Bernt Schiele
Gerhard Sagerer (Eds.)

LNCS 2095

Computer Vision Systems

Second International Workshop, ICVS 2001
Vancouver, Canada, July 2001
Proceedings

Springer

Bernt Schiele Gerhard Sagerer (Eds.)

Computer Vision Systems

Second International Workshop, ICVS 2001
Vancouver, Canada, July 7-8, 2001
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Bernt Schiele
ETH Zentrum, IFW B26.1
8092 Zürich, Switzerland
E-mail: schiele@inf.ethz.ch

Gerhard Sagerer
Universität Bielefeld
Technische Fakultät, Angewandte Informatik
Postfach 100 131, 33501 Bielefeld, Germany
E-mail: sagerer@techfak.uni-bielefeld.de

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Computer vision systems : second international conference ; proceedings / ICVS 2001, Vancouver, Canada, July 7 - 8, 2001. Bernt Schiele ; Gerhard Sagerer (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2095)
Literaturangaben
ISBN 3-540-42285-4

CR Subject Classification (1998): I.4, I.2.9-10, I.5.4-5, I.3.1-2, D.2

ISSN 0302-9743
ISBN 3-540-42285-4 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN 10839427 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2095
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Lecture Notes in Computer Science

For information about Vols. 1–2003
please contact your bookseller or Springer-Verlag

Vol. 2004: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing. Proceedings, 2001. XII, 528 pages. 2001.

Vol. 2006: R. Dunke, A. Abran (Eds.), New Approaches in Software Measurement. Proceedings, 2000. VIII, 245 pages. 2001.

Vol. 2007: J.F. Roddick, K. Hornsby (Eds.), Temporal, Spatial, and Spatio-Temporal Data Mining. Proceedings, 2000. VII, 165 pages. 2001. (Subseries LNAI).

Vol. 2009: H. Federrath (Ed.), Designing Privacy Enhancing Technologies. Proceedings, 2000. X, 231 pages. 2001.

Vol. 2010: A. Ferreira, H. Reichel (Eds.), STACS 2001. Proceedings, 2001. XV, 576 pages. 2001.

Vol. 2011: M. Mohnen, P. Koopman (Eds.), Implementation of Functional Languages. Proceedings, 2000. VIII, 267 pages. 2001.

Vol. 2012: D.R. Stinson, S. Tavares (Eds.), Selected Areas in Cryptography. Proceedings, 2000. IX, 339 pages. 2001.

Vol. 2013: S. Singh, N. Murshed, W. Kropatsch (Eds.), Advances in Pattern Recognition – ICAPR 2001. Proceedings, 2001. XIV, 476 pages. 2001.

Vol. 2014: M. Moortgat (Ed.), Logical Aspects of Computational Linguistics. Proceedings, 1998. X, 287 pages. 2001. (Subseries LNAI).

Vol. 2015: D. Won (Ed.), Information Security and Cryptology – ICISC 2000. Proceedings, 2000. X, 261 pages. 2001.

Vol. 2016: S. Murugesan, Y. Deshpande (Eds.), Web Engineering. IX, 357 pages. 2001.

Vol. 2018: M. Pollefeys, L. Van Gool, A. Zisserman, A. Fitzgibbon (Eds.), 3D Structure from Images – SMILE 2000. Proceedings, 2000. X, 243 pages. 2001.

Vol. 2019: P. Stone, T. Balch, G. Kraetzschmar (Eds.), RoboCup 2000: Robot Soccer World Cup IV. XVII, 658 pages. 2001. (Subseries LNAI).

Vol. 2020: D. Naccache (Ed.), Topics in Cryptology – CT-RSA 2001. Proceedings, 2001. XII, 473 pages. 2001

Vol. 2021: J. N. Oliveira, P. Zave (Eds.), FME 2001: Formal Methods for Increasing Software Productivity. Proceedings, 2001. XIII, 629 pages. 2001.

Vol. 2022: A. Romanovsky, C. Dony, J. Lindskov Knudsen, A. Tripathi (Eds.), Advances in Exception Handling Techniques. XII, 289 pages. 2001

Vol. 2024: H. Kuchen, K. Ueda (Eds.), Functional and Logic Programming. Proceedings, 2001. X, 391 pages. 2001.

Vol. 2025: M. Kaufmann, D. Wagner (Eds.), Drawing Graphs. XIV, 312 pages. 2001.

Vol. 2026: F. Müller (Ed.), High-Level Parallel Programming Models and Supportive Environments. Proceedings, 2001. IX, 137 pages. 2001.

Vol. 2027: R. Wilhelm (Ed.), Compiler Construction. Proceedings, 2001. XI, 371 pages. 2001.

Vol. 2028: D. Sands (Ed.), Programming Languages and Systems. Proceedings, 2001. XIII, 433 pages. 2001.

Vol. 2029: H. Hussmann (Ed.), Fundamental Approaches to Software Engineering. Proceedings, 2001. XIII, 349 pages. 2001.

Vol. 2030: F. Honsell, M. Miculan (Eds.), Foundations of Software Science and Computation Structures. Proceedings, 2001. XII, 413 pages. 2001.

Vol. 2031: T. Margaria, W. Yi (Eds.), Tools and Algorithms for the Construction and Analysis of Systems. Proceedings, 2001. XIV, 588 pages. 2001.

Vol. 2032: R. Klette, T. Huang, G. Gimel'farb (Eds.), Multi-Image Analysis. Proceedings, 2000. VIII, 289 pages. 2001.

Vol. 2033: J. Liu, Y. Ye (Eds.), E-Commerce Agents. VI, 347 pages. 2001. (Subseries LNAI).

Vol. 2034: M.D. Di Benedetto, A. Sangiovanni-Vincentelli (Eds.), Hybrid Systems: Computation and Control. Proceedings, 2001. XIV, 516 pages. 2001.

Vol. 2035: D. Cheung, G.J. Williams, Q. Li (Eds.), Advances in Knowledge Discovery and Data Mining – PAKDD 2001. Proceedings, 2001. XVIII, 596 pages. 2001. (Subseries LNAI).

Vol. 2037: E.J.W. Boers et al. (Eds.), Applications of Evolutionary Computing. Proceedings, 2001. XIII, 516 pages. 2001.

Vol. 2038: J. Miller, M. Tomassini, P.L. Lanzi, C. Ryan, A.G.B. Tettamanzi, W.B. Langdon (Eds.), Genetic Programming. Proceedings, 2001. XI, 384 pages. 2001.

Vol. 2039: M. Schumacher, Objective Coordination in Multi-Agent System Engineering. XIV, 149 pages. 2001. (Subseries LNAI).

Vol. 2040: W. Kou, Y. Yesha, C.J. Tan (Eds.), Electronic Commerce Technologies. Proceedings, 2001. X, 187 pages. 2001.

Vol. 2041: I. Attali, T. Jensen (Eds.), Java on Smart Cards: Programming and Security. Proceedings, 2000. X, 163 pages. 2001.

Vol. 2042: K.-K. Lau (Ed.), Logic Based Program Synthesis and Transformation. Proceedings, 2000. VIII, 183 pages. 2001.

Vol. 2043: D. Craeynest, A. Strohmeier (Eds.), Reliable Software Technologies – Ada-Europe 2001. Proceedings, 2001. XV, 405 pages. 2001.

Vol. 2044: S. Abramsky (Ed.), Typed Lambda Calculi and Applications. Proceedings, 2001. XI, 431 pages. 2001.

Vol. 2045: B. Pfitzmann (Ed.), Advances in Cryptology – EUROCRYPT 2001. Proceedings, 2001. XII, 545 pages. 2001.

Vol. 2047: R. Dumke, C. Rautenstrauch, A. Schmietendorf, A. Scholz (Eds.), Performance Engineering. XIV, 349 pages. 2001.

Vol. 2048: J. Pauli, Learning Based Robot Vision. IX, 288 pages. 2001.

Vol. 2051: A. Middeldorp (Ed.), Rewriting Techniques and Applications. Proceedings, 2001. XII, 363 pages. 2001.

Vol. 2052: V.I. Gorodetski, V.A. Skormin, L.J. Popyack (Eds.), Information Assurance in Computer Networks. Proceedings, 2001. XIII, 313 pages. 2001.

Vol. 2053: O. Danvy, A. Filinski (Eds.), Programs as Data Objects. Proceedings, 2001. VIII, 279 pages. 2001.

Vol. 2054: A. Condon, G. Rozenberg (Eds.), DNA Computing. Proceedings, 2000. X, 271 pages. 2001.

Vol. 2055: M. Margenstern, Y. Rogozhin (Eds.), Machines, Computations, and Universality. Proceedings, 2001. VIII, 321 pages. 2001.

Vol. 2056: E. Stroulia, S. Matwin (Eds.), Advances in Artificial Intelligence. Proceedings, 2001. XII, 366 pages. 2001. (Subseries LNAI).

Vol. 2057: M. Dwyer (Ed.), Model Checking Software. Proceedings, 2001. X, 313 pages. 2001.

Vol. 2059: C. Arcelli, L.P. Cordella, G. Sanniti di Baja (Eds.), Visual Form 2001. Proceedings, 2001. XIV, 799 pages. 2001.

Vol. 2060: T. Böhme, H. Unger (Eds.), Innovative Internet Computing Systems. Proceedings, 2001. VIII, 183 pages. 2001.

Vol. 2062: A. Nareyek, Constraint-Based Agents. XIV, 178 pages. 2001. (Subseries LNAI).

Vol. 2064: J. Blanck, V. Brattka, P. Hertling (Eds.), Computability and Complexity in Analysis. Proceedings, 2000. VIII, 395 pages. 2001.

Vol. 2065: H. Balster, B. de Brock, S. Conrad (Eds.), Database Schema Evolution and Meta-Modeling. Proceedings, 2000. X, 245 pages. 2001.

Vol. 2066: O. Gascuel, M.-F. Sagot (Eds.), Computational Biology. Proceedings, 2000. X, 165 pages. 2001.

Vol. 2068: K.R. Dittrich, A. Geppert, M.C. Norrie (Eds.), Advanced Information Systems Engineering. Proceedings, 2001. XII, 484 pages. 2001.

Vol. 2070: L. Monostori, J. Váncza, M. Ali (Eds.), Engineering of Intelligent Systems. Proceedings, 2001. XVIII, 951 pages. 2001. (Subseries LNAI).

Vol. 2071: R. Harper (Ed.), Types in Compilation. Proceedings, 2000. IX, 207 pages. 2001.

Vol. 2072: J. Lindskov Knudsen (Ed.), ECOOP 2001 – Object-Oriented Programming. Proceedings, 2001. XIII, 429 pages. 2001.

Vol. 2073: V.N. Alexandrov, J.J. Dongarra, B.A. Juliano, R.S. Renner, C.J.K. Tan (Eds.), Computational Science – ICCS 2001. Part I. Proceedings, 2001. XXVIII, 1306 pages. 2001.

Vol. 2074: V.N. Alexandrov, J.J. Dongarra, B.A. Juliano, R.S. Renner, C.J.K. Tan (Eds.), Computational Science – ICCS 2001. Part II. Proceedings, 2001. XXVIII, 1076 pages. 2001.

Vol. 2075: J.-M. Colom, M. Koutny (Eds.), Applications and Theory of Petri Nets 2001. Proceedings, 2001. XII, 403 pages. 2001.

Vol. 2076: F. Orejas, P.G. Spirakis, J. van Leeuwen (Eds.), Automata, Languages and Programming. Proceedings, 2001. XIV, 1083 pages. 2001.

Vol. 2077: V. Ambriola (Ed.), Software Process Technology. Proceedings, 2001. VIII, 247 pages. 2001.

Vol. 2078: R. Reed, J. Reed (Eds.), SDL 2001: Meeting UML. Proceedings, 2001. XI, 439 pages. 2001.

Vol. 2081: K. Aardal, B. Gerards (Eds.), Integer Programming and Combinatorial Optimization. Proceedings, 2001. XI, 423 pages. 2001.

Vol. 2082: M.F. Insana, R.M. Leahy (Eds.), Information Processing in Medical Imaging. Proceedings, 2001. XVI, 537 pages. 2001.

Vol. 2083: R. Goré, A. Leitsch, T. Nipkow (Eds.), Automated Reasoning. Proceedings, 2001. XV, 708 pages. 2001. (Subseries LNAI).

Vol. 2084: J. Mira, A. Prieto (Eds.), Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence. Proceedings, 2001. Part I. XXVII, 836 pages. 2001.

Vol. 2085: J. Mira, A. Prieto (Eds.), Bio-Inspired Applications of Connectionism. Proceedings, 2001. Part II. XXVII, 848 pages. 2001.

Vol. 2089: A. Amir, G.M. Landau (Eds.), Combinatorial Pattern Matching. Proceedings, 2001. VIII, 273 pages. 2001.

Vol. 2091: J. Bigun, F. Smeraldi (Eds.), Audio- and Video-Based Biometric Person Authentication. Proceedings, 2001. XIII, 374 pages. 2001.

Vol. 2092: L. Wolf, D. Hutchison, R. Steinmetz (Eds.), Quality of Service – IWQoS 2001. Proceedings, 2001. XII, 435 pages. 2001.

Vol. 2095: B. Schiele, G. Sagerer (Eds.), Computer Vision Systems. Proceedings, 2001. X, 313 pages. 2001.

Vol. 2096: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. Proceedings, 2001. XII, 456 pages. 2001.

Vol. 2097: B. Read (Ed.), Advances in Databases. Proceedings, 2001. X, 219 pages. 2001.

Vol. 2098: J. Akiyama, M. Kano, M. Urabe (Eds.), Discrete and Computational Geometry. Proceedings, 2000. XI, 381 pages. 2001.

Vol. 2099: P. de Groote, G. Morrill, C. Retoré (Eds.), Logical Aspects of Computational Linguistics. Proceedings, 2001. VIII, 311 pages. 2001. (Subseries LNAI).

Vol. 2105: W. Kim, T.-W. Ling, Y-J. Lee, S.-S. Park (Eds.), The Human Society and the Internet. Proceedings, 2001. XVI, 470 pages. 2001.

Vol. 2110: B. Hertzberger, A. Hoekstra, R. Williams (Eds.), High-Performance Computing and Networking. Proceedings, 2001. XVII, 733 pages. 2001.

Vol. 2119: V. Varadharajan, Y. Mu (Eds.), Information Security and Privacy. Proceedings, 2001. XI, 522 pages. 2001.

Preface

Following the highly successful International Conference on Computer Vision Systems held in Las Palmas, Spain (ICVS'99), this second *International Workshop on Computer Vision Systems,* ICVS 2001 was held as an associated workshop of the International Conference on Computer Vision in Vancouver, Canada. The organization of ICVS'99 and ICVS 2001 was motivated by the fact that the majority of computer vision conferences focus on component technologies. However, Computer Vision has reached a level of maturity that allows us not only to perform research on individual methods and system components but also to build fully integrated computer vision systems of significant complexity. This opens a number of new problems related to system architecture, methods for system synthesis and verification, active vision systems, control of perception and action, knowledge and system representation, context modeling, cue integration, etc. By focusing on methods and concepts for the construction of fully integrated vision systems, ICVS aims to bring together researchers interested in computer vision systems.

Similar to the previous event in Las Palmas, ICVS 2001 was organized as a single-track workshop consisting of high-quality, previously unpublished papers on new and original research on computer vision systems. All contributions were presented orally. A total of 32 papers were submitted and reviewed thoroughly by program committee members. Twenty of them have been selected for presentation. We would like to thank all members of the organizing and program committee for their help in putting together a high-quality workshop.

The workshop was sponsored by the IEEE Computer Society.

We hope all participants enjoyed a successful and interesting workshop.

July 2001

Bernt Schiele and Gerhard Sagerer

Organization

The *Second International Workshop on Computer Vision Systems, ICVS 2001*, held in Vancouver, Canada, July 7–8, 2001 was organized as an associated workshop of the International Conference on Computer Vision 2001.

Organizing Committee

General Co-chairs:	Kevin Bowyer (Univ. of South Florida, USA) Henrik Christensen (KTH, Sweden)
Program Co-chairs:	Bernt Schiele (ETH Zurich, Switzerland) Gerhard Sagerer (Univ. of Bielefeld, Germany)
Publication:	Jorge Cabrera (ULPGC, Spain)

Program Committee

Claus Andersen (Dator A/S, Denmark)
Aaron Bobick (Gorgia Tech, USA)
Kevin Bowyer (Univ. of South Florida, USA)
Jorge Cabrera (ULPGC, Spain)
Henrik Christensen (KTH, Sweden)
Patrick Courtney (Visual Automation, UK)
James L. Crowley (INPG, France)
Ernst Dickmanns (Univ. BW Munich, Germany)
Rüdiger Dillmann (Univ. Karlsruhe, Germany)
Bruce Draper (Univ. Colorado, USA)
Jan-Olof Eklundh (KTH, Sweden)
Robert Fisher (Univ. of Edinburgh, UK)
Catherine Garbay (TIMC-IMAG, France)
Domingo Guinea (IAI-CSIC, Spain)
Kurt Konolige (SRI, USA)
Claus Madsen (Univ. Aalborg, Denmark)
Stefan Posch (Univ. Halle, Germany)
Alex Pentland (MIT, USA)
Claudio Pinhanez (IBM Watson, USA)
Gerhard Sagerer (Univ. of Bielefeld, Germany)
Giulio Sandini (Univ. Genova, Italy)
Bernt Schiele (ETH Zurich, Switzerland)
Chris Taylor (Univ. Manchester, UK)
Monique Thonnat (INRIA Sophia-Antipolis, France)
Mohan Trivedi (UC San Diego, USA)
John Tsotsos (York Univ., Canada)
David Vernon (NUIM, Ireland)

Table of Contents

Real-Time Vision Modules

Recognition

Exploration and Navigation

A Vision System for Autonomous Ground Vehicles with a Wide Range of Maneuvering Capabilities

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
`http://www.unibw-muenchen.de/campus/LRT/LRT13/english/index.html`

Abstract. This paper gives a survey on UBM's new Expectation-based Multi-focal Saccadic Vision (EMS-Vision) system for autonomous vehicle guidance. The core element of the system is a new camera arrangement, mounted on a high bandwidth pan and tilt head (TaCC) for active gaze control. Central knowledge representation and a hierarchical system architecture allow efficient activation and control of behavioral capabilities for perception and action. The system has been implemented on commercial off-the-shelf (COTS) hardware components in both UBM test vehicles. Results from autonomous turn-off maneuvers, performed on army proving grounds, are discussed.

1 Introduction

Since nearly two decades autonomous systems are a topic of intense research. From the mid 80ies on, several national and international programs have been initiated all around the world, like AVCS in Asia, IVHS [1] and PATH in the United States or DRIVE and PROMETHEUS in Europe. The main goals in these programs have been to increase safety and efficiency in normal traffic. Thus, many research groups concentrated on the development of functionalities for autonomous road vehicles being able to interact with other vehicles in a co-operative manner. The key problem of an intelligent system is the acquisition of reliable and precise information about the actual situation. This includes information about ego-state as well as about position and velocity relative to the road and other road-users. Ego-state can be measured easily, but the determination of the vehicle's state relative to other objects and the geometry of these objects is much more difficult. Vision systems have proved their special aptitude for this kind of tasks, e.g. the determination of road curvature, without the need for expensive additional infrastructure. Countless approaches have been developed and often abandoned over the years. Impressive results have been demonstrated by the most successful groups, like from Carnegie Mellon University (CMU) C. Thorpe et al.[2], from University of Parma A. Broggi et al. [3], from the "Fraunhofer-Institut für Informations- und Datenverarbeitung" (IITB) [4] in

B. Schiele and G. Sagerer (Eds.): ICVS 2001, LNCS 2095, pp. 1–20, 2001.

Karlsruhe, from Daimler-Benz Forschung U. Franke et al. [5] and from UBM [6]. For the next step on the way to a really autonomous road vehicle, the navigation on road networks, more challenging problems like intersection recognition and complex driving maneuvers had to be solved. Only few groups have been able to present results from real driving tests like CMU [7], IITB [8] und UBM [9].

With the latest generation implementation, the Expectation-based Multi-focal Saccadic Vision (`EMS-Vision`) system, UBM has set a new course of development. Contrary to former implementations, mostly working with static configurations and optimized for specific tasks or domains, `EMS-Vision` is a flexible system, which is able to configure itself dynamically during operation depending on the actual situation. The explicit representation of the system's capabilities allows direct activation of specific capabilities just in time when needed. Special decision units responsible for different functions assign tasks to specialized experts and supervise their actions. Thus limited computational resources can be exploited very efficiently. The system has been implemented on standard, Intel-based PC hardware components running a general purpose operating system in both UBM test vehicles VaMoRs and VaMP.

The turn-off maneuver as an essential part of a complex driving mission is chosen to illustrate the interaction between the different modules.
The paper is organized as follows: Section 2 gives a review on the different hardware architectures used in former UBM systems. In section 3 the hardware concept of the `EMS-Vision` system is explained. While section 4 gives an overview over general system aspects, in section 5 and 6 individual components are explained in detail. In section 7 experimental results of an autonomous turn-off maneuver are presented. Section 8 concludes the contribution.

2 Development Steps at UBM

In 1985, after 8 years of simulation studies, UBM started to develop an autonomous vehicle based on the results achieved in the field of dynamic machine vision. At that time the computational needs could only be satisfied by specialized hardware. The system chosen then was based on an Intel MULTIBUS I architecture. It consisted of a maximum of 13 microprocessors Intel 80x86 and was extended by a videobus architecture. The system, called BVV, had been developed by the 'Institut für Meßtechnik' at UBM [10]. So called parallel processors (PP) were dedicated to feature extraction and on the other side of the MULTIBUS, the object processors (OP) were used for recursive estimation purposes. For proper operation, the system was equipped with specialized firmware for bootstrapping and communication. Connection to the outside world was done via IEC-bus to a PC. This PC hosted the 'vehicle control' (VC) process and sensor and actuator interfaces to the vehicle. Problems with the bus-based architecture of the BVV were the restricted communication bandwidth, limited scalability and missing robustness.

In 1991 a new approach was taken based on the transputer concept, which satisfied all requirements like higher computing performance, higher communication bandwidth, compactness and robustness, a lower price, and a homogeneous system incorporating all interfaces and modules. In the last configuration, the overall system integrated in VaMP comprised about 60 transputers T-222 (16 bit, for image processing and communication) and T-800 (32 bit, for number crunching and knowledge processing). Additionally a PC host was needed.

Although the scalability of the transputer system had been quite satisfying, the overhead of developing parallel algorithms lead to the wish to have more computing power on the chip available. So the next generation of transputers, the T9000, in combination with a Transputer Image Processing (TIP) videobus system was chosen to be the next hardware platform. As the development of the T9000 was unpredictably delayed, the TIP system could only be used in conjunction with the earlier transputer generation. In 1995 the video processing nodes were substituted by IBM/Motorola PowerPC 601 based components with performance improvements per processor of more than one order of magnitude. However, as further developments of the TIP system had been stopped due to pin-incompatibility of the PowerPC 603 chip, a new hardware platform had to be found.

3 EMS-Vision

Based on the experience made with these systems and the results achieved, new goals were set for the development of `EMS-Vision`:

1. Exclusive use of COTS components to reduce hardware costs.
2. More processing power per node.
3. Recognize a greater variety of object classes.
4. Develop decision units to reach goal driven behavior and knowledge based control of perception subsystems.
5. Develop a uniform representation for all object classes.
6. Introduce a Dynamic Object-Data Base (DOB) for central storage of actual states of objects.

Several computer architectures have been investigated to find the appropriate hardware platform allowing the realization of `EMS-Vision`. The PC platform has finally been chosen, after the usability for automotive applications had been demonstrated with a prototype system. Although Windows NT does not supply any methods for hard real-time applications, it has shown sufficient for fulfilling the smoother real-time constraints here:

1. Applications with hard real-time constraints (control loops) are running on real-time hardware (transputer).
2. The 4D approach allows time-delay compensations.
3. Due to efficient image processing, CPU capacity is not exhausted (this would lead to indeterministic losses of cycles).

3.1 Sensor Concept

The EMS-Vision system design has been kept as general as possible to be able to cope with a variety of scenarios. Thus, the requirements for the vision system are manifold: 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 terrain, inertial stabilization of the viewing direction is necessary in order to reduce motion blur in the images (especially the tele-ones).

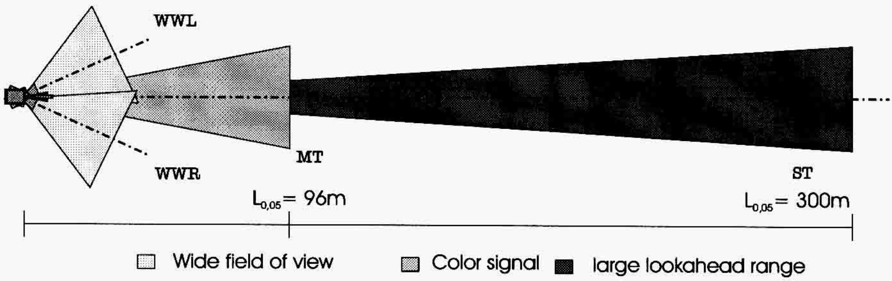

Fig. 1. MarVEye camera configuration

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 assembly of functions which nature combined into the vertebrate eye, closely interconnected to the vestibular system for inertial sensing. The MarVEye camera arrangement combines a wide field of view (f.o.v.) nearby with central areas of high resolution. Coarse resolution monochrome peripheral vision with a wide f.o.v. ($> 100°$), realized by a pair of cameras in a divergent arrangement, is accompanied by high resolution foveal color imaging with a f.o.v. of 23°. A high sensitivity b/w-camera with a f.o.v. of $\sim 7°$ for large lookahead distances completes the system. Figure 1 shows the MarVEye camera arrangement. At $L_{0.05}$ one pixel in the image corresponds to $5cm$ in the real world, so that a typical lane marking is covered by 3 pixels.

3.2 Hardware Concept

The MarVEye camera configuration had major influence on the design of the computational part of the EMS-Vision system. As a PCI bus master transfer of one monochrome video image (768 by 572 pixels) into host memory takes approximately 12 ms (and the memory interface being blocked in the meantime), it was clear that for reaching a cycle time of 40ms a multi computer architecture would be necessary.

Figure 2 shows the actual hardware architecture.

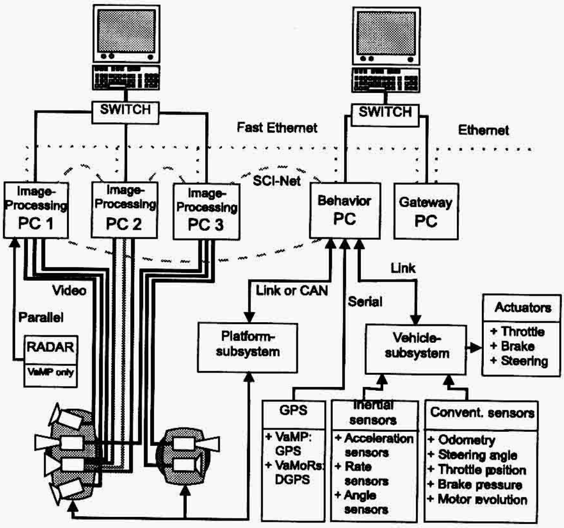

Fig. 2. Hardware architecture

The 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) for fast data exchange 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 used for consistent storage of system software only. Via Fast-Ethernet the system software is distributed to all other computers. The 10BaseT-Ethernet of the "Gateway PC" serves as a connection to external networks.

All "Image Processing PCs" are equipped with frame-grabbers for digitizing analog videostreams. The video signals of all cameras are synchronized. UBM's experimental vehicle VaMoRs is equipped with a two axes pan-tilt head, whereas VaMP has two single axis pan heads, covering front and rear hemisphere. Sensors for angular position and rate for each axis are mounted on the head. Signals from these sensors are used by the "Platform Subsystem" to control gaze. This subsystem is connected to the "Behavior PC" by CAN bus or transputer-link. Via a transputer-link the "Vehicle Subsystem" is also coupled to the "Behavior PC". The "Vehicle Subsystem" consists of a small transputer net which guarantees real-time feedback control loops. Actuators for steering, brake and throttle are controlled by this subsystem. Additionally, inertial and odometry data and other sensor signals are read in. Another sensor directly connected to the "Behavior PC" is a GPS receiver for global navigation.

Figure 2 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.

4 System Overview

This section presents some general system aspects.

4.1 Knowledge Representation

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 specified perception experts.

Dynamic 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 can at least handle 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 canceling 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 (`Proj.`) into an image coordinate system. 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