Abstract

Interpretation of complex scenes involves analyzing multiple objects being composed of several parts. Since different objects often have parts in common it is useful to share resources representing these identical substructures. This is quite a difficult task for artificial neural networks (ANNs) [4], but can be handled with semantic networks. They are a well established tool for representing and organizing scene models [1, 7]. However, signal interpretation needs to be robust against distortions and adaptive in different environments. These properties are characteristic advantages of ANNs [12] which suggests to combine them with semantic nets.

In this paper we propose a combination of ANNs and semantic networks attempting to combine the benefits of both approaches. First results on visual recognition of composed objects are promising.

1 Introduction

Describing a given scene is the purpose of computer vision systems. Classification of simple objects from extracted features is feasible without explicit knowledge about the structure of the domain. ANNs show impressive ability to accomplish this using only labeled samples of the domain. The human vision engineer just needs to supply suitable signal preprocessing resulting in a set of potentially useful features, and a labeled set of test patterns.

However, the much more difficult task of describing a scene of complex objects needs an explicit representation of the structural relations between an object and all its components. Furthermore, computer vision systems have to allow explicit structuring using human knowledge.

In artificial domains as well as in natural ones, many objects are built according to some “blueprint”. The objects then consist of similar basic construction elements. This gives the possibility to recognize different objects sharing some of the resources.
Additionally, since there may be more than one object of a certain class in the scene it is necessary to have multiple access to the shared recognizer modules. All these arguments demand for a powerful control mechanism with the representational power of a decomposition hierarchy. This in principle can also be done using ANNs [4]. There are first steps in this direction [18, 16]. However, a copying mechanism for whole neural nets is extremely difﬁcult to realize with ANNs and it can be argued that the need for a special ANN at each position of the signal is far from being convincing as an elegant mechanism, and also is very ineﬃcient. Using semantic networks of conventional technology, much more powerful control structures are available as we will show in this paper.

Besides these technical arguments, from a cognitive science viewpoint, the way human beings analyze things can be separated in a distinction between rational and prerational inference [4]. With regard to human visual cognition, the differentiation between a perception-related level and a concept-related level corresponds to the division postulated in the field of cognitive psychology into parallel, pre-attentive processes on the one hand and attention-driven object integration on the other. This discussion is based in particular on Treisman’s experiments on feature integration theory [17].

This suggests to combine semantic and neural nets in a hybrid system that may serve as a model for the interaction of the two different inference mechanisms.

In the next section we describe the tools and algorithms for the semantic and the neural networks we use as a starting point. Subsequently we propose an integration of both systems and describe an application of the hybrid system in a computer vision task.

2 Formalisms for Neural and Semantic Networks

2.1 The ERNEST-Tool for Semantic Nets

A semantic network is a special graph representing domain knowledge using a decomposition and specialization hierarchy of concepts. In contrast to other approaches like KL-ONE or PSN in the ERNEST semantic network language we are using 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. Concepts, the ﬁrst node type, represent classes of objects, events, or abstract conceptions having some common properties. The interpretation of the sensor signal in terms of concepts of the knowledge base is an important step in the context of image understanding. These extensions of a concept are represented by the second node type, the instance. 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. As in all approaches to semantic networks the part link decomposes a concept into its natural components. Another well-known link type is the specialization with a related inheritance mechanism by which a special concept inherits all properties of the general one. For a clear distinction of knowledge of diﬀerent levels of abstraction the link type concrete is introduced. In addition to its links, a concept is described by attributes representing mainly numerical features and
restrictions on these values according to the modeled term. Furthermore, relations defining constraints for the attributes of a concept and its parts or concretes 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 and concretes as a prerequisite. For the new instance, attribute values are computed and the fulfillment of relations is tested. The required procedural knowledge is attached to the concept by problem dependent functions. If only some of the prerequisites for an instantiation are available it is still possible to compute a modified concept incorporating the available information. This is accomplished by a data-driven modification of concepts activating the same procedural knowledge as above and yields a modified concept with tighter restrictions for valid attribute values. Given a modified concept it is also possible to apply top-down propagation of constraints to parts and concretes using inverse computations of attributes and relations. In the network language of ERNEST, these ideas are expressed by six problem-independent inference rules.

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 see [9, 7, 6].

2.2 Artificial Neural Networks

In ANNs knowledge is represented in an implicit way and can be acquired by relatively simple learning-rules [12]. We use two types of ANNs in our approach. For classifying simple objects based on feature vectors we use Local Linear Maps (LLM) and for representing geometric relationships in 3D-space we use parametrized self-organizing maps (PSOM).

The PSOM is a recently proposed type of ANN [13, 14] that creates a dimension reducing mapping from a feature space onto a non-linear map manifold. The PSOM generalizes the scheme of the Kohonen self-organizing map (SOM) [5]. In contrast to the SOM, which requires the iterative adaptation of a usually rather large number of weight or “reference” vectors before it can be used, a PSOM can be constructed in one step and from a rather small number of reference vectors (which play the role of the “training examples”). This is feasible by representing the map manifold of a PSOM not directly in terms of the reference vectors (as in the SOM), but instead as a superposition of a limited number of “basis manifolds” such that the resulting (hyper)surface interpolates smoothly between the reference vectors. This procedure works well for manifold dimensions of up to about six (as compared to about two or three for the standard SOM), while the embedding (feature-)space may be of much higher dimensionality.

The image of a feature vector is defined as that point of the map manifold that has minimal distance to the input vector. It is determined by a gradient-descent procedure, which can be viewed as the dynamics of a recurrent network with node activities represented parametrically by the map coordinate vector.

In the context of this paper, the most important feature of the PSOM is its ability to complete partial feature vectors. In this case the distance between the input vector and
the map manifold is minimized only in the corresponding subspace of the specified feature components. Let the feature space be \( n \)-dimensional, and the map manifold \( d \)-dimensional. If at least \( d \) components of the feature vector are specified, in the non-degenerate case, this will constrain the possible solutions to a zero-dimensional subset of the map manifold. The associated vector on the map manifold can be considered as the completion of the partial inputs. This property is particularly useful if the map manifold is the graph of a function, since this allows to arbitrarily split the \( n \)-dimensional feature vector into a \( d \)-dimensional subset of independent input variables and \( n - d \) dependent output variables.

Thus the PSOM can be considered as a continuous map which not only can be inverted but be used in a “multiway-fashion”. Therefore PSOMs provide a very convenient tool for the tasks outlined below.

The LLM network architecture [12] is related to the ”self-organizing maps” of Kohonen [5] and to the GBRF-approach [10]. This sort of network consists of units that are significantly more complex than the usually employed sigmoid neurons. Therefore, a moderate number of units is sufficient for many tasks.

Each of these units processes the same input vector \( \mathbf{x} \) of dimensionality \( L \) and computes a node response \( \mathbf{y} \) of dimensionality \( M \). Each LLM-unit is characterized by three components: an input weight vector \( \mathbf{w}_{r}^{(in)} \in \mathbb{R}^{L} \), an output weight vector \( \mathbf{w}_{r}^{(out)} \in \mathbb{R}^{M} \) and a \( M \times L \)-matrix \( \mathbf{A}_{r} \). The matrix \( \mathbf{A}_{r} \) implements a locally valid linear mapping. The node response \( \mathbf{y} \) of a unit \( r \) is determined by

\[
\mathbf{y}_{r} = \mathbf{w}_{r}^{(out)} + \mathbf{A}_{r}(\mathbf{x} - \mathbf{w}_{r}^{(in)}).
\]

For computing the final net output two different variants can be distinguished. If the LLM network acts like a “winner-takes-all” network the node response of one single unit is used as final output. Otherwise, a weighted superposition of several node responses is used. The contribution of each node to the superposition can depend e.g. on the distance between the input vector and the input weight vector of the node and can also be influenced by the previous node response. In this way, a short-term memory-effect might be produced.
3 A Hybrid Approach and its Application

3.1 Attaching ANNs to Semantic Nets

In an earlier project we used ANNs to holistically model objects as an alternative to its decomposition into parts, thus facilitating a direct instantiation of concepts [8]. In this work we pursue a different, complementary approach to integrate ANNs and semantic networks. We will use PSOMs and LLMs to realize procedural knowledge attached to concepts, resulting in a tighter coupling of both techniques.

An important part of the domain specific procedural knowledge is represented in functions computing judgments, attribute values, and the fulfillment of relations for each concept. While the first function is used to guide the A*-like control-algorithm, the later ones reflect the dependencies between attributes and relations of a concept, its parts and concretes, and the sensor data. In computer vision domains, these dependencies are mainly based on geometric considerations. In a pure semantic network approach, the functions are either derived analytically or based on expert knowledge. The first possibility often relies on idealizations of the true relationship, and furthermore deviations due to noise are hard to account for. The problem of knowledge acquisition is even more imminent for the second choice.

In contrast, ANNs have shown robust performance, especially with regard to noise and signal variance. Explicit derivation or acquisition of expert knowledge is replaced by learning from examples. Therefore, we propose to use ANNs as procedural knowledge in our semantic network to exploit the advantages of neural networks. Thus the capabilities of neural nets for robust performance and learning from samples is combine in the knowledge network.
base structured with semantic network techniques. This advantage becomes even more apparent if we consider top-down modification of concepts as mentioned in section 2.1. In this situation, the available information from already computed attributes, parts, and concretes is used in the inverted functions. These inverted functions are of course not inverse in a strict mathematical sense. Rather they facilitate constraint propagation in a top-down fashion according to the current state of the analysis and result in a tighter restriction of valid attribute values. Therefore, for each argument of a function an inverted function has to be supplied in order to exploit this mechanism of constraint propagation. Since PSOMs may be inverted as outlined above (see section 2.2), they can be used to generate prototypical values of unknown input variables.

3.2 The Computer Vision Application

An image with parts of a construction-set is segmented into regions of homogeneous colour [11] (see Figure 2). Shape parameters are computed for each region.

In our approach, objects are modelled by means of individual entities which can be robustly detected and which specify the object redundantly. In this, lighting conditions and perspective are taken into consideration on the perceptive level. In Figure 3, this corresponds to level PE of the knowledge base. An object corresponds to only a few percepts, since only a small number of topologically differing views can be derived from the contour structure of an object.

For example, a rhomb-nut is modelled by its upper side, the hole and the two visible sides in front, in a particular spatial arrangement. In ERNEST, a spatial arrangement of this kind can be represented by a relation between attributes within concepts.

The interface with the segmentation processes on the signal level (level I) in the present system is given by regions of homogeneous colour. For more accurate scene reconstruction and for cases with occlusion contour chains will be added. Using a selective perception mechanism [3] spatially-oriented attention with low resolution (level S) provides an important initial indication for subsequent focusing mechanisms (level A). In the present system the initial cues are given by a global color thresholding step, this is a simplified version of color indexing [2]. Object-related attention is realised in the PE module that has been described above. This architecture allows an interaction of low resolution and colour with subsequent focusing and shape processing. As such, it represents an abstraction and coarsening of biologically motivated architectures which model saccadic eye movements [15].

To concentrate on the main issue, we now describe the part of the semantic network modelling a connecting bar (see figure 1 for the corresponding part of the knowledge base and figure 2 for an image). The bar is a flat wooden cuboid with three holes of identical diameter. Therefore, on the conceptional level of physical objects we have three different concepts in our knowledge base, the bar, the wood, and the hole. On the next conceptional level (features in the image matrix) we model an ellipse and a parallelogram as special cases of regions. The attributes shape parameters and location are inherited via the specialisation links from the concept region. On the level of physical objects, the concepts hole and wood have the three-dimensional position of the center of mass, the orientation in space and size as attributes. The bar has the same attributes and a relation defining the geometrical constraint for the center of mass and orientation of its parts.
Figure 2: A color segmentation into regions
Figure 3: The knowledge base (simplified)
Now we will describe a typical interpretation process and will elaborate on the procedural attachment of PSOMs and LLMs to the semantic net. However, for the sake of clearness, we only explain the optimal path expanded by the $A^*$-control and omit competing interpretations. The problem independent control starts by applying a data-driven rule and seeks to instantiate concepts without concretes and without parts. In our example, competing instances for the concept parallelogram are created and the attached LLM is activated. It takes the shape parameters of each segmented region as input and computes the judgment function estimating the degree of certainty that the region is a parallelogram. Since the only concrete of the concept wood has been instantiated, it is also possible to create an instance $I_W$ of wood. The attributes of wood are again estimated by an attached LLM. Now one part of the concept bar is available and the control creates a modified concept $M_B$ of bar representing the hypothesis of a bar with wood $I_W$ as one part. In this step, appropriate bottom-up constraint propagation restricts valid attribute values of $M_B$ according to the attribute values of the instance $I_W$.

At this state of the analysis the control starts a model-driven pass. The relation arrangement of the concept bar ensures that only geometrically valid arrangements of hole and wood are accepted as bar. This relation is represented by a PSOM. The control applies this PSOM in inverted mode to supply information about missing input values. In tasks of computer vision such a model-driven restriction typically consists of an area of interest in the input image, but many also restrict other attributes like size or colour. Thus, the inverted PSOM computes a certain point as a guess for the location of the missing feature. In our example, three areas of interest for hole ($M_H$) were generated. Thus, exactly one instance ($I_E$) for each modified concept of hole is found on the image feature level. A last bottom-up pass uses the attributes of all the $I_E$ to generate $I_H$ and finally an instance $I_B$ of bar.

### 4 Conclusion

It is commonly accepted, that explicit knowledge representation with appropriate inference and control mechanisms is required for the analysis of complex scenes. Semantic networks are well suited for structuring this knowledge in various hierarchies and to model the necessary features and constraints. While acquisition of the declarative knowledge may be accomplished by a knowledge engineer in quite a natural manner, this is more intricated for the procedural components. One serious problem is the parametrization of these functions to account for e.g. noise and signal variation. Furthermore, to exploit the potentials of flexible control using data and model driven strategies, propagation of the constraints has to be accomplished in both directions. This can not be reduce to standard constraint propagation techniques, but amounts to inverting mappings in continuous domains. However, PSOMs are a new type of ANNs well suited for this problem. A constraint can be represented in a uniform way by one PSOM. This PSOM can be used in a "multiway-fashion" for the "associative" completion of varying sets of missing variables for the problem of constraint propagation. Additionally, PSOMs can be trained from few examples and allow an easy parametrization of the procedural knowledge. Thus, by the proposed integration of semantic and neural networks we combine the benefits of both techniques in a hybrid system.
5 Acknowledgment

This work has been supported by the German Research Foundation (DFG) in the SFB 360 project.

References