Aspects of Learning-Based Robot Vision

Josef Pauli

Robots are now employed to carry out well-defined tasks in customized static environments, often at a high speed and an astonishing level of precision. However, these robots usually depend totally in their actions on a detailed control scheme developed in advance during an off-line planning phase. Due to recent progress in electronics and computing power, in control and agent technology, and in computer vision and machine learning, the development of autonomous robots capable of solving high-level deliberate tasks in natural environments can now be approached seriously. This article provides essential vision- and learning-related aspects for developing autonomous camera-equipped robot systems.

1 Introduction

There is no generally accepted methodology for developing embedded systems such as autonomous camera-equipped robot systems. Development is based on pre-specified models which are difficult to obtain for various applications due to complexity and imponderables of the environmental world. For designing and developing autonomous camera-equipped robot systems we propose a methodology which is based on libraries of *learning tools* and *architecture patterns* [26]. In an experimental phase prior to the application phase, one must demonstrate relevant objects, critical situations, and purposive situation-action pairs. Then the learning tools are responsible for acquiring image operators and mechanisms of visual feedback control. The learned functions are deployed and thereby adapting generic modules into concrete ones, i.e. specializing the architecture patterns, which leads to task-solving competences in the real environment.

Sections 2 and 3 characterize *Robot Vision* and *Autonomous Camera-Equipped Robot Systems*, with the former being an integral part of the latter. Sections 4 and 5 introduce essential components involved in Robot Vision, i.e. *compatibilities* and *manifolds*, which must be learned for the actual application. Section 6 treats the *development of task-solving competences* by making use of and specializing dynamic vector fields and architecture patterns.

2 Robot Vision

According to Ballard and Brown [2], *Computer Vision is the construction of explicit, meaningful descriptions of physical objects from images.* However, current Computer Vision systems (in industrial use) only work well for specific scenes under specific imaging conditions. New design principles for more general and flexible systems are necessary in order to overcome to a certain extent the large gap between wishful thinking and reality. These principles can be summarized briefly by *animated attention*, *purposive perception*, *visual demonstration*, *learning compatibilities*, *learning manifolds*, *signal-response transformations*, and *feedback analysis*. The succinct term *Robot Vision* is used for systems which take these principles into account.

Animated vision by attention control
It is assumed that most of the three-dimensional vision-related applications must be treated by analyzing images at different viewing angles and/or distances [1]. Through exploratory controlled camera movement the system gathers information incrementally, i.e. the environment serves as external memory from which to read on demand. This paradigm of animated vision also includes mechanisms of selective attention and space-variant sensing [7]. Generally, a two-phase strategy is involved consisting of attention control and detailed treatment of the most interesting places [35]. This approach is a compromise for the trade-off between effort of computations and sensing at high resolution.

Purposive visual information
Only that information of the environmental world must be extracted from the images which is relevant for the vision task. The type of that information can be of quantitative or qualitative nature. In various sub-tasks of a Robot Vision task different information is useful, e.g. color information for tracking robot fingers, and geometric information for grasping objects. The *minimalism principle* emphasizes to solve the task by using features as basic as possible [17], i.e. avoiding time-consuming, erroneous data abstraction and high-level image representation.

Symbol grounding by visual demonstration
Models, which represent target situations, will only prove useful if they are acquired in the same way or under the same circumstances, as when the system perceives the scene in real application [11]. It is important to have a close relation between physically grounded task specifications and the appearance of actual situations. Furthermore, it is easier for a person to specify target situations by demonstrating examples instead of describing visual tasks symbolically. Therefore, *visual demonstration* overcomes the necessity of determining quantitative theories of image formation.

Compatibility between geometry and photometry

In the imaging process, certain *compatibilities* hold between the (global) geometric shape of the object surface and the (local) gray value structure in the photometric image [20]. However, there is no one-to-one correspondence between surface discontinuities and extracted gray value edges, e.g. due to texture, uniform surface color, or lighting conditions. Consequently, qualitative compatibilities must be exploited, which are generally valid for certain classes of regular objects and certain types of camera objectives, in order to bridge the global-to-local gap of representation.

Compatibilities versus manifolds
Compatibilities are general constraints in the process of image formation which do hold more or less under task-relevant or accidental variations of the imaging conditions [26, pp. 25-99]. Based on learned degrees of compatibilities, one can choose those image operators together with parametrizations, which are expected to be most adequate for treating the underlying task. On the other hand, significant variations of image features are represented as *manifolds*. They may originate from changes in the spatial relation among robot effectors, cameras, and environmental objects. Learned manifolds are the basis for acquiring image operators for task-relevant object or situation recognition [22].

Learning signal-response transformations
The signal coming from the imaging process must be transformed into 2D or 3D features, whose meaning and role depend on the task at hand, e.g. serving as motor signal for robot control, or serving as symbolic description for a user. This transformation must be learned on the basis of samples, as there is no theory for determining it a priori. The signal can be regarded as a point in an extremely high-dimensional space, but only a very small fraction of the signal space, i.e. small signal sub-space, is relevant and will be approximated through the samples of the transformation [24]. From the mathematical point of view, the basic matter of learning consists of degrees of compatibilities and approximations of manifolds (see previous subsection). These are the foundation for obtaining the *signal-response transformations* [10].

Feedback-based autonomous image analysis
The analysis algorithms used for signal transformation require the setting or adjustment of parameters [30], such as segmentation thresholds. A *feedback mechanism* is needed to reach autonomy instead of adjusting the parameters interactively. A cyclic process of quality assessment, parameter adjustment, and repeated application of the algorithm can serve as backbone of an automated system.

For the vast majority of vision-related tasks only Robot Vision systems (as opposed to specialized Computer Vision systems) can provide pragmatic solutions. The possibility of camera control and selective attention should be exploited for resolving ambiguous situations and for completing task-relevant information. The successful execution of the visual task is critically based on autonomous learning from visual demonstration. The online adaptation of visual procedures takes possible deviations between learned and actual aspects into account. Learning and adaptation are biased under general compatibilities between geometry and photometry of image formation, which are assumed to hold for a category of similar tasks and a category of similar camera objectives.

3 Autonomous Camera-Equipped Robot Systems

Advanced robot systems are under development which will be equipped with a sensor or camera system for perceiving the environmental scene. Based on perception, the sensor or camera system must impart to the robot an impression of the situation wherein it is working, and thus the robot can take appropriate actions for more flexibly solving a task. Autonomous robot systems can not emerge by simply combining results from research on Artificial Intelligence and Computer Vision. Research in both fields concentrated on reconstructing symbolic models and reasoning about abstract models, which was quite often irrelevant due to unrealistic assumptions. Instead of that, an intelligent system must interface directly to the real world through perception and action. This challenge can be handled by considering four basic characteristics (adopted from Brooks [6]), i.e. *situatedness*, *embodiment*, *emergence*, and *competence*. Taking the four basic characteristics into account, autonomous robot systems must be designed according to a layered *behavioral organization*, and developed and deployed with the use of *learning mechanisms*.

Behavioral organization
Each behavior is based on an activity producing subsystem, featuring sensing, processing, and acting capabilities. The *organization of behaviors* begins on the bottom level with very simple but complete subsystems, and follows an incremental path ending at the top-level with complex autonomous systems. In this layered organization, all behaviors have permanent access to the specific sensing facility and compete in gaining control over the effector or cooperate or being sequenced for realizing high-level tasks. In order to achieve a reasonable global behavior, a ranking of importance is considered for all behaviors, and only the most important ones have a chance to become active. The relevant behavior or subset of behaviors are triggered on occasion of specific sensations in the environment. For example, the obstacle-avoiding behavior must become active before collision, in order to guarantee the survival of the robot, otherwise the original task-solving behavior would be active.

Learning-based development
The development of a (semi-) autonomous camera-equipped robot must be grounded on an infrastructure, based on which the system can *acquire and/or adapt task-relevant competences* autonomously. This infrastructure consists of technical equipment to support the presentation of real world training samples, various *learning mechanisms* for automatically acquiring function approximations, and testing methods for evaluating the quality of the learned functions. Accordingly, for developing autonomous camera-equipped robot systems

one must first demonstrate relevant objects, critial situations, and purposive situation-actions pairs in an experimental phase prior to the application phase. Second, the learning mechanisms are responsible for acquiring image operators and mechanisms of visual feedback control based on supervised experiences in the task-relevant, real environment. Apart from supervised approaches, competences are also acquired by reinforcement learning mechanisms and others.

Role of robots in camera-equipped systems
In camera-equipped systems the robots can be used for two alternative purposes leading to a *robot-supported vision system (robot-for-vision tasks)* or to a *vision-supported robot system (vision-for-robot tasks)*. In the first case, a purposive camera control is the primary goal. For the inspection of objects, factories, or processes, the cameras must be agile for taking appropriate images. A separate actuator system, i.e. a so-called *robot head*, is responsible for the control of external and/or internal camera parameters. In the second case, cameras are fastened on a stable tripod (e.g. *eye-off-hand system*) or fastened on an actuator system (e.g. *eye-on-hand system*), and the images are a source of information for the primary goal of executing robot tasks autonomously. For example, a manipulator may handle a tool on the basis of images taken by an eye-off-hand or an eye-on-hand system. In both cases, a dynamic relationship between camera and scene is characteristic, e.g. inspecting situations with active camera robots, or handling tools with vision-based manipulator robots. For more complicated applications the cameras must be separately agile in addition to the manipulator robot, i.e. having a robot of its own just for the control of the cameras. For those advanced arrangements, the distinction between robot-supported vision system and vision-supported robot system no longer makes sense, as both types are fused.

4 Learning compatibilities

By eliciting fundamental principles underlying the process of image formation, one can make use of a generic bias, and thus reduce the role of object-specific knowledge for structure extraction and object recognition in images [23, pp. 26-30]. Theoretical assumptions (e.g. projective invariants) concerning the characteristic of image formation which can be proven nicely for simulated pinhole cameras, generally do not hold in practical applications. Instead, realistic qualitative assumptions (*compatibilities*) must be learned in an offline phase prior to online application.

Regularities under geometric projection
Shapes of objects should be described by features which are *invariant* under geometric projection and change of view. Examples are so-called *regularity features*, such as parallelism and symmetry of boundary lines, inherent in the shape of many natural or man-made objects. The importance of regularities is two-fold. First, the perceptual organization of line segments into complex two-dimensional constructs, which originate from the surface of three-dimensional objects, can be based on invariant shape regularities. For example, simple constructs of parallel line segment pairs, sophisticated constructs of repeated structures, or rotational symmetries are used. Second, invariant shape regularities are constant descriptions of certain shape classes and, therefore, can be used as indices for recognition [31]. A real camera, however, executes a projective transformation in which shape regularities are relaxed in the image, e.g. three-dimensional symmetries are transformed into two-dimensional skewed symmetries [12]. More generally, projective *quasi-invariants* must be considered instead of projective *invariants* [3].

By demonstrating sample objects including typical regularities and visually perceiving the objects using actual cameras, one makes measurements of real deviations from regularities (in the image), and thus learn the degree of *compatibility of regularities under geometric projection*.

Object surface and photometric invariants
Approaches for recognition and/or tracking of objects in images are confronted with variations of the gray values, caused by changing illumination conditions. The object illumination can change directly with daylight and/or the power of light bulbs, or can change indirectly by shadows arising in the spatial relation between effector, camera, and object. The problem is to convert color values or gray values, which depend on the illumination, into descriptions that do not depend on the illumination. However, solutions for perfect color constancy are not available in realistic applications [8], and therefore approximate photometric invariants are of interest. For example, normalizations of the gray value structure by standard or central moments of second order can improve the reliability of correlation techniques [32].

By demonstrating sample objects under typical changes of the illumination one can make measurements of real deviations from exact photometric invariants, and thus learn the degree of *compatibility of object surface and photometric invariants*.

Geometric and photometric image features
The general assumption behind all approaches of object detection and boundary extraction is that three-dimensional surface discontinuities should have corresponding gray value edges in the image. Based on this, a compatibility between the geometric and photometric type of object representation must hold in the image. For example, the orientation of an object boundary line in the image must be similar to the orientation of a gray value edge of a point on the line [27]. A further example, the junction angle of two boundary lines must be similar to the opening angle of the gray value corner at the intersection point of the lines. The geometric line features are computed globally in an extended patch of the image, and the photometric edge or corner feature are computed locally in a small environment of a point. Consequently, by the common consideration of geometric and photometric features one also verifies the compatibility between global and local image structure.

By demonstrating sample objects including typical edge curvatures and extracting geometric and photometric image features, one can compare the real measurements and learn the degree of *compatibility of geometric and photometric image features*.

Motions in space and changes in view sequence

In an autonomous camera-equipped robot system, the spatial relation between camera(s) and object(s) changes continually. The task-solving process could be represented by a discrete series of changes in this spatial relation, e.g. one could consider the changing relations for the task of moving the robot hand of a manipulator towards a target object while avoiding obstacle objects. Usually, there are different possibilities of taking trajectories subject to the constraint of solving the task. A cost function must be used for determining the cheapest course. Beside the typical components of the cost function, i.e. distance to goals and obstacles, it must also include a measure of difficulty of extracting and tracking task-relevant image features. This aspect is directly related with the computational effort of image sequence analysis and, therefore, has influence on the real-time capability of an autonomous robot system. By constraining the possible camera movements appropriately, the flow vector fields originating from scene objects are easy to represent. For example, a straight camera movement parallel over a plane face of a three-dimensional object should reveal a uniform flow field at the face edges. A further example, if a camera is approaching an object or is rotating around the optical axis which is normal to the object surface, then *log-polar transformation* can be applied to the gray value images. The motivation lies in the fact that during the camera movement, simple shifts of the transformed object pattern occur without any pattern distortions [5]. However, in the view sequence these invariants only hold for a simulated pinhole camera whose optical axis must be kept accurate normal to the object surface while moving the camera.

By demonstrating sample objects and executing typical camera movements relative to the objects, one can make measurements of real deviations from uniformity of the flow field in original gray value or in transformed images, and thus learn the degree of *compatibility between 3D motions and changes in 2D view sequences* [33].

Invariants are special cases of compatibilities

In classical approaches of Computer Vision, invariants are constructed for a group of transformations, e.g. by eliminating the transformation parameters [21]. In real applications, however, the actual transformation formula is not known, and for solving a certain robot task only a relevant subset of transformations should be considered (possibly lacking characteristics of a group). The purpose of visual demonstration is to consider the real corporeality of robot and camera by learning realistic compatibilities (involved in the imaging process) instead of assuming non-realistic invariants. Mathematically, a compatibility must be attributed with a *statistical probability distribution*, which represents the probabilities that certain degrees of deviation from a theoretical invariant might occur in reality. Gaussian probability distributions may be considered, and based on that, the Gaussian extent value σ can be used to define a confidence value for the adequacy of a theoretical invariant. The lower the value of σ, the more confident is the theoretical invariant, i.e. the special case of a compatibility with σ equal to 0 characterizes a theoretical invariant. In an experimentation phase, the σ values of interesting compatibilities are determined by visual demonstration and learning, and in the successive application phase, the learned compatibilities are used in various image operators and servoing cycles. This methodology of *acquiring and using compatibilities* replaces the classical concept of non-realistic, theoretical invariants.

The first attempt of relaxing invariants has been undertaken by Binford and Levitt, who introduced the concept of *quasi-invariance* under transformations of geometric features [3]. The compatibility concept in our work is a more general one, because more general transformations can be considered, maybe with different types of features prior and after the mapping.

5 Learning manifolds

Besides the concept of *compatibilities* we have in the paradigm of Learning-Based Robot Vision also the concept of *manifolds*. Manifolds of *features* play a central role in acquiring functions for object or situation detection in the images (i.e. localization, classification, identification). Additionally, there are manifolds of *signal-response associations* which may combine attributes extracted from images and steering signals for robots. They represent associations between image and environment respectively robot effector.

Feature manifolds for situation detection

For the detection of situations in an image, i.e. in answer to the question *"Where is which situation ?"*, one must acquire models of target situations in advance. There are two alternatives for acquiring such model descriptions. In the first approach, detailed models of 3D target situations and projection functions of the cameras are requested from the user, and from that the relevant models of 2D target situations are computed [16]. In many real world applications, however, the gap between 2D image and 3D world situations is problematic, i.e. it is difficult, costly, and perhaps even impossible to obtain realistic 3D models and realistic projection functions.[1] In the second approach, descriptions of 2D target situations are acquired directly from image features based on visual demonstration of 3D target situations and *learning of feature manifolds under varying conditions* [22]. For many tasks to be carried out in typical scenes, this second approach is preferable, because actual objects and actual characteristics of the cameras are considered directly to model the 2D target situations. A detection function must localize meaningful patterns in the image and classify or evaluate the features as certain model situations. The number of task-relevant image patterns is small in proportion to the overwhelming number of all possible patterns [28], and therefore a detection function must represent the manifolds of relevant image features implicitly.

[1] Recent approaches of this kind use more general, parametric models which express certain unknown variabilities, and these are verified and fine-tuned under the actual situations in the images [18]. With regard to the qualitativeness of the models, these new approaches are similar to the concept of compatibilities in our work, as discussed above.

In the following we use the term *feature* in a general sense (for easy reading). An image pattern or a collection of elementary features extracted from a pattern will simply be called a feature. What we really mean by a feature is a vector or even a complicated structure of elementary (scalar) attributes.

Learning feature manifolds of classified situations
The classification of a feature means assigning it to those model situation whose *feature manifold* contains the relevant feature most probably, e.g. recognize a feature in the image as a certain object [25]. Two criteria should be considered simultaneously, robustness and efficiency, and a measure is needed for both criteria in order to judge different feature classifiers. For the robustness criterion, a measure can be adopted from the literature on statistical learning theory [34] by considering the definition of *probably approximately correct learning (PAC-learning)*. A set of model situations is said to be *PAC-learned*, if a *maximum* percentage E of features is classified erroneous which holds *at least* with the probability P. Robustness can be defined reasonably by the quotient of P by E, i.e. the higher this quotient, the more robust is the classifier. It is conceivable that high robustness requires an extensive amount of attributes for describing the classifier. In order, however, to reduce the computation effort of classifying features, a *minimum description length* of the classifier is prefered [29]. For the obvious conflict between robustness and efficiency a compromise is needed.

By demonstrating appearance patterns of *classified situations*, one can experimentally learn several versions of classifiers and finally select the ones which carry out the best compromise between robustness and efficiency.

Learning feature manifolds of scored situations
Task-relevant changes of 3D spatial relations between effector, camera(s), and/or object(s) must be controlled on the basis of information extracted from the stream of images, e.g. assessing (scoring) consecutive 2D situations relative to the 2D goal situation. The intermediate situations are considered as discrete steps in a course of *scored situations* up to the main goal situation [13]. Classified situations (see above) are a special case of scored situations with just two possible scores, e.g. values 0 or 1. In the continual process of *robot servoing*, e.g. for arranging, grasping, or viewing objects, the differences between consecutive 2D situations in the images must correlate with certain changes between consecutive 3D spatial relations. Geometry-related features in the images include histograms of edge orientation, results of line Hough transformation, responses of situation-specific Gabor filters, etc. Feature manifolds must characterize scored situations, e.g. the gripper is 30 percent off from the optimal grasping situation. A course of scored situations is said to be *PAC-learned*, if a *maximum* deviation D from the actual score is obtained which holds *at least* with probability P.

By demonstrating appearance patterns of scored situations, the system learns several versions of scoring functions and finally selects the best ones according to the PAC-based evaluation. Gaussian basis function networks and principal component analysis may serve as basic mathematical tools for manifold approximation [4]. By exploiting the coherence of consecutive situations one can approximate the relevant manifolds more accurately which improves the robustness of the scoring functions (respectively recognition functions).

Learning environment-effector-image associations
The effector interacts with a small environmental part of the world. For manipulating or inspecting objects in this environment, their coordinates must be determined relative to the effector. Furthermore, the transformation between the effector coordinate system and the coordinates in the image coordinate system must be determined [14]. The relevant function can be learned automatically by controlled effector movement and observation of a calibration object.

For an *eye-off-hand* system, the gripper of a manipulator can be used as calibration object which is observed by cameras without physical connection to the robot arm. The gripper is steered by a robot program through the working space, and the changing image and manipulator coordinates of it are used as samples for learning the relevant function.

For an *eye-on-hand* system the camera(s) is (are) fastened on the actuator system for controlling inspection or manipulation tasks. A natural or artificial object in the environment of the actuator system serves as calibration object. First, the effector is steered by the operator (manually using the control panel) into a certain relation to this calibration object, e.g. touching it or keeping a certain distance to it. In this way, the goal relation between effector and an object is stipulated, something which must be known in the application phase of the task-solving process. Specifically, a certain environmental point will be represented more or less accurately in actuator coordinates. Second, the effector is steered by a robot program through the working space, and the changing image coordinates of the calibration object and position coordinates of the effector are used as samples for learning the relevant function (self-calibration using defined motion).

These strategies of learning environment-effector-image associations are advantageous in several aspects. First, by controlled effector movement, the relevant function of coordinate transformation is learned directly, without computing the intrinsic camera parameters and avoiding artifical coordinate systems (e.g. external world coordinate system). Second, the density of training samples can easily be changed by different discretizations of effector movements. Third, a natural object can be used instead of an artificial calibration pattern. Fourth, task-relevant goal relations are demonstrated instead of modeling them artificially. The learned function is used for transforming image coordinates of objects into the coordinate system of the actuator system.

6 Development of task-solving competences

The *dynamical systems theory* seems to be a powerful framework for commonly representing and performing the designing, the implementation, and the application phase of vision-based robot competences. *Learning and planning procedures* are applied for aquiring task-relevant functions based on measurement compaigns performed in the real environment. Furthermore, generic modules are prepared which serve as *architecture patterns or frameworks* for certain categories of tasks. The outcome of the designing and implementation phase is a configuration of *task-specific modules*. These realize deliberate strategies, parameterized control procedures, and situation detection functions for solving in the successive application phase the underlying, specific task.

Dynamic vector field framework

The variable relation between robot effectors and environmental objects can be represented by vectors with variable tail positions, head positions, directions, and lengths, according to the changing position of effectors and objects. A vector-based representation, as applied in the dynamical systems theory, plays an important role in autonomous robot systems. A layered configuration of dynamic vector fields may uniformly represent deliberate strategies (determined in the designing phase) on the one hand and perception-action cycles (occuring in the application phase) on the other hand. Therefore, a seamless transition between designing and application phase and an uniform integration of reactive and deliberate robot competences can be achieved. A typical task is the navigation of an effector towards target objects while avoiding obstacle objects. *Attractor vector fields* are virtually put at the positions of target objects, and *repellor vector fields* are virtually placed at the positions of obstacle objects [19, pp. 295-355]. By summarizing all contributing vector fields we obtain useful hypotheses of goal-directed, obstacle-avoiding navigation trajectories. If target and obstacle positions are read out offline from maps and the effector position is simulated, then the related vector field is of type *deliberate*. On the other hand, if at least the effector position is acquired online from cameras in a perception-action cycle, then the related vector field is of type *reactive*.

Generic modules as patterns for system development

Different combinations of various types of deliberate and reactive vector fields can be used for configuring a set of generic modules [26, pp. 171-253]. The generic modules serve as library of architecture patterns from which to finally implement *task-specific modules* in order to simplify system development for a specific robotic task. This methodology is similar to the use of general *design patterns* for the development of object-oriented software products [9]. However, we propose application-specific design patterns, i.e. *architecture frameworks*, for the development of autonomous, camera-equipped robot systems.[2] As a result of several case studies on solving high-level, deliberate tasks (using autonomous camera-equipped robot systems) we have discovered three categories of generic modules. First, the *instructional modules* are based solely on deliberate vector fields and are responsible for robotic actions without any visual feedback control. Second, the *behavioral modules* deploy reactive vector fields and possibly are also based on deliberate vector fields, and thus combine deliberate strategies with visual feedback control. Third, the *monitoring modules* are responsible for supervising the task-solving process.

These generic modules must be adapted by considering specific implementations and setting specific parametrizations in order to obtain task-specific modules.

7 Conclusion

We presented aspects for designing and developing advanced robot systems for solving high-level, cognitive tasks. Autonomous camera-equipped robot systems should be designed and developed with learning techniques for exploiting task-relevant and environment-related knowledge. More concretely, the matter of learning consists of compatibilities and manifolds which represent realistic variations of certain features and associations. Architecture patterns and frameworks simplify the system development by adapting generic modules into task-specific modules.

Striving for fully autonomous camera-equipped robot systems seems to be hopeless and ridiculous. However, pattern- and learning-based development tools can help to produce flexible systems at high degrees of sophistication, which should be really useful for solving complex tasks in natural environments.

References

[1] Y. Aloimonos, editor: *Active Perception*. Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey, 1993.

[2] D. Ballard and C. Brown: *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[3] T. Binford and T. Levitt: Quasi-invariants – Theory and exploitation. In *Image Understanding Workshop*, pages 819–829, Morgan Kaufmann Publishers, San Francisco, 1993.

[4] Ch. Bishop: *Neural Networks for Pattern Recognition*. Clarendon Press, London, England, 1995.

[5] M. Bolduc and M. Levine: A review of biologically motivated space-variant data reduction models for Robot Vision. In *Computer Vision and Image Understanding*, volume 69, pages 170–184, 1998.

[6] R. Brooks: Intelligence without reason. In *International Joint Conference on Artificial Intelligence*, pages 569–595, Morgan Kaufmann Publishers, San Francisco, 1991.

[7] C. Colombo, M. Rucci, and P. Dario: Attentive behavior in an anthropomorphic Robot Vision system. In *Robotics and Autonomous Systems*, volume 12, pages 121–131, 1994.

[8] G. Finlayson, B. Funt, and K. Barnard: Color constancy under varying illumination. In *International Conference on Computer Vision*, pages 720–725, IEEE Computer Society, Los Alamitos, California, 1995.

[2] Recently, architecture frameworks have been proposed for multi-agent systems [15].

[9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides: *Design Patterns*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1995.

[10] G. Granlund: The complexity level of vision. In *Signal Processing*, volume 74, pages 101–126, 1999.

[11] S. Harnad: The symbol grounding problem. In *Physica D*, volume 42, pages 335–346, 1990.

[12] P. Havaldar, G. Medioni, and F. Stein: Perceptual grouping for generic recognition. In *International Journal of Computer Vision*, volume 20, pages 59–80, 1996.

[13] G. Heidemann and H. Ritter: Learning to recognise objects and situations to control a robot end-efector. In *Künstliche Intelligenz*, Heft 2, 2003.

[14] R. Horaud and F. Dornaika: Hand-Eye calibration. In *International Journal of Robotics Research*, volume 14, pages 195–210, 1995.

[15] E. Horn and T. Reinke: Musterarchitekturen und Entwicklungsmethoden für Multiagentensysteme. In *Künstliche Intelligenz*, volume 4, pages 48-54, 2000.

[16] K. Ikeuchi and T. Kanade: Automatic generation of object recognition programs. In *Proceedings of the IEEE*, volume 76, pages 1016–1035, 1988.

[17] B. Julesz: Early vision is bottom-up, except for focal attention. *Cold Spring Harbor Symposia on Quantitative Biology*, volume LV, pages 973–978, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York, 1990.

[18] C. Lam, S. Venkatesh, and G. West: Hypothesis verification using parametric models and active vision strategies. In *Computer Vision and Image Understanding*, volume 68, pages 209–236, 1997.

[19] J.-C. Latombe: *Robot Motion Planning*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.

[20] D. Metaxas and D. Terzopoulos: *Computer Vision and Image Understanding*. Special journal issue on Physics-Based Modeling and Reasoning in Computer Vision, volume 65, pages 111–360, Academic Press, San Diego, California, 1997.

[21] J. Mundy and A. Zisserman: Introduction – Towards a new framework for vision. In J. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 1–39, The MIT Press, Cambridge, Massachusetts, 1992.

[22] H. Murase and S. Nayar: Visual learning and recognition of 3D objects from appearance. In *International Journal of Computer Vision*, volume 14, pages 5–24, 1995.

[23] S. Negahdaripour and A. Jain: *Final Report of the NSF Workshop on the Challenges in Computer Vision Research*. University of Miami, Florida, 1991.

[24] E. Oja: *Subspace Methods of Pattern Recognition*. Research Studies Press, Hertfordshire, England, 1983.

[25] L. Paletta and E. Rome: Learning of active object detection in mobile robots. In *Künstliche Intelligenz*, Heft 2, 2003.

[26] J. Pauli: *Learning-Based Robot Vision*. Springer Verlag, Berlin, LNCS 2048, 2001.

[27] J. Princen, J. Illingworth, and J. Kittler: A hierarchical approach to line extraction based on the Hough transform. In *Computer Vision, Graphics, and Image Processing*, volume 52, pages 57–77, 1990.

[28] R. Rao and D. Ballard: Object indexing using an iconic sparse distributed memory. In *International Conference on Computer Vision*, pages 24–31, IEEE Computer Society, Los Alamitos, California, 1995.

[29] J. Rissanen: Universal coding, information, prediction, and estimation. In *IEEE Transactions on Information Theory*, volume 30, pages 629–636, 1984.

[30] U. Rost: *Maschinelles Lernen für die Adaption von Parametern in Bildverarbeitungssystemen*. VDI Verlag, Düsseldorf, Fortschritt-Bericht 633, 2000.

[31] C. Rothwell: Hierarchical object description using invariants. In J. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, Springer Verlag, Berlin, LNCS 825, pages 397–414, 1993.

[32] B. Schiele and J. Crowley: Object recognition using multidimensional receptive field histograms. In *European Conference on Computer Vision*, Springer Verlag, Berlin, LNCS 1064, pages 610–619, 1996.

[33] V. Stephan and H.-M. Gross: Visuomotor anticipation - a powerful approach to behavior-driven perception. In *Künstliche Intelligenz*, Heft 2, 2003.

[34] V. Vapnik: *The Nature of Statistical Learning Theory*. Springer Verlag, Berlin, 1995.

[35] W. Zangenmeister, H. Stiehl, and C. Freksa: *Visual Attention and Control*. Elsevier Science Publishers, Amsterdam, The Netherlands, 1996.

Contact

Dr. habil. Josef Pauli
Fraunhofer IITB, Erkennungssysteme
Fraunhoferstr. 1, 76131 Karlsruhe
pauli@iitb.fraunhofer.de

Bild

Josef Pauli studied Computer Science, Master‘s degree in 1986. He was affiliated with the Computer Science institute at the Technische Universität München, Doctor‘s degree in 1992. Until August 2002 he was a staff member of the Computer Science institute at the Christian-Albrechts-Universität zu Kiel, habilitation degree in 1999. Then he moved to the Fraunhofer IITB in Karlsruhe as head of the department of Recognition Systems.