AI Magazine Volume 25 Number 2 (2004) (© AAAI)

Dynamic Vision-Based Intelligence

Ernst D. Dickmanns

■ A synthesis of methods from cybernetics and AI yields a concept of intelligence for autonomous mobile systems that integrates closed-loop visual perception and goal-oriented action cycles using spatiotemporal models. In a layered architecture, systems dynamics methods with differential models prevail on the lower, data-intensive levels, but on higher levels, AI-type methods are used. Knowledge about the world is geared to classes of objects and subjects. Subjects are defined as objects with additional capabilities of sensing, data processing, decision making, and control application. Specialist processes for visual detection and efficient tracking of class members have been developed. On the upper levels, individual instantiations of these class members are analyzed jointly in the task context, yielding the situation for decision making. As an application, vertebrate-type vision for tasks in vehicle guidance in naturally perturbed environments was investigated with a distributed PC system. Experimental results with the test vehicle VaMoRs are discussed.

During and after World War II, the principle of feedback control became well understood in biological systems and was applied in many technical disciplines for unloading humans from boring work load in system control and introducing automatic system behavior. Wiener (1948) considered it to be universally applicable as a basis for building intelligent systems and called the new discipline *cybernetics* (the science of systems control). After many early successes, these arguments soon were oversold by enthusiastic followers; at that time, many people realized that high-level decision making could hardly be achieved on this basis. As a consequence, with the advent of sufficient digital computing power, computer scientists turned to descriptions of abstract knowledge and created the field of AI (Miller, Gallenter, and Pribram 1960; Selfridge 1959).[1] With respect to results promised versus those realized, a similar situation to that with cybernetics developed in the last quarter of the twentieth century.

In the context of AI, the problem of computer vision has also been tackled (see, for example, Marr [1982]; Rosenfeld and Kak [1976]; Selfridge and Neisser [1960]). The main paradigm initially was to recover three-dimensional (3D) object shape and orientation from single images or from a few viewpoints. In aerial or satellite remote sensing, the task was to classify areas on the ground and detect special objects. For these purposes, snapshot images taken under carefully controlled conditions sufficed. *Computer vision* was the proper name for these activities because humans took care of accommodating all side constraints observed by the vehicle carrying the cameras.

When technical vision was first applied to vehicle guidance (Nilsson 1969), separate viewing and motion phases with static image evaluation (lasting for many minutes on remote stationary computers) were initially adopted. Even stereo effects with a single camera moving laterally on the vehicle between two shots from the same vehicle position were investigated (Moravec 1983). In the early 1980s, digital microprocessors became sufficiently small and powerful, so that on-board image evaluation in near real time became possible. The Defense Advanced Research Projects Agency (DARPA) started its program entitled On Strategic Computing in which vision architectures and image sequence interpretation for ground-vehicle guidance were to be developed (autonomous land vehicle [ALV]) (AW&ST 1986). These activities were also subsumed under the title *computer vision.* This term became generally accepted for a broad spectrum of applications, which makes sense as long as dy-

namic aspects do not play an important role in sensor signal interpretation.

For autonomous vehicles moving under unconstrained natural conditions at higher speeds on nonflat ground or in turbulent air, it is no longer that the computer "sees" on its own. The entire body motion as a result of control actuation and perturbations from the environment has to be analyzed based on information coming from many different types of sensors. Fast reactions to perturbations have to be derived from inertial measurements of accelerations and the onset of rotational rates because vision has a rather long delay time (a few tenths of a second) until the enormous amount of data in the image streams has been interpreted sufficiently well. This concept is proven in biological systems operating under similar conditions, such as the vestibular apparatus of vertebrates.

If internal models are available in the interpretation process for predicting future states (velocity, position, and orientation components) based on inertial measurements, two essential information cross-feed branches become possible: (1) Own body pose need not be derived from vision but can be predicted from the actual state and inertially measured data (including the effects of perturbations) and (2) slow drift components, occurring when only inertial data are used for state prediction, can be compensated for by visual observation of stationary objects far away in the outside world. Thus, quite naturally, the notion of an extended presence is introduced because data from different points in time (and from different sensors) are interpreted in conjunction, taking additional delay times for control application into account. Under these conditions, it no longer makes sense to talk about computer vision. It is the vehicle with an integrated sensor and control system, which achieves a new level of performance and becomes able to see, also during dynamic maneuvering. The computer is the hardware substrate used for data and knowledge processing.

The article is organized as follows: First, a review of the 4D approach to vision is given, and then dynamic vision is compared to *computational vision,* a term that is preferred in the AI community. After a short history of experiences with the second generation of dynamic vision systems according to the 4D approach, the design goals of the third-generation system are discussed, emphasizing dynamic aspects. A summary of the requirements for a general sense of vision in autonomous vehicles and the natural environment is then followed with the basic system design. Discussions of the realization of the concept on commercial off-the-shelf (COTS) hardware (a distributed PC system) and the experimental results with VaMoRs conclude the article.

Figure 1. Test Vehicle VaMP for Highway Driving.

Review of the 4D Approach to Vision

This approach with temporal models for motion processes to be observed was quite uncommon when our group started looking at vision for flexible motion control in the late 1970s and early 1980s. In the mid-1980s, it was realized that full reward of this approach could be obtained only when a single model for each object was installed covering visual perception and spatiotemporal (4D—integrated models in three-dimensional space and time, the fourth dimension) state estimation (Dickmanns 1987; Wuensche 1988). Also, the same model was to be used for computation of expectations on higher system levels for situation assessment and behavior decision, which includes the generation of hypotheses for intended behaviors of other subjects in standard situations. The big advantage of this temporal embedding of single-image evaluation is that no previous images need to be stored. In addition, 3D motion and object states can more easily be recovered by exploiting continuity conditions according to the 4D models used (3D shape and dynamic models for 3D motion behavior over time).

Joint inertial and visual sensing and recursive state estimation with dynamic models are especially valuable when only partially predictable motion has to be observed from a moving platform subject to perturbations. This

is the general case in real life under natural conditions in the outside world. It was soon realized that humanlike performance levels could be expected only by merging successful methods and components both from the cybernetics approach and the AI approach. For about a decade, the viability of isolated perception and motion control capabilities for single tasks in vehicle guidance has been studied and demonstrated (Dickmanns 1995; Dickmanns and Graefe 1988). Figure 1 shows one of the test vehicles used at University of the Bundeswehr, Munich (UBM).

Basic knowledge had to be built up and, lacking performance of electronic hardware did not yet allow for the realization of large-scale vision systems; this lack of performance was true for many of the subfields involved, such as frame grabbing, communication bandwidth, digital processing power, and storage capabilities.

At many places, special computer hardware with massively parallel (partially single-bit) computers has been investigated, triggered first by insight into the operation of biological vision systems (for example in the DARPA project on strategic computing [AW&ST 1986; Klass 1985]). This line of development lost momentum with the rapid development of general-purpose microprocessors. Video technology available and driven by a huge market contributed to this trend. At least for developing basic vision algorithms and understanding characteristics of vision paradigms, costly hardware developments were no longer necessary. Today's workstations and PCs in small clusters allow investigating most of the problems in an easy and rapid fashion. Once optimal solutions are known and have to be produced in large numbers, it might be favorable to go back to massively parallel vision hardware.

It is more and more appreciated that feedback signals from higher interpretation levels allow making real-time vision much more efficient; these problems are well suited for today's microprocessors. Within a decade or two, technical vision systems might well approach the performance levels of advanced biological systems. However, they need not necessarily follow the evolutionary steps observed in biology. On the contrary, the following items might provide a starting level for technical systems beyond the one available for most biological systems: (1) high-level 3D shape representations for objects of generic classes; (2) homogeneous coordinate transformations and spatiotemporal (4D) models for motion; (3) action scripts (feed-forward control schemes) for subjects of certain classes; (4) powerful feedback control laws for subjects in certain situations; and (5) knowledge databases, including explicit representations of perceptual and behavioral capabilities for decision making.

As some researchers in biology tend to speculate, the conditions for developing intelligence might be much more favorable if tasks requiring both vision and locomotion control are to be solved. Because biological systems almost always have to act in a dynamic way in 3D space and time, temporal modeling is as common as spatial modeling. The natural sciences and mathematics have derived differential calculus for compact description of motion processes. In this approach, motion state is linked to temporal derivatives of the state variables, the definition of which is that they cannot be changed instantaneously at one moment in time; their change develops over time. The characteristic way in which this takes place constitutes essential knowledge about "the world." Those variables in the description that can be changed instantaneously are called *control variables*. They are the basis for any type of action, whether they are goal oriented or not. They are also the ultimate reason for the possibility of free will in subjects when they are able to derive control applications from mental reasoning with behavioral models (in the fuzzy diction mostly used). It is this well-defined terminology, and the corresponding set of methods from systems dynamics, that allows a unified approach to intelligence by showing the docking locations for AI methods neatly supplementing the engineering approach.

Recursive estimation techniques, as introduced by Kalman (1960), allow optimal estimation of state variables of a well-defined object under noisy conditions when only some output variables can be measured. In monocular vision, direct depth information is always lost; nonetheless can 3D perception be realized when good models for ego motion are available (so-called *motion stereo*). This fact has been the base for the success of the 4D approach to image sequence analysis (Dickmanns and Wünsche 1999).

First, applications in the early 1980s had been coded for systems of custom-made (8-to 32-bit) microprocessors in Fortran for relatively simple tasks. Driving at speeds as high as 60 miles per hour (mph) on an empty highway was demonstrated in 1987 with the 5-ton test vehicle VaMoRs; both lateral (lane keeping) and longitudinal control (adapting speed to road curvature) was done autonomously with half a dozen Intel 8086 microprocessors (Dickmanns and Graefe 1988).

Second-generation vision systems by our group (around 1990 to 1997) were implemented in the programming language C on "transputers" (microprocessors with four parallel links to neighboring ones in a network). Bifocal vision with two cameras, coaxially arranged on gaze control platforms, have been the standard (Schiehlen 1995). The focal length of their lenses differed by a factor of 3 to 4 so that a relatively large field of view was obtained nearby, but in a correspondingly smaller field of view, higher-resolution images could be obtained. In the framework of the PROMETHEUS Project (1987–1994), more complex tasks were solved by using about four dozen microprocessors. For information on road and lane recognition, see Behringer (1996) and Dickmanns and Mysliwetz (1992); for information on observing other vehicles in the front and the rear hemispheres, see Thomanek (1996) and Thomanek, Dickmanns, and Dickmanns (1994). Another one dozen transputers were in use for conventional sensor data processing, situation assessment, and control output (throttle, brakes, and steering of the car as well as gaze control for the active vision system).

At the final demonstration of the EUREKA Project, two Mercedes 500 SEL sedans (VaMP [figure 1] and VITA [the Daimler-Benz twin vehicles]) showed the following performance in public three-lane traffic on Autoroute A1 near Paris: (1) lane keeping at speeds as high as 130 km/h; (2) tracking and relative state estimation of (as many as) six vehicles in each hemisphere in the autonomous vehicle's and two neighboring lanes; (3) transition to convoy driving at speed-dependent distances to the vehicle in front; and (4) lane changing, including decision making about whether the lane changes were safe (Dickmanns 1995; Dickmanns et al. 1994).

In figure 2, a summary is given on the general system concept as derived from ground and air vehicle (Schell and Dickmanns 1994) applications (except the upper right part, which is more recent). The lower left part, including inertial sensing, was developed for air vehicles first and added later to the ground vehicle systems. The lower part essentially works with systems engineering methods (recursive state estimation and automatic control). The vision part is based on computer graphics approaches adapted to recursive estimation (object-oriented coordinate systems for shape representation, homogeneous coordinate transformations [HCTs] for linking points in the image to object representation in 3D space, and simulation models for object motion). It should be noted that HCTs allow for easy implementation of spatial scaling by just one number, the scale factor. The series of arrows, shown directed upward in the center left of the figure, is intended to indicate that object data come from a distributed processor system. In the system developed for our industrial partner, even diverse makes with arbitrary programming methods could be docked here. The dynamic database (DDB as it was called at that time, now dynamic object database [DOB] shown by the horizontal spatial bar, left) functioned as a distribution platform to all clients; in a transputer network, this function could be performed by any processing node. The upper part was not well developed; elements necessary for the few functions to be demonstrated in each test were programmed in the most efficient way (because of time pressure resulting from the tight project plan). On the basis of this experience, a third-generation system was to be developed in an object-oriented (C++) programming paradigm using commercial off-the-shelf PC systems and frame-grabbing devices, starting in 1997. It was designed for flexible use, and for easy add-ing on of components as they become available on the market. In addition, system autonomy and self-monitoring were to achieve a new level. The structure of several software packages for classes of objects to be perceived (these might be called "agents for perception") had proven to be efficient. These packages combine specific feature-extraction algorithms; generic shape models with likely aspect conditions for hypothesis generation; dynamic models for object motion; and recursive estimation algorithms, allowing the adjustment of both shape parameters and state variables for single objects. *N* of these objects can be instantiated in parallel. Communication is done through a dynamic object database (DOB) updated 25 times a second (video rate). More details on the system can be found in Gregor et al. (2001); Pellkofer 2003; Pellkofer, Lützeler, and Dickmanns (2001); Rillings and Broggi (2000) and Rieder (2000). On the higher system levels, more flexibility and easy adaptability in mission performance, situation assessment, and behavior decision were to be achieved.

Starting from activities of a physical unit (the body with its functioning subsystems) instead of beginning with abstract definitions of terms seems to be less artificial because the notion of verbs results in a natural way. This start is achieved by introducing symbols for activities the system already can perform because of the successful engineering of bodily devices. This abstraction of symbols from available behavioral capabilities provides the symbol

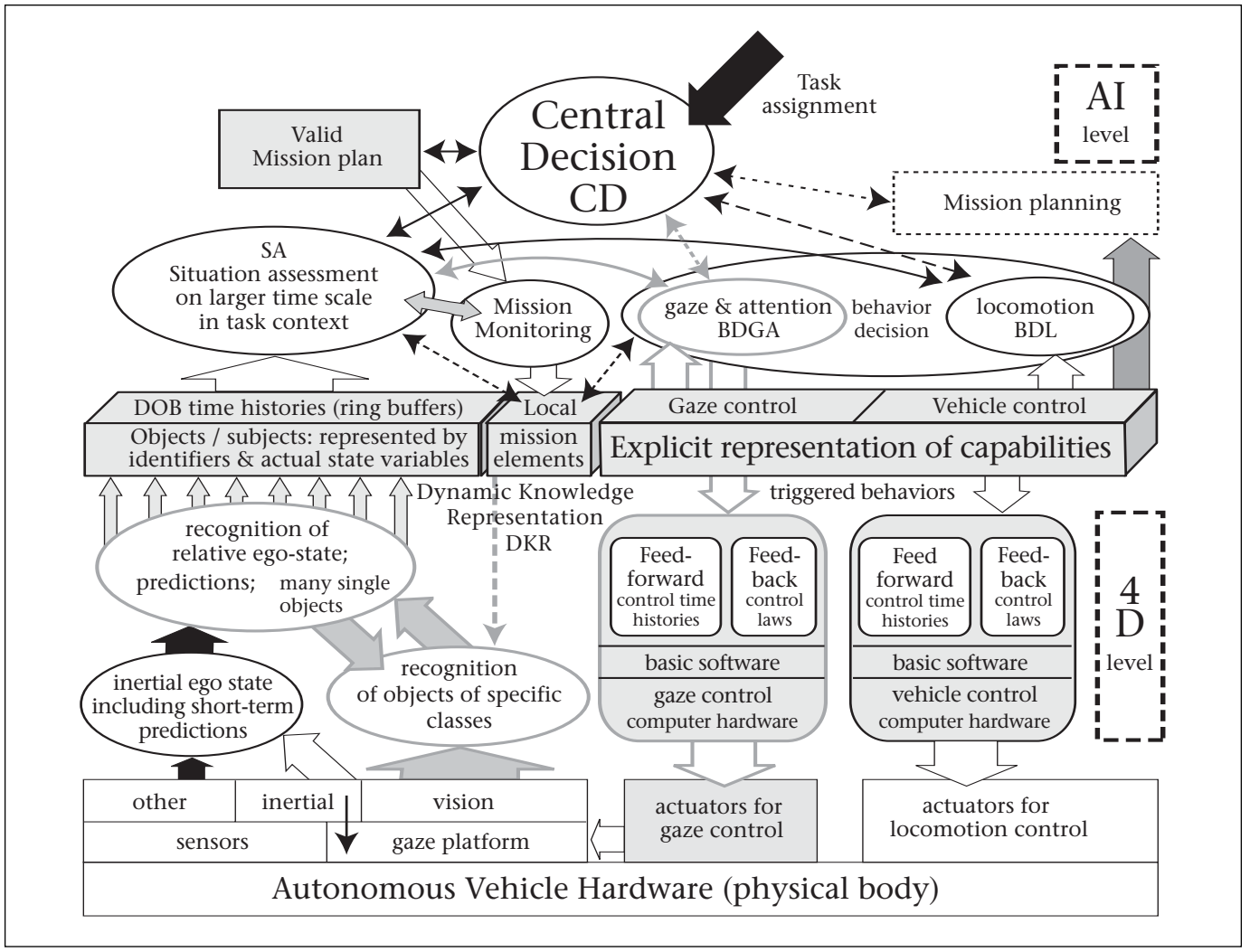

Figure 2. Dynamic Vision System with Active Gaze Stabilization and Control for Vehicle Guidance under Strongly Perturbed Conditions.

Control engineering methods predominate in the lower part (below knowledge representation bar) and AI-type methods in the upper part.

grounding often complained to be missing in AI. It is interesting to note that Damasio (1999) advocates a similar shift for understanding human consciousness.

The rest of the article discusses the new type of expectation-based, multifocal, saccadic (EMS) vision system and how this system has led to the modified concept of technical intelligence. EMS vision was developed over the last six years with an effort of about 35 person-years. During its development, it became apparent that explicit representation of behavioral capabilities was desirable for achieving a higher level of system autonomy. This feature is not found in other system architectures such as discussed by Arkin (1998) and Brooks and Stein (1994). Albus and Meystel (2001) use multiple representations on different grid scales; however, their approach is missing the nice scalability effect of homogeneous coordinates in space and does not always provide the highest resolution along the time axis. In the existing implementation at UBM, explicit representation of capabilities on the mental level has been fully realized for gaze and locomotion control only. Several processes in planning and perception have yet to be adapted to this new standard. The right center part of figure 2 shows the functions implemented.

Dynamic versus Computational Vision

Contrary to the general approach to real-time vision adopted by the computer science and AI communities, dynamic vision right from the

beginning does embed each single image into a time history of motion in 3D space for which certain continuity conditions are assumed to hold. Motion processes of objects are modeled, rather than objects, which later on are subjected to motion. For initialization, this approach might seem impossible because one has to start with the first image anyway. However, this argument is not quite true. The difference lies in when to start working on the next images of the sequence and how to come up with object and motion hypotheses based on a short temporal sequence. Instead of exhausting computing power with possible feature combinations in a single image for deriving an object hypothesis (known as *combinatorial explosion*), recognizable characteristic groupings of simple features are tracked over time. Several hypotheses of objects under certain aspect conditions can be started in parallel. It is in this subtask of hypothesis generation that optical (feature) flow might help; for tracking, it loses significance.

The temporal embedding in a hypothesized continuous motion then allows much more efficient pruning of poor hypotheses than when you proceed without it. These motion models have to be in 3D space (and not in the image plane) because only there are the Newtonian second-order models for motion of massive objects valid in each degree of freedom. Because of the second order (differential relationship) of these models, speed components in all six degrees of freedom occur quite naturally in the setup and will be estimated by the least squares fit of features according to the recursive estimation scheme underlying the extended Kalman filter. Prediction error feedback according to perspective mapping models is expected to converge for one object shape and motion hypothesis. There are several big advantages to this approach:

First, state-prediction and forward-perspective mapping allow tuning of the parameters in the feature-extraction algorithms; this tuning might improve the efficiency of object recognition by orders of magnitude! It also allows implementing the idea of Gestalt in technical vision systems, as discussed by psychologists for human vision.

Second, in parallel to the nominal forward-perspective mapping, additional mappings are computed for slight variations in each single (hypothesized) value of the shape parameters and state components involved. From these variations, the Jacobian matrix of first-order derivatives on how feature values in the image plane depend on the unknown parameters and states is determined. Note that because the model is in 3D space, there is very rich information available about how to adjust parameters to obtain the best-possible spatial fit. (If you have experienced the difference between looking at a monitor display of a complex spatial structure of connected rods when it is stationary and when it is in motion, you immediately appreciate the advantages.)

Third, checking the elements in each row and column of the Jacobian matrix in conjunction allows decision making about whether a state or shape parameter should be updated or whether continuing the evaluation of a feature does make sense. By making these decisions carefully, many of the alleged cases of poor performance of Kalman filtering can be eliminated. If confusion of features might occur, it is better to renounce using them at all and instead look for other features for which correspondence can be established more easily.

Fourth, characteristic parameters of stochastic noise in the dynamic and measurement processes involved can be used for tuning the filter in an optimal way; one can differentiate how much trust one puts in the dynamic model or the measurement model. This balance might even be adjusted dynamically, depending on the situation encountered (for example, lighting and aspect conditions and vibration levels).

Fifth, the (physically meaningful) states of the processes observed can be used directly for motion control. In systems dynamics, the best one can do in linear feedback control is to link corrective control output directly to all state variables involved (see, for example, Kailath [1980]). This linear feedback allows shifting eigenvalues of the motion process arbitrarily (as long as the linearity conditions are valid).

All these items illustrate that closed-loop perception and action cycles do have advantages. In the 4D approach, these advantages have has been exploited since 1988 (Wuensche 1988), including active gaze control since 1984 (Mysliwetz 1990). Visual perception for landing approaches under gusty wind conditions could not be performed successfully without inertial measurements for counteracting fast perturbations (Schell 1992). This feature was also adopted for ground vehicles (Werner 1997).

When both a large field of view close up and a good resolution farther away in a limited region are requested (such as for high-speed driving and occasional stop-and-go driving in traffic jams or tight maneuvering on networks of minor roads), peripheral-foveal differentiation in visual imaging is efficient. Figure 3 shows the aperture angles required for mapping the same circular area (30 meter diameter) into a full image from different ranges.

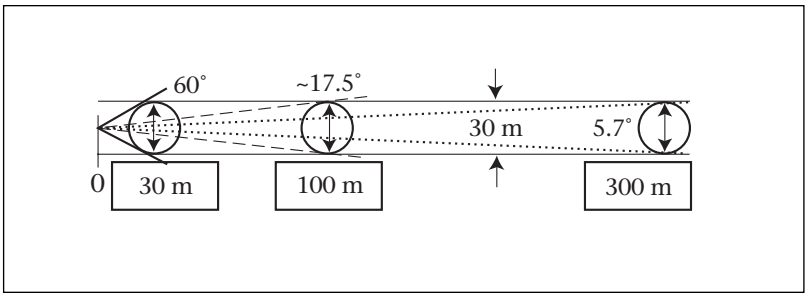

Figure 3. Scaling of Lenses Required for Mapping the Same Area at Different Ranges into an Image of Given Size.

With approximately 760 pixels to an image row, at each of the distances given, about 4 centimeters normal to the optical axis are mapped into a single pixel. The wide-angle lens (60 degrees) would map 40 centimeters at a distance of 300 meters onto a single pixel. In this case, saccadic viewing direction control is advantageous because it trades about two orders of magnitude in data flow requirement for short delay times until the high-resolution imager has been turned to point to the area of special interest. These two-axis pointing devices do have their own dynamics, and image evaluation has to be interrupted during phases of fast rotational motion (blur effects); nonetheless, an approximate representation of the motion processes under observation has to be available. This requirement is achieved by a standard component of the recursive estimation process predicting the states with the dynamic models validated to the present point in time. This very powerful feature has to rely on spatiotemporal models for handling all transitions properly, which is way beyond what is usually termed *computational vision*. *Dynamic vision* is the proper term for this approach, which takes many different aspects of dynamic situations into account for perception and control of motion.

Requirements for a General Sense of Vision for Autonomous Vehicles

The yardstick for judging the performance level of an autonomous vehicle is the human one. Therefore, the vehicle's sense of vision should at least try to achieve many (if not all) of the average human capabilities.

The most important aspects of human vision for vehicle control are as follows: First, a wide field of view (f.o.v.) nearby for regions greater than 180 degrees, as with humans, is not required, but greater than about 100 degrees is desirable. Second, a small subarea of this f.o.v. needs to have much higher spatial resolution, such that at several hundred meters distance, an object of the size of about 0.1 meter in its smaller dimension should be detectable. Third, the pointing direction should be quickly controllable to track fast-moving objects and reduce motion blur, which is also required for compensating disturbances on the autonomous vehicle body. In case some new, interesting features are discovered in the wide f.o.v., the top speed of the saccades for shifting viewing direction should limit delay times to a few tenths of a second (as with the human eye). Fourth, the capability of color vision should be at least in part of the f.o.v. Fifth, there should be a high dynamic range with respect to light intensity distribution in the f.o.v. (in humans to 120 decibels). Sixth, for maneuvering in narrow spaces or reacting to other vehicles maneuvering right next to oneself (for example, lane changes), stereo vision is advantageous. Stereo ranges up to 10 meters might be sufficient for these applications. Under the flat-ground assumption, good monocular range estimation (accuracies of a few percent) to vehicles has been achieved reliably as long as the point where the vehicles touch the ground can be seen. Seventh, degradation of the vision process should be gradual if environmental conditions deteriorate (dusk, rain, snowfall, fog, and so on). Multiple cameras with different photometric and perspective mapping properties can help achieve robustness against changing environmental conditions.

Based on these considerations, an eye for road vehicles to be assembled from standard cameras and lenses was designed in 1995 for quick testing of the approach. It was dubbed the multifocal, active-reactive vehicle eye (MarVEye) and is not considered a prototype for a product, just a research tool. Figure 4 shows some of the properties. It allows investigating and proving all basic requirements for vision system design; further development steps toward a product seem justified only in case it appears to be promising from a price-performance point of view.

Beside these specifications regarding sensor properties, equally important features regarding the knowledge-based interpretation process should be met. It relies on schemata for classes and relationships between its members.

Objects of certain classes are known generically, and visual interpretation relies on tapping this source of knowledge, which encompasses both 3D shape and likely visual appearance as well as typical motion patterns over time. In the generic models, some adapt-

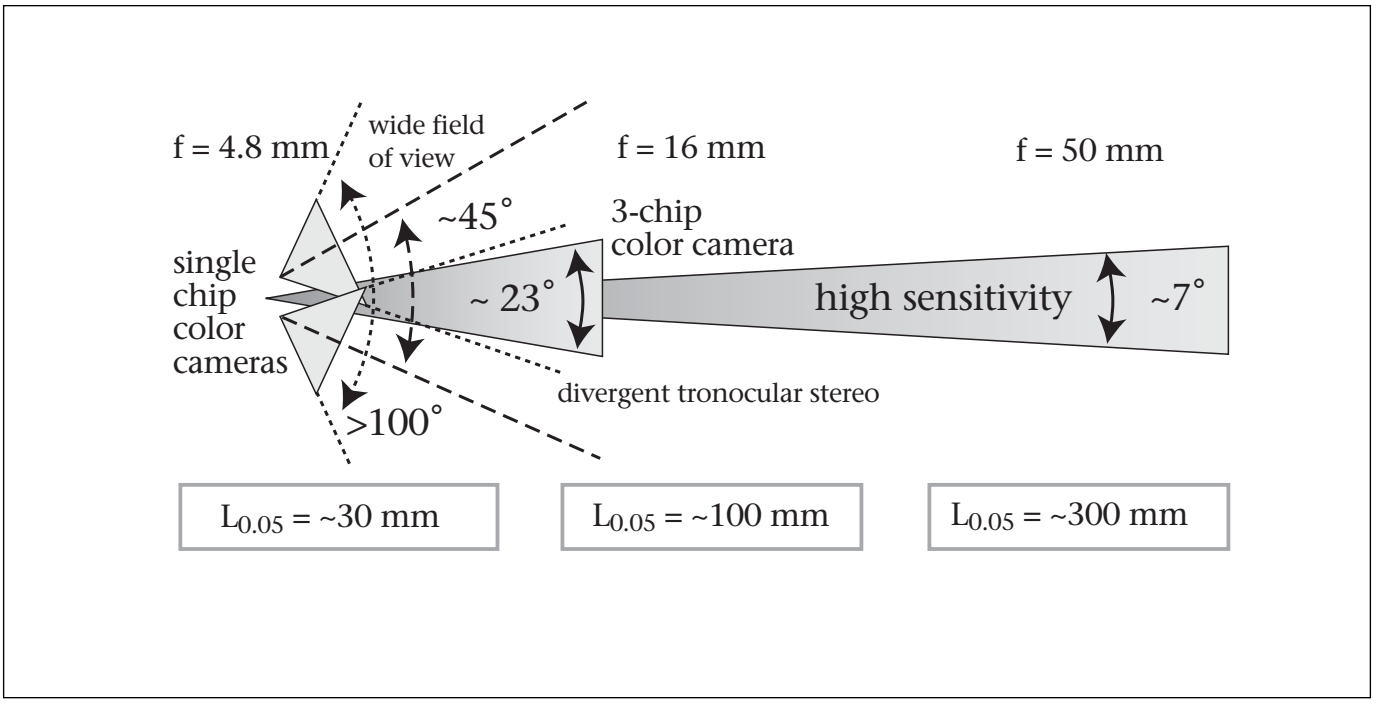

Figure 4. Visual Ranges of the MARVEYE Concept with Four Cameras.

able parameters, the actual aspect conditions, and the motion state variables have to be derived from actual observations.

Objects of certain classes do not follow simple laws of motion. Subjects capable of initiating maneuvers (actions) on their own form a wide variety of different classes and subclasses. Each subclass again can have a wide variety of members: All animals capable of picking up information from the environment by sensors and making decisions, how to control parts of their body or the whole body, are members of these subclasses. Also, other autonomous vehicles and vehicles with human drivers inside form specific subclasses.

Most of our knowledge about the world (the mesoscale world of everyday life under discussion here) is geared to classes of objects and subjects as well as to individual members of these classes. The arrangement of individual members of these classes in a specific environment and a task context is called a *situation.*

Three areas of knowledge can be separated on a fine-to-coarse scale in space and time (table 1): (1) the feature/(single object) level ["here and now" for the actual point in space and time, upper left corner]; (2) the generic object (subject) class level, combining all possible properties of class members for further reference and communication (central row); and (3) the situation level with symbolic representations of all single objects (subjects) relevant for decision making. In addition to their relative spatial arrangement and motion state, the maneuver activity they are actually involved in and the relative state within the maneuver should also be available (lower right corner in table 1).

On the feature level, there are two (partially) separate regimes for knowledge representation: First, what are the most reliable and robust features for object detection? For initialization, it is important to have algorithms available, allowing an early transition from collections of features observed over a few images of a sequence (spanning a few tenths of a second at most) to hypotheses of objects in 3D space and time. Three-dimensional shape and relative aspect conditions also have to be hypothesized in conjunction with the proper motion components and the underlying dynamic model. Second, for continued tracking with an established hypothesis, the challenge is usually much more simple because temporal predictions can be made, allowing work with special feature-extraction methods and parameters adapted to the actual situation. This adaptation was the foundation for the successes of the 4D approach in the early and mid-1980s.

For detecting, recognizing, and tracking objects of several classes, specific methods have been developed and proven in real-world applications. In the long run, these special perceptual capabilities should be reflected in the knowledge base of the system by explicit representations, taking their characteristics into ac-

Column no range in time ↓ in space →	1 point in time	2 local time integral	3 extended time integrals	4
point in 3D space	"here & now" measurements	transition matrices for local objects	--------	features
local differential environment	edge positions, regions, angles, curvatures	change of feature parameters	feature history	features
local space integrals	object states, feature distribution, shape	state transitions, new aspect condit. "central hub"	state predictions, object state histories	objects
maneuver space	local situation	1 step prediction of situation (usually not done)	multiple step predictions; monitoring; mission performance	situations
mission space	actual global situation	--------		situations

Table 1. Multiple-Scale Recognition and Tracking of Features, Objects, and Situations in Expectation-Based, Multifocal, Saccidic (EMS) Vision.

count for the planning phase. These characteristics specify regions of interest for each object tracked. This feature allows flexible activation of the gaze platform carrying several cameras so that the information increase by the image stream from the set of cameras is optimized. These decisions are made by a process on the upper system level dubbed behavior decision for gaze and attention (BDGA in upper right of figure 2 [Pellkofer 2003; Pellkofer, Lützeler, and Dickmanns 2003).

On the object level, all relevant properties and actual states of all objects and subjects instantiated are collected for the discrete point in time now and possibly for a certain set of recent time points (recent state history).

For subjects, the maneuvers actually performed (and the intentions probably behind them) are also inferred and represented in special slots of the knowledge base component dedicated to each subject. The situation-assessment level (see later discussion) provides these data for subjects by analyzing state-time histories and combining the results with background knowledge on maneuvers spanning certain time scales. For vehicles of the same class as oneself, all perceptual and behavioral capabilities available for oneself are assumed to be valid for the other vehicle, too, allowing fast in-advance simulations of future behaviors, which are then exploited on the situation level.

For members of other classes (parameterized), stereotypical behaviors have to be stored or learned over time (this capability still has to be implemented). This animation capability on the subject level is considered essential for deeply understanding what is happening around oneself and arriving at good decisions with respect to safety and mission performance.

On the situation level (lower right corner in table 1), the distinction between relevant and irrelevant objects-subjects has to be taken, depending on the mission to be performed. Then, the situation has to be assessed, further taking a wider time horizon into account, and decisions for safely achieving the goals of the autonomous vehicle mission have to be taken. Two separate levels of decision making have been implemented: (1) optimizing perception (mentioned earlier as BDGA (Pellkofer 2003; Pellkofer, Lützeler, and Dickmanns 2001) and (2) controlling a vehicle (BDL [behavior decision for locomotion] (Maurer 2000; Siedersberger 2003)) or monitoring a driver. During mission planning, the overall autonomous mission is broken down into mission elements that can be performed in the same control mode. Either feed-forward control with parameterized time histories or feedback control depending on deviations from the ideal trajectory is applied for achieving the subgoal of each mission element. Monitoring of mission progress and the environmental conditions can lead to events calling for replanning or local deviations from the established plan. Safety aspects for the autonomous vehicle body and other participants always have a high priority in behavior decision. Because of unforseeable events, a time horizon of only several seconds or a small fraction of a minute is usually possible in ground vehicle guidance. Viewing direction and visual attention have to be controlled so that sudden surprises are minimized. When special maneuvers such as lane changing or turning onto a crossroad are intended, visual search patterns are followed to avoid dangerous situations (defensive-style driving).

A replanning capability should be available as a separate unit accessible from the "central decision," part of the direct overall system (see figure 2, top right); map information on road networks is needed, together with quality criteria, for selecting route sequences (Gregor 2002).

The system has to be able to detect, track, and understand the spatiotemporal relationship to all objects and subjects relevant for the task at hand. To not overload the system, tasks are assumed to belong to certain domains for which the corresponding knowledge bases on objects, subjects, and possible situations have been provided. Confinement to these domains

keeps knowledge bases more manageable. Learning capabilities in the 4D framework are a straightforward extension but still have to be added. It should have become clear that in the dynamic vision approach, closed-loop perception and action cycles are of basic importance, also for learning improved or new behaviors.

In the following section, basic elements of the EMS vision system are described first; then, their integration into the overall cognitive system architecture are discussed.

System Design

The system is object-oriented in two ways: First, the object-oriented–programming paradigm as developed in computer science has been adopted in the form of the C++ language (grouping of data and methods relevant for handling of programming objects, exploiting class hierarchies). Second, as mentioned earlier, physical objects in 3D space and time with their dynamic properties, and even behavioral capabilities of subjects, are the core subject of representation.[2] Space does not allow a detailed discussion of the various class hierarchies of relevance.

Basic Concept

According to the data and the specific knowledge levels to be handled, three separate levels of visual perception can be distinguished, as indicated in figure 5 (right). Bottom-up feature extraction over the entire area of interest gives the first indications of regions of special interest. Then, an early transition to objects in 3D space and time is made. Symbols for instantiated objects form the central representational element (blocks 3 to 5 in figure 5). Storage is allocated for their position in the scene tree (to be discussed later) and the actual best estimates of their yet to be adapted shape parameters and state variables (initially, the best guesses).

The corresponding slots are filled by specialized computing processes for detecting, recognizing, and tracking members of their class from feature collections in a short sequence of images (agents for visual perception of class members); they are specified by the functions coded. The generic models used on level 2 (blocks 3 and 4) consist of one subsystem for 3D shape description, with special emphasis on highly visible features. A knowledge component for speeding up processing is the aspect graph designating classes of feature distributions in perspective images. The second subsystem represents temporal continuity conditions based on dynamic models of nth order for object motion in space. In a video data stream consisting of regularly sampled snapshots, discrete dynamic models in the form of (n x n) transition matrices for the actual sampling period are sufficient. For rigid objects with 3 translational and 3 rotational degrees of freedom, these are 12×12 matrices because of the second-order nature of Newtonian mechanics. If the motion components are uncoupled, only 6 (2×2) blocks along the diagonal are nonzero. For practical purposes in noise-corrupted environments, usually the latter model is chosen; however, with a first-order noise model added in each degree of freedom, we have 6 blocks of (3×3) transition matrices.

By proper tuning of the noise-dependent parameters in the recursive estimation process, in general, good results can be expected. If several different cameras are being used for recognizing the same object, this object is represented multiple times, initially with the specific measurement model in the scene tree. A supervisor module then has to fuse the model data or select the one to be declared the valid one for all measurement models. In the case of several sensors collecting data on the same object instantiated, one Jacobian matrix has to be computed for each object-sensor pair. These processes run at video rate. They implement all the knowledge on the feature level as well as part of the knowledge on the object level. This implies basic generic shape, the possible range of values for parameters, the most likely aspect conditions and their influence on feature distribution, and dynamic models for temporal embedding. For subjects with potential control applications, their likely value settings also have to be guessed and iterated.

As a result, describing elements of higher-level percepts are generated and stored (3D shape and photometric parameters as well as 4D state variables [including 3D velocity components]). They require several orders of magnitude less communication bandwidth than the raw image data from which they were derived. Although the data stream from 2 black-and-white and 1 (3-chip) color camera is in the 50 megabyte a second range, the output data rate is in the kilobyte range for each video cycle. These percepts (see arrows leaving the 4D level upward in figure 2 [center left] and figure 5 [into block 5]) are the basis for further evaluations on the upper system level. Note that once an instance of an object has been generated and measurement data are not available for one or a few cycles, the system might nonetheless continue its regular mission, based on predictions according to the temporal model available. The actual data in the corresponding object slots will then be the predicted val-

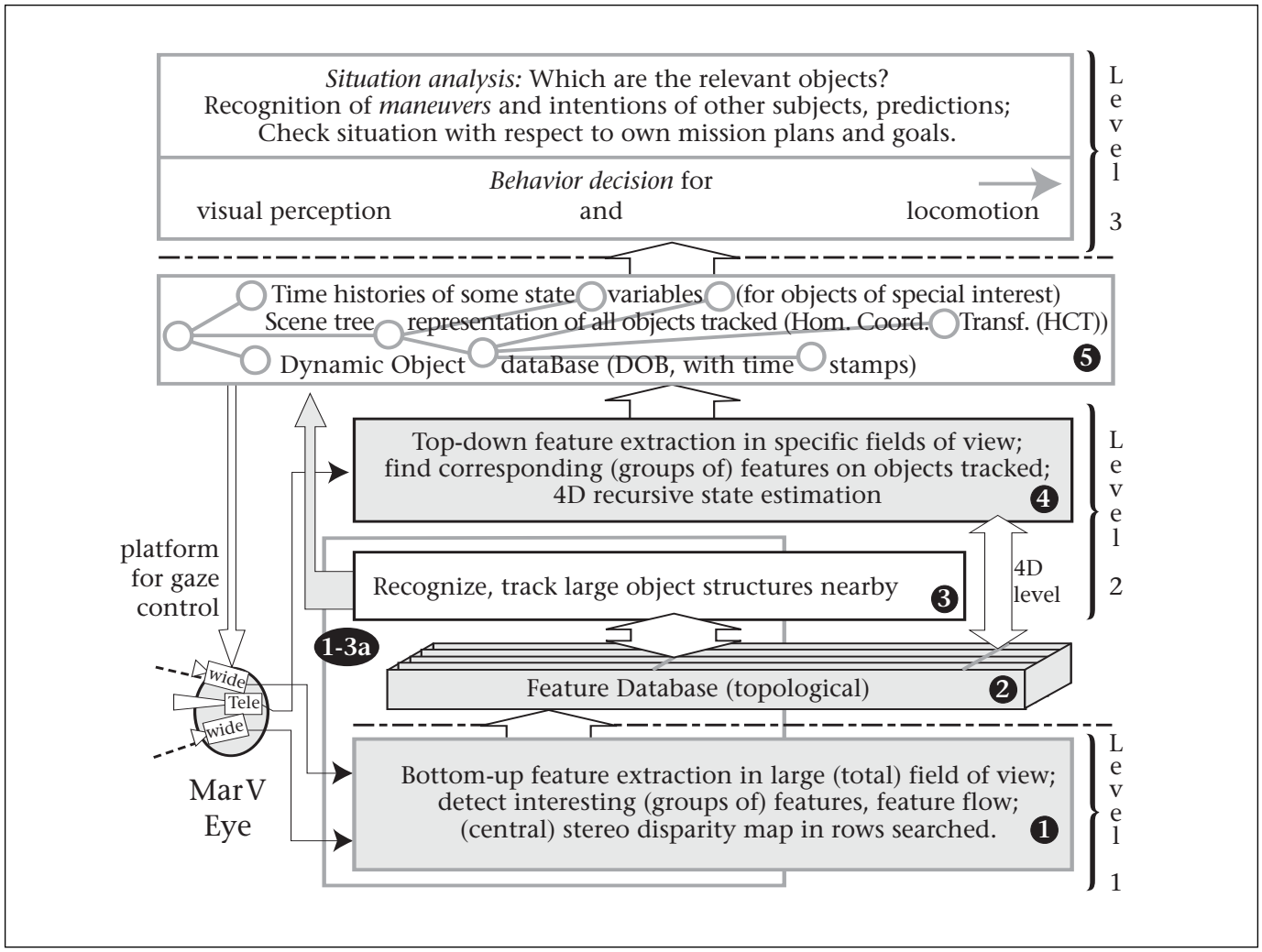

Figure 5. Three Distinct Levels of Visual Perception.

1. Bottom-up detection of standard features. 2. Specific features for individual objects under certain 3D aspect conditions. 3. Time histories of state variables for recognizing maneuvers and mission elements (trajectory and intent recognition at larger spatial and temporal scales).

ues according to the model instantiated. Trust in these predicted data will decay over time; therefore, these periods without new visual input are steadily monitored. In a system relying on saccadic vision, these periods occur regularly when the images are blurred because of high rotational speeds for shifting the visual area of high resolution to a new (wide-angle) image region of special interest. This fact clearly indicates that saccadic vision cannot be implemented without temporal models and that the 4D approach lends itself to saccadic vision quite naturally.

In a methodologically clean architecture, the blocks numbered 1 to 3 in the dark circular regions in figure 5 should be separated completely. Because of hardware constraints in frame grabbing and communication bandwidth, they have been lumped together on one computer in actual implementation. The elliptical dark field on the shaded rectangular area at the left-hand side (1–3a) indicates that object recognition based on wide-angle images is performed directly with the corresponding features by the same processors for bottom-up feature detection in the actual implementation. In the long run, blocks 1 and 2 are good candidates for special hardware and software on plug-in processor boards for a PC.

The 4D level (2) for image processing (labeled 3 and 4 in the dark circles) requires a completely different knowledge base than level 1. It is applied to (usually quite small) special image regions discovered in the feature database by interesting groups of feature collections. For these feature sets, object hypotheses in 3D space and time are generated. These 4D object hypotheses with specific assumptions

for range and other aspect conditions allow predictions of additional features to be detectable with specially tuned operators (types and parameters), thereby realizing the idea of Gestalt recognition. Several hypotheses can be put up in parallel, if computing resources allow is; invalid ones will disappear rapidly over time because the Jacobian matrices allow efficient use of computational resources.

Determining the angular ego state from inertial measurements allows a reduction of search areas in the images, especially in the tele-images. The dynamic object database (DOB, block 5 in figure 5) serves the purpose of data distribution; copies of this DOB are sent to each computer in the distributed system. The DOB isolates the higher-system levels from high visual and inertial data rates to be handled in real time on the lower levels. Here, only object identifiers and best estimates of state variables (in the systems dynamics sense), control variables, and model parameters are stored.

By looking at time series of these variables (of limited extension), the most interesting objects of the autonomous vehicle and their likely future trajectories have to be found for decision making. For this temporally deeper understanding of movements, the concept of maneuvers and mission elements for achieving some goals of the subjects in control of these vehicles has to be introduced. Quite naturally, this requirement leads to explicit representation of behavioral capabilities of subjects for characterizing their choices in decision making. These subjects might activate gaze control and select the mode of visual perception and locomotion control. Realization of these behavioral activities in the physical world is usually performed on special hardware (real-time processors) close to the physical device (see figure 2, lower right). Control engineering methods such as parameterized stereotypical feedforward control-time histories or (state- or output-) feedback control schemes are applied here. However, for achieving autonomous capabilities, an abstract (quasistatic) representation of these control modes and the transitions possible between them has been implemented recently on the upper system levels by AI methods according to extended state charts (Harel 1987; Maurer 2000a; Siedersberger 2003). This procedure allows more flexible behavior decisions, based also on maneuvers of other subjects supposedly under execution. In a defensive style of driving, this recognition of maneuvers is part of situation assessment. For more details on behavior decision for gaze and attention (BDGA), see Pellkofer (2003) and Pellkofer, Lützeler, and Dickmanns (2001).

The upper part of figure 5 is specific to situation analysis and behavior decision. First, trajectories of objects (proper) can be predicted in the future to see whether dangerous situations can develop, taking the motion plans of the autonomous vehicle into account. For this reason, not just the last best estimate of state variables is stored, but on request, a time history of a certain length can also be buffered. Newly initiated movements of subjects (maneuver onsets) can be detected early (Caveney, Dickmanns, and Hedrik 2003). A *maneuver* is defined as a standard control time history resulting in a desired state transition over a finite period of time. For example, if a car slightly in front in the neighboring lane starts moving toward the lane marking that separates its lane from the one of the autonomous vehicle, the initiation of a lane-change maneuver can be concluded. In this maneuver, a lateral offset of one lane width is intended within a few seconds. This situational aspect of a started maneuver is stored by a proper symbol in an additional slot; in a distributed realization, pointers can link this slot to the other object properties.

A more advanced application of this technique is to observe an individual subject over time and keep statistics on its way of behaving. For example, if this subject tends to frequently close in on the vehicle in front at unreasonably small values, the subject might be classified as aggressive, and one should try to keep a safe distance from him or her. A great deal of room is left for additional extensions in storing knowledge about behavioral properties of other subjects and learning to know them not just as class members but as individuals.

Arranging of Objects in a Scene Tree

To classify the potentially large decision space for an autonomous system into a manageable set of groups, the notion of *situation* has been introduced by humans. The underlying assumption is that for given situations, good rules for arriving at reasonable behavioral decisions can be formulated. A situation is not just an arrangement of objects in the vicinity but depends on the intentions of the autonomous vehicle and those of other subjects. To represent a situation, both the geometric distribution of objects-subjects with their velocity components, their intended actions, and the autonomous vehicle's actions have to be taken into account.

The relative dynamic state of all objects-subjects involved is the basic property for many aspects of a situation; therefore, representing this state is the central task. In the 4D approach, Dirk Dickmanns (1997) introduced the

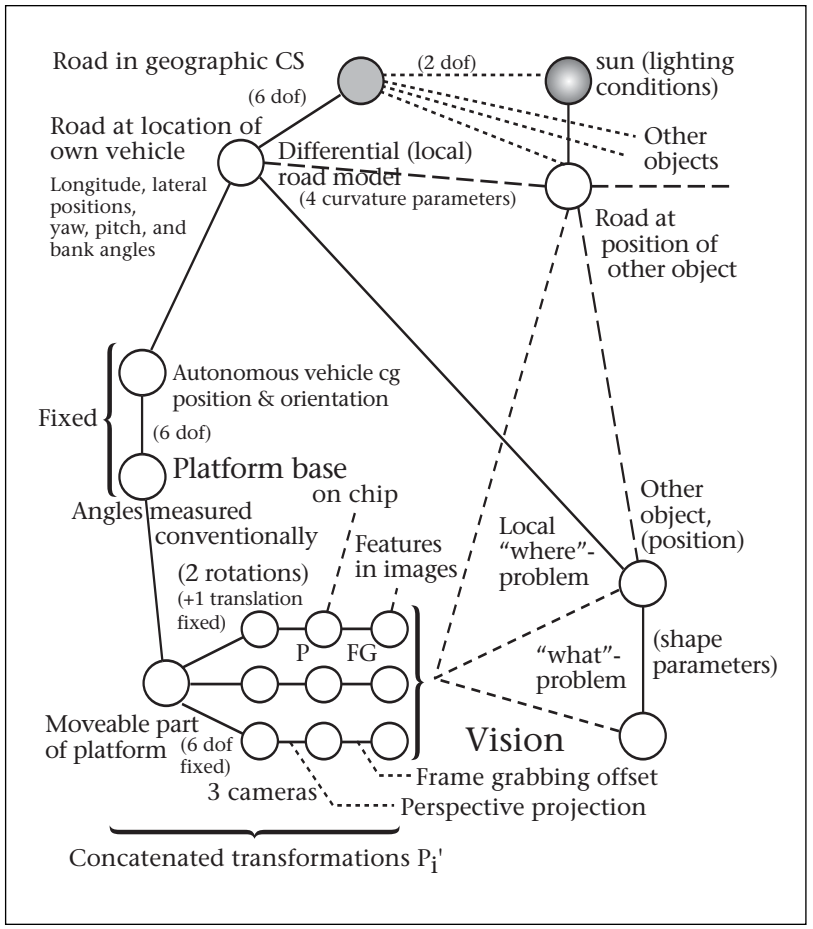

Figure 6. Scene Tree for Visual Interpretation in Expectation-Based, Multifocal, Saccadic Vision Showing the Transformations from the Object in Space to Features in the Image.

so-called *scene tree* for achieving this goal in an efficient and flexible way. In figure 6, an example is given for a road scene with other vehicles and shadows. As in computer graphics, each node represents an object, a separately movable subobject, or a virtual object such as a coordinate system or the frame grabber. Its homogeneous coordinate transformations (HCTs), using 4×4 matrices and represented by graph edges, are exactly as used in computer graphics; however, in computer vision, many of the variables entering the transformation matrices between two elements are the unknowns of the problem and have to be iterated separately. By postponing concatenation and forming the derivatives of each matrix with respect to the variables involved, the elements of the first-order overall derivative matrix linking changes of feature positions in the image to changes in the generic shape parameters or state variables can easily be computed. However, this process involves a heavy computational load. Both numeric differencing and analytic differentiation of each single HCT, followed by the concatenations required, have been tried and work well. From systematic changes of the full set of unknown parameters and state variables, the Jacobian matrix rich in first-order information on all spatial relationships is obtained. Under ego motion, it allows motion stereo interpretation with a single camera.

The lower left part of figure 6 shows all relevant relationships within the vehicle carrying the three cameras. The perspective projection step *P* models the physics of light rays between objects in the outside world (road and other vehicles) and the sensor elements on the imager chip. When transfer into computer memory is performed by the frame grabber, additional 2D shifts can occur, represented by transformation *FG*. In the upper part of figure 6, the extended road is represented, both with its nearby region yielding information on where the autonomous vehicle is relative to it and with its more distant parts connected to the local road by a generic curvature model (Dickmanns and Zapp 1986).

When absolute geographic information is needed, for example, for navigating with maps or determining orientation from angles to the Sun, corresponding elements of planetary geometry have to be available. It is interesting to note that just five HCTs and a few astronomical parameters allow representing the full set of possible shadow angles for any point on a spherical Earth model at any arbitrary time. For a given point on Earth at a given time, of course, only two angles describe the direction to the Sun (azimuth and elevation yielding actual, local shadow angles).

Other objects can be observed relative to the autonomous vehicle's position or relative to their local road segment (or some other neighboring object). The problem of where another object is located essentially results from summing positions of features belonging to this object. For understanding the type of object, essentially differencing of feature positions and shape interpretation are required. In biological systems, these computations are done in separate regions of the brain, which is why the naming of "where" and "what" problems has been introduced and adopted here (right part of figure 6). The number of degrees of freedom of relative position and orientation modeled is shown at each edge of the scene tree; 6 degrees of freedom (3 translations and 3 rotations) is the maximum available for a rigid object in 3D space. (More details are given in Dickmanns [1997] and Dickmanns and Wuensche [1999].)

Representation of Capabilities, Mission Requirements

Useful autonomous systems must be able to accept tasks and solve them on their own. For this purpose, they must have a set of perceptual and behavioral capabilities at their disposal. Flexible autonomous systems are able to recognize situations and select behavioral components that allow efficient mission performance. All these activities require goal-oriented control actuation. Subjects can be grouped to classes according to their perceptual and behavioral capabilities (among other characteristics such as shape), so that the scheme developed and discussed later can be used for characterizing classes of subjects. This scheme is a recent development; its explicit representation in the C++ code of EMS vision has been done for gaze and locomotion behavior (figure 7).

Two classes of behavioral capabilities in motion control are of special importance: (1) so-called *maneuvers*, in which special control time histories lead to a desired state transition in a finite (usually short) amount of time, and (2) *regulating activities*, realized by some kind of feedback control, in which a certain relative state is to be achieved and/or maintained; the regulating activities can be extended over long periods of time (like road running). Both kinds of control application are extensively studied in control engineering literature: maneuvers in connection with optimal control (see, for example, Bryson and Ho [1975]) and regulating activities in the form of linear feedback control (Kailath [1980] only one of many textbooks). Besides vehicle control for locomotion, both types are also applied in active vision for gaze control. Saccades and search patterns are realized with prespecified control time histories (topic 1 earlier), and view fixation on moving objects and inertial stabilization are achieved with proper feedback of visual or inertial signals (topic 2).

Complex control systems are realized on distributed computer hardware with specific processors for vehicle- and gaze-control actuation on the one side and for knowledge processing on the other. The same control output (maneuver, regulatory behavior) can be represented internally in different forms, adapted to the subtasks to be performed in each unit. For decision making, it is not required to know all data of the actual control output and the reactions necessary to counteract random disturbances as long as the system can trust that perturbations can be handled. However, it has to know which state transitions can be achieved by which maneuvers, possibly with favorable parameters depending on the situation (speed of vehicle, friction coefficients to the road, and so on). This knowledge is quasistatic and is typically handled by AI methods (for example, state charts [Harel 1987]).

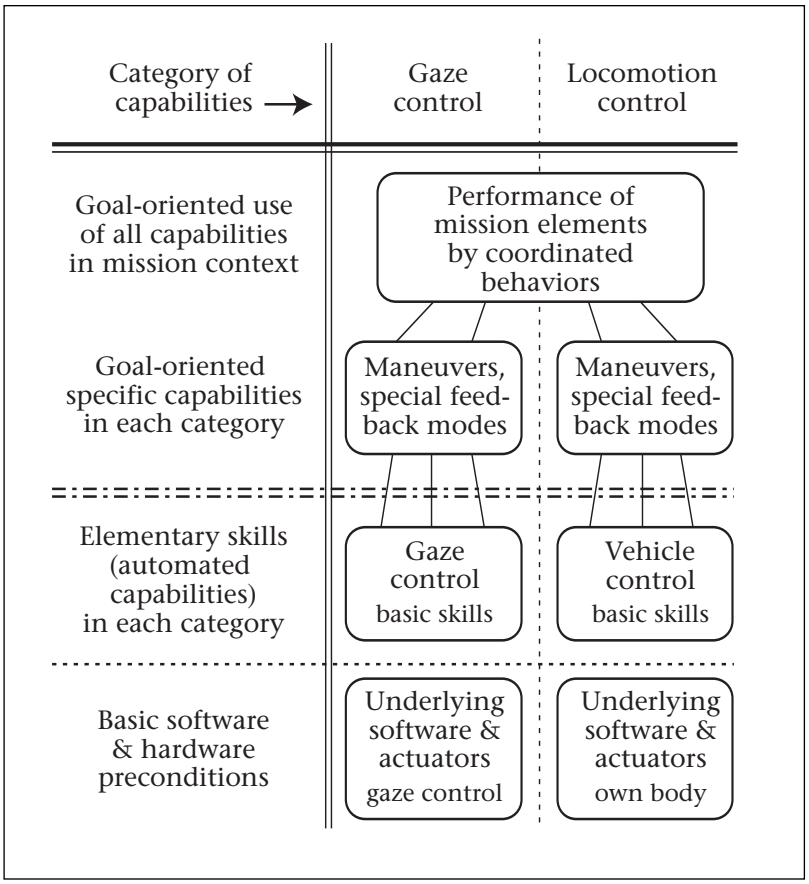

Figure 7. Capability Matrix as Represented in Explicit Form in the EMS Vision System for Characterizing Subjects by Their Capabilities in Different Categories of Action.

Actual implementations of maneuvers and feedback control are achieved by tapping knowledge stored in software on the processors directly actuating the control elements (lower layers in figures 2, 7, and 8 right). For maneuvers, numerically or analytically coded time histories can be activated with proper parameters; the nominal trajectories resulting can also be computed and taken as reference for superimposed feedback loops dealing with unpredictable perturbations. For regulatory tasks, only the commanded (set) values of the controlled variables have to be specified. The feedback control laws usually consist of matrices of feedback gains and a link to the proper state or reference variables. These feedback loops can be realized with short time delays because measurement data are directly fed into these processors. This dual representation with different schemes on the level close to hardware (with control engineering–type methods)

and on the mental processing level (with quasistatic knowledge as suited for decision making) has several advantages. Besides the practical advantages for data handling at different rates, this representation also provides symbol grounding for verbs and terms. In road vehicle guidance, terms such as *road running* with *constant speed* or with *speed adapted to road curvature, lane changing, convoy driving,* or the complex (perception and vehicle control) maneuver for *turning off onto a crossroad* are easily defined by combinations or sequences of feed-forward and feedback control activation. For each category of behavioral capability, network representations can be developed showing how specific goal-oriented capabilities (third row from bottom of figure 7) depend on elementary skills (second row). The top layer in these network representations shows how specific capabilities in the different categories contribute to overall goal-oriented use of all capabilities in the mission context. The bottom layer represents necessary preconditions for the capability to be actually available; before activation of a capability, this fact is checked by the system through flags, which have to be set by the corresponding distributed processors. For details, the interested reader is referred to Pellkofer (2003) and Siedersberger (2004).

Overall Cognitive System Architecture

The components for perception and control of an autonomous system have to be integrated into an overall system, allowing both easy human-machine interfaces and efficient performance of the tasks for which the system has been designed. To achieve the first goal, the system should be able to understand terms and phrases as they are used in human conversations. Verbs are an essential part of these conversations. They have to be understood, including their semantics and the movements or changes in appearance of subjects when watching these activities. The spatiotemporal models used in the 4D approach should alleviate the realization of these requirements in the future.

In general, two regimes in recognition can be distinguished quite naturally: First are observations to grasp the actual kind and motion states of objects or subjects as they appear here and now, which is the proper meaning of the phrase *to see*. Second, when an object-subject has been detected and recognized, further observations on a larger time scale allow deeper understanding of what is happening in the world. Trajectories of objects can tell something about the nature of environmental perturbations (such as a piece of paper moved around by wind fields), or the route selected by subjects might allow inferring pursued intentions. In any case, these observations can affect autonomous vehicle decisions.

Recent publications in the field of research on human consciousness point in the direction that the corresponding activities are performed in different parts of the brain in humans (Damasio 1999).

Figure 2 shows the overall cognitive system architecture as it results quite naturally when exploiting recursive estimation techniques with spatiotemporal models for dynamic vision (4D approach). However, it took more than a decade and many dissertations to get all elements from the fields of control engineering and AI straightened out, arranged, and experimentally verified such that a rather simple but very flexible system architecture resulted. Object-oriented programming and powerful modern software tools were preconditions for achieving this goal with moderate investment of human capital. An important step in clarifying the roles of control engineering and AI methods was to separate the basic recognition step for an object-subject here and now from accumulating more in-depth knowledge on the behavior of subjects derived from state variable time histories.

The horizontal bar half way up in figure 2, labeled "Dynamic Knowledge Representation (DKR)," separates the number-crunching lower part, working predominantly with systems dynamics and control engineering methods, from the mental AI part on top. It consists of three (groups of) elements: (1) the scene tree for representing the properties of objects and subjects perceived (left); (2) the actually valid task (mission element) for the autonomous vehicle (center); and (3) the abstract representations of autonomous vehicle's perceptual and behavioral capabilities (right).

Building on this wealth of knowledge, situation assessment is done on the higher level independently for gaze and vehicle control because MarVEye with its specialized components has more precise requirements than just vehicle guidance. It especially has to come up with the decision when to split attention over time and perform saccadic perception with a unified model of the scene (Pellkofer 2003; Pellkofer, Lützeler, and Dickmanns 2001). A typical example is recognition of a crossroad with the telecamera requiring alternating viewing directions to the intersection and further down into the crossroad (Lützeler 2002). If several specialist visual recognition processes request contradicting regions of attention, which cannot be satisfied by the usual perceptual capabilities, the central decision unit has to set priorities in

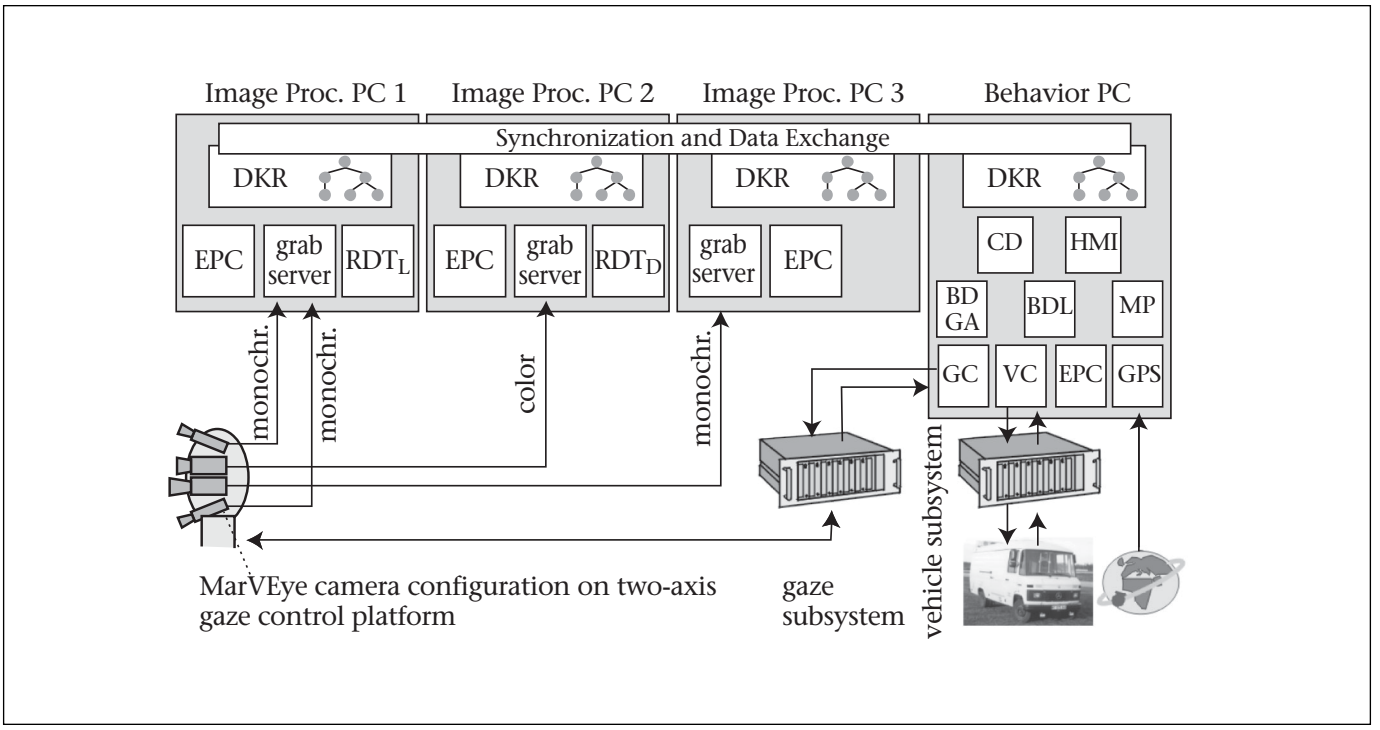

Figure 8. Distributed Commercial Off-the-Shelf (COTS) Computer System with Four Dual PCs, Three Frame Grabbers, and Two Transputer Systems on Which Expectation-Based, Multifocal, Saccadic Vision Has Been Realized and Demonstrated.

the mission context. Possibly, it has to modify vehicle control such that perceptual needs can be matched (see also figure 2, upper right); there is much room for further developments.

The results of situation assessment for vehicle guidance in connection with the actual mission element to be performed include some kind of prediction of the autonomous vehicle trajectory intended and of several other ones for which critical situations might develop. Background knowledge on values, goals, and behavioral capabilities of other vehicles allow assuming intentions of other subjects (such as a lane-change maneuver when observing systematic deviations to one side of its lane). On this basis, fast in-advance simulations or quick checks of lookup tables can clarify problems that might occur when the autonomous vehicle maneuver is continued. Based on a set of rules, the decision level can then trigger transition to a safer maneuver (such as deceleration or a deviation to one side).

The implementation of a new behavior is done on the 4D level (lower center right in figure 2), again with control engineering methods allowing the prediction of typical frequencies and damping of the eigenmodes excited. Designing these control laws and checking their properties is part of conventional engineering in automotive control. What has to be added for autonomous systems is (1) specifying situations when these control laws are to be used with which parameters and (2) storing actual results of an application together with some performance measures about the quality of resulting behavior. The definition of these performance measures (or pay-off functions, possibly geared to more general values for judging behavior) is one of the areas that needs more attention. Resulting maximum accelerations and minimal distance to obstacles encountered or to the road shoulder might be some criteria.

Realization with Commercial Off-the-Shelf Hardware

The EMS vision system has been implemented on two to four DualPentium PCs with a scalable coherent interface (SCI) for communication rates up to 80 megabytes a second and with NT as the operating system. This simple solution has only been possible because two transputer subsystems (having no operating system at all) have been kept as a hardware interface to conventional sensors and actuators from the previous generation (lower layers in figure 8). Spatiotemporal modeling of processes allows for adjustments to variable cycle times and various delay times in different branches of the system.

Three synchronized frame grabbers handle video data input for as many as 4 cameras at a

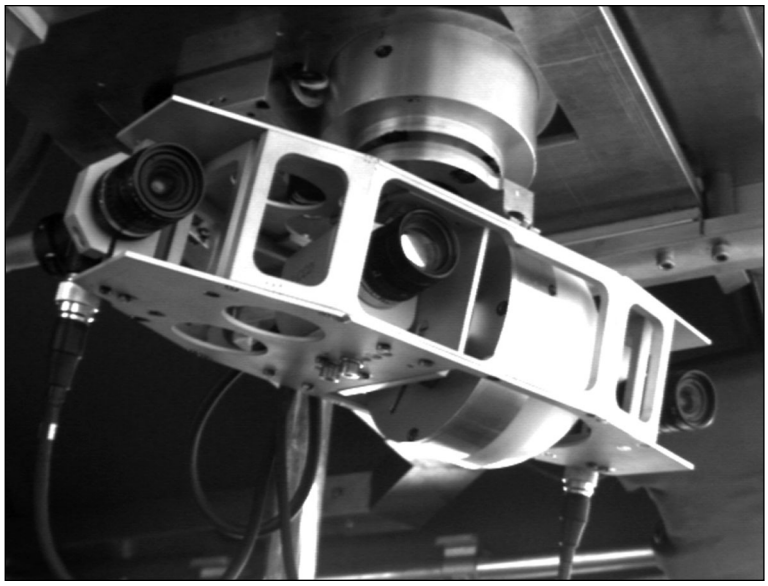

Figure 9. One Version of MarVEye as Tested in VaMoRs.

rate of as high as 50 megabytes a second. The first one transfers only 2 video fields every 40 milliseconds into PC1 (of the 4 odd and even ones grabbed every 20 microseconds). The two black-and-white cameras with wide-angle lenses are divergently mounted on the pointing platform. The second frame grabber does the same with (triple) red-green-blue fields of a three-chip color camera with a mild telelens on PC2, and the third one handles a black-and-white video field of a camera with a strong telelens on PC3. The embedded PC (EPC) DEMON process allows starting and controlling of all PCs from a single user interface on the behavior PC, on which all higher-level processes for behavior decision (both central decision [CD] and those for gaze and attention (BDGA) as well as for locomotion [BDL]) run. The subsystems for gaze control (GC) and vehicle control (VC), as well as the global positioning system (GPS) sensor, are connected to this PC (figure 8, right). It also serves as human-machine-interface (HMI) and mission planning (MP). In the lower right, a small picture of the test vehicle VaMoRs can be seen.

System integration is performed through dynamic knowledge representation (DKR) exploiting SCI, the bar shown running on top through all PCs, thereby implementing the corresponding level in figure 2 (for details, see Rieder [2000]).

MarVEye

Several realizations of the idea of a multifocal active-reactive vehicle eye (MarVEye) with standard video cameras have been tested. Figure 9 shows one version with degrees of freedom in yaw and pitch (pan and tilt), as used in the five-ton test vehicle VaMoRs. Its stereo base is about 30 centimeters. For achieving a large stereo field of view, the optical axes of the outer (wide-angle) cameras are almost parallel in the version shown. Divergence angles as high as ± ~20° have been investigated for a larger total field of view, accepting a reduced stereo field. Both extreme versions have even been implemented together with three more cameras mounted at the bottom of the carrier platform. Because of the increased inertia, time for saccades went up, of course. For high dynamic performance, a compromise has to be found with smaller cameras and a smaller stereo base (entering inertia by a square term in lateral offset from the axis of rotation). (Note that the human stereo base is only 6 to 7 centimeters).

Smaller units for cars with reduced stereo base (~15 cm) and only 1 degree of freedom in pan have also been studied. In the latest version for cars with a central zoom camera mounted vertically in the yaw axis, viewing direction in pitch is stabilized and controlled by a mirror (Dickmanns 2002).

Experimental Results

Both driving on high-speed roads with lane markings and on unmarked dirt roads, as well as cross country, have been demonstrated with EMS vision over the last three years. Recognizing a crossroad without lane markings and turning onto it, jointly using viewing direction and vehicle control, is shown here as one example (Lützeler and Dickmanns 2000). Figure 10 shows three snapshots taken at different time intervals, with the telecamera performing a saccade during the approach to an intersection. Saccading is performed to see the crossroad both at the intersection and at a distance into the intended driving direction for the turnoff. This procedure allows determining the intersection angle and geometry more precisely over time. When these parameters and the distance to the crossroad have been determined, the turning-off maneuver is initiated by fixing the viewing direction at some suitable distance from the vehicle into the crossroad; now the gaze angle relative to the vehicle body increases as the vehicle continues driving straight ahead. Based on the turning capability of the vehicle, a steer angle rate is initiated at a proper distance from the center of the intersection, which makes the vehicle start turning such that under normal conditions, it should end up on the proper side of the crossroad.

During this maneuver, the system tries to find the borderlines of the crossroad as a new

Figure 10. Teleimages during Saccadic Vision When Approaching a Crossroad.
The center image during a saccade is not evaluated (missing indicated search paths).

reference in the wide-angle images. Figure 11 shows this sequence with simultaneously taken images left and right in each image row and a temporal sequence at irregular intervals from top down. The upper row (figure 11a) was taken when the vehicle was still driving essentially in the direction of the original road, with the cameras turned left (looking over the shoulder). The left image shows the left A-frame of the body of VaMoRs and the crossroad far away through the small side window. The crossroad is measured in the right window only, yielding the most important information for a turnoff to the left.

In figure 11b, the vehicle has turned so far that searching for the new road reference is done in both images, which has been achieved in figure 11c of the image pairs. Now, from this road model for the near range, the road boundaries are also measured in the teleimage for recognizing road curvature precisely (left image in figure 11d). The right subimage here shows a bird's eye view of the situation, with the vehicle (marked by an arrow) leaving the intersection area.

This rather complex perception and locomotion maneuver can be considered a behavioral capability emerging from proper combinations of elementary skills in both perception and motion control, which the vehicle has or does not have. The development task existing right now is to provide autonomous vehicles with a sufficient number of these skills and behavioral capabilities so that they can manage to find their way on their own and perform entire missions (Gregor 2002; Gregor et al. 2001). As a final demonstration of this project, a complete mission on a road network was performed, including legs on grass surfaces with the detection of ditches by EMS vision (Siedersberger et al. 2001). For more robust performance, the EMS vision system has been beefed up with special stereo hardware from Pyramid Vision System (PVS ACADIA board). The board was inserted in one of four PCs. Detection of a ditch at greater distances was performed using proven edge-detection methods, taking average intensity values on the sides into account; discriminating shadows on a smooth surface and nearby tracking of the ditch had to be done on the basis of the disparity image delivered. The ditch is described by an encasing rectangle in the ground surface, the parameters and orientation of which are determined visually. An evasive maneuver with view fixation on the nearest corner was performed (Hofmann and Siedersberger 2003; Pellkofer, Hofmann, and Dickmanns 2003)].

Conclusions and Outlook

The integrated approach using spatiotemporal models, similar to what humans do in everyday life (subconsciously), seems more promising for developing complex cognitive robotic systems than separate subworlds for automatic control and knowledge processing. Both the cybernetics approach developed around the middle of the last century and the AI approaches developed in the latter part seem to have been too narrow minded for real success in complex environments with noise-corrupted processes and a large spectrum of objects and subjects. Subjects encountered in our everyday environment range from simple and dumb (animals on lower evolutionary levels or early robots) to intelligent (such as primates); autonomous vehicles can augment this realm. Classes of subjects can be distinguished (beside shape) by their potential perceptual and behavioral capabilities. Individuals of each class can actually only have some of these capabilities at their disposal. For this reason (among others), the explicit representation of capabilities has been introduced in the EMS visual perception system.

This advanced system mimicking vertebrate vision trades about two orders of magnitude reduction in steady video data flow for a few

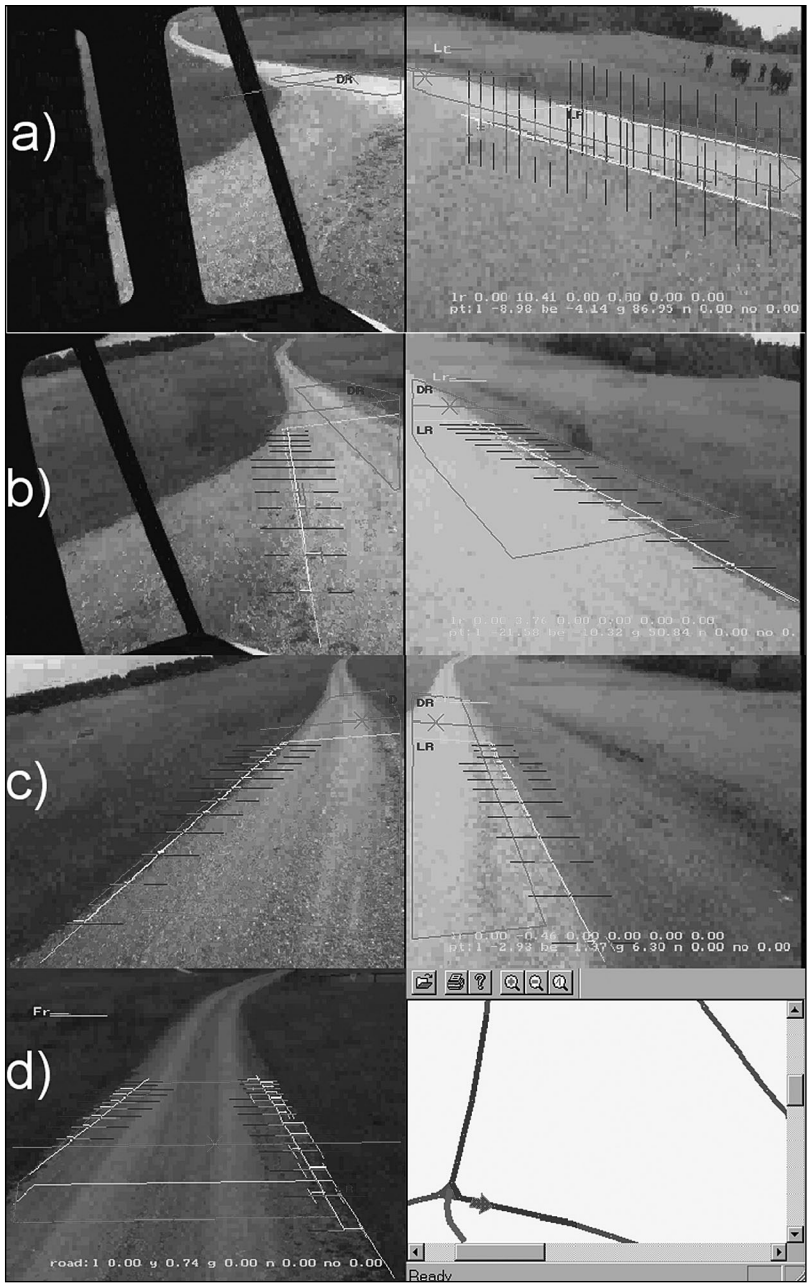

Figure 11. MarVEye Snapshots for Turning Left into a Crossroad (unsealed, gravel).

A. The turn maneuver just started; the right wide-angle camera tracks the crossroad; the left one shows the side window of the vehicle. B. Halfway into the turn-off. C. Both wide-angle cameras track the new road reference. D. *Left:* Tele-image after resuming "road running" on the crossroad. *Right:* Bird's eye view of turn-off trajectory.

tenths of a second delay time in attention-focused, high-resolution perception. This concept allows a wide field of view (> ~ 100° by 45°) at moderate data rates, including a central area of stereo vision that can be redirected by about ± 70 degrees in pan. All video signals are interpreted in conjunction on the so-called 4D system level by distributed processing taking specific sensor locations and parameters into account; basic visual perception is done (except during saccades with fast gaze direction changes) by class-specific agents having generic shape and motion models at their disposal. During saccades, the system works with predictions based on the dynamic models (including animation of typical behaviors). The results are accumulated in a shared scene tree displaying the actual best estimates of geometric shape parameters and (3D) dynamic state variables of all objects-subjects.

This so-called DOB is the decoupling layer with respect to the upper AI-type level on which the data stream concerning objects and subjects is monitored and analyzed (maybe on a different time scale) with special respect to the tasks to be solved. The object data rate is reduced by several orders of magnitude compared to video input rate. The object-oriented representation in connection with the task at hand yields the situation and determines the decisions for efficient use of perceptual and behavioral capabilities.

Behavioral capabilities are implemented on the 4D layer with its detailed representation of dynamic properties. Most of the stochastic perturbations encountered in real-world processes are counteracted on this level by feedback control geared directly to desired state-variable time histories (trajectories). All these capabilities are represented on the higher decision level by their global effect in state transition (performing maneuvers) or by their regulatory effects in counteracting perturbations while performing tasks for achieving the goals of the mission elements. Switching between capabilities is triggered by events or finished mission elements. Switching and transition schemes are represented by state charts.

The system has been implemented and validated in the two autonomous road vehicles VaMoRs (5-ton van) and VaMP (Mercedes S-class car) at UBM. Driving on high-speed roads, turning off onto crossroads in networks of minor roads, leaving and entering roads for driving cross country, and avoiding obstacles (including ditches as negative obstacles) have been demonstrated.

By defining proper performance criteria in special categories and more general values for the overall system, the quality of the actual functioning of these capabilities can be judged in any actual case. This procedure can form a starting point for learning, alone or in a team, based on perception and locomotion capabilities. Slightly varied control output in certain maneuvers (or, more general, actions) will lead

to different values of the performance measure for this maneuver. By observing which changes in control have led to which changes in performance, overall improvements can be achieved systematically. Similarly, systematic changes in cooperation strategies can be evaluated in light of some more refined performance criteria. On the top of figure 7, layers have to be added for future extensions toward systems capable of learning and cooperation. The explicit representation of other capabilities of subjects, such as perception, situation assessment, planning, and evaluation of experience made, should be added as additional columns in the long run. This approach will lead to more flexibility and extended autonomy.

Acknowledgments

This work was funded by the German Bundesamt fuer Wehrtechnik und Beschaffung (BWB) from 1997 to 2002 under the title "Intelligent Vehicle Functions." Contributions by former coworkers S. Baten, S. Fuerst, and V. v. Holt are gratefully acknowledged here because they do not show up in the list of references.

Notes

1. Also of interest is McCarthy, J.; Minsky, M.; Rochester, N.; and Shannon, C. 1955. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, Aug. 31.

2. It is interesting to note here that in the English language, this word is used both for the acting instance (a subject [person] doing something) and for the object being investigated (subjected to investigation). In the German language, one would prefer to call this latter one the object, even if it is a subject in the former sense (der Gegenstand der Untersuchung; *Gegenstand* directly corresponds to the Latin word *objectum*).

References

Albus, J. S., and Meystel, A. M. 2001. *Engineering of Mind—An Introduction to the Science of Intelligent Systems*. New York: Wiley Interscience.

Arkin, R. C. 1998. *Behavior-Based Robotics*. Cambridge, Mass.: MIT Press.

AW&ST. 1986. Researchers Channel AI Activities toward Real-World Applications. *Aviation Week and Space Technology,* Feb. 17: 40–52.

Behringer, R. 1996. Visuelle Erkennung und Inter-pretation des Fahrspurverlaufes durch Rechnersehen für ein autonomes Straßenfahrzeug (Visual Recognition and Interpretation of Roads by Computer Vision for an Autonomous Road Vehicle). Ph.D. diss., Department LRT, UniBw Munich.

Brooks R., and Stein, L. 1994. Building Brains for Bodies. *Autonomous Robots* 1(1): 7–25.

Bryson, A. E., Jr., and Ho, Y.-C. 1975. *Applied Optimal Control. Optimization, Estimation, and Control*. rev. ed. Washington D.C.: Hemisphere.

Caveney, D.; Dickmanns, D.; and Hedrik, K. 2003. Before the Controller: A Multiple Target Tracking Routine for Driver Assistance Systems. *Automatisierungstechnik* 51 (5/2003): 230–238.

Damasio, A. 1999. *The Feeling of What Happens. Body and Emotion in the Making of Consciousness*. New York: Harcourt Brace.

Dickmanns, E. D. 2002. The Development of Machine Vision for Road Vehicles in the Last Decade. Paper presented at the IEEE Intelligent Vehicle Symposium, 18–20 June, Versailles, France.

Dickmanns, E. D. 1998. Vehicles Capable of Dynamic Vision: A New Breed of Technical Beings? *Artificial Intelligence* 103(1–2): 49–76.

Dickmanns, D. 1997. Rahmensystem für visuelle Wahrnehmung veränderlicher Szenen durch Computer (Framework for Visual Perception of Changing Scenes by Computers). Ph.D. diss., Department Informatik, UniBw Munich.

Dickmanns, E. D. 1987. 4D Dynamic Scene Analysis with Integral Spatiotemporal Models. In *the Fourth International Symposium on Robotics Research,* eds. R. Bolles and B. Roth, 311–318. Cambridge Mass.: The MIT Press.

Dickmanns, E. D. 1995. Performance Improvements for Autonomous Road Vehicles. In Proceedings of the International Conference on Intelligent Autonomous Systems (IAS-4), 27–30 March, Karlsruhe, Germany.

Dickmanns, E. D., and Graefe, V. 1988a. Application of Dynamic Monocular Machine Vision. *Journal of Machine Vision and Application* 1: 223–241.

Dickmanns, E. D., and Graefe, V. 1988b. Dynamic Monocular Machine Vision. *Journal of Machine Vision and Application* 1:242–261.

Dickmanns, E. D., and Mysliwetz, B. 1992. Recursive 3D Road and Relative Ego-State Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (Special issue on Interpretation of 3D Scenes) 14(2): 199–213.

Dickmanns, E. D., and Wünsche, H.-J. 1999. Dynamic Vision for Perception and Control of Motion. In *Handbook of Computer Vision and Applications, Volume 3,* eds. Bernd Jähne, Horst Haußecker, and Peter Geißler, 569– 620. San Diego, Calif.: Academic Press.

Dickmanns, E. D., and Zapp, A. 1986. A Curvature-Based Scheme for Improving Road Vehicle Guidance by Computer Vision. In Mobile Robots, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), Volume 727, 161–168. Bellingham, Wash.: SPIE—The International Society for Optical Engineering.

Dickmanns, E. D.; Mysliwetz, B.; Christians, T. 1990. Spatiotemporal Guidance of Autonomous Vehicles by Computer Vision. *IEEE Transactions on Systems, Man, and Cybernetics* (Special issue on Unmanned Vehicles and Intelligent Robotic Systems) 20(6): 1273–1284.

Dickmanns, E. D.; Behringer, R.; Dickmanns, D.; Hildebrandt, T.; Maurer, M.; Thomanek, F.; and Schiehlen, J. 1994. The Seeing Passenger Car. Paper presented at the Intelligent Vehicles '94 Symposium, 24–26 October, Paris, France.

Gregor, R. 2002. Fähigkeiten zur Missionsdurchführung und Landmarkennavigation (Capabilities for Mission Performance and Landmark Navigation). Ph.D. diss., Department LRT, UniBw Munich.

Gregor, R., and Dickmanns, E. D. 2000. EMS Vision: Mission Performance on Road Networks. In Proceedings of the IEEE Intelligent Vehicles Symposium, 140–145. Washington, D.C.: IEEE Computer Society.

Gregor, R.; Lützeler, M.; Pellkofer M.; Siedersberger, K.-H.; and Dickmanns, E. D. 2001. A Vision System for Autonomous Ground Vehicles with a Wide Range of Maneuvering Capabilities. Paper presented at the International Workshop on Computer Vision Systems, 7–8 July, Vancouver, Canada.

Gregor, R., Lützeler, M.; Pellkofer, M.; Siedersberger, K. H.; and Dickmanns, E. D. 2000. EMS Vision: A Perceptual System for Autonomous Vehicles. In Proceedings of the IEEE Intelligent Vehicles Symposium, 52–57. Washington, D.C.: IEEE Computer Society.

Harel, D. 1987. State Charts: A Visual Formalism for Complex Systems. *Science of Computer Programming* 8(3):231–274.

Hofmann, U., and Siedersberger, K.-H. 2003. Stereo and Photometric Image Sequence Interpretation for Detecting Negative Obstacles Using Active Gaze Control and Performing an Autonomous Jink. Paper presented at the SPIE AeroSense Unmanned Ground Vehicles Conference, 22–23 April, Orlando, Florida.

Hofmann, U.; Rieder, A.; and Dickmanns,

E. D. 2000. EMS Vision: An Application to Intelligent Cruise Control for High Speed Roads. In Proceedings of the IEEE Intelligent Vehicles Symposium, 468–473. Washington, D.C.: IEEE Computer Society.

Kailath, T. 1980. *Linear Systems*. Englewood Cliffs, N.J.: Prentice Hall.

Kalman, R. 1960. A New Approach to Linear Filtering and Prediction Problems. In Transactions of the ASME Journal of Basic Engineering, 35–45. Fairfield, N.J.: American Society of Mechanical Engineers.

Klass, P. J. 1985. DARPA Envisions New Generation of Machine Intelligence. *Aviation Week and Space Technology.* 122(16) (22 April 1985)

Lützeler, M. 2002. Fahrbahnerkennung zum Manövrieren auf Wegenetzen mit aktivem Sehen (Road Recognition for Maneuvering on Road Networks Using Active Vision). Ph.D. diss., Department LRT, UniBw Munich.

Lützeler, M., and Dickmanns E. D. 2000. EMS Vision: Recognition of Intersections on Unmarked Road Networks. In Proceedings of the IEEE Intelligent Vehicles Symposium, 302–307. Washington, D.C.: IEEE Computer Society.

Marr, D. 1982. *Vision*. San Francisco, Calif.: Freeman.

Maurer, M. 2000a. Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen (Flexible Automation of Road Vehicles by Computer Vision). Ph.D. diss., Department LRT, UniBw Munich.

Maurer, M. 2000b. Knowledge Representation for Flexible Automation of Land Vehicles. In Proceedings of the IEEE Intelligent Vehicles Symposium, 575–580. Washington, D.C.: IEEE Computer Society.

Miller G.; Galanter, E.; and Pribram, K. 1960. *Plans and the Structure of Behavior.* New York: Holt, Rinehart, and Winston.

Moravec, H. 1983. The Stanford Cart and the CME Rover. *Proceedings of the IEEE* 71(7): 872–884.

Mysliwetz, B. 1990. Parallelrechner-basierte Bildfolgen-Interpretation zur autonomen Fahrzeugsteuerung (Parallel Computer-Based image Sequence Interpretation for the Guiding of Autonomous Vehicles). Ph.D. diss., Department LRT, UniBw Munich.

Newell, A., and Simon, H. 1963. GPS: A Program That Simulates Human Thought. In *Computers and Thought,* 279–293, eds. E. Feigenbaum and J. Feldman. New York: McGraw-Hill.

Nilsson N. J. 1969. A Mobile Automaton: An Application of Artificial Intelligence. In Proceedings of the First International Joint Conference on Artificial Intelligence, 509–521. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Pellkofer, M. 2003. Verhaltensentscheidung fuer autonome Fahrzeuge mit Blickrichtungssteuerung (Behavior Decision for Autonomous Vehicles with Gaze Control). Ph.D. diss., Department LRT, UniBw Munich.

Pellkofer, M., and Dickmanns, E. D. 2000. EMS Vision: Gaze Control in Autonomous Vehicles. In Proceedings of the IEEE Intelligent Vehicles Symposium, 296–301. Washington, D.C.: IEEE Computer Society.

Pellkofer, M.; Hofmann, U.; and Dickmanns E. D. 2003. Autonomous Cross Country Driving Using Active Vision. Paper presented at Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision, 27–30 October, Providence, Rhode Island.

Pellkofer, M.; Lützeler, M.; and Dickmanns, E. D. 2001. Interaction of Perception and Gaze Control in Autonomous Vehicles. Paper presented at the XX SPIE Conference on Intelligent Robots and Computer Vision: Algorithms, Techniques, and Active Vision, 28 October–2 November, Boston, Massachusetts.

Pomerleau, D. A. 1992. Neural Network Perception for Mobile Robot Guidance. Ph.D. diss., Computer Science Department, Carnegie Mellon University, Pittsburgh, Penn.

Rieder, A. 2000. Fahrzeuge sehen—Multisensorielle Fahrzeug-erkennung in einem verteilten Rechnersystem für autonome Fahrzeuge (Seeing Vehicles—Multi-Sensory Vehicle Recognition in a Distributed Computer System for Autonomous Vehicles). Ph.D. diss., LRT, UniBw Munich.

Rosenfeld, A., and Kak, A. 1976. *Digital Picture Processing*. San Diego, Calif.: Academic.

Schell, F. R. 1992. Bordautonomer automatischer Landeanflug aufgrund bildhafter und inertialer Meßdatenauswertung (Onboard Autonomous Automatic Landing Approached. Based on Processing of Iconic (Visual) and Inertial Measurement Data). Ph.D. diss., LRT, UniBw Munich.

Schell, F. R., and Dickmanns, E. D. 1994. Autonomous Landing of Airplanes by Dynamic Machine Vision. *Machine Vision and Application* 7(3): 127–134.

Schiehlen, J. 1995. Kameraplattformen für aktiv sehende Fahrzeuge (Camera Pointing Platforms for Vehicles with Active Vision). Ph.D. diss., LRT, UniBw Munich.

Selfridge, O. 1959. Pandemonium: A Paradigm for *Learning. In Proceedings of the Symposium on the Mechanization of Thought Processes., 511–529,* ed. D. V. Blake and A. M. Uttley. London, U.K.: Her Majesty's Stationary Office.

Selfridge, O., and Neisser, U. 1960. Pattern Recognition by Machine. *In Computers and Thought,* ed. E. Feigenbaum and J. Feldman, 235–267. New York: McGraw-Hill.

Siedersberger, K.-H. 2003. Komponenten zur automatischen Fahrzeugführung in sehenden (semi-) autonomen Fahrzeugen (Components for Automatic Guidance of Autonomous Vehicles with the Sense of Vision). Ph.D. diss., LRT, UniBw Munich. LRT.

Siedersberger, K.-H., and Dickmanns, E. D. 2000. EMS Vision: Enhanced Abilities for Locomotion. In Proceedings of the IEEE Intelligent Vehicles Symposium, 146–151. Washington, D.C.: IEEE Computer Society.

Siedersberger, K.-H.; Pellkofer, M.; Lützeler, M.; Dickmanns, E. D.; Rieder, A.; Mandelbaum, R.; and Bogoni, I. 2001. Combining EMS Vision and Horopter Stereo for Obstacle Avoidance of Autonomous Vehicles. Paper presented at the International Workshop on Computer Vision Systems, 7–8 July, Vancouver, Canada.

Thomanek, F. 1996. Visuelle Erkennung und Zustandsschätzung von mehreren Straßenfahrzeugen zur autonomen Fahrzeugführung (Visual Recognition and State Estimation of Multiple Road Vehicles for Autonomous Vehicle Guidance). Ph.D. diss., LRT, UniBw Munich.

Thomanek, F.; Dickmanns, E. D.; and Dickmanns, D. 1994. Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance. Paper presented at the 1994 Intelligent Vehicles Symposium, 24–26 October, Paris, France.

Werner, S. 1997. Maschinelle Wahrnehmung für den bordautonomen automatischen Hubschrauberflug (Machine Perception for On-board Autonomous Automatic Helicopter Flight). Ph.D. diss., LRT, UniBw Munich.

Wiener, N. 1948. *Cybernetics*. New York: Wiley.

Wuensche, H.-J. 1988. Bewegungssteuerung durch Rechnersehen (Motion Control by Computer Vision). Band 20, Fachber. Messen-Steuern–Regeln. Berlin: Springer.

Ernst D. Dickmanns studied aerospace and control engineering at RWTH Aachen and Princeton University. After 14 years in aerospace research with DFVLR Oberpfaffenhofen in the field of optimal trajectories, in 1975, he became full professor of control engineering in the Aerospace Department at the University of the Bundeswehr near Munich. Since then, his main research field has been real-time vision for autonomous vehicle guidance, pioneering the 4D approach to dynamic vision. His e-mail address is Ernst.Dickmanns@unibw-muenchen.de.