Intelligent Autonomous System 2 (IAS-2)
Amsterdam, 11-14 December 1989

Relative 3D-State Estimation for Autonomous Visual Guidance of Road Vehicles

E.D. Dickmanns, Th. Christians

Universität der Bundeswehr München
Department of Aero-Space Engineering*

Abstract: The integrated spatio-temporal approach to real-time machine vision, which has allowed outstanding performance with moderate computing power, is extended to obstacle recognition and relative spatial state estimation using monocular vision. A modular vision system architecture is discussed centering around features and objects. Experimental results with VaMoRs, a 5-ton test vehicle are given. Stopping in front of obstacles of at least 0.5 m^2 cross section has been demonstrated on unmarked two-lane roads at velocities up to 40 km/h.

1. Introduction

Fully autonomous visual road vehicle guidance has received considerable attention since DARPA started its 'Autonomous Land Vehicle (ALV)' project in 1983 as a demonstrator in its program 'On Strategic Computing', and especially in Europe since the Eureka-project Prometheus was initiated in 1986.

In Japan, Tsugawa e.a. [1,2] had been working on this problem since the late 70ies. At the authors' institution, simulation studies started in 1978 [3]; the test vehicle to become 'VaMoRs' was developed from 1984 to 86. The vision system hardware had been under development from the beginning of this decade [4]. Contrary to the AI-oriented approaches in the DARPA projects [5,6,7], our approach has been control-oriented, exploiting the well developed linear system theory for recursive state estimation.

The superiority of this approach, at least for well structured scenes like roads, has become apparent over the last three years. A survey on this method is given in [4]; speeds close to 100 km/h on a freeway and up to 60 km/h on an unmarked two-lane country road have been demonstrated repeatedly.

In the next section, a brief review of this 4D-approach is given; then, the method is extended for dealing with obstacles, leading to a modular processing structure to be discussed in the following section. After an explication of the signal flow in real-time operation, results with VaMoRs will be discussed.

---

*Werner- Heisenberg-Weg 39, D-8014 Neubiberg, FRG
This research has been supported by BMFT and the Daimler-Benz AG: Grant No. ITM 8503E

2. The 4D-integrated approach

Figure 1 summarizes the model based spatio-temporal approach to real-time vision: Parallel to the real world (upper left rectangle) a 'mental' world representation is maintained in the interpretation process (upper right), which is manipulated by prediction error feedback in such a way as to duplicate (symbolically) the real world with respect to those objects of highest significance for the task at hand.

Single objects are represented as units existing in 3D-space and time. The spatial shape is impoverished to a distribution of visual features relative to some centroid. Motion, i.e. time, is idealised to translation of and rotation around the center of gravity (c.g.). Temporally fixed sampling of the scene by a TV-camera with period T is assumed. We confine ourselves at present to rigid objects. The motion capabilities of objects, which are constraints characterizing the object, are represented by generic models via difference equations with discretization period T, the socalled dynamical model.

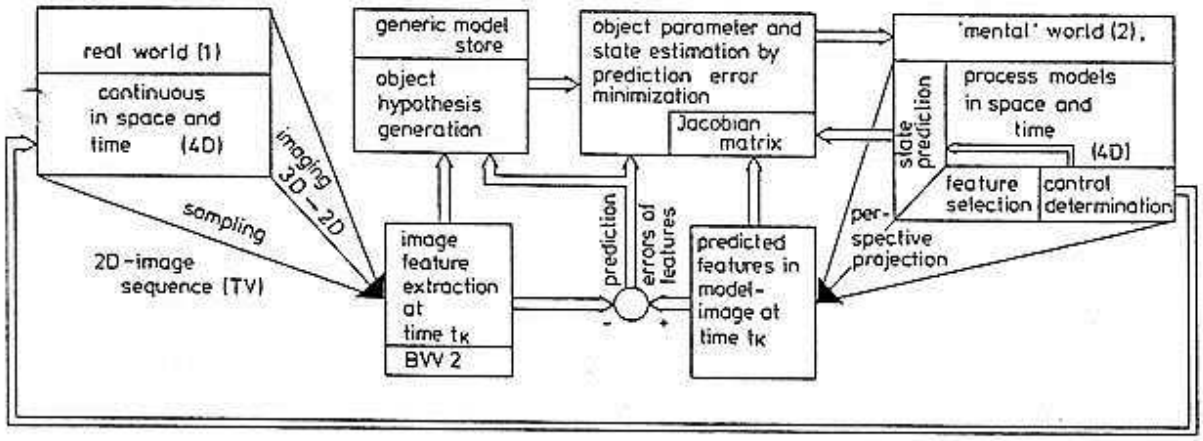


Figure 1: Basic scheme for 4D-image sequence understanding by prediction error minimization

Given a complete state description at time t, the state at the next sampling time t + T can be predicted exploiting the dynamical model. For the features to be tracked, their position and orientation are determined using the simple forward perspective projection (lower center right in fig. 1). At the same time the Jacobian matrix of the feature positions in the image with respect to the state variable components in 3D-space are determined from the model.

Following Kalman's idea of a recursive least squares measurement data fit using a generic model [9], a numerically very efficient estimation of the full 3D-motion is obtained, bypassing the inversion of the perspective mapping. For details see [4].

Figure 2 shows the system architecture realising this approach for road vehicle guidance: In the upper right, the scene as imaged by one wide angle- and one tele-camera is shown. The cameras are mounted together on a two-axis pan and tilt platform (ZP, top center). Their signals are digitized and fed onto a video-bus in the image sequence multi-processor system BVV (left). Parallel processors (PP_i) for feature extraction grab picture elements (pel) belonging to a rectangular sub-area (socalled windows) and determine the position of linear edge elements with a preset direction by simplified correlation [4]. The window shape, the direction parameter and the search path may be controlled by the higher interpretation levels; in the image, several such windows (1 through 9) are shown.

In an initialization phase, using intelligent search strategies, feature groupings belonging to one object have to be detected. This is achieved by the object processor GPP_i (dashed and dotted curves around

several PP and one GPP, fig. 2 left) having knowledge about how spatial objects look in the image under certain aspect conditions. The GPP also contains the dynamical model for introducing the temporal constraints into the interpretation process.

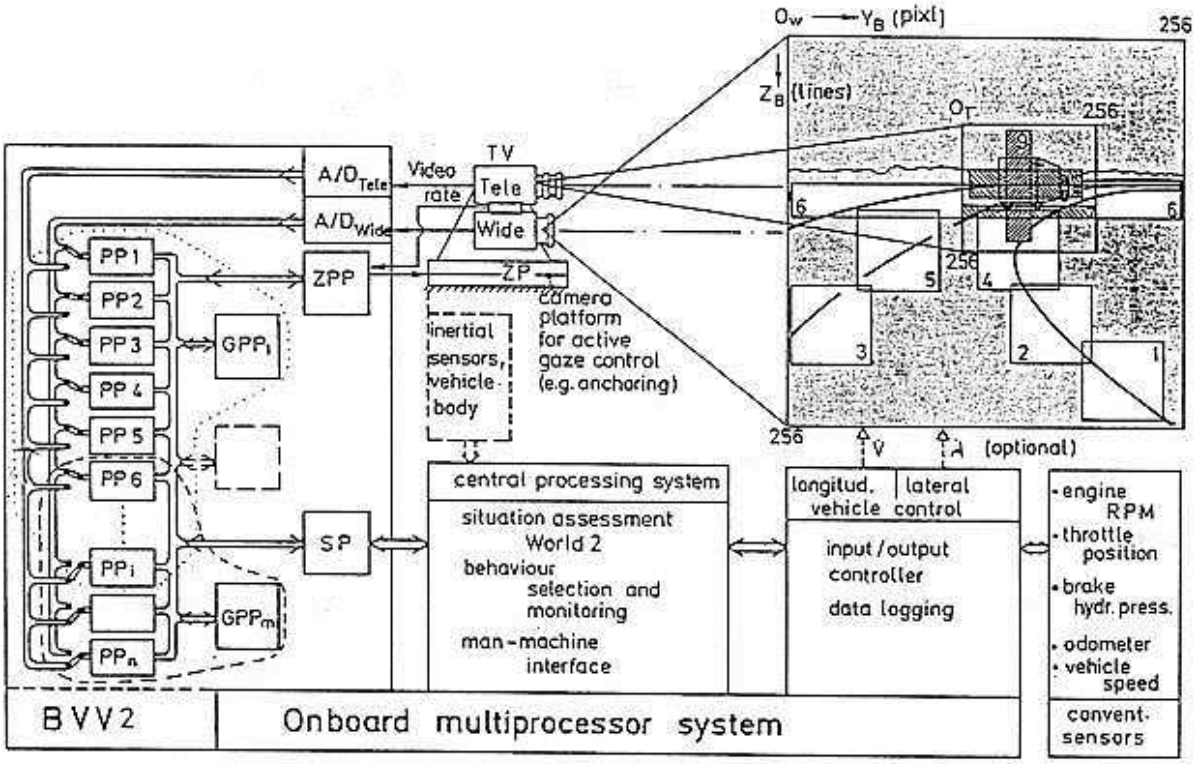


Figure 2: System architecture for the automatic control of road vehicles by visual feedback

In the tracking phase, each PP_i locks onto its feature set and communicates the feature data to the GPP for recursive state update. Usually, two to three times the number of features minimally required for object tracking are evaluated for achieving robust motion recognition. PP's are Intel 8086 or 286 single board computers, while GPP's are 386 ones.

The GPP delivers the best estimate for the object state via the communication processor SP to the main computer (lower center, central processing system). Here the situation is assessed, integrating data from other sensors (lower right); then the control for the own vehicle is determined. This is done at two levels. For a given situation, a state-variable-feedback-law is applied yielding fast reflexlike behavioral competences (10 to 20 Hz update rate). If a situation is changing, e.g. from lane following to lane changing as a mission element, more complex control sequences have to be applied. The situation assessment rate may be slower than the normal control rate.

Lane following with speed adapted to road curvature has been demonstrated for the speed range of VaMoRs (up to 100 km/h); also the lane changing capability has been proven.

3. Extension for dealing with obstacles

The general scheme implemented for road curvature detection and for vehicle ego-state estimation relative to the road has been extended in 1988/89 to obstacle handling. This is done in three steps: 1. detection of obstacle candidates, 2. recognition of object size and location in the image plane and

3. estimation of the spatial state relative to the ego-vehicle and of the physical dimensions of the obstacle.

Steps 1 and 2 have been treated in [10] to some extend; here step 3 is detailed. It deals with a feature set determined from windows 7 to 9 in the telecamera image in fig. 2. Vertical and horizontal feature pairs, centering each other crosswise, are taken as input to the object processor. If these features belong to the same object they should move in conjunction, except for changes in the aspect conditions which induce relative changes between the feature positions.

3.1 Geometry model

In fig. 3 the nomenclature used is given. Besides the object dimension, also the left and right road boundaries at the position of the lower end of the obstacle are determined (y_{BRl}, y_{BRr}). This immediately yields the object size in units of the road width, and it allows to determine the lateral position of the object on the road, needed for determining the reaction of the vehicle to the obstacle to be executed. Observing these variables over time, together with the conventionally measured egomotion-speed and distance travelled allows to determine the absolute longitudinal speed of the obstacle and its lateral speed component relative to the road. Note that this latter information cannot be derived from a rangefinder if road and shoulders are planar. In our approach, additionally range and range rate can be determined from prediction error feedback exploiting the dynamical model over time, utilizing a monocular intensity-image sequence only.

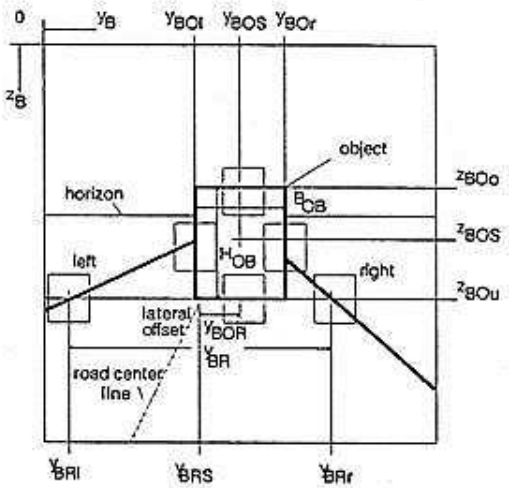


Figure 3: Feature based obstacle recognition, nomenclature

Figure 4 shows the perspective mapping of significant feature positions onto the image plane in a top down (a) and a side view (b). Only the backplane of the object, which is considered to have a shape close to a parallelepiped (rectangular box), is depicted.

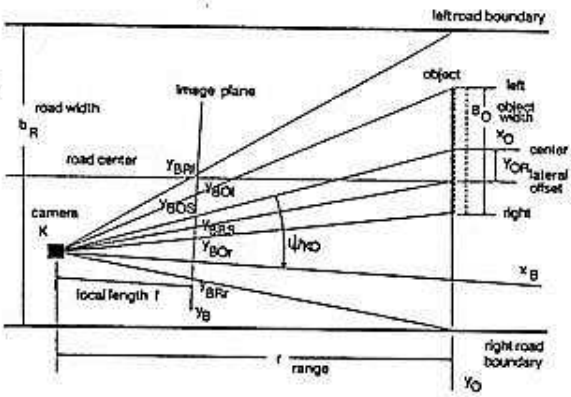


Figure 4a: top down view

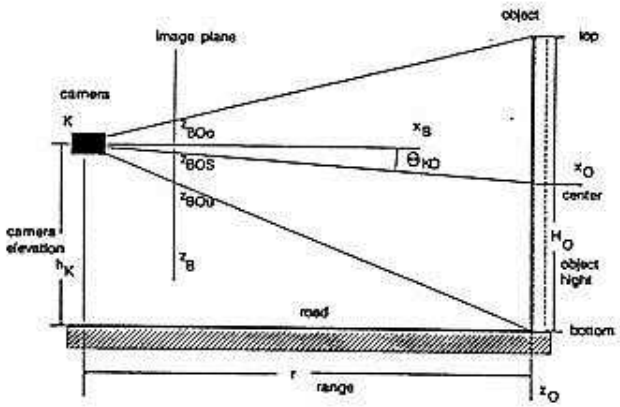


Figure 4b: side view

Figure 4: Measurement- and relative state variables camera to object

Assuming a planar road surface and small angles (cos ≈ 1, sine ≈ argument), all mapping conditions are simple and need not be detailed here. Sums and differences of feature positions yield the position and size information looked for.

3.2 The dynamical model for relative state estimation

Of prime interest are the range r and the range rate $\dot{r}$ to the obstacle; the former is the integral of the latter. Then, the lateral motion of the object relative to the road v_{OR} is important for deriving the proper reaction for the own vehicle. Since the control inputs to the other object are not known in general, they are modeled by stochastic disturbance variables s_i. This yields the dynamical model (v = speed along the road)

$$\dot{r} = v_O - v + s_r \qquad \text{(Index O = Object)} \qquad (1)$$

$$\dot{v}_O = s_{vO} \qquad (2)$$

$$\dot{y}_{OR} = v_{OR} \qquad \text{(Index R = Road)} \qquad (3)$$

$$\dot{v}_{OR} = s_{yOR}\,. \qquad (4)$$

In addition, for determining the obstacle size and the viewing direction relative to its center, the following four state variables are added

$$\dot{H}_O = s_{HO} \qquad (5)$$

$$\dot{B}_O = s_{BO} \qquad (6)$$

$$\dot{\psi}_{KO} = s_{\psi KO} \qquad \text{(Index K = camera)} \qquad (7)$$

$$\dot{\Theta}_{KO} = s_{\Theta KO} \qquad (8)$$

where again the s_i are assumed to be unknown Gaussian random noise. In shorthand vector notation these eqs are written in the form

$$\dot{x}(t) = f[x(t), u(t), s(t)] \qquad (9)$$

with the state variables

$$x = (r, v_O, \psi_{KO}, \Theta_{KO}, B_O, H_O, y_{OR}, v_{OR})\,. \qquad (10)$$

After transformation into the discrete state transition form, standard methods for state estimation are applied.

3.3 The estimation process

Figure 5 left shows the window arrangement set up for relative obstacle state determination. The initialization is performed by steps 1 and 2 mentioned above; then, the object processor GPP2 computes starting values for the spatio-temporal iteration based on the known road width determined by GPP1 in the near range (right half). Note that the range to the obstacle can also be derived from the vertical position of the lower feature where the obstacle touches the ground.

The estimation cycle on GPP2, an 80386 microprocessor, runs at 25 Hz (40 ms) while the feature extraction and -tracking runs at video rate (50 Hz) on 8086's. The initial transient takes 10 to 20 cycles.

In a prediction step, the expected position of features for the next measurement is computed by applying forward perspective projection to the object as 'imagined' by the interpretation process. Only those feature position candidates delivered by the PP's which are close to these values, are accepted; others are rejected as outliers. This contributes considerably to stabilizing the interpretation in noisy natural environments.

Due to the integral-relationships in eqs (1) and (3) also the speed components can be estimated in a consistent manner using only position data over time.

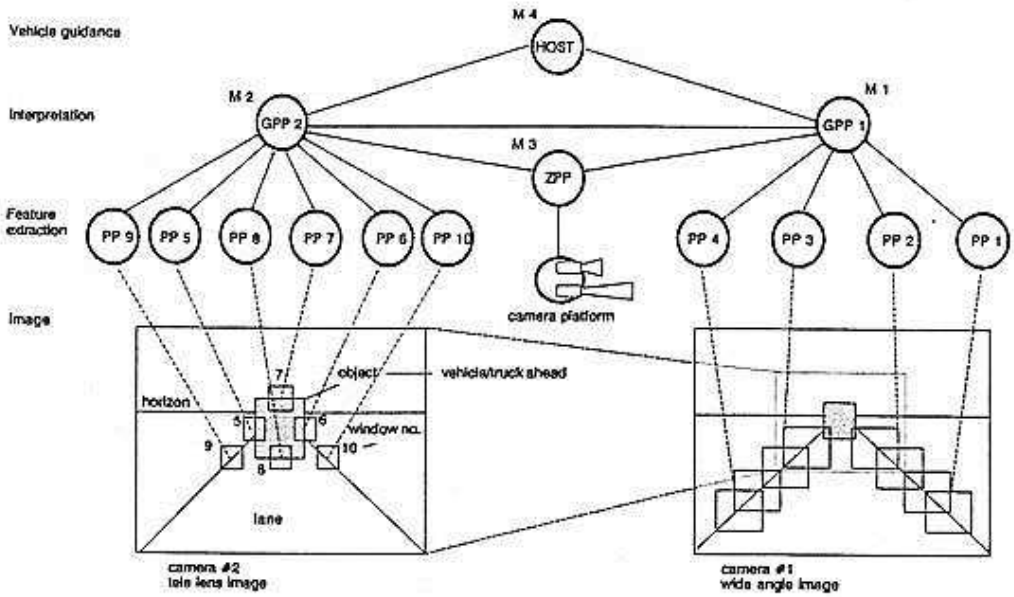


Figure 5: Modular processing structure for visual road vehicle guidance

4. Modular processing structure

In figure 5 the modular processing structure resulting naturally in this object-oriented 4D-approach is emphasized. There are four processing layers shown: the pel-level (bottom), where 2D-spatial data structures (intensity images and subimages) have to be handled. Then, at the PP-level, edge-element, corner- and adjacent intensity-features are extracted with respect to 2D-position and orientation, still missing any relation to 3D-space or time. Only in the third layer, implementing object interpretations on the GPP, spatial and temporal constraints are introduced for associating objects with groupings of features and their relative change over time. In our case, objects are the road, the ego-vehicle and obstacles on the road, right now just one; it is easily seen, how this approach can be extended to multiple objects by adding more groups of processors.

It may be favourable for the initialization phase to insert an additional 2D-object layer between layers 2 and 3 as given here; this is still under discussion [11]. Also layer 4 for vehicle guidance as implemented now, will have to be expanded in the future in order to be able to deal with more complex tasks and situations. Table 1 shows the structure under development. In addition to the multiprocessor system for image sequence processing and object state estimation (BVVx) there will be a second

one (termed MPS in column 2) for situation assessment, relative goal state evaluation (inclusive planning) and control decisions as well as implementation.

	acitvity level	processors	operation	result
scene understanding	control level	MPS	compute expectations control viewing direction apply vehicle control	-> action
	↑		↑	
	task level	MPS	relative goal state evaluation <-	-> planning, decisions
	↑		↑	↑
	object level	MPS	-> situation assessment parameter adaptation <-	-> situation <-
state estimation	↑		↑	↑
	feature level	GPP	-> feature aggregation	-> objects in space/time <-
	↑		↑	↑
	pel level	PP	-> feature extraction	-> features in image plane <-

Table 1: Modular processing structure for complex tasks

5. Signal flow in real-time operation

In order to better visualize the operational interconnections between the subsytems, fig. 6 shows a block diagram of the system for dynamic vision: In the lower center the real world and the feature

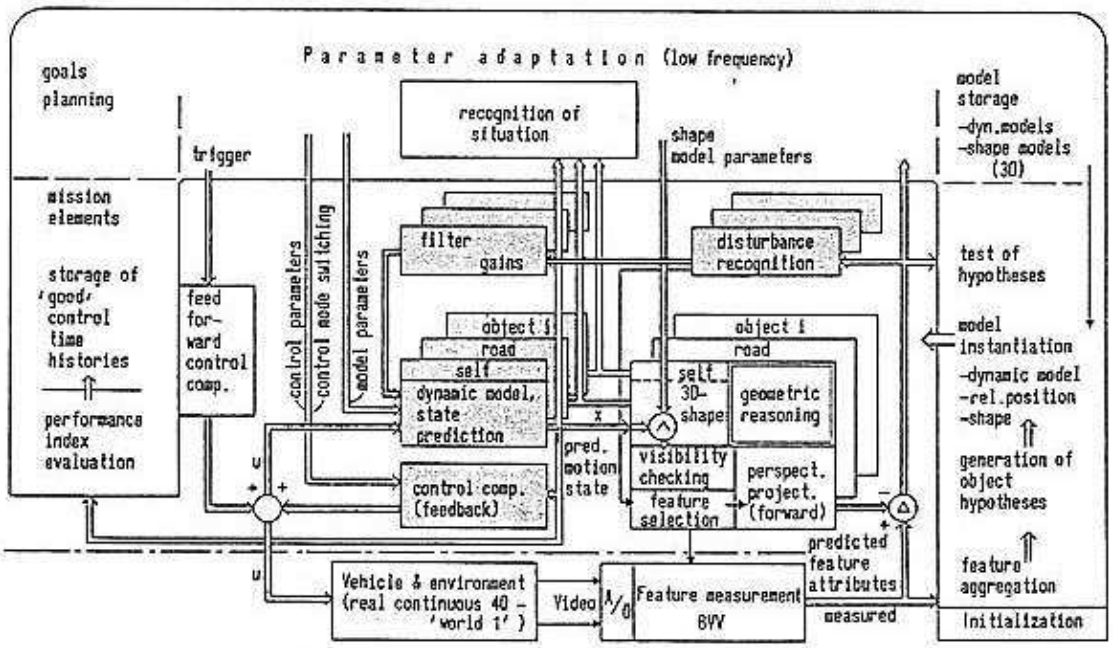


Figure 6: Block diagram of dynamic vision; control oriented AI (4D prediction error minimization)

extraction by the BVV is depicted. Initially, via the horizontal data path to the right, object hypotheses have to be generated through feature aggregation (right column). The models to be instantiated have at least three essential ingredients: a dynamical model for temporal behavior, a 3D-shape model and the aspect conditions. Now the recursive estimation loop can be started which is shown in the center of the image. The shaded areas correspond to the well known control engineering methods (Kalman filter and derivatives [12]) while the geometric reasoning block is a new extension for image sequence processing.

The prediction errors (arrow at right, showing upward) are used for additional purposes besides the computations of the innovation. Since the scene is time varying and new features belonging to yet unknown objects may occur, a steady monitoring has to be done in order to detect new objects or to adapt parameters for generically known objects. All these parts in the upper and left outer layer of fig. 6 have to be further developed in the future. This may be done by a hybrid approach exploiting fitting parts from engineering and Artificial Intelligence (AI) methods. The left column is intended for learning good control strategies from autonomous experimenting in similar situations and evaluation of the results with respect to goal functions (upper left).

Figure 7 shows a simplified block diagram of this approach which is oriented towards a display which Rasmussen used for discussing human behavorial modes when dealing with the real world; the three layers are the same, details of implementation are different, of course. Our approach yields a simple means for switching control laws depending on the situation, thereby yielding flexible behavioral competences for classes of tasks (lower center in fig. 7). For each active control law, fast reflex-like behavior is achieved.

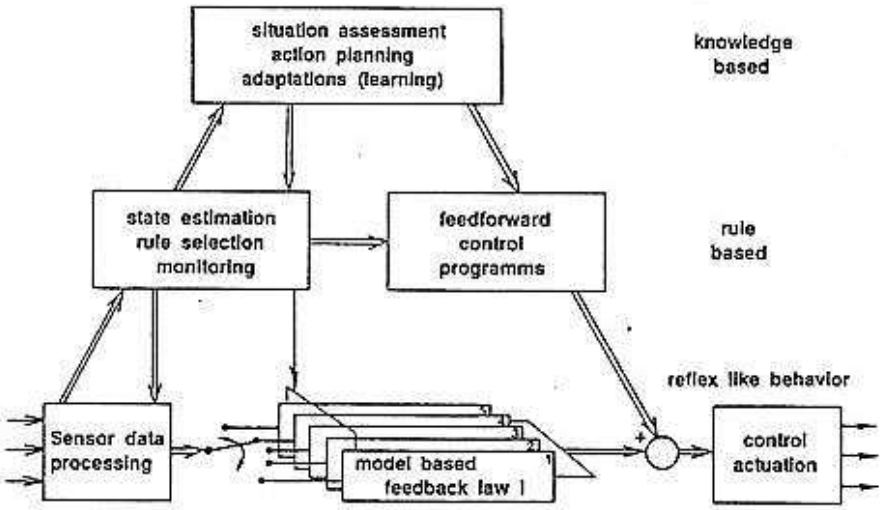


Figure 7: Idealized layered system architecture for fast and flexible realization of behavioral modes

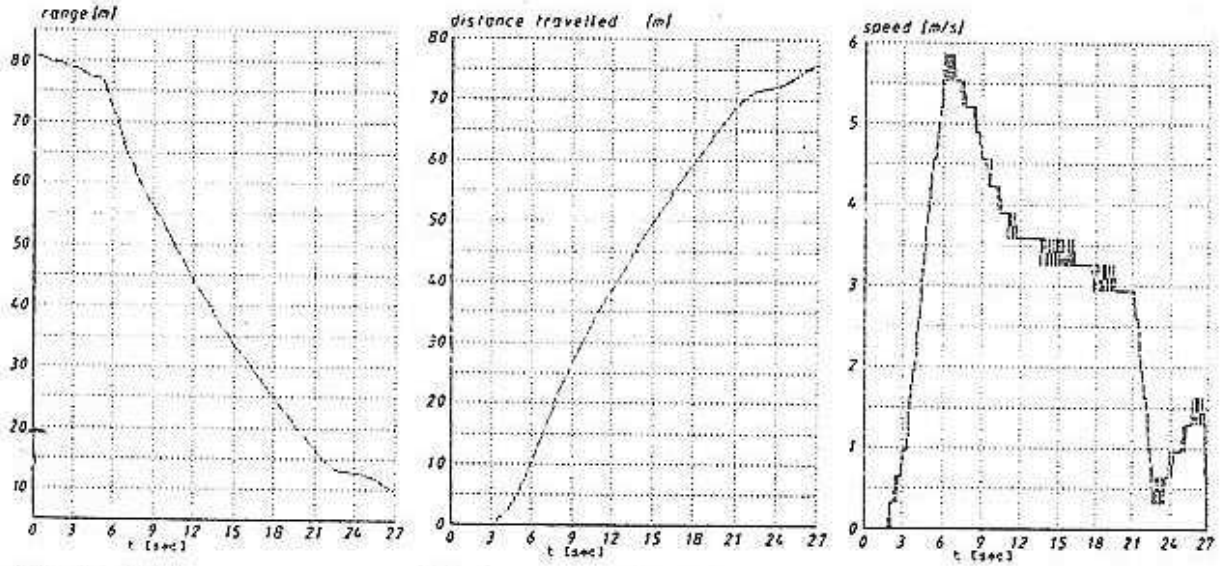
At present, our system is almost completely lacking the highest level. The development of the two basic layers has progressed to a state, however, where the evolution of the upper one is the natural next step. This can be based on the notion of objects existing in space and time as well as on knowledge of their spatial shape and their temporal behavior.

6. Experimental results

The system described has been tested both in simulation with real image sequence processing hardware (BVV 2) in the real-time loop and with two test vehicles in real scenes: our 5-ton van VaMoRs and a 10-ton bus of the Daimler-Benz AG equipped with our vision system.

6.1 Autonomous longitudinal control from rest

Figure 8 shows results of a test, where the autonomous vehicle initially stood still at a large distance from the obstacle having about 5 m^2 cross-section (another bus).



a) range $r^*(t)$ b) distance travelled $s(t)$ c) speed $v(t)$

Figure 8: Autonomous approach of an obstacle to a preset distance (10 m)

The initial distance was estimated at 81 m. The task for the vehicle was to approach the obstacle to a distance of 10 m as close as possible. At about 2s the vehicle starts moving (fig. 8c) and accelerates to a speed of 5.7 m/s (≈ 20 km/h) achieved at 6 to 7 s; then speed is decreased by the longitudinal controller depending on the distance to the obstacle. The precise control to the final stop is done at a low speed (≈ 1 m/s) for about the last 3 meters (23 to 27 s). The range after stop has been measured to be within about 5 % of the value specified.

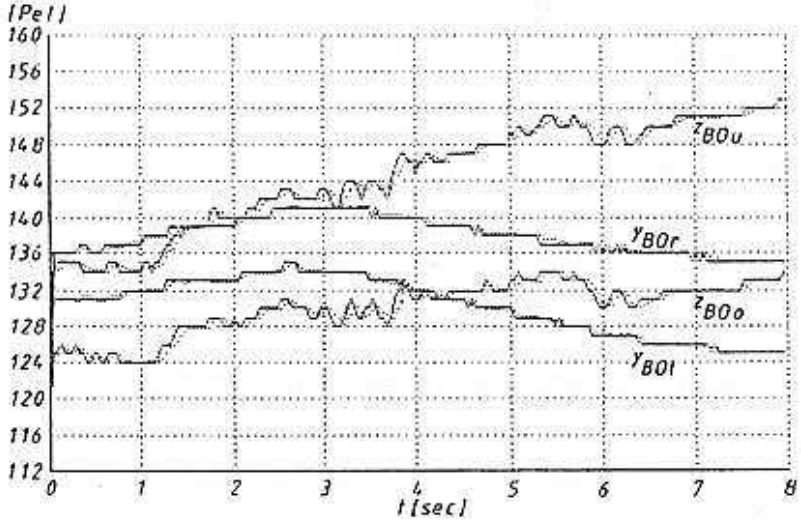
Since the overall range difference estimated $(r^*(o) - r^*(t_f)) = 71$ m differs by 5 m from the distance travelled, measured by the odometer of the vehicle, the intial estimate $r^*(o)$ must have been 4 to 5 m wrong (≈ 6 %). This error has been reduced to 1/5 at 18 s when the range had decreased to about 25 m. This is a typical behavior for monocular range estimation for known obstacle size, since the angle subtended by the obstacle - and therewith the measurement accuracy - increases during the approach; at 25 m range this angle is about 5 to 6°.

For stopping distance control this is an acceptable relationship since accuracy becomes better when it is most needed (nearby).

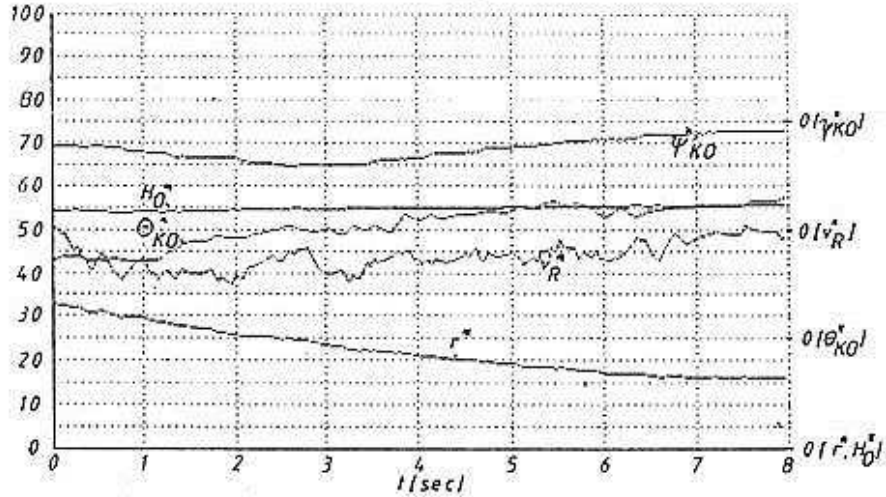
6.2 Obstacle detection while driving

The three-stage obstacle detection, recognition and relative spatial state estimation process has been tested with VaMoRs on an unmarked two-lane campus road at speeds up to 40 km/h with an obstacle of about 0.5 m^2 cross-section (a trash can). The detection range was set to about 35 to 40 m. As figure 9 shows, the range estimation started with $r^* = 33$ m. The initial speed in this test was about 4 m/s. The filter, however, was started with an initial value of zero in order to test the transient behavior (see curve v_R in fig. 9b).

It took about ten to fifteen videocycles, i.e. 0.2 to 0.3 seconds from detection for hypothesis verification and set up of the obstacle-processor group until first results for the spatial interpretation became available.



a) measured (—) and estimated (···) feature positions over time



b) time histories of estimated state variables range r^*/[m], range rate v^*_R/[0.3 m/s], obstacle height H^*_O/[0.02/m], azimuth direction to obstacle ψ^*_{KO}/[0.1°], pitch angle to obstacle centroid Θ^*_{KO}/ [0.1°]

Figure 9: Relative state estimation to obstacle

During the shown test portion, range decreased from 33 to 16 m. The object height was estimated very stable at 1.1 m. The pitch angle Θ^*_{KO} increased in magnitude during the approach since the elevation of the camera above the ground (1,8 m) was higher than the object centroid height. Apparently a slight curve was steered since the azimuth angle ψ^*_{KO} shows a dip (of less than 1°). Speed estimation in this run had been tuned to be rather sensitive as seen by the frequent oscillation. In the version presently running, speed is taken from the tachometer and the absolute velocity of the obstacle is being estimated.

7. Conclusions

The 4D approach to real-time machine vision has been shown to be well suited for depth estimation in monocular vision. Since image sequence evaluation is done with time explicitely represented in the model underlying the recognition process, motion stereo is an inherent property of the approach. Accuracies in the percent range have been demonstrated, becoming better the closer the obstacle is approached.

The feature based approach is computationally very economic and leads to a processor architecture oriented towards physical objects in a modular way. State of the art single board computers as MIMD processing elements allow cycle times of 40 ms (25 Hz).

Literature

[1] Tsugawa, S. e.a.: *An Automobile with Artificial Intelligence*. Proc. of the 6th IJCAI, Tokyo, Japan, 1979, pp 893-895.

[2] Tsugawa, S. e.a.: *An Intelligent Vehicle with Obstacle Detection and Navigation Functions*. Proc. of the IECON'84, Tokyo, Japan, Oct. 1984, pp 303-308.

[3] Meissner, H.G.: *Steuerung dynamischer Systeme aufgrund bildhafter Informationen*. Dissertation, Fakultät für Luft- und Raumfahrttechnik der Universität der Bundeswehr München.

[4] Dickmanns, E.D. and Graefe, V.: *Dynamic Monocular Machine Vision* and *Applications of Dynamic Monocular Machine Vision*. Int. Journal of Machine Vision & Application, Vol. 1, Springer-Internat., N.Y, 1988, pp 223-261.

[5] Turk, M. e.a.: *Video Road Following for the Autonomous Land Vehicle*. Proc. IEEE Int. Conf. on Robotics and Automation, Raleigh, NC, April 1987.

[6] Thorpe, C.E. and Kanade, T.: *Vision and navigation for the CMU Navlab*. SPIE Conf. 727 on 'Mobile Robots', Cambridge, Ma., USA, Oct. 1986.

[7] Seida, St. e.a.: *Vision-based Road following in the Autonomous Land Vehicle*. Proc. IEEE 26th Conf. on Decision and Control, Los Angeles, Dec. 1987.

[8] Mysliwetz, B. and Dickmanns, E.D.: *Distributed Scene Analysis for Autonomous Road Vehicles Guidance*. Proc. SPIE Conf. 852 on 'Mobile Robots', Cambridge, Mass., 1987, pp 72-79.

[9] Kalman, R.E.: *A New Approach to Linear Filtering and Prediction Problems*. Trans. ASME, Series D, J. Basic Engineering, 1960, pp 35-45.

[10] Graefe, V. and Regensburger, U.: *Analysis and Measurement of Objects in the Path of a Vision Guided Mobile Robot*. 2nd Workshop on Manipulators, Sensors and Steps toward Mobility. Manchester, Nov. 1988.

[11] Graefe, V.: *Dynamic Vision Systems for Autonomous Mobile Robots*. IEEE Workshop on Intelligent Robots and Systems IROS'89, Tsukuba, Sept. 1989.

[12] Maybeck, P.S.: *Stochastic Models, Estimation and Control*. Vol. 1, Academic Press, 1979.