SEMANTIC MODELS AND OBJECT RECOGNITION IN COMPUTER VISION

G. Sagerer, F. Kummert, G. Socher
University of Bielefeld
Technische Fakultät, AG Angewandte Informatik
Postfach 100131, 33501 Bielefeld, Germany

KEY WORDS: Vision, Modeling, Knowledge Base, Image Understanding, Hybrid Systems, Semantic Networks

ABSTRACT:

Object recognition has a long history in pattern recognition and computer vision. A major problem addressed is the development of models which are suitable for recognition and scene interpretation tasks. Two principal paradigms are emphasized. On the one hand side statistical and neural models making use of representative training samples optimizing parameters of decision functions. Contrarily, knowledge based techniques build explicit representations by modeling the structure and the constraints associated with a specific task. The main point of this paper is to show that both paradigms shall and can be incorporated to achieve efficient problem solutions for complex problems. According to this goal the basic techniques for object recognition and scene interpretation will be presented and discussed. Based on this evaluation a hybrid system has been evolved which tries to combine the advantages of the fundamental paradigms. The system is derived from the knowledge representation scheme of procedural semantic networks integrating the advantages of neural network approaches for classification and scoring purposes. Thus, explicit semantic models are combined with learning sample dependent analogous representations. One application of this environment the reconstruction of three dimensional scenes illustrates that this approach is appropriate for complex tasks. Furthermore, the accuracy of the results shows that hybrid and distributed modeling of objects and scenes is a powerful and efficient technique for scene interpretation tasks.

1 Introduction

Object recognition and scene interpretation are great challenges in various scientific fields. The development of algorithms and system architectures is mainly influenced by research activities in pattern recognition, neural computer science, and artificial intelligence. Although the problem is addressed by a large number of projects, there is not a unique baseline algorithm or a general paradigm to solve these complex tasks. According to the classical problem solving techniques of the three mentioned disciplines, approaches have been developed for many applications. Additionally, the aspect of suitable models for image processing, image understanding, or computer vision is increasingly emphasized. Two main streams are considered. Semantic models are derived from representation techniques mainly developed in artificial intelligence research. Statistical and neural network based approaches are studied mainly in the context of computer vision providing algorithms for classification and localization of objects. While semantic models emphasize on structural properties, both analogous techniques concentrate on a holistic viewpoint of an entire object.

In this paper the cooperation and integration of these basic techniques is proposed. In our opinion, the development of systems which are able to solve complex tasks requires the study of various approaches and their use for subtasks within the overall task of a system. In order to substantiate this, we discuss the problem on three different levels. First of all, the baselines of the general paradigms are discussed. Of course it is not possible to give a complete review of all algorithms, representation schemes, and languages. The presentation aims at the general ideas, advantages, and methodology with respect to the task of object recognition and scene interpretation. Detailed descriptions on pattern recognition techniques are given in [15, 25, 5, 7, 18], artificial neural networks with different models and applications are discussed in [22, 23, 12, 19, 29]. General knowledge representation is addressed in [24, 21, 6, 27]. [2, 16, 3] discussing semantic models for scene understanding. Computer vision methodology is outlined in [1, 2, 13]. Secondly, with respect to this discussions a hybrid representation system for object recognition and scene interpretation is presented. It combines and integrates semantic networks as a knowledge based approach with artificial neural networks.

Finally, we demonstrate the use of this environment for the detection of 2D-objects and for the reconstruction of 3D-scenes.

2 The Basic Techniques

Although object recognition has a long history in the field of pattern recognition and computer vision there is no unique solution. Presently, three paradigms of algorithms are in some sense competing with each other but also work together in what is called hybrid approaches. Before discussing these paradigms, namely statistical methods, artificial neural networks, and knowledge based approaches, the general baselines and goals of each family of algorithms are presented.
The common goal is the automation of perceptive skills. Therefore, the environment of a pattern recognition system is at least restricted by the set of measurable quantities. As a matter of fact, it is not possible to construct a system which is able to interpret all potential measurements in an arbitrary situation. It is necessary to restrict on a certain small type of sensor quantities as well as on a specific task Ω which is called the problem domain. The elements of this set are called *patterns*. Of course, this set is not explicitly available. It may be either described by a representative sample or by explicit knowledge about the domain. Viewed as measurements, a pattern is given by a function $\mathbf{f}(\mathbf{x})$ with analogous or discrete domain and range both possibly of higher dimensions. It is assumed that for one task the dimensions of $\mathbf{f}$ and $\mathbf{x}$ are arbitrary but fixed. However, this restriction must not hold for a complete system. Sensor data fusion can be acquired at different levels of interpretation. For example, complex systems may integrate visual and acoustic signals. But for each of the subsystems the restriction must be fulfilled at least in processing steps which handle the signals or derive features. A pattern is restricted to a certain domain. Only within this domain it has a meaning and can be associated with a concept of the task. Thus, a pattern or object is a triple

$$\mathbf{M} = (\mathbf{f}(\mathbf{x}), \Omega, \mathbf{B}) \tag{1}$$

covering the measured signal $\mathbf{f}(\mathbf{x})$, i.e. the appearance of the object, the problem domain Ω, and its description $\mathbf{B}$. This description does not only depend on the problem domain but also on the specific task of a system. For instance, it may be sufficient for a certain task to detect a set of objects whereas another one also requires the estimation of the object locations and a detailed analysis of its features. The most simple description of a pattern is given by the name of its class. For example, reading machines have to classify letters. In such a situation, we are talking about *simple patterns*. 'If a detailed individual description is asked for we refer to *complex patterns*. Here, features, attributes, parts, and relationships within the pattern build up the description.

Classical statistical pattern recognition deals with classification tasks. The entire pattern given by measurements $\mathbf{f}(\mathbf{x})$ is mapped into a class Ω_k. The name of this class gives the description. Each pattern is assumed to be member of one unique class. The problem domain is completely characterized by a fixed number of classes. The measurements of a pattern are modeled by stochastic processes each one generating a certain class Ω_k. It is assumed that numerical features can be extracted from the sensor data. The features of patterns having the same class should be neighbored in the feature space, and patterns of different classes are separated in this space. The relationship between measurement and classes is given by a suitable number of examples. The parameters of the classification system are adjusted according to a learning sample.

Contrarily, the generation of individual symbolic descriptions of patterns require explicit assumptions about their structural properties. In this sense, a complex pattern has parts which are related to each other and together compose the entire pattern. Parts can be complete objects as a whole or object components which do not occur independently of an entire object. The arrangement of the parts as well as the mutual location of objects are restricted by the problem domain. This fact forms the basis for knowledge based approaches. The collection of restrictions, which are the parts of an object, how they are arranged, which relationships between them must be fulfilled, etc., is viewed as knowledge about the problem domain. Explicit representation of these facts requires a knowledge representation scheme. An adequate representation formalism must be able to cover both structural and numerical properties of an object. Furthermore, it must provide algorithms associated with the scheme to make use of stored knowledge in order to achieve symbolic descriptions efficiently.

Whereas knowledge based techniques are based on explicit models, artificial neural networks claim the opposite way. The use of "know how of the nature" is the overall idea. Principals how animals perceive and act shall form the basis also for technical systems. Taking into account the ability of neural networks to learn they are a flexible instrument for object recognition and computer vision. But similar to statistical approaches a representative sample is required. Learning is equivalent to parameter adjustment in a statistical sense. In this way, neural networks and statistical methods make both use of implicit representation techniques.

2.1 Statistical Pattern Recognition

Statistical pattern recognition techniques are characterized by looking at patterns as high dimensional random variables. Decomposition into parts or structural properties are not taken into account. If such classifiers shall be applied on complex scenes, segmentation is required. Then, each resulting area can be classified as an entire unit. It is mapped onto a class as one entity. The architecture of such a classification system is outlined in Fig. 1. Given the measurement of an entity to be classified a feature vector $\mathbf{c}$ is calculated. This point in the N-dimensional feature space is the argument of a decision function. We assume that the feature vectors are already available and we concentrate on the decision function. The task is to construct a mapping from the feature space into the set of indices characterizing the classes Ω_k. The decision function is denoted by

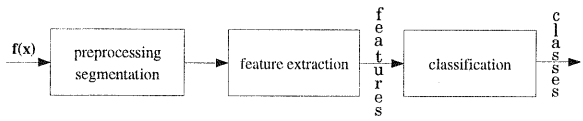

Figure 1: Architecture of a Classification System

$$D(\mathbf{c}) = k, \quad k \in \{1, \ldots, K\} \; . \tag{2}$$

In order to optimize this function with respect to a given learning sample it is convenient to use a decision vector in the following way

$$\mathbf{d}(\mathbf{c}) = (d_1(\mathbf{c}), \ldots, d_K(\mathbf{c})) \quad \text{and} \quad \sum_{k=1}^{K} d_k(\mathbf{c}) = 1 \; . \tag{3}$$

The choice of an optimization criterion determines the functions d_k. Both classical approaches, i.e. minimizing a cost function and approximation of the perfect decision function, will be outlined.

To minimize the costs of decisions it is required that the density functions $p(\mathbf{c}|\mathbf{\Omega_k})$, the a priori probabilities p_k, and the pairwise error classification costs $r_{kl}, 0 \leq r_{kk} < r_{kl} \leq 1$ are known. r_{kl} denotes the costs of a classification of a pattern belonging to class l into class k. The average costs evoked by the decision function are therefore given by

$$V(\mathbf{d}) = \sum_{k=1}^{K} p_k \sum_{l=1}^{K} r_{lk} \int p(\mathbf{c}|\Omega_k) d_l(\mathbf{c}) d\mathbf{c} \; . \tag{4}$$

The costs are minimal if the decision function is chosen to

$$d_k(\mathbf{c}) = \begin{cases} 1 \text{ if } k = argmin_l \{ \sum_{j=1}^{K} r_{lj} p_j p(\mathbf{c} \mid \Omega_j) \} \; , \\ 0 \text{ otherwise} \end{cases}$$

$$D(\mathbf{c}) = argmax_k \{ d_k(\mathbf{c}) \} \; . \tag{5}$$

Based on this general decision optimization, special variants of classifiers can be derived. Restricting the costs to $r_{kk} = 0$ and $r_{kl} = 1, l \neq k$ the Bayes classification rule of maximum a posteriori probability is given. Fixing $p(\mathbf{c}|\mathbf{\Omega_k})$ to be Gaussian results in the normal distribution decision rule.

The perfect decision function

$$\delta_k(\mathbf{c}) = \begin{cases} 1 & \text{if} \quad \mathbf{c} \in \Omega_k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

is approximated by a polynomial decision rule. This function is to be defined according to the learning sample. The decision functions d_k approximating the perfect ones make use of polynomial expansions of the feature vectors $\mathbf{c}$. Given an arbitrary but fixed polynomial expression $\mathbf{x}(\mathbf{c})$ over the coefficients of $\mathbf{c}$ the decision functions are expressed by

$$d_k(\mathbf{c}) = \mathbf{a}_k^T \mathbf{x}(\mathbf{c}) \; . \tag{7}$$

Rewriting in vectorial form leads to

$$\mathbf{d}(\mathbf{c}) = \mathbf{A}^T \mathbf{x}(\mathbf{c}) \; . \tag{8}$$

Learning or adjusting the classification rule is equivalent to the estimation of the parameter matrix $\mathbf{A}$. According to the Weierstrass Theorem, arbitrary functions can be approximated where the accuracy only depends on the degree of the polynomial $\mathbf{x}(\mathbf{c})$. The optimal matrix $\mathbf{A}^*$ is that one minimizing the error between the perfect and the estimated decision rule. Therefore, it has to fulfill the criterion

$$\varepsilon(\mathbf{A}^*) = \min_{\mathbf{A}} E\{(\delta(\mathbf{c}) - \mathbf{A}^T \mathbf{x}(\mathbf{c}))^2\} \; . \tag{9}$$

A closed form solution can be achieved resulting in the simple scheme

$$\mathbf{A}^* = (\frac{1}{N} \sum_{j}^{N} \mathbf{x}(\mathbf{c}_j) \mathbf{x}(\mathbf{c}_j)^T)^{-1} (\frac{1}{N} \sum_{j}^{N} \mathbf{x}(\mathbf{c}_j) \delta(\mathbf{c}_j)^T) . \tag{10}$$

The only assumption is that the matrix to be inverted is not singular. But this is not a serious problem if a representative learning sample and therefore a sufficient number of feature vectors and their corresponding classes are available.

Both classification rules depend on the learning sample. The parameters of the decision rule results from an optimization process. Above, an off-line estimation has been presented. Nevertheless, there exist recursive estimation procedures for both approaches. They can be applied in a supervised or un-supervised training. In the latter case, a sufficient initial estimation is required. An incoming feature vector is classified according to the present parameters. The result is used to update the parameters of a certain class. This class can be the result of classification or can be achieved by a randomized decision which must take into account the values of the decision vector $\mathbf{d}(\mathbf{c})$. It should be pointed out, that both approaches are optimal with respect to the chosen criteria. The semantics of a domain is

reflected by the training sample and the perfect decision rule. An implicit distributed representation of the objects is used.

2.2 Artificial Neural Networks

While most classical statistical pattern recognition systems follow the sequential architecture outlined in Fig.1, the development of architectures based on simple units is one of the main goals of neural networks research activities. As in biological systems information about objects or classes is represented in a distributed fashion. The basic processing of a system is performed by units which adopt models of neurons. Although such artificial neurons do not provide calculations with high precision their mutual interaction results in systems of globally high performance. A network of neurons establishes an ensemble of nonlinear joint processes. Architecture deals with the arrangement of neural units and their synchronization in the network. As a consequence, a system is viewed as a strong interrelationship between structure and functionality. Subnets form specialized modules. Because of the simple basic units, an artificial neural network has a high connectivity and it provides a massive parallel computing environment.

Models for neurons are based on the principle of synaptic summation. The following processing steps are carried out: The input vector $\mathbf{x}$ is manipulated with respect to a weight vector $\mathbf{w}$ giving a scalar value $s(\mathbf{x}, \mathbf{w})$. Then a bias term is subtracted. The third step provides a nonlinear mapping which may be enriched by stochastic processes. Hence, the neural model can be expressed as a function $y(\mathbf{x})$ defined by

$$y(\mathbf{x}) = f(s(\mathbf{x}, \mathbf{w}) - \theta) \tag{11}$$

Frequently used combinations for s are the Euklidian distance or the scalar product of $\mathbf{x}$ and $\mathbf{w}$, whereas the so called activation function f is realized by the sign-function, tanh, Fermi, or Gaussian. Its argument characterizes the state of the neuron. Three major values are distinguished for this state z. If $z > 0$ the neuron is called active, for $z = 0$ quiet, and for $z < 0$ obstructing. Based on such kinds of units an artificial neural network is constructed as a directed graph defined by a set of states Z for each node and a set of states W providing the weights associated with the links between nodes, i.e. neurons, and a set of input and output variables. The state set of a network covering k nodes and k^2 links is represented by

$$\Sigma = Z^k \times W^{k^2} , \tag{12}$$

where one state $\sigma = (\mathbf{z}, \mathbf{W}) \in \Sigma$. The activities are denoted by a vector $\mathbf{z} \in Z^k$ and the weights in a matrix $\mathbf{W} \in W^{k^2}$. According to this connectivity matrix $\mathbf{W} = [w_{ij}]$, the basic types of architectures can be described

- complete connected network: $w_{ij} \neq 0 \forall i, j$
- isolated neurons: $\mathbf{W}$ diagonal matrix
- weak connected network: $w_{ij} \neq 0$ for only a few pairs i, j
- forward connectivity: $w_{ij} = 0$ if $i < j$
- small range connectivity: $\mathbf{W}$ band matrix

The processing behavior of neural networks is characterized by state transitions, i.e. from a state σ^t at time t a new state σ^{t+1} at time $t+1$ is achieved. Like the states, also the transitions are divided into two components. Changes of activities express the short term dynamics of the network. For a certain neuron i it can be described by

$$z_i^{(t+1)} = f_i(s(\mathbf{z}, \mathbf{w}) - \theta_i) \tag{13}$$

Long term dynamics refer to changes of weights according to

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + g_{ij}(w_{ij}^{(t)}, z_i^{(t)}, z_j^{(t)}) \tag{14}$$

Both types of dynamics are of local character and distribute domain knowledge several units. The parameters, i.e. the weights and thresholds shall be learned automatically based on a training sample. Therefore, two a priori decisions are necessary at the present state of the art. First of all a unique type of model neurons has to be selected. Although a few examples of learning the topology for a network exists, in most cases the number of units and their connectivity is also fixed a priori. The training phase therefore adjusts the parameters in a sense comparable to statistical classifiers.

A large number of neural network architectures is covered by the description above. But it should be mentioned, that further types have been developed and are in successful use. As a few examples there are Hebb networks, Kohonen maps, and associative memories. Another type, the so called local linear maps will be described in the context of hybrid systems.

2.3 Knowledge Based Interpretation

Knowledge based techniques are influencing computer vision and object recognition approaches for nearly two decades. The goal is to generate individual symbolic descriptions of domain entities, i.e. objects. Contrarily to classical AI approaches which deal with symbol to symbol transformations semantic models for object recognition are concerned with the transformation of numerical data into symbolic descriptions. It is not possible to achieve the overall goal by optimizing one decision function or by adjusting weights to a given problem, although both techniques presented so far are of great importance

for intermediate steps. The task must be decomposed into several processing steps. But due to the variability of the sensor data of an object and the aim of individual descriptions, the sequence of processing as well as the transformation steps to be applied can not be fixed a priori. Therefore, we have to deal with a search process where for each step the following question must be answered: What is the best transformation at the present state to achieve the overall goal of the analysis process? The search process must be guided and restricted by information about the problem domain and the specific task. This knowledge about objects, events, structural properties, and constraints must be explicitly represented in such away that it can be efficiently used for the interpretation process. A knowledge base for object recognition and description tasks must cover semantic models which enable to establish connections between numerical sensor data and symbolic entities. Fig. 2 reflects the two main lines which must be incorporated in the construction of semantic models and a knowledge base.

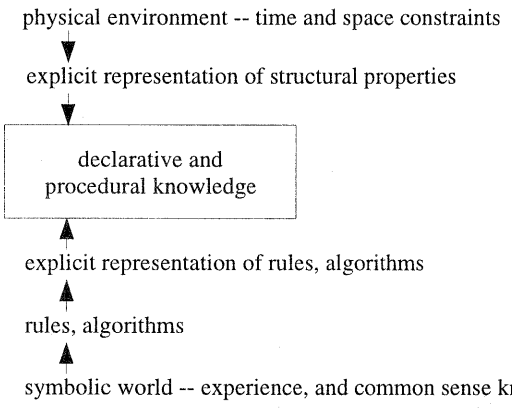

Figure 2: Knowledge for Object Recognition and Description

The base line of knowledge based object interpretation systems is given by a state search approach. The initial state is given by some complex pattern $\mathbf{f(c)}$ and the knowledge base. This covers the semantic models of objects, procedures, and functions which realize transformations between and inside both the numerical and the symbolic world, and inference processes. An inference process provides a state transformation by generating new or manipulating data. If $data(i)$ denotes the knowledge base and already achieved intermediate results of state i and $\mathbf{T} = \{\mathbf{T_1}, ..., \mathbf{T_N}\}$ is the set of transformations, the complete interpretation process can be outlined by a search tree as depicted in Fig. 3. The initial state includes the input pattern and a final state its description. In general, several transformations can be applied to a certain state. They compete with each other and the successful and optimal sequence of transformations forms a path in the search tree.

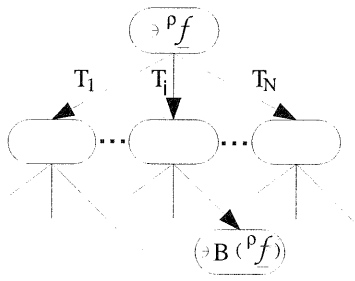

Figure 3: Search Tree of an Interpretation Process

One of the major problems in dealing with such systems concerns their architecture and functional organization. A widely used bases is the decomposition of knowledge based systems into functional modules. An adaptation to pattern interpretation task is shown in Fig. 4. A centralized control module supervises the process by activating suitable transformations and methods with respect to

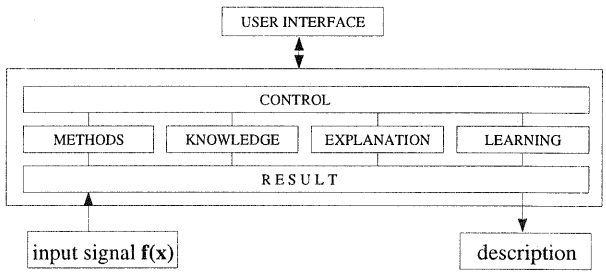

Figure 4: Functional Modules of a Knowledge Based System

the achieved intermediate results and the knowledge base. Further modules for knowledge acquisition, explanation, and user interfaces complete the architecture. An orthogonal viewpoint addresses the hierarchy of processing steps. Fig. 5 shows this model. Each level is characterized by the knowledge and processes available and the results which can be achieved at this processing step. Again, a centralized control module is used. It should be mentioned that this model only determines hierarchies but not necessarily the direction of information flow. Data as well as expectation guided strategies can be implemented.

Organization and use of knowledge is not reflected by both approaches. It is often assumed that a certain problem domain forms a homogeneous uniform type of knowledge or that it can be separated into hierarchies or functional units. Contrarily, inquiries on knowledge representation point out that different types of knowledge must be distinguished. Except declarative models and procedural knowledge, types like definitions, descriptions, and constraints must be separated. Furthermore, each such type provides its own inferences and consequently its results. Each

declarative knowledge	inferences (procedural knowledge)	(intermediate-)results
concepts (terms, objects, events)	data-driven: instantiation	instances
	model-driven: propagation of constraints	modified concepts concepts
attributes (numerical, symbolic features)	data-driven: extraction, combination	values
	model-driven: to constrain	range of values for attributes
scoring calculus	algorithms	values
conditions (structural relations)	data-driven: tests	valid / not valid judgment
	model-driven: to constrain	range of values for attributes modified concepts

Figure 6: Knowledge Types for Image Understanding

	KNOWLEDGE	PROCESSES	RESULTS	
level n	task	generation	output: high level description B	
...	...	...	...	CONTROL
level i	objects	matching	symbolic names	
...	...	...	...	
level 2	segmentation	segmentation	segmented pattern	
level 1	distortion	normalization	enhanced pattern	
level 0	pattern formation	recording	pattern **f(x)**	

Figure 5: Hierarchical Processing Model

one can be used in a top-down and bottom-up fashion. Fig. 6 illustrates such knowledge types for image understanding. The basic entities are concepts modeling objects and events. The models are used for verification and propagation purposes. Features are described by attributes, again both directions of data flow are performed. In a similar way, constraints as relationships between concepts and attributes are established. One of the most serious problems when dealing with noisy uncertain data is scoring. The efficiency of a system strongly depends on an adequate scoring calculus. It provides the hints which transformation should be applied next, which knowledge should be evaluated, and what intermediate results are to be processed.

This depiction of knowledge types forms the basis for a hybrid knowledge representation system as described in the next section. It allows us to handle the complete interpretation process as state search and as an optimization problem regarding a certain scoring calculus. Assuming a knowledge base is constructed according to Fig. 6, the optimal interpretation can be characterized by

$$B^*(\mathbf{f}(\mathbf{x})) = argmax_B\{g(B, \mathcal{W}, \mathbf{f}(\mathbf{x}))\} \qquad (15)$$

where B denotes the implicitly given set of potential descriptions, $\mathcal{W}$ the knowledge base, and g a scoring function of the chosen calculus.

3 A Hybrid Representation System

The development of systems for object recognition and scene interpretation requires the representation of logical and numerical models. Therefore, cooperation of both knowledge based and statistical/neural techniques is necessary. Representations based on the statistical evaluation of a training sample are the backbone for holistic object recognition based on numerical features. Explicitly represented knowledge provides the decomposition of objects into parts and of scenes into objects. Furthermore, it enables the use of constraints and relationships which describe the structural properties of a problem domain and a special task.

A hybrid representation system is described according to the basic discussion above. Its architecture and overall organization of explicit knowledge has been developed regarding the distinction of knowledge types as outlined in Fig. 6. Within this paradigm the integration of analogous models like statistical classifiers and artificial neural networks is achieved in a very natural way. Whereas the knowledge based components deal with structural properties, neural networks are concerned with holistic classification and scoring tasks.

In the next subsection the semantic network language ERNEST is described which builds the framework for the hybrid representation system. Afterwards, the hybrid approach is outlined combining ERNEST and artificial neural networks[1] (ANNs).

[1] In an analogous way statistical classifiers can be incorporated. For simplicity, however, we only refer to ANNs in the following.

3.1 A Semantic Network Language

In contrast to other approaches like KL-ONE or PSN, in the ERNEST semantic network language only three different types of nodes and three different types of links exist. They have well defined semantics and we believe that these structures are adequate to represent the knowledge for different pattern understanding tasks. The node type *Concept* represents classes of objects, events, or abstract conceptions having some common properties. In the context of image understanding an important step is the interpretation of the sensor signal in terms modeled in the knowledge base. The second node type, called *instance*, represents these extensions of a concept. It associates certain areas of the image with concepts of the knowledge base. It is a copy of the related concept where common property descriptions of a class are substituted by values derived from the signal. In an intermediate state of processing instances of some concepts may not be computable because certain prerequisites are missing. Nevertheless, the available information can be used to constrain an uninstantiated concept. This is done via the node type *modified concept* which represents modifications of a concept due to intermediate results of the analysis.

As in all approaches to semantic networks the *part* link decomposes a concept into its natural components (i.e. CAR $\xrightarrow{part}$TYRE). However, in image understanding it often occurs that a certain concept is only defined in the context of another one. For example, if you want to find a spare tyre in an image it only can be identified as a spare tyre in the context of a related vehicle. Contrarily, an ordinary tyre can be recognized without any context as the definition of that term is independent of relationships to other ones. However, the term front tyre is context–dependent as this property can be only determined by an appropriate context. To model this, fact a part can be marked as *context–dependent* and vice versa a context can be explicitly inserted in a concept. That means, SPARE_TYRE is for instance a context–dependent part ($\xrightarrow{cdpart}$) of JEEP and in SPARE_TYRE the concept JEEP is inserted as a possible context. Another well-known link type is the *specialization* which connects a concept with a more general concept (i.e. CAR $\xrightarrow{spec}$JEEP). Closely related to that type of link is an inheritance mechanism by which a special concept inherits all properties of the general one, unless they are explicitly modified. In order to motivate the third link type, the description of aggregation in [14] is reported: "for example, the parts of John Smith, viewed as a physical object, are his head, arm, etc. When viewing as a social object, they are its address, social insurance number, etc." Two *conceptual systems* are distinguished in this example. A concept modeling a person has different parts within each of these systems. Parts in the social system are social conceptions, parts in the physical system are physical conceptions. In complex applications, more than one such conceptual system will occur, i.e., in image understanding, lines, geometry, named objects, or motions. Relationships between concepts belonging to different conceptual systems are only established by the link type *concrete*. Therefore, part and specialization are restricted in the way that they are only allowed inside the same conceptual system. For example, the concepts TYRE and CIRCLE represent terms of different conceptual systems because bar belongs to "named object", while rectangle belongs to "geometry". According to the fact that circle is more concrete to the signal than tyre, the following link TYRE $\xrightarrow{con}$CIRCLE is established.

In addition to its links, a concept is described by attributes representing qualitative or numerical features and restrictions on these values according to the modeled term. Furthermore, relations defining constraints for the attributes can be specified and must be satisfied for valid instances.

The creation of modified concepts and instances constitutes the knowledge utilization in the semantic network. For the creation of instances, this process is based on the fact that the recognition of a complex object needs the detection of all its parts as a prerequisite. For concepts which model terms only defined within a certain context the instantiation process must proceed in the opposite direction. In this case the context must exist before an instance of the context-dependent concept can be created. In the network language, these ideas are expressed by six problem-independent inference rules. Context-independent parts, contexts, and concretes are the prerequisites for the creation of instances and modified concepts in a data-driven strategy. The opposite link directions are used for model-driven inferences. Since the results of an initial segmentation are not perfect, the definition of a concept is completed by a judgment function estimating the degree of correspondence of an image area to the term defined by the related concept. On the basis of these estimates and the inference rules an A*-like control algorithm is applied. For a detailed description of the network language and the control algorithm see [17, 9].

3.2 A Hybrid Approach

To overcome the respective disadvantages of knowledge based and neural techniques we propose a hybrid system combining neural and semantic networks. The main idea is to associate or attach ANNs as holistic models to concepts of the semantic network, with both components modeling the same object[2]. That is, the interface between the different

[2]The same applies to other concepts modeled in the semantic network, like events or abstract conceptions. For sake of simplic-

network types is not defined at one fixed level of the segmentation hierarchy, rather it is determined as appropriate for the given task, knowledge base, or the current state of the analysis process. Given such a hybrid knowledge base, different options are available to recognize a modeled object in a model-driven strategy. If a concept node is to be instantiated the associated ANN can be activated and the object is recognized in a fast and robust way without the necessity to detect the parts of the object as modeled by the semantic network. If no ANN has been attached to a concept node the analysis works in the usual manner pursuing the decomposition hierarchy. In this mode of operation the semantic network is mainly used to control the analysis process and focus the various ANNs attached to the semantic network on different image regions. If in a later phase of the analysis process information about parts and attributes of an object is required which was holistically instantiated by an ANN then the knowledge about the structure of objects modeled in the semantic network can still be exploited. An example for such a situation is the detection of gripping positions to guide a robot hand after the object has been detected holistically by a neural network. In a data driven analysis strategy the interaction works in a similar way. After an object has been recognized by an ANN the corresponding concept can be instantiated even if its parts are not (yet) detected. In a mixed strategy the instantiated objects recognized by ANNs can be used to select appropriate goal concepts from more abstract levels of the semantic network. In this way the number of competing interpretations is drastically reduced and the analysis process can be restricted propagating the constraints from the estimated goal concepts and the instantiated objects.

As indicated above, it is not necessary to attach an ANN to each concept of the semantic network. Rather, one might choose to first train and associate ANNs for objects that occur frequently or that are difficult to recognize by a semantic network. In cases when sufficient training data are not available for a successful training of an ANN, no ANN is bound to the corresponding concept. On the other hand, the hybrid approach gives the option not to fully decompose some of the objects alleviating the effort to acquire and adapt the knowledge base of the semantic network.

Further extensions of the hybrid approach include the utilization of neural networks to compute attributes and judgments during analysis as well as to learn control information to guide the analysis. This gives more possibilities to exploit the learning capabilities and robustness of neural networks for semantic nets. Another option is to explore additional ways to adapt ANNs: As indicated above it is usually not feasible to train an ANN for each object to be expected in a complex scene. However, the results of the analysis of an image sequence can be used to adapt ANNs to objects occurring frequently in the sequence.

ity, however, we only refer to objects in the following.

4 Semantic Models for Object Recognition

The work described in the following is embedded in a special research project studying advanced human-machine communication. The machine should be able to process acoustic and visual input and react meaningfully by producing speech output or by manipulating objects in the environment of the communicating partners. The domain was chosen to be the cooperative construction of a toy-airplane with parts from a wooden construction-kit for children. Object recognition and 3D-scene reconstruction are necessary prerequisites for a robot to grasp parts in a scene. Fig. 7 shows the main part of the hybrid knowledge base solving these tasks.

Currently, the network consists of three levels of abstraction namely the image level (indicated by the prefix I_), the level of perception (indicated by the prefix PE_), and the level of 3D-reconstruction (indicated by the prefix RC_). The concept I_FOCUS mainly allows to focus on certain areas in the image to restrict the object recognition task. This focus can be established by an utterance or a gesture during the construction dialogue (not yet considered at the moment) or by the objects detected so far. This concept has two context–dependent parts namely I_REGION representing a color segmented region and I_OBJECT representing an object hypothesis. According to our hybrid approach both concepts are associated with a numerical classifier performing a holistic instantiation of a colored region or of an object, respectively. Region segmentation is done by a polynomial classifier whereas object detection is done by a special form of neural networks called Local-Linear-Map (LLM) [23]. From a color segmentation algorithm realized on a special hardware platform the neural network gets blob centers as 'focus points'. At each focus point and based on an edge enhanced intensity image a feature vector is extracted by 16 Gabor filter kernels. This is the input for the LLM-network calculating up to three competing object hypotheses [8] For each competing LLM-hypothesis an instance I_OBJECT$^{(I)}$ is created which are stored in competing search tree nodes. Dependent on the object type detected by the LLM-network the corresponding concept in the perceptual level is selected to verify the object hypothesis according to the structural knowledge stored in the semantic network. That means if an instance I_OBJECT$^{(I)}$ with type 'bolt' exists then a modified concept PE_BOLT$^{(M)}$ is created with the concrete I_OBJECT$^{(I)}$. This link is inherited from the concept PE_OBJECT. In the next step the control algorithm

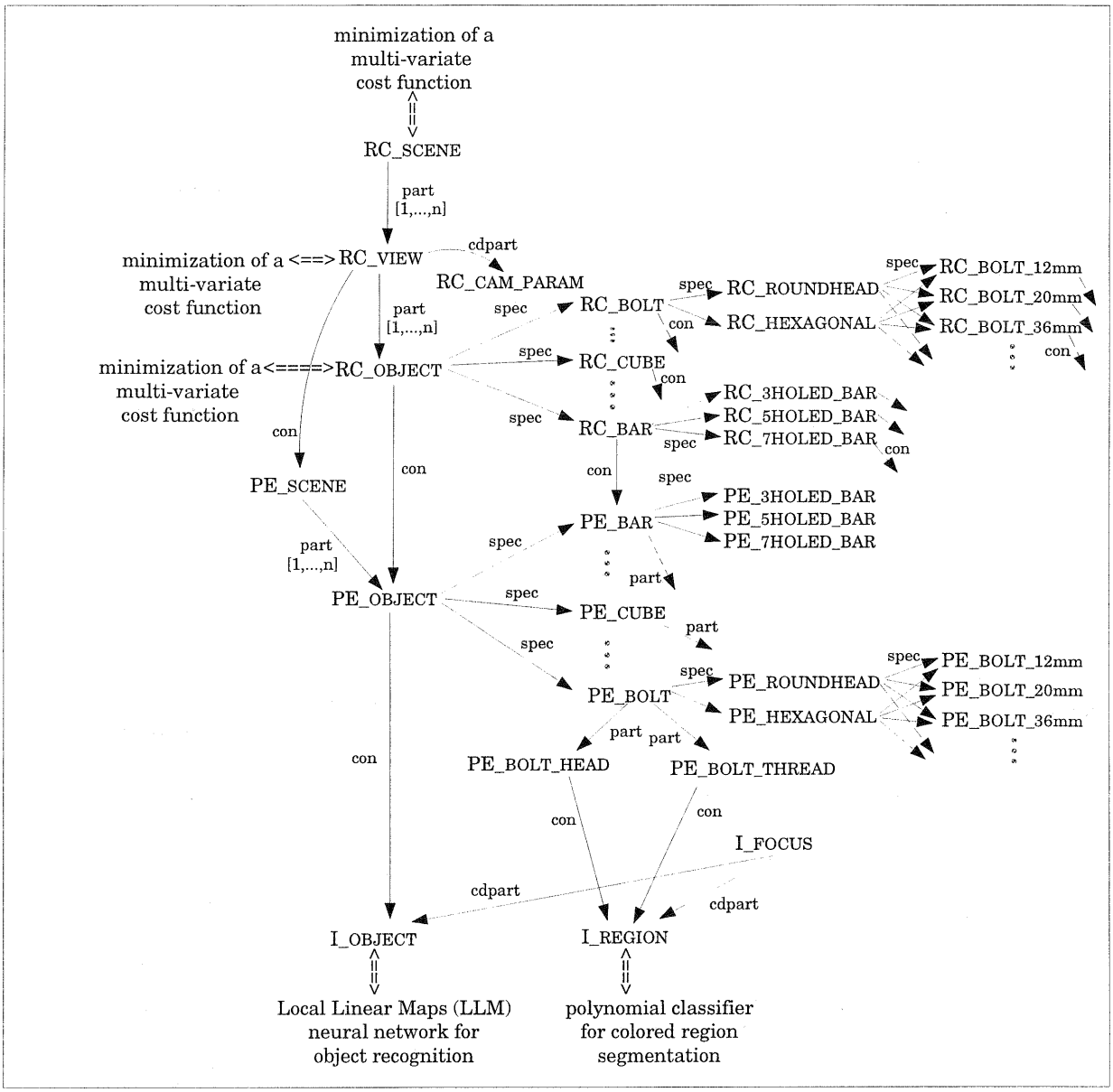

Figure 7: The hybrid knowledge base for object recognition and 3D-reconstruction

tries to detect the parts of a modified perceptual object as they are modeled in the semantic network. For our bolt example this yields in instances for 'bolt head' and 'bolt thread'. Every instance has a concrete of I_REGION which is created by a polynomial classifier of sixth degree using intensity, hue, and saturation as features for classification. During this instantiation process restrictions for position, color, and shape are propagated in a model–driven way. Additionally, the restrictions of the current focus are taken into account. If a successful instance of a perceptual object is created then it is added as part of PE_SCENE which refers to all objects in the scene detected so far. After this step the focus is adapted according to the newly detected object and the next object hypotheses — created by the LLM-network and yielding into instances of I_OBJECT — are processed. In parallel, the corresponding concept on the 3D-reconstruction level can be instantiated activating the individual reconstruction of the detected object. This process is described in the next section.

5 Semantic Models for 3D Reconstruction and Camera Calibration

Depth estimation is a well known problem in computer vision. Various approaches using different assumptions and heuristics try to reconstruct the missing depth information from images. Model-based approaches use a priori known geometric object models. Model-based 3D reconstruction is a quantitative method to estimate simultaneously the best viewpoint of all cameras and the object pose param-

eters by fitting the projection of a three-dimensional model to two-dimensional features detected in the image(s). We represent the 3D model information in the reconstruction level of the hybrid knowledge base (see Fig. 7).
For each object in the domain, there is one concept (e.g. RC_3HOLED_BAR) in the knowledge base where the necessary geometric information is stored. These concepts are linked by specialization links to the generic object concept RC_OBJECT. The same specialization hierarchy exists in the PE-concrete level. So, direct links connect the 3D object models and the reconstructed objects to the recognized objects with all their detected image features. While the concept RC_VIEW collects the reconstructed objects per camera view, the concept RC_SCENE establishes the connection between all camera views (e.g. stereo images) and stands for a 3D representation of the observed scene. The concept RC_CAM_PARAM is a context-dependent part of each camera view. This concept models the external camera parameters and the focal length. Our method holds for one ore more views of the scene. All concepts in the reconstruction level are associated with a numerical model-fitting procedure which minimizes a multi-variate cost functions measuring all differences between projected model and detected image features as a function of the objects' pose and the camera parameters[3]. Common features in the scenes we are dealing with are points and circles.

5.1 Projection of model points

The projection of a model point is the transformation of the point $\boldsymbol{x}_o$ from model coordinates o to the camera coordinate system l and the subsequent projection onto the image plane b_l. This can be expressed in homogeneous coordinates[4] as

$$\begin{aligned}\tilde{\boldsymbol{x}}_{b_l} &= \mathcal{P}^p_{b_l o}(\boldsymbol{x}_o) = \Phi^{-1}\left(\mathcal{T}_{b_l l}\cdot\mathcal{T}_{lo}\cdot\Phi(\boldsymbol{x}_o)\right)\\ &= \Phi^{-1}\left(\begin{pmatrix}\frac{f}{d_x} & 0 & C_x & 0\\ 0 & \frac{f}{d_y} & C_y & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\cdot\right.\\ &\left.\begin{pmatrix}c(\theta)c(\phi) & c(\theta)s(\phi) & -s(\theta) & t_x\\ s(\psi)s(\theta)c(\phi)-c(\psi)s(\phi) & s(\psi)s(\theta)s(\phi)+c(\psi)c(\phi) & s(\psi)c(\theta) & t_y\\ c(\psi)s(\theta)c(\phi)+s(\psi)s(\phi) & c(\psi)s(\theta)s(\phi)-s(\psi)c(\phi) & c(\psi)c(\theta) & t_z\\ 0 & 0 & 0 & 1\end{pmatrix}\cdot\right.\\ &\left.\Phi(\boldsymbol{x}_o)\right) \qquad \text{with} \quad s(x)=\sin(x) \text{ and } c(x)=\cos(x) \qquad (16)\end{aligned}$$

Φ is a function for the transformation from affine to homogeneous coordinates. The projection of a model point in a second image plane b_r needs one additional transformation $\mathcal{T}_{rl}$ from the reference coordinate system which we place in the first camera coordinate system l to the second camera coordinate system r,

$$\tilde{\boldsymbol{x}}_{b_r} = \mathcal{P}^p_{b_r o}(\boldsymbol{x}_o) = \Phi^{-1}\left(\mathcal{T}_{b_r r}\cdot\mathcal{T}_{rl}\cdot\mathcal{T}_{lo}\cdot\Phi(\boldsymbol{x}_o)\right) \quad (17)$$

5.2 Projection of model circles

The perspective projection of circles which are planar figures can be understood as a collineation in the projective plane $\mathbb{P}^2$. The quadratic form of a projected model circle is easily computed using four projected points on the circle and the corresponding cross ratio (see [26] for further details).
The projection of a model circle to the first and to the second image plane are denoted by

$$\tilde{\boldsymbol{x}}_{b_l} = \mathcal{P}^e_{b_l o}(\boldsymbol{x}_o) = \Gamma_b\left(\mathcal{T}_{b_l l}\cdot\mathcal{T}_{lo}\cdot\Gamma_c(\boldsymbol{x}_o)\right) \quad (18)$$

$$\tilde{\boldsymbol{x}}_{b_r} = \mathcal{P}^e_{b_r o}(\boldsymbol{x}_o) = \Gamma_b\left(\mathcal{T}_{b_r r}\cdot\mathcal{T}_{rl}\cdot\mathcal{T}_{lo}\cdot\Gamma_c(\boldsymbol{x}_o)\right) \quad (19)$$

Γ_b is the function realizing the transformation of the projected model circle in homogeneous coordinates to the ellipse representation as center point, radii and orientation. A model circle $\boldsymbol{x}_o$ is characterized by its center point, the radius and a normal vector in model coordinates o. The function Γ_c calculates the four points that are projected and their cross ratio in homogeneous coordinates. This formulation of the perspective projection of a model circle allows us to measure easily the deviation of projected and detected ellipses comparing five parameters.

5.3 Model-fitting

The pose of an object is well estimated from the image data if the value of the non-linear multi-variate cost function

$$\mathrm{C}(\boldsymbol{a}) = \sum_{i=1}^{N}\sum_{j\in B}\left(\boldsymbol{x}_{b_{j_i}} - \mathcal{P}^i_{b_j o}(\boldsymbol{a}, \boldsymbol{x}_{o_i})\right)^T\cdot\mathcal{K}^{-1}\cdot\left(\boldsymbol{x}_{b_{j_i}} - \mathcal{P}^i_{b_j o}(\boldsymbol{a}, \boldsymbol{x}_{o_i})\right) \quad (20)$$

is minimal. The cost function C measures the deviation of projected model features $\boldsymbol{x}_{o_i}$ – these can be points or circles – from the corresponding image features. The vector $\boldsymbol{a}$ contains all unknown parameters. B is the set of images of a scene. N is the number of corresponding model and image feature pairs. Depending on the feature, the vectors $\boldsymbol{x}_{b_{j_i}}$ and $\boldsymbol{x}_{o_i}$ contain different representations and the projection functions $\mathcal{P}^i_{b_j o}$ are the respective transformations. $\mathcal{K}$ is a covariance matrix which is used to model the admissible tolerance with respect to deviation from projected model to detected image features.

[3] The specializations of RC_OBJECT inherit this feature.

[4] Homogeneous transformations are denoted by $\mathcal{T}$ with subscripts indicating destination and source coordinate frame of the transformation.

5.4 Minimization

The main problem of non-linear parameter estimation is to find a method which guarantees convergence of the cost function (eq. 20) to a global minimum. The minimization using the Levenberg-Marquardt method (see [20]), which is a combination of Newton's method and a gradient descent, converges to the nearest local minimum. The global minimum is found with good initial parameter values. However, we do not have initial parameter estimates. Thus, we divide the global model fitting problem into three steps to enhance and monitor the parameter estimates.

Step I: In the first step, the poses of all objects are reconstructed individually, and separately for each camera view. This procedural knowledge belongs to the concept RC_OBJECT and is inherited by every specialization. The projection of one object model depends on 7 parameters. As few parameters are to be estimated, the individual reconstructions are performed very quickly; however the minimizations have to be monitored in order not to let them converge to false local minima because of inappropriate initial values. If the focal length leaves an admissible range (10-100mm in our case), the object is rotated by negating two rotational parameters and the minimization is restarted with the other parameters reset to their original initial values. The cost function is also monitored during minimization. If the process converges to a local minimum with inadmissible high costs, the z-translation parameter is modified according to a predefined scheme. This monitored Levenberg-Marquardt iteration is stopped if either the change of the parameter estimates from one iteration step to the next is less than a given threshold, or if the model fitting does not succeed, i.e. if a maximum number of iterations is reached or if the same local minimum is found despite modified parameter values.

Step II: If a successful instance of a reconstructed object is created then it is added as part of RC_VIEW. This concept performs step II of the minimization process. For a given camera view the median of all estimates of the focal length from step I is fixed at this step and it is used to reconstruct the pose of each object in the scene. So during this step, better initial estimates for objects' poses are derived for each view of the scene.

Step III: The median focal length and the resulting objects' poses of step II are used as initial values for global model fitting. It is possible to estimate the relative pose between different cameras from the object correspondences. This step is part of the procedural knowledge of the concept RC_SCENE. Within this step it is possible to instantiate the concept RC_CAM_PARAM.

5.5 Camera Parameter Estimation

Classical camera calibration methods (e.g. [28]) can not be performed on-line as they demand a special calibration pattern. Depth estimation is then a two-step process and it may lead to suboptimal solutions. We have explicitly modeled the camera parameters in our projection functions and thus they are estimated using the knowledge of the 3D structure of the objects in the scene as part of the procedural knowledge of the concepts RC_SCENE and RC_CAM_PARAM. We estimate the external camera parameters and the focal length. The results show that principal point and scale factors are stable enough for our off-the-shelf CCD cameras to assume fixed values. The influence of lens distortion to the results of our approach is quite small. Nevertheless, it is possible to model the estimation of lens distortion in a manner similar to that of [10].
Tsai [28] shows that full camera calibration is possible with five coplanar reference points. A solution for calibration derived with four coplanar points is unique because four coplanar points determine a collineation in a plane and any further imaginary points in that plane as intersections of lines between lines through the four points can be derived. Six non coplanar points determine a unique solution as well (see [30]).
Scene reconstruction is possible with one camera view. Taking a stereo image leads to much more robust results. Furthermore, the pose of a circle with known radius can not be computed uniquely from one view (see [11]). Taking at least two images for reconstruction, the pose of a circle in space is, if the focal lengths are known, uniquely defined up to the direction of its normal vector (ref. [4]). The sign of the normal can be determined due to the visibility of the projected ellipse.

5.6 Results

Fig. 8 shows the object recognition results and the 3D reconstruction of a stereo image typical for our scenario. In Fig. 8 a) and b) the instances of the corresponding specializations of the concept PE_OBJECT (names in German) and their image regions, obtained by the color segmentation, are visualized. All objects are recognized correctly. Only in the right image the small ring is missing. This is corrected taking the left image in the 3D reconstruction processes. Fig. 8 c) shows the final result of the 3D scene reconstruction (instance of RC_SCENE). The geometric object models are projected onto the right image. The projected object models fit very well to the objects in the images.

6 Conclusion

Based on a detailed discussion of object modeling for object recognition and scene interpretation, a

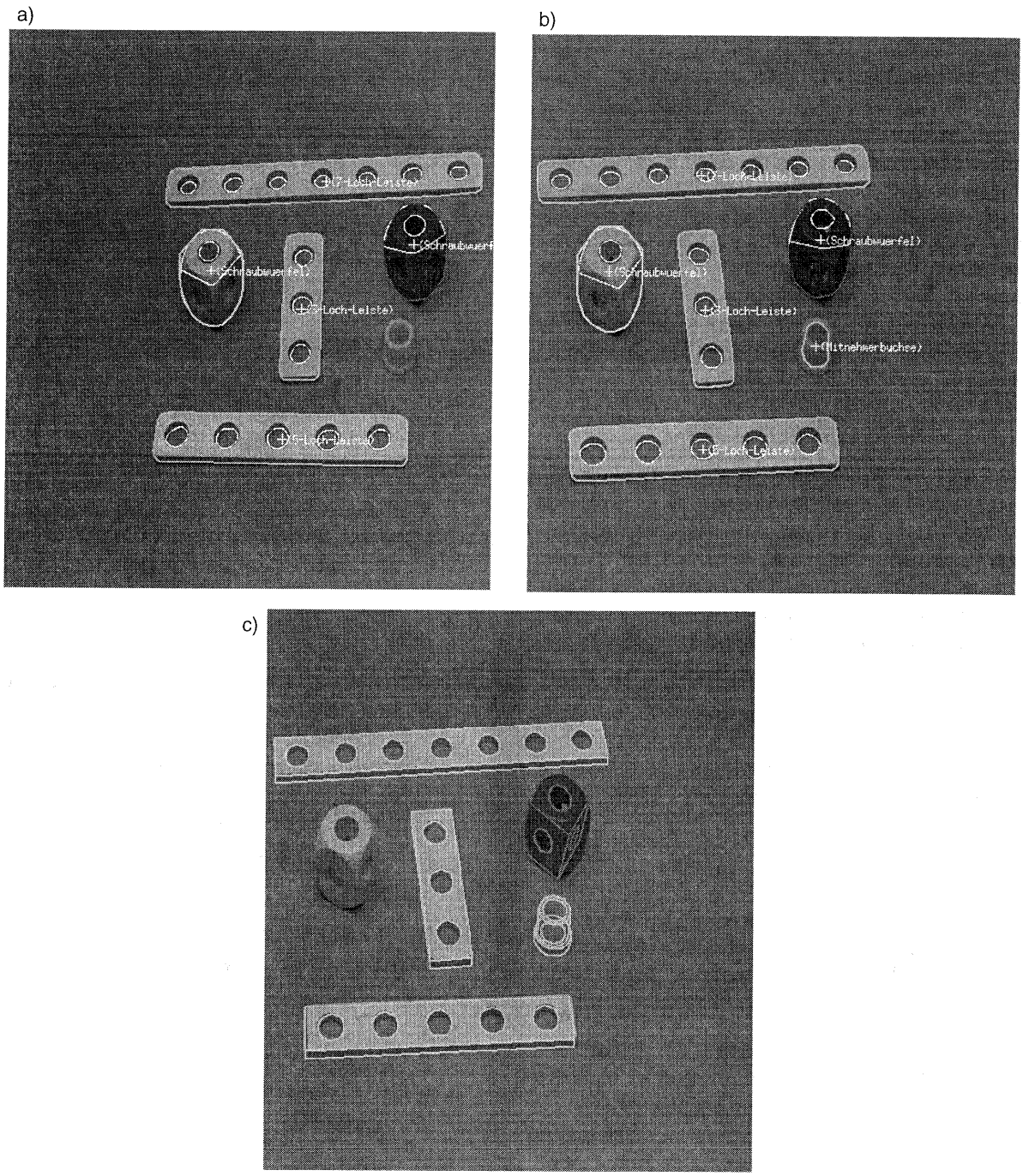

Figure 8: Results for a typical stereo image: a) result of the 2D object recognition for the left image; b) 2D object recognition for the right image; c) result of the 3D reconstruction process projected onto the right image.

hybrid system has been developed. It combines and integrates semantic networks for explicit knowledge representation with artificial neural networks which provide an analogous holistic object representation. Besides the representation formalism, the described system includes problem independent inference rules as well as a judgment based control algorithms. As an example for its abilities and efficiency for the interpretation of complex scenes, a system for the three-dimensional reconstruction of scenes has been presented. Further investigations will integrate image sequences and will also emphasize the use of neural network learning algorithms for the training of symbolic semantic networks.

Acknowledgments

This work has been supported by the German Research Foundation (DFG) in the project SFB 360 "Situated Artificial Communicators". Among the people involved in the topic described, we want to thank especially to Gernot Fink, Gunther Heidemann, and Nils Jungclaus for their contributions on the development of the entire hybrid distributed system.

REFERENCES

[1] J. Aloimonos and D. Shulman. *Integration of visual Modules.* Academic Press, 1989.

[2] D. Ballard and C. Brown. *Computer Vision.* Prentice Hall, New York, 1982.

[3] C. Brown. Issues in selective perception. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 21–30, 1992.

[4] M. Dhome, J. T. Lapreste, G. Rives, and M. Richetin. Spatial localization of modelled objects of revolution in monocular perspective vision. In *Proc. First European Conference on Computer Vision*, pages 475–485, 1990.

[5] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* J. Wiley, New York, 1972.

[6] N. V. Findler, editor. *On the Epistemological Status of Semantic Networks.* Academic Press, New York, 1979.

[7] K. S. Fu. *Digital Pattern Recognition.* Springer Verlag, Berlin, Heidelberg, New York, 1976.

[8] G. Heidemann and H. Ritter. Objekterkennung mit neuronalen Netzen. Report 96/2 – Situierte K"unstliche Kommunikatoren, SFB 360, Universität Bielefeld, 1996.

[9] F. Kummert, H. Niemann, R. Prechtel, and G. Sagerer. Control and Explanation in a Signal Understanding Environment. *Signal Processing, special issue on 'Intelligent Systems for Signal and Image Understanding'*, 32:111–145, 1993.

[10] Mengxiang Li. Camera Calibration of a Head-Eye System for Active Vision. In *Proc. 3rd European Conference on Computer Vision*, volume I, pages 543–554, ECCV'94, Stockholm, Sweden, May 2-6, 1994.

[11] Song De Ma. Conics-based stereo, motion estimation, and pose determination. *International Journal of Computer Vision*, 10(1):7–25, 1993.

[12] H. A. Mallot. Frühe Bildverarbeitung in neuronaler Architektur. In B. Radig, editor, *Mustererkennung 91, 13. DAGM-Symposium München, Informatik-Fachberichte*, pages 19–34. Springer-Verlag, Berlin, 1991.

[13] D. Marr. *Vision.* Freeman, San Francisco, 1982.

[14] J. Mylopoulos and H. J. Levesque. An overview of knowledge representation. In M. Brodie, J. Mylopoulos, and J. V. Schmidt, editors, *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages.* Springer-Verlag, New York, 1983.

[15] H. Niemann. *Klassifikation von Mustern.* Springer-Verlag, Berlin, 1983.

[16] H. Niemann. *Pattern Analysis and Understanding.* Springer-Verlag, Berlin, zweite edition, 1990.

[17] H. Niemann, G. Sagerer, S. Schröder, and F. Kummert. ERNEST: A Semantic Network System for Pattern Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):883–905, 1990.

[18] E. A. Patrick. *Fundamentals on Pattern Recognition.* Prentice Hall, Englewood Cliffs N. J., 1972.

[19] D.A. Pomerleau. Rapidly adapting artificial neural networks for autonomous navigation. In R.P. Lippmann, J.E. Moody, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 429–435. Morgan Kaufman Publishers, San Mateo, CA, 1991.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[21] M. M. Richter. *Prinzipien der Künstlichen Intelligenz.* Teubner Verlag, Stuttgart, 1989.

[22] H. Ritter, T. Martinetz, and K. Schulten. *Neuronale Netze.* Addison-Wesley, Reading, MA, 1990.

[23] H. Ritter, T. Martinetz, and K. Schulten. *Neural Computation and Self-organizing Maps.* Addison-Wesley, Reading, MA, 1992.

[24] P. Schefe. *Künstliche Intelligenz – Überblick und Grundlagen.* Bibliographisches Institut, Mannheim, 1986.

[25] J. Schürmann and U. Kreßel. Mustererkennung mit statistischen Methoden. Vorlesungsskript Sommersemester 1992 und 1993, Daimler-Benz AG, Forschungszentrum Ulm, Institut für Informationstechnik, 1992.

[26] G. Socher, T. Merz, and S. Posch. 3-D Reconstruction and Camera Calibration from Images with known Objects. In D. Pycock, editor, *Proc. British Machine Vision Conference*, pages 167–176, Birmingham, UK, Sept. 11-14, 1995.

[27] J. F. Sowa. *Principles of Semantic Networks.* Morgan Kaufmann Publishers, Inc., Philadelphia, Pennsylvania, 1991.

[28] R. Tsai. A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology using Off-the-Shelf TV Cameras and Lenses. Technical report, Research Report, 1985.

[29] W. von Seelen and H. Janßen. Structural principles in visually guided autonomous vehicles. In *11th International Conference on Pattern Recognition*, volume I, pages 302–311, The Hague, 1992.

[30] J. Yuan. A general photogrammetric method for determing object position and orientation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-5(2):129–142, 1989.