-D model edge segments are obtained which are matched to straight-line edge segments, so called data segments, extracted from the image. This feeds into a state MAP-update step. Kalman-Filter prediction is performed by using a motion model.

Compared with the system built by [Koller *et al.* 93] we substituted both low-level image analysis modules. First, the blob feature based component for the estimation of displacement vectors (see [Koller *et al.* 91]) was replaced by an optical flow estimation module (see [Otte & Nagel 94]). The optical flow field is denser so that a redesigned clustering algorithm enables us now to obtain significantly better initial pose estimates for object candidates. This facilitates to tighten the thresholds and Kalman-Filter parameters which in turn stabilizes the tracking process. Second, the straight line segment detection process used by [Koller *et al.* 93] has been supplanted by the edge detector reported by [Otte & Nagel 92a + 92b], which provides more data segments based on image structures which improves the matching process. This in turn contributes to an enhanced a-posteriori state estimation in the Kalman-Filter update step. Moreover, by interpolating the interlaced half-frames we doubled the sampling rate from 25 Hz to 50 Hz. As a consequence of all these improvements, all vehicles in our test image sequence can be tracked with the same Kalman-Filter parameter set.

In order to associate motion verbs and verb phrases with the notably better trajectory segments, a complete rework of previously published methods [Koller *et al.* 91; Heinze *et al.* 91] was necessary. The set of German verbs was translated into English wherever translation was possible. Fuzzy sets instead of the former threshold decision approach are used to associate trajectory attributes and verbs in order to cope with the inherent vagueness of natural language descriptions. As a consequence, the automata for incremental occurrence recognition had to be redesigned. Moreover, the conceptual descriptions are visualized in the image which enables us to more thoroughly inspect the system output.

This paper is organized as follows: A brief overview regarding related work on object recognition is presented in Section 2. The improvements of our detection and tracking system are described in Sections 3 and 4. The subsequent Sections 5 – 7 deal with the extraction of conceptual descriptions from image signals. The results of our experiments are illustrated in Section 8.

2 Object segmentation and pose estimation approaches

A review of relevant literature can be found in [Koller *et al.* 93]. Here we confine ourselves to recent publications. [Zhang *et al.* 93] propose a view-independent relational model (VIRM) for 3-D object recognition. The VIRM of an object is

represented as a hypergraph with attached weights to represent the covisibility of model features and with associated procedural constraints to represent view independent relationships between model features, e. g. parallelism or relative size. Given a CAD-wireframe model, their system constructs off-line a view-independent relational model, which can be applied for pose estimation without the need for information about the position and orientation of the camera (due to the VIRM).

[Tan *et al.* 92] propose a non-statistical linear algorithm for object pose estimation. They have no motion model and no prediction. The correspondences between data and model segments are established interactively. In [Tan *et al.* 93] the matching is performed automatically by histogram voting based on a generalized Hough transform. This pose estimation approach is used to extend their traffic vision system to multiple cameras and to track articulated objects, such as a lorry and a trailer. First results are reported by [Worrall *et al.* 93].

In contrast to interpretation-tree matching approaches, where the resulting computational costs can be reduced, for instance, by using a best-first search [Lowe 92], [Du *et al.* 93] establish the 3-D grouping of line segments by monotonically improving compliance with a viewpoint consistency constraint. By means of an experimental study they illustrate that their approach is more robust than Lowe's in the presence of errors of data segments.

[Liu & Huang 93] propose a vehicle centered motion model by representing a 3-D motion as a rotation around an axis through the vehicle center followed by a translation. By adding several constraints on the rotation and translation, they obtain different motion types. However, their 3-D motion estimation approach has so far only been tested on five image frames containing one vehicle.

In comparison with the above described approaches, our scenes are more complex. By exploiting the information from optical flow, we obtain a good initial guess and therefore avoid trying all possible angles for the positions of the model. Although our trajectory data are not computed in real-time, they are more densely sampled – 50 Hz – compared with 0.48 Hz [Tan *et al.* 92] and 5 Hz [Tan *et al.* 93; Worrall *et al.* 93]. Moreover, we do not restrict ourselves to a single aspect of image sequence analysis but present a system that covers all analysis steps from the gray value data up to conceptual descriptions.

3 Exploiting the information of a segmented optical flow field to initialize a model-based tracking system

The displacement vectors in the approach reported by [Koller *et al.* 91] were obtained by matching blobs generated by the monotonicity operator [Kories & Zimmermann 86] in two consecutive frames. In lots of experiments in which we exercised the system described by [Koller *et al.* 93], it turned out that the blob-based motion segmentation step used in their approach did not provide a very exact initial guess for each object nor detected it every object.

Reporting problems by using a similar displacement vector estimation module, [Gong & Buxton 93] improve the segmentation and 2-D tracking of moving objects by incorporating more contextual knowledge about the scene even at the earliest stages of visual processing. In contrast, we are able to simplify the cluster analysis by improving the low-level image analysis module.

We replaced this module by a more time consuming optical flow vector estimation module related to the approach of [Campani & Verri 92], which has been extended to include partial derivatives with respect to time by [Otte & Nagel

94]. This optical flow field estimation enables us to compute better initial pose estimates and thus more appropriate object hypotheses.

The optical flow field restricted to vectors exceeding a minimum magnitude, which, moreover, survived a singular value threshold, is juxtaposed to the displacement vector field used by [Koller *et al.* 91; Koller *et al.* 93] in Figure 1. Details of the clustering analysis originally developed by [Sung 88] and subsequently improved significantly by [Koller 92] can be found in Appendix A. The resulting optical flow field is significantly denser than the displacement vector field and overlaps each moving vehicle to such an extent that the subsequent cluster analysis step is significantly simplified: neighboring vectors with approximately the same magnitude and orientation are grouped into object image can-

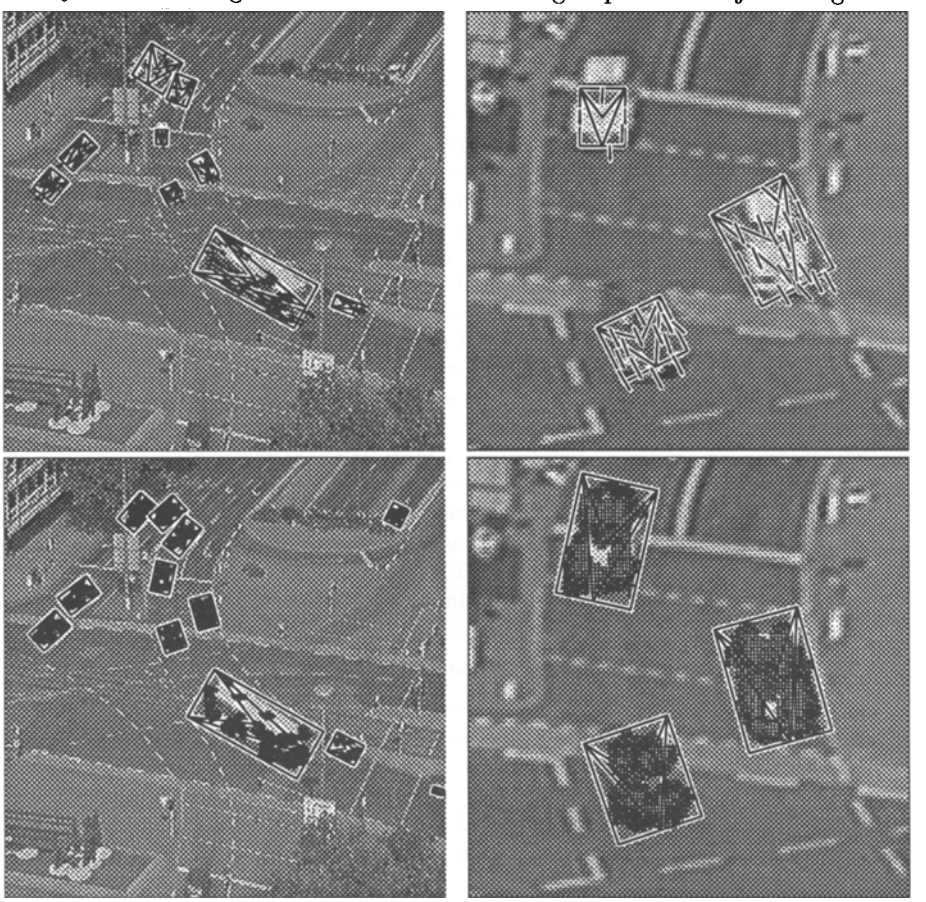

Fig. 1. Improvements in the initialization step: The first row shows the results of the displacement vector field estimation module used by [Koller *et al.* 91; Koller *et al.* 93] and the clustered vectors; each cluster is marked by a circumscribed rectangle. A section from the lower right part of the upper left quadrant in the first row is given on the right hand side of the first row. The displacement vectors are the results of tracking blob features along four consecutive frames. The second row shows the output of the clustering step applied to an optical flow vector field related to the approach of [Campani & Verri 92], extended and implemented by [Otte & Nagel 94]. The smaller optical flow vectors show the gray value displacements in one half-frame.

didates. Moreover, the detection rate increases significantly, too. In contrast to the approach of [Koller *et al.* 93], the information from the optical flow estimation is not just exploited to obtain initial values for position and orientation of a vehicle, but also to estimate the magnitude of the velocity v for each object. Therefore, our initialization is more homogeneous than that of [Koller *et al.* 93] whose displacement vector field could only be used to separate moving regions from the static image background. Their bootstrap phase was performed using the first two or three frames in order to estimate initial magnitudes of the translational and angular velocities v and ω, respectively. Since our initial estimates are more reliable, we have been able to tighten the tolerances which in turn resulted in a more efficient exclusion of outliers.

4 Computing data segments

[Koller *et al.* 93] extracted line segments fitted to thresholded edge elements which in turn are detected as local maxima of the gray-value gradient magnitude in gradient direction. In low contrast image regions, thresholding the gradient magnitude may suppress not only noise, but also edge elements which are part of a significant image structure and may thus result in the fragmentation or total loss of edge segments. In contrast to the traditional pixel oriented gradient magnitude thresholding, [Otte & Nagel 92a + 92b] proposed to chain edge elements to edge element chains and vertices without any thresholding. The evaluation of chain properties such as average gradient magnitude, length of chains and second moments of gradient direction change rates allows to either reject edge element chains as noisy or to accept them as a structure underlying the original image. Edge element chains include much more global information as compared to the information about a single edge element. The extraction of line segments, therefore, can be improved. For this reason, we replaced the line extraction process by the novel approach of [Otte & Nagel 92b].

Furthermore, instead of selecting uncertainties of data segments interactively (as e.g. [Tan *et al.* 93; Deriche & Faugeras 90]), we estimate them from the image data and, therefore, we are able to reduce the set of free parameters. Using the midpoint representation of line segments as described in [Deriche & Faugeras 90], we calculate the smaller eigenvalue in an eigenvector line fitting process to a set of edge elements to estimate the uncertainties perpendicular to a line segment.

5 Trajectory attributes based on fuzzy sets

In the following, fuzzy sets are used to abstract from quantitative details in geometrical descriptions obtained by automatic image sequence analysis.

For every object its world coordinates $(x(t_k), y(t_k))$, its speed $v(t_k)$, orientation $\theta(t_k)$ and its angular velocity $\omega(t_k)$ are extracted from an image sequence, sampled with 50 Hz. These trajectory data are characterized by attributes modeled by fuzzy sets. For example, the attribute *A_Speed* that characterizes the speed of the agent is modeled by the fuzzy membership functions shown in Figure 2, where the speed limit of 50 km/h on German downtown roads is taken into account. Other attributes are described in [Nagel & Kollnig 94]. The changing, increasing, decreasing, staying equal, or becoming unequal of the attribute values, below defined as monotonicity conditions, are depicted by fuzzy membership functions, too.

6 Discourse world and definition of occurrences

Human beings describe important occurrences by verbs. However, the exact meaning of a verb often depends on the subjective impression of the speaker.

Table 1. Definitions of agent reference occurrences. A dash '—' denotes the irrelevance of the attribute in the occurrence definition.

occurrences	agent reference attributes A_Speed		
	Pre_C	Mon_C	Post_C
be standing	zero	—	zero
drive off	zero	increasing	≥small
accelerate	≥small	increasing	≥small
drive slowly	small	—	small
drive at regular speed	normal	—	normal
run fast	fast	—	fast
run very fast	very fast	—	very fast
drive at constant speed	≥small	staying equal	≥small
brake	≥small	decreasing	≥small
stop	≥small	decreasing	zero

To avoid ambiguity we describe occurrences detected in an image sequence not just by motion verbs but also by verb phrases. Scanning a German dictionary with 150,000 entries yielded about 9,200 verb entries. Using a set of criteria – for instance, our occurrences should be elementary, i.e. not composed of other occurrences – all those verbs are selected from this set of 9,200 verbs which describe vehicle motions for downtown roads and road intersections. After the removal of synonyms, our system retains 67 verbs, listed in Appendix B.

Each occurrence is defined by three predicates, a *precondition (Pre_C)* that determines the attribute constellation necessary for the beginning of the occurrence, a *monotonicity condition (Mon_C)* that indicates the direction and amount of change during the validity of the occurrence, and a *postcondition (Post_C)*, defining the end of the occurrence. We divided the occurrences with respect to their reference into four classes:

- **Agent Reference**: the occurrence refers only to the agent (i.e. 'to brake'),
- **Location Reference**: in addition to the agent, the occurrence refers to a location (i.e. 'to arrive at a location '),
- **Road Reference**: in addition to the agent, the occurrence refers to the road or lane (i.e. 'to leave a driving lane'),
- **Object Reference**: in addition to the agent, the occurrence refers to another object (i.e. 'to follow a car').

By combining several attributes, we obtain an occurrence definition scheme. For example, the agent reference occurrences are tabulated in Tab. 1.

7 Automata for incremental occurrence recognition

To be able to extend our system to react to evaluated image sequence data in real-time, we prefer an incremental scene analysis instead of an retrospective or a-posteriori analysis where the motion descriptions are extracted on completed trajectories. Therefore, at each half-frametime during the evaluation of an image sequence, the attribute values are determined. For each occurrence, the values of Pre_C, Mon_C, and $Post\_C$ are determined. Multiple entries are combined

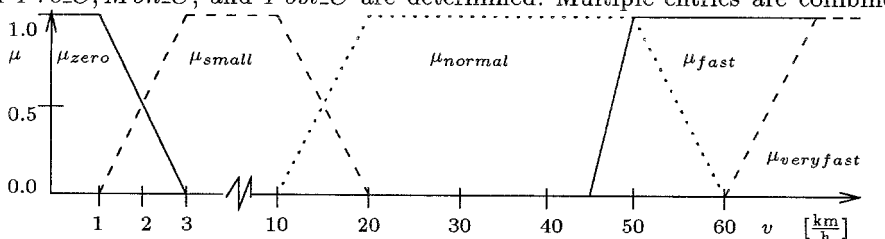

Fig. 2. Fuzzy membership functions of the attribute values *zero, small, normal, fast* and *very fast* for *A_Speed* as a function of the estimated vehicle speed.

by choosing the minimum of the attribute values, irrelevant entries are set to 1. The automata for incremental occurrence recognition are divided into four classes according to the aspect of the occurrences: details about the extraction of description of mutative, perpetuative, resultative, and inchoative occurrences can be found in [Kollnig & Nagel 93]. Each one of our occurrence descriptions contains a start frame number, an end frame number, and a degree λ of estimated validity ($0 < \lambda \leq 1$).

8 Experiments and results

As an experiment we used an image sequence of about 50 frames of a multi-lane street intersection in Karlsruhe. The size of the images of the moving vehicles varies from 25×30 to 30×35 (apart from the bus: 110×110) pixels in a frame. The smallest car images are even smaller than the 20×40 pixels in [Koller *et al.* 93].

The state vector $\boldsymbol{x}_k$ at time t_k used by our Kalman-Filter tracking module (for details see [Koller *et al.* 93]) is a five-dimensional vector consisting of the position (p_{x_k}, p_{y_k}) and orientation ϕ_k of the model as well the magnitudes v_k and ω_k of the translational and angular velocities. Due to our more precise initialization, we were able to decrease — compared with [Koller *et al.* 93] — the entries in the start covariance matrix by a factor 100. We used the initial values $\sigma_{p_{x_0}} = \sigma_{p_{y_0}} = 0.05$ m, $\sigma_{\phi_0} = 0.01$ rad, $\sigma_{v_0} = 0.032$ m/frame, $\sigma_{\omega_0} = 0.032$ rad/frame (apart from the 3 times larger bus: $\sigma_{p_{x_0}} = \sigma_{p_{y_0}} = 0.16$ m, $\sigma_{\phi_0} = 0.032$ rad). We use a process noise of $\sigma_v = 10^{-3}$ (m/frame) and $\sigma_\omega = 10^{-4}$ (rad/frame). The threshold d_τ for the computed Mahalanobis distance used for establishing correspondences between model and data segments could be doubled — compared with [Koller *et al.* 93] — due to the more robust initialization and better data segments. The values of our estimated errors perpendicular to the data segments are often only one half of the value which was interactively chosen by [Koller *et al.* 93]. We set $d_\tau = 10$. To track the bus – despite the fact that it is partially occluded by a street-lamp post – we were forced to use $d_\tau = 12$. Hereby, we compensate for the fact that wheels and doors of the bus are not yet modeled and, therefore, only comparatively long line segments are expected according to our very simple box model of the bus.

The computed trajectories for each moving car are given in Fig. 3. There still remain some problems. Obj. #1 can be correctly tracked only 40 half-frames due to its partial occlusion by a road sign. Obj. #3 cannot be tracked because it emerges from a tunnel underneath the intersection on a not yet calibrated lane. Obj. #11 cannot be tracked because its image is not correctly covered by the initially detected moving region due to the street-lamp post; obj. #12 has almost left the field of view before the tracking could stabilize. Fig. 4 shows an enlarged section of the image shown in Fig. 3. The results of the conceptual description extraction module for this image area are given in Fig. 5. To be able to verify the system output, the agent trajectory is colored depending on the extracted occurrences. The degree of estimated validity for each occurrence associated with a trajectory segment is visualized by the thickness of the trajectory. If more than one description is valid at one half-frametime, the translated trajectories are projected with different colors. Fig. 6 shows the visualization of the contents of Fig. 5.

9 Acknowledgments

We gratefully acknowledge stimulating discussions with Konstantinos Daniilidis, Dieter Koller, and Karl Schäfer. Our thanks go to Markus Maier, Harald Damm, and Martin Tonko for their support in this research.

A Clustering analysis of optical flow vectors

[Koller *et al.* 93] used the following cluster analysis, originally developed by [Sung 88]: First of all, each optical flow vector is considered as a cluster seed. Around such a seed, all vectors are clustered for which the conjunction of the following three predicates is satisfied.

- Two vectors satisfy the **neighboring** predicate, if the Euclidean distance of their footpoints does not exceed a threshold t_n.
- Two vectors satisfy the **parallel** predicate, if the absolute difference of their orientations does not exceed a threshold t_p.
- Two vectors satisfy the **same_length** predicate, if their relative length difference with respect to the first vector does not exceed a threshold t_l.

Second, we create maximal disjoint clusters by merging recursively all clusters with a non-empty intersection. Third, the footpoints of all vectors in each cluster are enclosed by a rectangle. Again, it is tested if one rectangle contains a vector of another rectangle. In this case, these clusters are merged and the enclosing rectangle for the merged clusters is determined. In our experiments we used $t_n = 1, t_p = 15°, t_l = 15\%$. Due to the now available dense optical flow fields, the threshold t_n could be set to one, compared to $t_n = 15$ in [Koller *et al.* 93] (see first row in Fig. 1).

B List of occurrences in the discourse world

Agent Reference: be standing, drive off, accelerate, drive slowly, drive at regular speed, run fast, run very fast, drive at constant speed, brake, stop, run straight ahead, turn right, turn left, revolve around a vertical axis, slide, skid, reverse, run forward (18 occurrences).
Location Reference: drive to location, pass location, arrive at location, depart from location, run over location, stop at location, park at location, leave location, leave location behind (9 occurrences).
Road Reference: leave driving lane, enter lane, turn, change section, drive on lane, cross a lane (6 occurrences).
Object Reference: catch up with obj, fall behind, follow, follow closely, run into obj, pull out from behind obj, get out of the way of obj, cut in in front of obj, slip in in front of obj, pull up to, flank, move past, let run into, pass, drive in front of, lose a lead on, draw ahead of obj, approach oncoming obj, make way for oncoming obj, leave an obj driving off in opposite direction, approach crossing obj, close up to obj, merge in front of obj, leave crossing obj, move towards stationary obj, stop behind stationary obj, be standing near stationary obj, start in front of stationary obj, pull out behind stationary obj, drive around stationary obj, pass stationary obj, merge in front of stationary obj, move away from stationary obj, collide with obj (34 occurrences).

References

[André *et al.* 88] E. André, G. Herzog, T. Rist, On the Simultaneous Interpretation of Real World Image Sequences and their Natural Language Descriptions: The System Soccer, in Y. Kodratoff (ed.), *Proc. 8th Europ. Conf. on Artificial Intelligence*, München, Aug. 1-5, 1988, pp. 449-454.

[Birnbaum *et al.* 93] L. Birnbaum, M. Brand, P. Cooper, Looking for Trouble: Using Causal Semantics to Direct Focus of Attention, in *Proc. Int. Conf. on Computer Vision (ICCV '93)*, Berlin, Germany, May 11-14, 1993, pp. 49–56.

[Campani & Verri 92] M. Campani, A. Verri, Motion Analysis from First-Order Properties of Optical Flow, *Computer Vision, Graphics, and Image Processing* **56** (1992) 90–107.

[Deriche & Faugeras 90] R. Deriche, O. Faugeras, Tracking line segments, *Image and Vision Computing* **8**:4 (1990) 261–270.

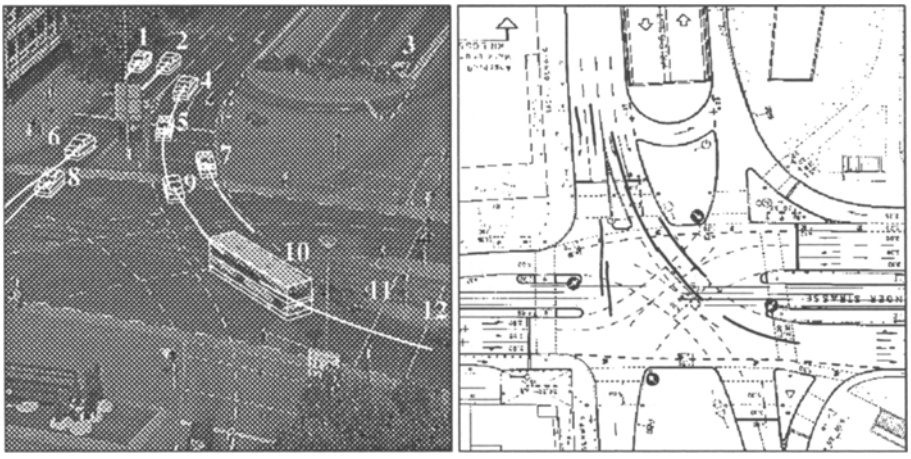

Fig. 3. The estimated trajectories of each moving vehicle of our test image sequence and a projection of the estimated trajectories into the street plane, superimposed to a digitized image of an official map for this intersection. The vehicles are referred to by numbers indicated in the left frame.

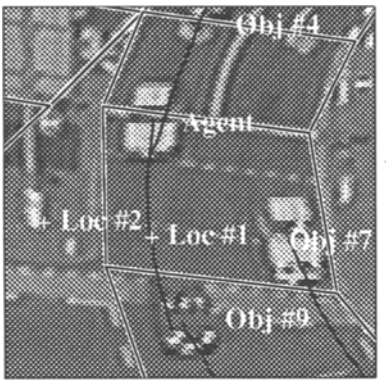

Fig. 4. Enlarged section of the image shown in Fig. 3. Obj. # 5 was selected as agent and two locations are marked as '+ Loc #1' and '+ Loc #2'. An interactively created road model is superimposed, representing road sections as polygons.

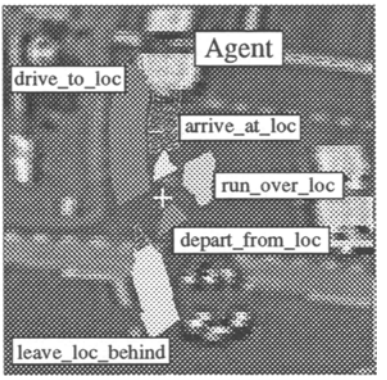

Fig. 6. The trajectory of agent #5 shown in Fig. 4 is colored by location occurrences involving location #1 shown in Fig. 5.

Agent Reference

```
  7 :   24 ! turn_left(obj_5). % 1
  7 :   25 ! turn_left(obj_5). % 0.98
  7 :   26 ! turn_left(obj_5). % 0.82
  7 :   27 ! turn_left(obj_5). % 0.69
  7 :   28 ! turn_left(obj_5). % 0.58
  7 :   29 ! turn_left(obj_5). % 0.48
 32 :   48 ! turn_left(obj_5). % 0.4
 31 :   48 ! turn_left(obj_5). % 0.39
  7 :   48 ! turn_left(obj_5). % 0.39
  7 :   49 ! turn_left(obj_5). % 0.38
 63 :   79 ! accelerate(obj_5). % 0.94
 64 :   80 ! run_straight_ahead(obj_5). % 0.26
 64 :   81 ! run_straight_ahead(obj_5). % 0.22
 63 :   81 ! run_straight_ahead(obj_5). % 0.21
 63 :   82 ! run_straight_ahead(obj_5). % 0.21
 93 :   83 ! run_straight_ahead(obj_5). % 0.2
 72 :   88 ! turn_left(obj_5). % 0.56
 72 :   89 ! turn_left(obj_5). % 0.52
 71 :   89 ! turn_left(obj_5). % 0.52
  7 :   93 ! drive_at_regular_speed(obj_5). % 1
  7 :   93 ! run_forward(obj_5). % 1
```

Object Reference

```
  7 :   64 ! drive_in_front_of(obj_5, obj_4). % 1
 16 :   73 ! follow(obj_5, obj_3). % 1
```

Road Reference

```
 63 :   63 ! turn(obj_5). % 0.75
 63 :   64 ! turn(obj_5). % 0.65
 63 :   65 ! turn(obj_5). % 0.53
 63 :   65 ! turn(obj_5). % 0.47
 66 :   66 ! change_section(obj_5). % 0.53
 65 :   66 ! change_section(obj_5). % 0.47
 64 :   66 ! change_section(obj_5). % 0.35
 63 :   66 ! change_section(obj_5). % 0.21
  7 :   93 ! drive_on_lane(obj_5). % 1
```

Location Reference

```
  7 :   52 ! drive_to_location(obj_5, loc_1). % 1
 43 :   49 ! arrive_at_location(obj_5, loc_1). % 0.65
 43 :   50 ! arrive_at_location(obj_5, loc_1). % 0.74
 43 :   51 ! arrive_at_location(obj_5, loc_1). % 0.81
  9 :   52 ! arrive_at_location(obj_5, loc_1). % 0.86
  8 :   52 ! arrive_at_location(obj_5, loc_1). % 0.87
 52 :   58 ! run_over_location(obj_5, loc_1). % 0.91
 52 :   59 ! run_over_location(obj_5, loc_1). % 0.9
 51 :   59 ! run_over_location(obj_5, loc_1). % 0.82
 51 :   60 ! run_over_location(obj_5, loc_1). % 0.81
 50 :   60 ! run_over_location(obj_5, loc_1). % 0.74
 50 :   61 ! run_over_location(obj_5, loc_1). % 0.72
 49 :   61 ! run_over_location(obj_5, loc_1). % 0.65
 49 :   62 ! run_over_location(obj_5, loc_1). % 0.62
 48 :   62 ! run_over_location(obj_5, loc_1). % 0.58
 48 :   63 ! run_over_location(obj_5, loc_1). % 0.54
 47 :   63 ! run_over_location(obj_5, loc_1). % 0.53
 47 :   64 ! run_over_location(obj_5, loc_1). % 0.45
 46 :   64 ! run_over_location(obj_5, loc_1). % 0.44
 46 :   65 ! run_over_location(obj_5, loc_1). % 0.36
 45 :   65 ! run_over_location(obj_5, loc_1). % 0.35
 44 :   65 ! run_over_location(obj_5, loc_1). % 0.27
 44 :   66 ! run_over_location(obj_5, loc_1). % 0.24
 60 :   66 ! depart_from_location(obj_5, loc_1). % 0.76
 60 :   67 ! depart_from_location(obj_5, loc_1). % 0.81
 59 :   68 ! depart_from_location(obj_5, loc_1). % 0.9
 58 :   69 ! depart_from_location(obj_5, loc_1). % 0.99
 56 :   69 ! depart_from_location(obj_5, loc_1). % 1
 58 :   93 ! leave_location_behind(obj_5, loc_1). % 1

  7 :   33 ! drive_to_location(obj_5, loc_2). % 1
 30 :   69 ! pass_location(obj_5, loc_2). % 0.26
 29 :   70 ! pass_location(obj_5, loc_2). % 0.34
 28 :   71 ! pass_location(obj_5, loc_2). % 0.42
 27 :   71 ! pass_location(obj_5, loc_2). % 0.45
 26 :   72 ! pass_location(obj_5, loc_2). % 0.49
 25 :   73 ! pass_location(obj_5, loc_2). % 0.56
 24 :   74 ! pass_location(obj_5, loc_2). % 0.61
 23 :   75 ! pass_location(obj_5, loc_2). % 0.67
 22 :   75 ! pass_location(obj_5, loc_2). % 0.72
 21 :   76 ! pass_location(obj_5, loc_2). % 0.77
 20 :   77 ! pass_location(obj_5, loc_2). % 0.84
 19 :   78 ! pass_location(obj_5, loc_2). % 0.9
 19 :   79 ! pass_location(obj_5, loc_2). % 0.9
 18 :   79 ! pass_location(obj_5, loc_2). % 0.95
 17 :   80 ! pass_location(obj_5, loc_2). % 1
 15 :   81 ! pass_location(obj_5, loc_2). % 1
 66 :   93 ! leave_location_behind(obj_5, loc_2). % 1
```

Fig. 5. The output of the computed occurrence descriptions after selecting object #5 as agent. The descriptions contain a time interval (before the exclamation mark), the involved objects and locations (in round brackets as arguments) and the fuzzy membership degree (following the percent symbol).

[Du *et al.* 93] L. Du, G. D. Sullivan, K. D. Baker, Quantitative Analysis of the Viewpoint Consistency Constraint in Model-Based Vision, in *Proc. Int. Conf. on Computer Vision (ICCV '93)*, Berlin, Germany, May 11-14, 1993, pp. 632–639.

[Gong & Buxton 93] S. Gong, H. Buxton, From Contextual Knowledge to Computational Constraints, in *Proc. Brit. Machine Vision Conf.*, Guildford, UK, Sept. 21-23, 1993, pp. 229–238.

[Heinze *et al.* 91] N. Heinze, W. Krüger, H.-H. Nagel, Berechnung von Bewegungsverben zur Beschreibung von aus Bildfolgen gewonnenen Fahrzeugtrajektorien in Straßenverkehrsszenen (in German), *Informatik - Forschung und Entwicklung* **6** (1991) 51–61.

[Herzog 92] G. Herzog, Utilizing Interval-Based Event Representations for Incremental High-Level Scene Analysis, in *Proc. Fourth European Workshop on Semantics of Time, Space and Movement and Spatio-Temporal Reasoning*, Château de Bonas, France, Sept. 4-8, 1992, pp. 425–435.

[Koller 92] D. Koller, *Detektion, Verfolgung und Klassifikation bewegter Objekte in monokularen Bildfolgen am Beispiel von Straßenverkehrsszenen* (in German), Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Karlsruhe, Juni 1992, available as vol. DISKI 13, *Dissertationen zur Künstlichen Intelligenz*, infix-Verlag, Sankt Augustin, Deutschland, 1992.

[Koller *et al.* 91] D. Koller, N. Heinze, H.-H. Nagel, Algorithmic Characterization of Vehicle Trajectories from Image Sequences by Motion Verbs, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR '91)*, Lahaina, Maui, Hawaii/HI, June 3-6, 1991, pp. 90–95.

[Koller *et al.* 93] D. Koller, K. Daniilidis, H.-H. Nagel, Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes, *Intern. Journal of Comp. Vision* **10**:3 (1993) 257–281.

[Kollnig & Nagel 93] H. Kollnig, H.-H. Nagel, Ermittlung von begrifflichen Beschreibungen von Geschehen in Straßenverkehrsszenen mit Hilfe unscharfer Mengen (in German), *Informatik – Forschung und Entwicklung* **8** (1993) 186–196.

[Kories & Zimmermann 86] R. Kories, G. Zimmermann, A Versatile Method for the Estimation of Displacement Vector Fields from Image Sequences, in *Proc. of IEEE Workshop on Motion: Representation and Analysis*, Kiawah Island Resort, Charleston/SC, May 7-9, 1986, pp. 101-106.

[Liu & Huang 93] Y. Liu, T.S. Huang, Vehicle-Type Motion Estimation from Multi-frame Images, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **PAMI-15**:8 (1993) 802–808.

[Lowe 92] D. G. Lowe, Robust Model-Based Motion Tracking Through the Integration of Search and Estimation, *International Journal of Computer Vision* **8**:2 (1992) 113–122.

[Mohnhaupt & Neumann 90] M. Mohnhaupt, B. Neumann, On the Use of Motion Concepts for Top-Down Control in Traffic Scenes, in O. Faugeras (ed.), *Proc. Second European Conference on Computer Vision (ECCV '90)*, Antibes, France, Apr. 23-26, 1990, Lecture Notes in Computer Science **427**, Springer-Verlag, Berlin, Heidelberg, New York/NY and others, 1990, pp. 598–600.

[Nagel 88] H.-H. Nagel, From Image Sequences towards Conceptual Descriptions, *Image and Vision Computing* **6**:2 (1988) 59–74.

[Nagel 91] H.-H. Nagel, La représentation de situations et leur reconnaissance à partir de séquences d'images — The Representation of Situations and their Recognition from Image Sequences, in *8e Congrès Reconnaissance des Formes et Intelligence Artificielle*, Lyon–Villeurbanne, 25–29 Novembre 1991, AFCET, 1991, pp. 1221–1229.

[Nagel & Kollnig 94] H.-H. Nagel, H. Kollnig, Description of the Motion of Road Vehicle Agglomerations in Image Sequences by Natural Language Verbs (1994). In preparation.

[Neumann & Novak 86] B. Neumann, H.-J. Novak, Naos: Ein System zur natürlichsprachlichen Beschreibung zeitveränderlicher Szenen, *Informatik - Forschung und Entwicklung* **1** (1986) 83-92.

[Otte & Nagel 92a] M. Otte, H.-H. Nagel, Extraction of Line Drawings from Gray Value Images by Non-Local Analysis of Edge Element Structures, in G. Sandini (ed.), *Proc. Second European Conference on Computer Vision (ECCV '92)*, S. Margherita Ligure, Italy, May 18-23, 1992, Lecture Notes in Computer Science **588**, Springer-Verlag, Berlin and others, 1992, pp. 687–695.

[Otte & Nagel 92b] M. Otte, H.-H. Nagel, *Verbesserte Extraktion von Strukturen aus Kantenelementbildern durch Auswertung von Kantenelementketten* (in German), Interner Bericht, Institut für Algorithmen und Kognitive Systeme, Fakultät für Informatik der Universität Karlsruhe (TH), Karlsruhe, Deutschland, Oktober 1992.

[Otte & Nagel 94] M. Otte, H.-H. Nagel, Optical Flow Estimation: Advances and Comparisons, in: *Proc. Third Europ. Conf. on Computer Vision ECCV '94*, Stockholm, Sweden, May 2-6, 1994.

[Retz-Schmidt 91] G. Retz-Schmidt, Recognizing Intentions, Interactions and Causes of Plan Failures, *User Modeling and User-Adapted Interaction* **1**:2 (1991) 173–202.

[Sung 88] C.-K. Sung, Extraktion von typischen und komplexen Vorgängen aus einer Bildfolge einer Verkehrsszene (in German), in H. Bunke, O. Kübler, P. Stucki (Hrsg.), *Mustererkennung 1988*, Zürich, Informatik-Fachberichte **180**, Springer-Verlag, Berlin u.a., 1988, pp. 90–96.

[Tan *et al.* 92] T. N. Tan, G. D. Sullivan, K. D. Baker, Linear Algorithms for Object Pose Estimation, in *Proc. British Machine Vision Conference*, Leeds, UK, Sept. 22-24, 1992, pp. 600–609.

[Tan *et al.* 93] T. N. Tan, G. D. Sullivan, K. D. Baker, Recognising Objects on the Ground Plane, in *Proc. British Machine Vision Conference*, Guildford, UK, Sept. 21-23, 1993, pp. 85–94.

[Toal & Buxton 92] A. F. Toal, H. Buxton, Spatio-temporal Reasoning within a Traffic Surveillance System, in G. Sandini (ed.), *Proc. Second European Conference on Computer Vision (ECCV '92)*, S. Margherita Ligure, Italy, May 18-23, 1992, Lecture Notes in Computer Science **588**, Springer-Verlag, Berlin, Heidelberg, New York/NY and others, 1992, pp. 884–892.

[Worrall *et al.* 93] A. D. Worrall, G. D. Sullivan, K. D. Baker, Advances in Model-Based Traffic Vision, in *Proc. Brit. Machine Vision Conf.*, Guildford, UK, Sept. 21-23, 1993, pp. 559–568.

[Zhang *et al.* 93] S. Zhang, G. D. Sullivan, K. D. Baker, The Automatic Construction of a View-Independent Relational Model for 3-D Object Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-15**:6 (1993) 531–544.