# EMS-Vision: A Perceptual System for Autonomous Vehicles

*R. Gregor, M. Lützeler, M. Pellkofer, K. H. Siedersberger*
and *E. D. Dickmanns*

Institut für Systemdynamik und Flugmechanik,
Universität der Bundeswehr München (UBM),
D-85577 Neubiberg, Germany

*Rudolf.Gregor@unibw-muenchen.de*

**Abstract**

*A survey is given on UBM's new Expectation-based Multi-focal Saccadic Vision (`EMS-Vision`) system. EMS-Vision is the 3rd generation dynamic vision system for road vehicle guidance following the 4-D approach. It combines a wide field of view (f.o.v.) nearby ($> 100°$, $L_{0.05} = 36m$, peripheral part) with central areas of high resolution: a 3-chip-color-camera with a f.o.v. of $23°$ ($L_{0.05} = 100m$) and a high sensitivity b/w-camera with a f.o.v. of $5.5°$ ($L_{0.05} = 300m$, foveal part). At $L_{0.05}$ a single pixel in the image corresponds to 5 cm in the real world. By active gaze control, this foveal cone can be inertially stabilized, be redirected to a point of interest in the wide f.o.v. (saccade), and locked onto a moving object for reducing motion blur (fixation). This vertebrate type of vision system allows new performance levels in machine vision. The system has been implemented on commercial off-the-shelf (COTS) components in both UBM test vehicles* VaMoRs *and* VaMP.

**Keywords** perceptual architecture, dynamic machine vision, autonomous vehicles

## 1 Introduction

UBM has been active in the field of dynamic machine vision since nearly two decades. Over the years, top rank performance has been demonstrated in various applications. Since the mid 80's, in the framework of 'BMFT-Verbundprojekte' with the industrial partner Daimler-Benz AG, UBM has pushed ahead with the development of autonomous road vehicles. In the meantime, seven test vehicles have been equipped with UBM vision systems. The two test vehicles VaMoRs and VaMP, owned by UBM, have covered several thousand kilometers in autonomous mode on public Autobahnen. The central element of all applications at UBM is the '4D-approach' to dynamic machine vision. In order to realize top performance, always the most powerful computer hardware available, complying with the special requirements of an autonomous system, had to be used. In 1996 the course was set for the development of a new generation of vision systems, EMS-Vision. In the first step, the computer market was studied to find the best low-cost hardware basis. Taking into account the extensive experience of operating with custom-made hardware for many years and considering the recent developments in the field of personal computers, special attention was paid to main stream computer hardware. A minimal system was built up to prove that the realization of a real-time vision system was possible by exclusive use of low-cost components [1].

With the state of development in micro-electronics, the hardware and the knowledge base needed for robust solution of a single automotive task in (civilized) natural environments is not much less than for a complex system. However, these allow adaptations to a variety of tasks depending on mission and situation, thereby distributing investment costs on a larger number of functions to be served. This flexible use for many applications may make vision economically viable and superior to any other sensory modality like in vertebrate (especially primate) systems in biology. Driving at high speeds requires large look-ahead distances on the trajectory planned in order to detect obstacles sufficiently early for collision avoidance. On uneven and rough ground, inertial stabilization of the viewing direction is necessary in order to reduce motion blur in the images (especially the tele-ones). In cluttered environments with many subjects moving in an unpredictable manner, a wide field of view is required for collision avoidance; the capability of stereo interpretation will help in these cases (in the near range) to understand the spatial arrangement and motion of objects quickly. All of these requirements have led to the design of the 'Multi-focal, active/reactive Vehicle Eye' MarVEye taking advantage of the as-

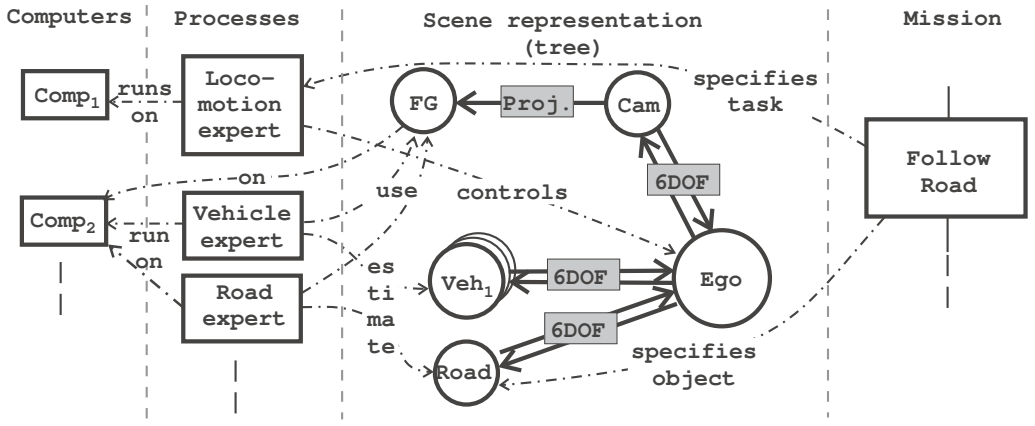


Figure 1: The EMS-Vision data bases for dynamic knowledge

sembly of functions which nature combined into the vertebrate eye, closely interconnected to the vestibular system for inertial sensing.

In this paper, a survey is given on EMS-Vision and some common implementation details of this 20-man-year effort. Detailed aspects of major subsystems and results achieved with this system are presented in five companion papers in this volume. While in [2], [3], [4] and [5] an autonomous driving mission is explained in detail, in [6] the results of an ACC system, developed together with the industrial partner VDO, are presented. A detailed overview of the internal knowledge representation for situation analysis and locomotion is given in [7].

## 2 Knowledge Representation

In order to perform complex tasks in dynamic environments, an autonomous agent needs several kinds of background knowledge. On the one side, an expectation based agent needs static background information about the objects in the environment he will have to cope with. On the other side, an intelligent robot with a variety of capabilities and (mostly) limited computational resources needs an internal representation of these capabilities to achieve optimal performance. During operation, an internal representation of the outside world - the scene-representation - is aggregated dynamically by specific perception experts.

The following section gives an overview of the contents of the different knowledge bases and how they are exploited for performing complex missions on road networks.

Knowledge representation and data exchange in the EMS-Vision system is object oriented. It consists of four specific sections for the distributed system:

**Every computer** in the system is represented by a computer object. This list is generated dynamically during system bootup.

**Every process** in the system is represented by a process object. The process objects contain both general information about the process itself and an interface for point to point communication. Each process object handle at least standardized administrative messages. Additionally, perception process objects contain information about the object classes they are specialized for. An interface allows assigning new perception tasks to them or cancelling running tasks.

**All nodes** of the scene tree represent physical objects (sub-objects) or virtual coordinate systems. Generally, the transformations between scene nodes are described by homogeneous coordinate transformations (HCT), as standard in computer graphics. HCTs can be used for the description of the relative position (`6DOF`) between physical objects as well as for perspective projection into the image coordinate systems. Each scene node offers methods for computing the HCT to its father node or vice versa. In this manner, an arbitrary point in a specific object coordinate system can be transformed into any other object coordinate system as long as all transformations are invertible, which perspective projection (`Proj.`) is not! Besides the relative position, scene nodes can also contain models for shape and dynamics or any other attributes of the represented objects, e. g. symbolic object information, meanings or control flow states. Figure 1 shows an example of a scene tree.

**The mission plan** describes the overall task. It is computed using digital maps containing roads

and landmarks as background knowledge [2]. It consists of a sequential list of mission elements containing planned tasks for locomotion and perception. During a mission only one mission element can be valid at a time.

Figure 1 shows the organisation of the knowledge bases. The central element, the scene tree, contains an internal representation of the own vehicle (here condensed to one node `Ego`) and other objects in the real world (`Road, Vehicles`). The `Ego` node is connected to one camera node (`Cam`); the signals of this camera are digitized by a frame grabber (`FG`).

The mission plan specifies the tasks "Follow Road" for the locomotion expert and gives a reference to the node containing road data. The locomotion expert controls the vehicle's motion. The perception modules aggregate data about physical objects within their specific field of expertise. They need access to digitized images and are therefore started on the computer equipped with the according frame grabber. These knowledge bases are stored and managed by the **d**ynamic **o**bject data**b**ase (`DOB`) process (see 5.2).

# 3 Perception

## 3.1 Basic Module Design

Perception modules have the task to process sensor signals in order to supply decision modules with information about relevant objects in the world (3D) and their dynamics for the mission at hand. Raw sensor signals, usually, are of little use as they are subject to noise and outliers; often, the required information is not readily available, for example geometry and position information in images. For processing noisy sensor signals, the well known technique of Extended Kalman Filtering is applied. In order to be able to handle signals ranging from odometry and accelerometers to radar and video cameras, separate measurement models for each sensor type are required. This comprises the object characteristics that form features for the respective sensor and a reliability indicator for the measurement.

The object, a perception module supplies information about, is characterized by its type, defining geometrical structure and dynamical model, its position and the current geometry parameters. The quality of estimation for each state is given by the variance, taken from the covariance matrices $P$ of the filters, see [8].

The basic estimation cycle, figure 2, is initiated by a new video image. A prediction step for the state variables and the covariance matrix between the last innovation and the time-stamp of the current image

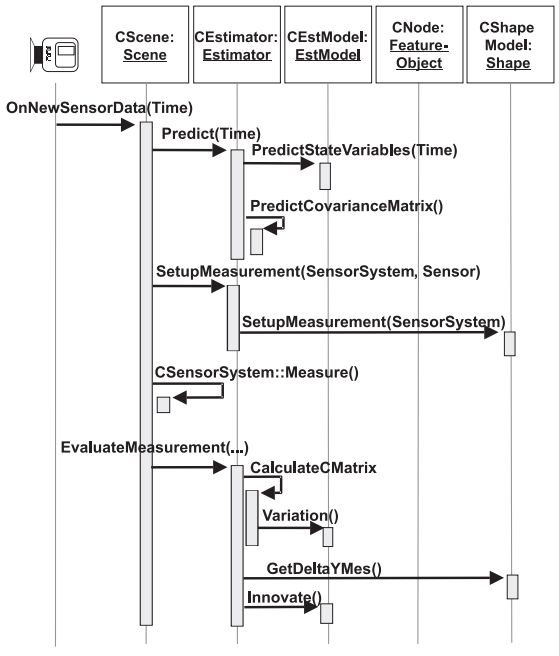


Figure 2: Perception cycle

preceeds the setup of measurement commands. Having gathered the requested measurements from all objects for one sensor, the image is processed and a matching procedure links measurement points with measurement results. Only for matched features the Jacobian row is calculated for the innovation step. This procedure is repeated for each sensor. As all information is fused on the physical object level the integration of multiple sensors can be achieved easily.

The perception modules are integrated in the system by the control flow described in section 4.

## 3.2 Sensor Concept

The main sources of information for the perception system are odometric, inertial and vision sensors. Inertial sensors supplying angular rates and linear accelerations provide information about the current interaction of the vehicle with its environment. Vision, in contrast, supplies information about the future environment the vehicle will meet. A pot-hole will be recognized in the output of inertially based state estimation but a vision system detecting it at $10m$ lookahead distance will have enabled vehicle control to initiate an avoidance maneuver. This recognition capability is improved by visual fixation on this object while approaching. UBM's experimental vehicles are equipped with up to four cameras for the front hemisphere, yielding information in a vertebrate eye fashion. Coarse resolution monochrome peripheral vision with a wide field of view is accompanied by high

resolution foveal color imaging. This concept, dubbed MarVEye is realized with lenses of different focal length, an example for highway driving is given in figure 3.

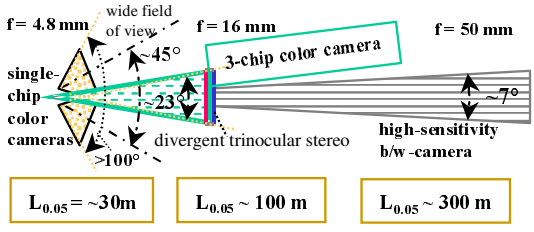


Figure 3: MarVEye camera configuration

## 4 Control Flow

The following section describes how tasks are assigned to the qualified experts. Task assignment in the EMS-Vision system can be divided into two parts: locomotion and perception. During a mission, the mission control module provides the system with planned tasks for locomotion according to the mission plan and the actual situation. In general, all tasks are related to physical objects, like "road" following. Due to this, the major part of the interfaces for task assignment is related to objects as well. The interface for the interaction between the `Attention Control (AC)` module, the perception experts and the `Gaze Control (GC)` module is divided into an object related and a process related part. The process node of each perception process contains information about its capabilities and methods for task assignment. The perception object, in the computer science sense, allows direct control of the perception expert's operation on this object. The perception expert uses the object to transmit its needs to `GC`. It specifies, which gaze direction provides optimal aspect conditions for the recognition of this object with a specific camera. Before tasks are assigned, the decision unit starts the qualified expert through the System Control module if necessary. In order to initiate object recognition, simply the corresponding object is inserted into the scene tree and initialized with background knowledge. This object insertion is the signal for `AC` to assign the according perception task for this object to exactly one expert. During object tracking, another expert may be activated to achieve best estimation results. Of course, several objects may be recognized in parallel.

Viewing direction control and communication with the embedded Two-Axes Camera Carrier (TaCC) controller system ('Platform Subsystem') is performed by the `GC` process. `GC` offers and performs gaze maneuvers, monitors the performance of active maneuvers and updates TaCC data in the `DOB`. Among others, these gaze maneuvers include the following functionality:

**Saccade:** With a Saccade an arbitrary camera can be oriented towards a physical object or a point in object coordinates. The start and the end of the saccade is signaled to the system.

**Smooth pursuit:** With a smooth pursuit a moving object can be kept in a camera image. If the discrepancy exceeds a certain threshold value, an intermediate saccade is started to center the object in the image.

**Searchpath:** With this functionality a certain part of the environment can be scanned with a high resolution sensor for detecting new objects. For this purpose, search paths are planned and performed by the TaCC.

More information about the gaze control unit may be found in [4].

## 5 Implementation on a Distributed Processing Network

### 5.1 Hardware Setup

Prior to explaining software realization aspects an overview of the hardware is given.

In 1996 UBM decided to use COTS computer hardware running under Windows NT as new basis for the development of the EMS-Vision system. First results were presented by Gregor et. al [1]. During the last years both experimental vehicles VaMoRs, a van (MB 508D) and VaMP, a passenger car (MB 500 SEL) have been equipped with the new hardware. Figure 4 shows the actual hardware architecture.

The actual computational part of the EMS-Vision system is a PC-net with 4 computers (three "Image Processing PCs" and one "Behavior PC"). They are connected by SCI (Scalable Coherent Interface) which is used to exchange data in real-time operation. Actually, two types of PCs are used, Dual Pentium II with 333MHz and Dual Pentium III with 450 MHz. The fifth PC ("Gateway PC") is only used for consistent storage of system software. Via FastEthernet the system software is distributed to all other computers. The 10BaseT-Ethernet of the "Gateway PC" serves as a connection to external networks.

"Image Processing PCs" are equipped with framegrabbers for digitizing analog videostreams of CCD cameras. It is also possible to grab image sequences from High Dynamic Range or Low-Light-Level-TV cameras. The video signals of all cameras are synchronized. The cameras are mounted on an

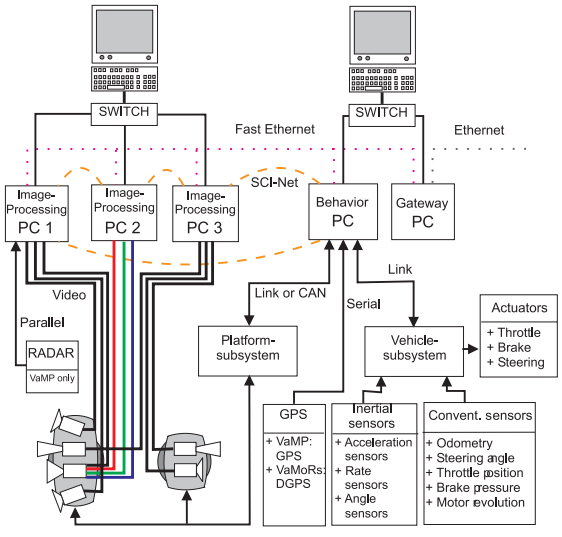


Figure 4: Hardware architecture

active gaze control camera carrier. As described in section 4 UBM's experimental vehicle VaMoRs is equipped with a two axes pan-tilt camera carrier, whereas VaMP has two single axis pan camera carriers, covering front and rear hemisphere. Sensors for angular position and angular rate for each axis are mounted on the carrier. Signals from these sensors are used by the "Platform Subsystem" to control gaze. This subsystem is connected to the "Behavior PC" by a CAN bus. Via a transputer-link the "Vehicle Subsystem" is also coupled to the "Behavior PC". The "Vehicle Subsystem" consists of a transputer net which guarantees real-time feedback control loops. Actuators like steering, brake or throttle are controlled by this subsystem. Additionally, inertial and odometry data and other sensor signals are read in. Another sensor directly connected to "Behavior PC" is a GPS receiver for global navigation.

Figure 4 shows that the entire computer network is connected to two terminals. Although one terminal is sufficient for running the EMS-Vision system, the second terminal is used for debugging and visualization purposes.

For time synchronized logging of digitized videostreams and vehicle sensor data a RAID-system has been integrated. The EMS-Vision system can be run in a simulation mode where logged data can be replayed synchronously.

### 5.2 Software Design

The following section describes some implementation details of basic system functionalities.

Communication in the EMS-Vision system is completely encapsulated in the `DOB` process. The aggregated system knowledge is distributed by the `DOB` to all connected clients on each computer. Within one cycle, the `DOB` collects changed data and sends it in datagrams to all remote computers at once. By bundling data, latancy time for communication is minimized. Every process has access to a copy of a part of the database. The `DOB` keeps the scene representations of all client processes (`EMS-process`) consistent by cyclic data exchange. On every computer of the PC-net an instance of the `DOB` is running. Between these instances point to point communication is performed. Each `EMS-process` is a client of its local `DOB`.

Data consistency is achieved via time management and data buffering over time. A dynamic variable is of value only, if the time is known at which the value has been measured or for which it has been estimated. Data are generated in distributed components. These components must be synchronized in order to make the time-stamps assigned by the different components comparable. The sync-signal of the video cameras, which indicates the presence of a new image every 40 ms (PAL-TV standard), is used as an external hardware timer. After a new sync-signal, the `DOB` raises a cycle-event and distributes it to all clients. The cycle-event triggers data exchange between processes and their local `DOB`; afterwards, data processing is initiated. A `DOB` initiates a new cycle when

- a video sync-signal occurs on the own computer,
- a `DOB` on a remote computer announces a new cycle,
- or if a timeout of $\sim 46ms$ has elapsed without a cycle event.

At the beginning of a cycle, the `DOB`s running on the different computers exchange their modified data. Afterwards, the `DOB`s send modified data to the processes running on their own computer. If a process generates data for cycle n, these data are available to every process in cycle n+1, even on remote computers.

Generally, multiple processes manipulate different data of one node. Therefore, every node possesses several atomic information units, called data items. Each data item has a time-stamp specifying the time of validity. With the help of buffered nodes it is possible to keep a history over time for dynamic variables. Reading processes can, therefore, access data generated several cycles before. Two different concepts of information flow exist:

**Broadcast data flow:** One writing process sends data of common interest to all other processes.

**Routed data flow:** Multiple writing processes precisely send object data to a single reading process.

The DOB offers special data items for both kinds of information flow. Processes can request the DOB to signal one of the following events:

- The beginning of a new cycle,
- the modification of a specified node by another process,
- the modification of a specified data item within a node and
- the insertion and removal of a node of a specified class.

This event-driven strategy leads to efficient data processing as time consuming polling is avoided. In spite of the high bandwidth of the SCI-net, communication still represents a bottleneck. Therefore, the DOB limits the data flow to a minimal extent by communicating only modified or preselected data. When a process connects to the DOB, it specifies the object classes that are required for processing. Therefore, the data flow between a local DOB and an EMS-process is limited to these object classes.

During operation, the complete system is administered by a System Control (SC) module. This process performs two tasks: on the one side it is responsible for starting and terminating processes on all computers. After booting a minimal system this may be done dynamically by request of a decision unit during a mission. On the other side, SC checks, if the processes running are still communicating regularly. Dead processes can be restarted automatically.

The system design calls for multiple processes on one computer node to perform image processing. This and the desire to transparently handle different frame grabbers has led to the implementation of an image abstraction layer. A family of **G**rab **D**evice **S**ervers (GDS) has been implemented to handle digitized images from various sources, e. g. different framegrabbers, bitmap files, or digitized videostreams. The interface supplied allows different clients to access the same image buffer in memory for processing and to display results as overlay on the video images. Additionally, system synchronisation is supported by the **GDS** by passing the video sync-signal to the DOB process.

## 6 Conclusions and Outlook

The system architecture presented here is considered to be rather general and scalable to actual needs. Besides the realization of autonomous road vehicles, aerospace applications for nap-of-the-earth helicopter guidance and for landing approaches have been investigated as well. The system has been used as basis for driver assistance as well as for the implementation of fully autonomous driving missions in various domains. Adaptation to different types of vehicles is simple due to well-separated modules for the representation of specific knowledge. The approach is object oriented both in a computer science and in a physical sense. Physical objects are represented in 3-D space and time. Sensor data fusion is realized by proper modeling of sensors taking the statistical properties of the measurement process fully into account. The approach has large growth potential for more elaborate area-based image sequence processing with the powerful microprocessors to come, and for learning on an elevated level based on understanding processes in 3-D space and time.

## References


[1] R. Gregor et al. A low cost vision system for automotive applications. In *30th International Symposion on Automotive Technology & Automation*, Florence, Italy, 1997.

[2] R. Gregor and E. D. Dickmanns. EMS-Vision: Mission performance on road networks. In *this volume*.

[3] M. Lützeler and E. D. Dickmanns. EMS-Vision: Recognition of intersections on unmarked road networks. In *this volume*.

[4] M. Pellkofer and E. D. Dickmanns. EMS-Vision: Gaze control in autonomous vehicles. In *this volume*.

[5] K. H. Siedersberger and E. D. Dickmanns. EMS-Vision: Enhanced abilities for locomotion. In Ichiro Masaki, editor, *Proc. Int. Symp. on Intelligent Vehicles*, Dearborn, (MI), September 2000. IEEE Industrial Electronics Society.

[6] U. Hofmann, André Rieder, and E. D. Dickmanns. EMS-Vision: An application to intelligent cruise control for high speed roads. In *this volume*.

[7] M. Maurer. Knowledge representation for flexible automation of land vehicles. In *this volume*.

[8] C. L. Thornton and G. J. Bierman. $UDU^T$-covariance factorization for Kalman filtering. In C. T. Leondes, editor, *Control and Dynamic Systems: Advances in Theory and Applications*. Academic Press, Inc., New York, USA, 1980.