

# Machine Vision Competence as a Function of Computing Power – Four Years Later

H.-H. Nagel<sup>†‡</sup>, Th. Müller<sup>†</sup>, V. Gengenbach<sup>‡</sup>,  
N. Andreff<sup>\*</sup>, A. Bachem<sup>†</sup>, R. Horaud<sup>\*</sup>, and H. Leuck<sup>†</sup>

<sup>†</sup> *Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe (TH)*  
*Postfach 6980, D-76128 Karlsruhe, Germany*

<sup>‡</sup> *Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB)*  
*Fraunhoferstr. 1, D-76131 Karlsruhe, Germany*

<sup>\*</sup> *Unité de Recherche INRIA Rhône-Alpes*  
*655, avenue de l'Europe, F-38330 Montbonnot St. Martin, France*

*E-mail: hhn@iitb.fhg.de, thmu@ira.uka.de, gen@iitb.fhg.de,*  
*nicolas.andreff@inrialpes.fr, bachem@ira.uka.de, Radu.Horaud@imag.fr, leuck@ira.uka.de*  
*Fax: +49 (0)721 6091-524*

*Abstract*— Annual increases in workstation capacity suggest that today between 5 and 15 times the computing power of 1994 should be available in a lab. We discuss the consequences for the specific case of robot control by real-time machine vision in an automatic disassembly cell, characterizing machine vision competence by speed, robustness, versatility, analyzability, and maintainability as in [Nagel 94].

*Keywords*— Machine-Vision-Based Robot Control, Cycle-Time, Latency, Robustness, Versatility

## 1. INTRODUCTION

At the IAR Annual Conference in 1994, the development of real-time machine vision has been linked to computing capacities which could be made available for the execution of related processes – see [Nagel 94]. The author characterized this development by a sigmoid relation between computing power allocated to machine vision processes and their competence: a computing power of considerable size must be made available in order to master even a rudimentary vision task in real-time. This threshold has been estimated roughly at about  $10^9$  (1 Giga) Operations Per Second (OPS), abbreviated as 1 GOPS. Attempts to refine this estimate provided strong clues that the threshold could be even higher.

Computing power which exceeds such a threshold value would initially be exploited to further speed up the execution of machine vision in order to approach processing at video frame rate. At that point, other aspects of machine vision competence such as robustness, versatility, analyzability, and maintain-

ability would receive increasing emphasis and would begin to absorb further increases of allocated computing power.

The current contribution returns to this hypothesis in order to re-evaluate it against the experience gathered during the past four years. As a preparation, the subsequent section sketches various estimates of the increase in computing power which one could expect over this period. Section 3 outlines the *actual* changes – including changes in computing capacity – for a *specific* installation, an experimental automatic disassembly cell for manufactured goods such as used cars, in particular their engine compartments.

The next section will be devoted to a step by step discussion regarding how additional computing power has been used in order to introduce and study new approaches towards machine vision control of the robots which form part of this disassembly cell. In Section 5, the modifications and extensions to our machine vision approaches will then be ‘projected’ onto a ‘5-dimensional competence space’ spanned by speed, robustness, versatility, analyzability, and

maintainability as mentioned above. Against this background, we shall comment on ideas about how to ‘quantify’ changes along these different ‘dimensions of competence’. We hope in this manner to contribute towards the foundation of an eventual *systems engineering* approach which should facilitate a more rational design of machine vision installations, including trade-offs and optimization.

## 2. ALTERNATIVE GROWTH ESTIMATIONS FOR THE ACCESSIBLE COMPUTING POWER

Various approaches can be defended in an attempt to estimate the increase in computing power due to technical improvements in design and production of highly integrated processor and memory chips. The following discussion assumes that the amount annually available to a university-type laboratory for investment in computers remains constant (in real terms, i. e. corrected for inflation). We are well aware that in times of a declining taxation substrate and shrinking state budgets for higher education, already this assumption may appear overly audacious – but let’s assume for a moment that we are involved in a typical academic discourse where such assumptions should not be ruled out a-priori! It is implied, however, that resources are limited and we can not simply order and install several supercomputers just because this might simplify some experiments.

A semilogarithmic plot of the number of instructions executed per second by a newly introduced processor chip versus its year of introduction exhibits a surprisingly linear relationship over the past 25 years, corresponding to an exponential increase by a factor of roughly 1.5 per year – see, e. g., [Tremblay et al. 95, top of right column on p. 1655]. As a result, one expects an increase by about a factor of 5 over a 4 year period.

Another – frequently quoted – ballpark figure assumes that the computing power doubles every eighteen months, resulting in an increase by a factor of 8 over 4-1/2 years. Experts, who take even the improvement of compiler techniques into account, claim that an annual doubling of the computing power of a top notch workstation would constitute an appropriate estimate. This latter assumption would lead one to expect a total increase by a factor of  $2^4 = 16$  over four years.

In summary, then, we expect that our total computing capacity increased during the last four years – just by keeping up with technological advances – by about an order of magnitude.

## 3. ACTUAL CHANGES IN OUR DISASSEMBLY CELL

The state of our disassembly cell in 1994 is shown in Figure 1 (top panel) whereas the current state is illustrated by the bottom panel of this Figure. The introduction of a separate 6 DoF robot, which carries a stereo-camera setup, allows to position the video-cameras for machine vision control of a manipulation by the original robot. This avoids the risk that crucial parts of the workpiece are occluded by the tool exactly when it matters most, namely immediately prior to the tool-workpiece engagement.

The computer configuration installed in 1994 is indicated by Figure 2. This configuration was connected to a Sun 4/330 which served for program development and interactive control of the disassembly cell. During a short period in 1995, we could temporarily install a SuperSPARC double-processor configuration in order to perform first experiments with a stereo-camera setup (see [Tonko et al. 95b]). Using a particularly fortuitous occasion in late 1995, we could replace the – difficult to program – signal processors permanently by the configuration shown in Figure 3, denoted as GigaMachine. It was a kind of prototype, offered to us without support. We thus could not be surprised to encounter smaller, but increasingly annoying difficulties about two years later in the fall of 1997. We decided, therefore, to complement this multiprocessor configuration by a 300 MHz UltraSPARC-II double processor workstation shown in Figure 4.

It is difficult to quantify the increase in effective computing power connected with each of the modifications sketched above. For simplicity’s sake, we neglect the improvement of computing power due to architectural advances from the SuperSPARC processors to the UltraSPARC-II processors. Computing power is assumed to be proportional to the operating frequency – possibly an underestimation of the actual increase.

Let a single 50 MHz SuperSPARC processor serve as a temporary unit of computing power. Then the three ‘computing nodes’ (RK1 through RK3) correspond together to 12 units and the two ‘interface nodes’ (IK1 and IK2) each add another three units – counting one 75 MHz SuperSPARC as 1.5 units yields 3 units for the two 75 MHz SuperSPARC processors at each interface node – for a total of 18 units. A single 300 MHz UltraSPARC-II processor would then correspond to six units, together they add up to 12 units. In order to correct for the previous ne-

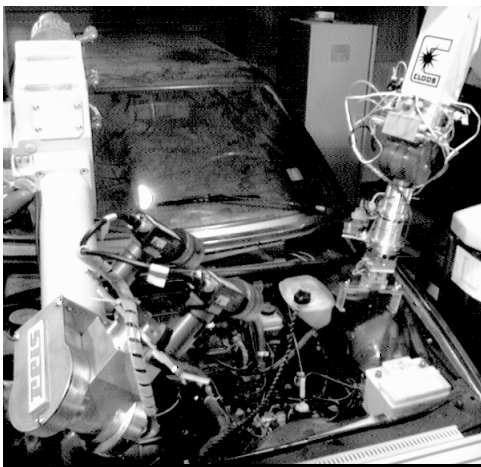
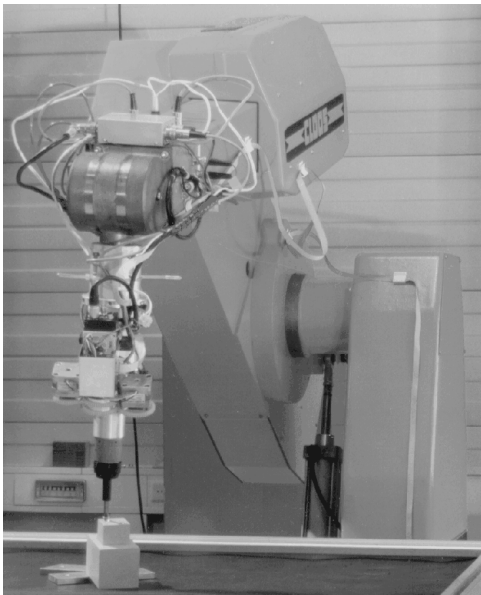


Fig. 1. The top panel shows the state of our experimental disassembly cell in the summer of 1994 when preparatory experiments were conducted on the automatic loosening of screws and the subsequent removal of the top block from a workpiece consisting of two polyhedral pieces joined by screws (from [Gengenbach 94]). The bottom panel (from [Tonko et al. 97c]) illustrates the current status, comprising the former robot for manipulation and a newly installed robot which carries a stereo-camera head in order to study optimal placement of video-cameras for machine vision control of robot disassembly operations in the engine compartment of a used car.

glect of architectural improvements in the transition from SuperSPARC to UltraSPARC-II processors, we could consider the latter to be 50 % more powerful *at the same frequency* – an assumption which again may be a bit bold but allows us to bracket the replacement as being equivalent in *raw* processor (as opposed to *processing*) power to between two thirds

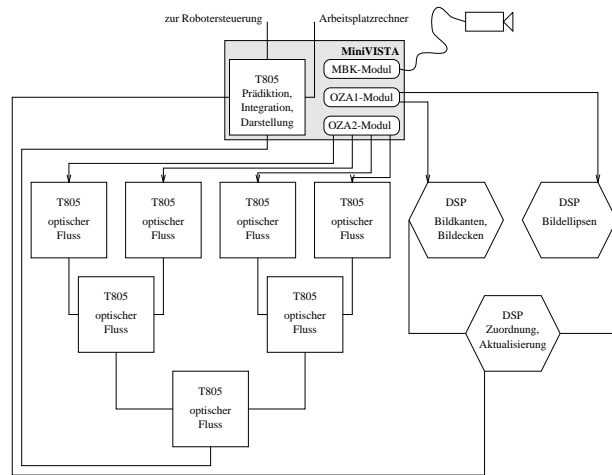


Fig. 2. The computer configuration installed in 1994 consisted of a special purpose computer (denoted here by *MiniVISTA*) for the computation of first and second partial derivatives of the grayvalue function, seven Transputers T805 for the estimation of optical flow vectors at selected image locations, another Transputer (in the *MiniVISTA*) for robot control, and three signal processors (DSP) for the extraction of edge elements, their aggregation into ellipses and the state-update ([Gengenbach 94]; see, too, [Tonko 97]).

and a full GigaMachine. As it turned out later, the efficiency of the GigaMachine was smaller than 1, a well-known phenomenon for those working with parallel processors linked by buses. Moreover, the complexity of the GigaMachine slowed down work on program optimization before we really could test its limits, despite the fact that it could be programmed in C and C++.

The crucial advantage of integrating the GigaMachine into our experimental setup has been twofold. It provided already in early 1996 the computing power necessary to push ahead with a number of methodological advances for machine vision. This would have been impossible with the signal processor configuration of mid-1994, both due to the difficulties to program and debug the signal processors as well as due to lack of computing power. In addition, it allowed us to experiment with significant parallelism (using a total of 16 SuperSPARC processors, albeit with different local environments and capabilities).

#### 4. METHODOLOGICAL CHANGES IN OUR SYSTEM SINCE 1994

Since space does not permit a detailed treatment of the various machine vision problems which have been studied in our disassembly cell since 1994, we

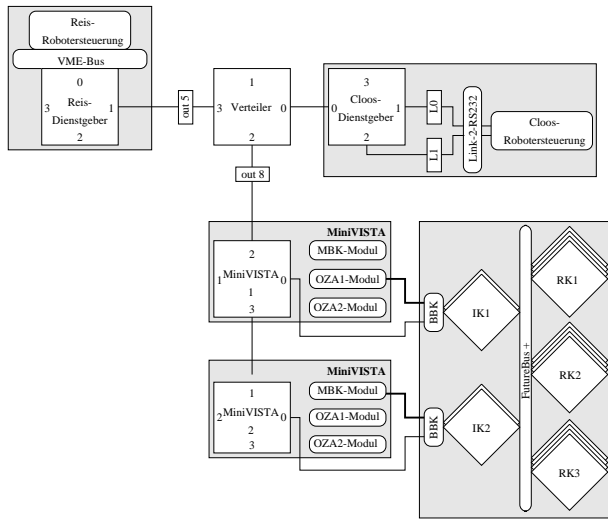


Fig. 3. In order to provide a substantial increase in computing power, the three signal processors in Figure 2 were replaced by a multiprocessor configuration comprising two 'interface nodes' (marked IK1 and IK2, with two 75 MHz SuperSPARC processors each) and three 'computing nodes' (marked RK1 through RK3, with four 50 MHz SuperSPARC processors each), connected by a 15 MByte/s FutureBus+. Since this configuration had to serve the two video cameras of the stereo setup, two MiniVISTA special purpose computers had to be incorporated. In addition, one can see that a dedicated control computer (Cloos-Dienstgeber) is allocated to the original (Cloos) robot as well as a separate second control computer (Reis-Dienstgeber) to the newly installed (Reis) robot which allows to move the stereo-camera setup around under program control (from [Tonko 97]).

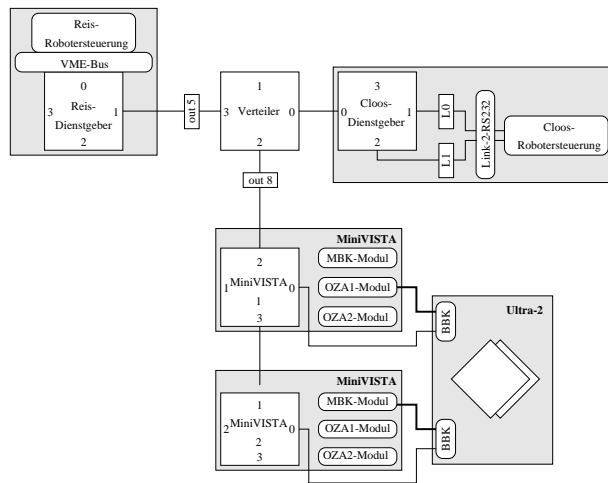


Fig. 4. The GigaMachine multiprocessor configuration shown in Figure 3 has been replaced by a symmetric multiprocessor configuration with two 300 MHz UltraSPARC-II processors.

indicate only the more important ones. In addition

to the references given, the reader is referred to [Tonko 97].

1. Fast image-based tracking, using optical flow estimation, in order not to lose an object image between the time-consuming update steps of a Kalman-Filter – see [Tonko et al. 95a].
2. Transition from an 'eye-on-hand'-camera arrangement to an independently mobile stereo camera system mounted on a separate 6 DoF robot (the 'Reis' robot) – see [Schäfer et al. 95].
3. Transition from a data-driven aggregation of edge *elements* into edge *segments* to a model-driven collection of edge elements. This aggregation step is based on *model* edge segments obtained after projection of the object model (according to the current state estimate) into the image plane and removal of hidden line segments. In this manner, the knowledge accumulated about the workpiece pose in the course of a tracking process can be exploited for edge element selection. This approach had been originally developed in the context of model-based tracking of intersections in video sequences recorded by a moving road vehicle ([Heimes et al. 98]) and has subsequently been adapted for machine-vision-based robot control – see [Tonko et al. 97a].
4. Incorporation of edge element *orientation* into the distance function used to assess the association and fit between edge elements and model segments – see [Tonko et al. 97a].
5. Transition to a consistent formulation of the distance function between an edge element and a model segment in the form of a Mahalanobis-distance which incorporated edge element location as well as its orientation – see [Tonko et al. 97a].
6. Transition from a simple, planar-faceted 'blockworld' workpiece to a quasi-polyhedral workpiece (a car battery) – see [Tonko et al. 97a], [Tonko et al. 97b].
7. Transition to a complex, *non-polyhedral* workpiece (an airfilter intake), represented by a finite-element approximation to a Constructive Solid Geometry model – see [Tonko & Nagel 98].
8. Suppression of model edge segments if their projection into the image plane would be too close neighbors and thus could cause mismatches – see [Tonko 97].
9. Development of a *multi-object* tracking approach by which workpiece and tool could be tracked even if they moved independently

from each other in the scene with respect to the recording camera – see [Tonko 97] and [Tonko et al. 97b].

10. Transition from a *monocular* to a model-based *binocular* tracking process without recourse to a data-driven correspondence search between the two images of the stereo-pair – see [Tonko 97] and [Tonko et al. 97b].
11. Incorporation of the relative pose parameters between the left and right stereo camera coordinate system into the set of parameters to be estimated during the tracking process. Since both workpiece and tool to be tracked are rigid bodies whose pose is described by six parameters, this incorporation of the relative pose increased the number of parameters to be estimated by 50 %, namely from 12 to 18 – see [Tonko 97] and [Tonko et al. 97b].
12. Investigation of camera placement algorithms in order to automatically obtain an optimal camera position for a particular operation of the manipulation robot – see [Keçeci et al. 98].
13. Investigation of *adaptive* gradient estimation and edge element determination processes. These allowed to increase the detection and tracking sensitivity for closely neighbored segments without simultaneously increasing the danger of incorrect associations between edge elements and model segments – see [Nagel et al. 98a]. This approach has been based on recent investigations regarding improved estimation of optical flow fields [Nagel & Gehrke 98].
14. A considerable reduction in the update time, increasing the update frequency for the monocular tracking process to almost 25 Hz, i. e. video-rate!
15. A considerable reduction of the latency until a new state estimation result becomes available.
16. We currently study various alternatives towards either a self-calibrating (stereo) camera setup (see, e. g., [Höraud et al. 94], [Höraud & Csurka 98], [Csurka et al. 98], [Andreff 97], [Andreff et al. 98], or [Wei et al. 98]) or calibration-free tracking approaches.

As shown by Figure 5, the system is able to track a complex, non-polyhedral workpiece by a monocular machine-vision process practically at video-rate, i. e. 25 Hz.

## 5. DISCUSSION AND OUTLOOK

The series of improvements outlined in the preceding section affects *all* performance aspects of machine

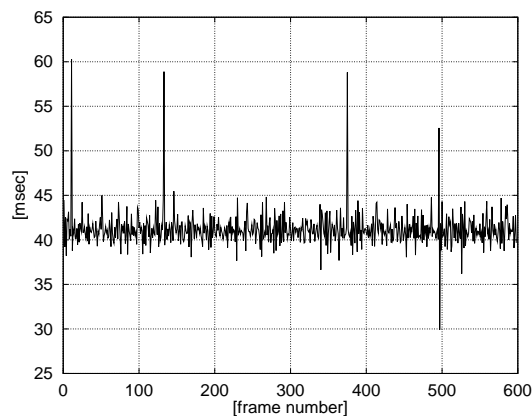


Fig. 5. The ordinate shows the cycle time [msec] of the monocular tracking process as a function of the frame number for 600 pictures recorded by a camera mounted on a robot while it moved continuously towards the airfilter in the engine compartment of a used car. Apart from occasional larger excursions which are attributed to synchronisation variations between the MiniVISTA used for edge element extraction and the Ultra-2 workstation, the measured cycle times fluctuate around 41 msec per frame. This implies that we practically track in video real-time despite the fact that more than thousand edge elements have to be estimated, associated, and matched to model segments for each frame.

vision control of robots.

The most obvious is the increase in speed which has been achieved. At first sight, the ‘speed aspect of machine vision competence’ appears easy to quantify, but upon further scrutiny, the picture gets much more murky. The comparisons discussed here all refer to the same objects with images recorded under the same conditions. Only the tracking programs have been either changed or transferred to a different computer configuration – or both! Since tracking of the complex object configurations treated recently would have been impossible with the initial computer configuration shown in Figure 2, we do not have the means to compare the increase in speed for control of the simple disassembly task comprising two polyhedral blocks as shown in Figure 1 (top panel): we discontinued the old control program when we switched to the GigaMachine.

Such phenomena occur over and over again: changes in computing environments disable the re-use of previous program versions. Their adaptation or re-implementation does not appear attractive enough in a research environment oriented towards *new* investigations rather than re-doing old ones which have been published long ago! From the point of view of systems engineering, this appears foolish since it prohibits the possibility to *quantitatively*

compare the results of any (alleged) improvement – except, perhaps, for those immediately following each other.

Apart from such problems to track the influence of increasing computing power on evaluation speed – and thus on the quality of machine-vision-based control – over a period of several years, further difficulties arise. If the required computing power is provided by a pipelined processor configuration, one needs to differentiate between the repetition frequency of a measurement and its latency time: as shown by [Tonko 97] and extended more recently in [Nagel et al. 98b], such measurements – if reported at all – may vary by an order of magnitude. It does not appear clear under which conditions comparable measurements can be obtained.

A further sobering note applies to a more detailed consideration of what precisely is measured. Even small modifications of the illumination may completely change the number of edge elements in an image and thereby greatly influence the computing power necessary to *select* those which should be matched to model segments, not to speak of the *matching* efforts themselves. It thus appears appropriate to somehow normalize the time taken for an update cycle to the number of edge elements. This would simultaneously take care to some extent of size variations due to changing projection conditions for a tool or workpiece which moves relative to the camera setup in the course of a tracking process. A further consideration in this context would be a normalization regarding the (accumulated) lengths of model segments since this again may influence the edge element selection and matching steps. As a consequence of these considerations, ‘speed’ does not (yet) constitute a simple to quantify aspect of real-time machine vision competence.

Considering additional aspects beyond speed, robustness is clearly affected by adaptive edge element extraction as well as by model-driven collection of edge elements. Regarding quantification of robustness, problems analogous to those mentioned in connection with speed will be encountered. It appears useful, therefore, to somehow specify the conditions under which two approaches should be compared. In order to be able to distinguish better between *robustness* and *versatility*, the following approach is suggested. Robustness will be determined only for objects which are precisely specified by a given (3D CAD-type) computer-internal model. One possibility to quantify robustness would consist in the determination of pose parameter ranges

within which proper initialization would still succeed (*capture range*). Another alternative would simply collect a test suite of images and determine the number of successful initialization and tracking attempts. Still another possibility could consist in the determination of shape and size parameter limits for which an actual object conforming to the nominal object model could be initialized and tracked properly.

The ability to incorporate the tracking of quasi-polyhedral and even non-polyhedral object models clearly increases the *versatility* of our approach. In view of the preceding discussion, a quantification of ‘versatility’ could be attempted in the following manner. One could construct a generic computer-internal description of objects to be handled in the application domain. The model required to successfully initialize and track an object would be derived by fixing parameters of the generic description to a particular value within the allowed range. The distinction between robustness and versatility would thus be based on the number of different object types a system might be capable to handle (in case of versatility) vs. the admissible variations of pose-, size-, and illumination-parameters for a particular object type (in case of robustness). A *model-based autocalibration capability* is thus more likely to support robustness whereas a *data-based* autocalibration capability (see, e. g., [Andreff et al. 98]) could also increase versatility.

Analyzability is improved by the incorporation of a Mahalanobis-distance for association and fitting of edge elements to model segments. The study of adaptive edge element extraction approaches, moreover, promises to replace difficult to analyze selection procedures by a much more homogeneous – and thus easier to analyze – treatment.

The replacement of the GigaMachine by a standard workstation comprising a symmetric multiprocessor with at least the same *effective* computing power greatly simplified the debugging compared to earlier phases where we had to cope with parallelism in loosely-coupled transputer networks or with special process decomposition and allocation of processes to different nodes of the GigaMachine. The transition to the UltraSPARC-II symmetric double processor thus resulted in improved maintainability. An attempt to quantify maintainability for studying its dependence on computing power leads to difficulties which are analogous to those discussed in connection with quantifying the ‘speed’ attribute of a machine vision approach to robot control. One could think of measuring the time taken to incorporate new ob-

ject models or illumination conditions. Since this time may heavily depend on the experience of the programmer asked to implement such modifications, one would like to become less dependent on human intervention for the quantification of maintainability. As an alternative, various software metrics – which are increasingly advocated for an ‘objective’ assessment of program quality – should be investigated. Can such metrics be adapted in order to quantify the changes which a program requires for incorporation of new models or generic object descriptions? A machine vision approach would be the better to maintain, the smaller any required modifications turn out according to the chosen software metric. The use of a software-supported version control tool such as CVS ([Cederqvist et al. 93]) could provide initial means to compare the maintainability of different machine vision approaches quantitatively.

One could argue that a separate ‘maintainability’ dimension might not be necessary in the future. Increases in computing power of standard processors could obviate the need to fall back on special purpose processors for real-time machine vision. Since then machine vision program structures must no longer be adapted to special hardware configurations, standard programming techniques are likely to prevail eventually. Maintaining a machine vision program will be no different from any other complex programming endeavor and thus will not need to be assessed separately in the future.

In summary, then, this reappraisal of an attempt to consider the systems engineering aspects of real-time machine vision in the light of four years of additional experience demonstrates that the ‘natural’ increase of workstation computing power gradually facilitates to replace difficult to program as well as difficult to analyze special purpose computer configurations by new ones, which are more amenable to quantitative analysis and optimization. It simultaneously shows that many questions remain for which even a *precise* problem *formulation* still appears to be part of the problem. Nevertheless, as expected four years ago ([Nagel 94]), the exponential increase in computing power – due to technological advances in the design and production of highly integrated circuits – begins to influence systems engineering considerations for machine vision control of robots in an already discernible manner.

#### REFERENCES

[Andreff 97] N. Andreff: *Towards the Embedding of On-Line Hand-Eye Calibration into Visual Servoing*. In R. Horaud and F. Chaumette (Eds.), Proc. Workshop on

- ‘New Trends in Image-Based Robot Servoing’ in connection with IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS’97), Grenoble, France, 7–11 September 1997, INRIA – Domaine de Voluceau B.P. 105, Le Chesnay, France, 1997, pp. 64–70.
- [Andreff et al. 98] N. Andreff, B. Espiau and R. Horaud: *Une méthode d’auto-étalonnage pince-caméra*. Rapport de recherche (Juin 1998), Unité de Recherche INRIA Rhône-Alpes, 655, avenue de l’Europe, F-38330 Montbonnot St. Martin, France.
- [Cederqvist et al. 93] P. Cederqvist et al.: *Version Management with CVS*. Signum Support AB, 1993. <ftp://ftp.gnu.org/pub/gnu/cvs-1.9.tar.gz>.
- [Csurka et al. 98] G. Csurka, D. Demirdjian, A. Ruf, and R. Horaud: *Closed-Form Solutions for the Euclidean Calibration of a Stereo Rig*. Proc. 5th European Conference on Computer Vision ECCV’98, 2–6 June 1998, Freiburg/Germany, Vol. I, H. Burkhardt and B. Neumann (Eds.), Lecture Notes in Computer Science **1406**, Springer-Verlag: Berlin Heidelberg New York/NY 1998, pp. 426–442.
- [Gengenbach 94] V. Gengenbach: *Einsatz von Rückkopplungen in der Bildauswertung bei einem Hand-Auge-System zur automatischen Demontage*. Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Juli 1994. Published in the series ‘Dissertationen zur Künstlichen Intelligenz’, Vol. DISKI 72, infix Verlag: Sankt Augustin/Germany 1994 (in German).
- [Heimes et al. 98] F. Heimes, H.-H. Nagel, and T. Frank: *Model-Based Tracking of Complex Innercity Road Intersections*. Mathematical and Computer Modelling **27**:9-11 (1998) 189–203.
- [Horaud & Csurka 98] R. Horaud and G. Csurka: *Self-Calibration and Euclidean Reconstruction Using Motions of a Stereo-Rig*. Proc. Sixth International Conference on Computer Vision (ICCV’98), 4–7 January 1998, Bombay/India, Narosa Publishing House: New Delhi a. o. 1998, pp. 96–103.
- [Horaud et al. 94] R. Horaud, F. Dornaika, B. Boufama, and R. Mohr: *Self Calibration of a Stereo Camera Head Mounted onto a Robot Arm*. In Proc. 3rd European Conference on Computer Vision ECCV’94, 2–6 May 1994, Stockholm/S, Vol. I, J.-O. Eklundh (Ed.), Lecture Notes in Computer Science **800**, Springer-Verlag: Berlin Heidelberg New York/NY 1994, pp. 455–462.
- [Keçeci et al. 98] F. Keçeci, M. Tonko, H.-H. Nagel, and V. Gengenbach: *Improving Visually Servoed Disassembly Operations by Automatic Camera Placement*. In Proc. IEEE Int. Conf. on Robotics and Automation, Leuven, Belgium, 16–20 May 1998, pp. 2947–2952.
- [Nagel 94] H.-H. Nagel: *Towards Real-Time Vision-Based Manipulation of Work Pieces*. In: IAR Jahresbericht /Rapport Annuel 1994 (ISSN 0947–0727), Deutsch-Französisches Institut für Automation und Robotik (IAR), Karlsruhe/Germany 1995, pp. 91–104.
- [Nagel & Gehrke 98] H.-H. Nagel and A. Gehrke: *Spatiotemporally Adaptive Filtering for Estimation and Segmentation of Optical Flow Fields*. Proc. 5th European Conference on Computer Vision ECCV’98, 2–6 June 1998, Freiburg/Germany, Vol. II, H. Burkhardt and B. Neumann (Eds.), Lecture Notes in Computer Science **1407**, Springer-Verlag: Berlin Heidelberg New York/NY 1998, pp. 86–102.
- [Nagel et al. 98a] H.-H. Nagel, Th. Müller, A. Gehrke, and V. Gengenbach: *Spatially-Adaptive Filtering in a Model-Based Machine Vision Approach to Robust Work-piece Tracking*. In M. Vincze and G.D. Hager (Eds.), Proc. Workshop WS2 ‘Robust Vision for Vision-Based Control of Motion’, 16 May 1998, Leuven/Belgium (in conjunction with IEEE International Conference on Robotics and Automation).

- [Nagel et al. 98b] H.-H. Nagel, V. Gengenbach, Th. Müller, A. Bachem, and M. Tonko: *Vigor – Visually Guided Robots Using Uncalibrated Cameras: Pose Estimation, Tracking and Manipulation Served by Machine Vision – State-of-the-Art*. KOGS contribution to *State-of-the-Art and Project Specification*, Version 1.4 (September 1998), <http://www.inrialpes.fr/VIGOR/intranet.html>.
- [Schäfer et al. 95] K. Schäfer, M. Tonko, V. Gengenbach, H.-H. Nagel, R. Horaud, R. Mohr: *Real-Time Pose Estimation of Workpiece and Actuator Using an Independently Mobile Camera System*. In: IAR Jahresbericht/Rapport Annuel 1995 (ISSN 0947-0727), Deutsch-Französisches Institut für Automation und Robotik (IAR), Karlsruhe/Germany 1996, pp. 259–263.
- [Tonko 97] M. Tonko: *Zur sichtsystemgestützten Demonstration am Beispiel von Altfahrzeugen*. Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Juni 1997. Published in the series ‘Dissertationen zur Künstlichen Intelligenz’, Vol. DISKI 166, infix Verlag: Sankt Augustin/Germany 1997 (in German).
- [Tonko & Nagel 98] M. Tonko and H.-H. Nagel: *Model-Based Stereo-Tracking of Non-Polyhedral Objects for Automatic Disassembly Experiments*. Submitted for publication (January 1998)
- [Tonko et al. 95a] M. Tonko, K. Schäfer, V. Gengenbach, and H.-H. Nagel: *Multi-level 3D-Tracking of Objects Integrating Velocity Estimation Based on Optical Flow and Kalman-Filtering*. In Proc. Fourth Int. Symp. on Experimental Robotics, Stanford, CA, 30 June–2 July 1995. O. Khatib and K. Salisbury (Eds.), Lecture Notes in Control and Information Sciences vol. **223**. Springer-Verlag: Berlin Heidelberg New York/NY 1997, pp. 212–221.
- [Tonko et al. 95b] M. Tonko, K. Schäfer, V. Gengenbach, H.-H. Nagel, R. Horaud, R. Mohr: *The Determination of Rigidly Moving Polyhedral Structures Based on Real-Time Stereo Vision*. In: IAR Jahresbericht /Rapport Annuel 1995 (ISSN 0947-0727), Deutsch-Französisches Institut für Automation und Robotik (IAR), Karlsruhe/Germany 1996, pp. 255–258.
- [Tonko et al. 97a] M. Tonko, K. Schäfer, F. Heimes, and H.-H. Nagel: *Towards Visually Served Manipulation of Car Engine Parts*. In Proc. IEEE Int. Conf. on Robotics and Automation, Albuquerque/NM, USA, 20–25 April 1997, R. W. Harrigan (Ed.), IEEE Computer Society Press: Los Alamitos/CA, USA, pp. 3166–3171.
- [Tonko et al. 97b] M. Tonko, J. Schürmann, K. Schäfer, H.-H. Nagel: *Visually Served Gripping of a Used Car Battery*. In C. Laugier (Ed.): Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Grenoble, France, 7–11 September 1997, INRIA – Domaine de Voluceau B.P. 105, Le Chesnay, France, 1997, pp. 49–54.
- [Tonko et al. 97c] M. Tonko, V. Gengenbach, H.-H. Nagel, K. Schäfer, S. Picard, R. Horaud, R. Mohr: *Towards the Integration of Object Recognition and Visual Servoing for Disassembly of Used Cars*. In: IAR Jahresbericht /Rapport Annuel 1997 (ISSN 0947-0727), Deutsch-Französisches Institut für Automation und Robotik (IAR), Karlsruhe/Germany 1998, pp. 79–84.
- [Tremblay et al. 95] M. Tremblay, D. Greenley, and K. Normoyle: *The Design of the Microarchitecture of Ultra-SPARC<sup>TM</sup>-I*. Proc. IEEE **83**:12 (1995) 1653–1663.
- [Wei et al. 98] G.-Q. Wei, K. Arbter, and G. Hirzinger: *Active Self-Calibration of Robotic Eyes and Hand-Eye Relationships with Model Identification*. IEEE Transactions on Robotics and Automation **RA-14**:1 (1998) 158–166.