A Problem-Independent Control Algorithm for Image Understanding

F. Kummert, G. Sagerer

University of Bielefeld
Technische Fakultät, Postfach 100131
4800 Bielefeld 1, Germany

H. Niemann

University of Erlangen
Informatik 5 (Mustererkennung)
Martensstr. 3, 8520 Erlangen, Germany

Abstract

The paper presents a problem-independent control algorithm for image understanding providing both data-driven and model-driven control structures. By an easy combination of these structures any mixed strategy can be achieved. The basis is a framework for the representation of declarative and procedural knowledge using a semantic network.

1 Introduction

Beside the acquisition and representation of task-specific knowledge the flexible and efficient use of the available knowledge sources is necessary for the automatic interpretation of sensor signals. According to the flow of information and the activity through the representational layers the data-driven (bottom-up) and model-driven (top-down) strategy are the two basic control paradigms. Unfortunately, the appropriate way strongly depends on a specific task-domain. For a problem with unambiguous results of preprocessing a data-driven analysis is suitable, because this leads to a small number of competing interpretations. On the other hand, a model-driven strategy is efficient for applications with a small and/or unambiguous knowledge base, because many hypotheses, incompatible with the model, can be excluded in an early state of the analysis. Otherwise, a mixed top-down and bottom-up strategy should be preferred. Therefore, a problem-independent control algorithm must incorporate both data-driven and model-driven control structures which can easily be combined to any mixed strategy.

In this paper we present a control algorithm taking the above requirements into account. The algorithm is embedded in a framework representing the declarative and procedural knowledge on the basis of a suitable definition of a semantic network.

2 The semantic network language

Contrary to other approaches, in our definition of the network there exist only three different types of nodes and three different types of links. They have a well defined semantics and we believe that these structures are adequate to represent the knowledge of different pattern understanding tasks. To handle such tasks a basic requirement is the ability to represent classes of objects, events, or abstract conceptions having some common properties. This is done by the node type **concept**. In the context of image or speech understanding an important step is the interpretation of the sensor signal in terms modeled in the knowledge base. That means, one connects certain areas of the signal with concepts of the knowledge base. For that reason, the second node type, called **instance**, is introduced representing an extension of a concept found in the sensor data. The instance is a copy of the related concept except that the general description is substituted by concrete values calculated from the signal data. In an intermediate state of processing it may occur that instances of some concepts cannot be computed 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. For a clear distinction between a term and the related model in the network, the following convention is used: the term xyz is represented by the concept *Xyz*. An instance of *Xyz* is denoted by I(*Xyz*), a modified concept by M(*Xyz*).

Like in all approaches to semantic networks there exists a link type **specialization** which connects a concept with a more general concept (i.e. *Car* $\xrightarrow{spec}$ *Jeep*). Closely related to this link is an inheritance mechanism by which a special concept inherits all properties of the general ones. Another well-known link type is **part** which decomposes a concept into its

0-8186-2910-X/92 $3.00 © 1992 IEEE

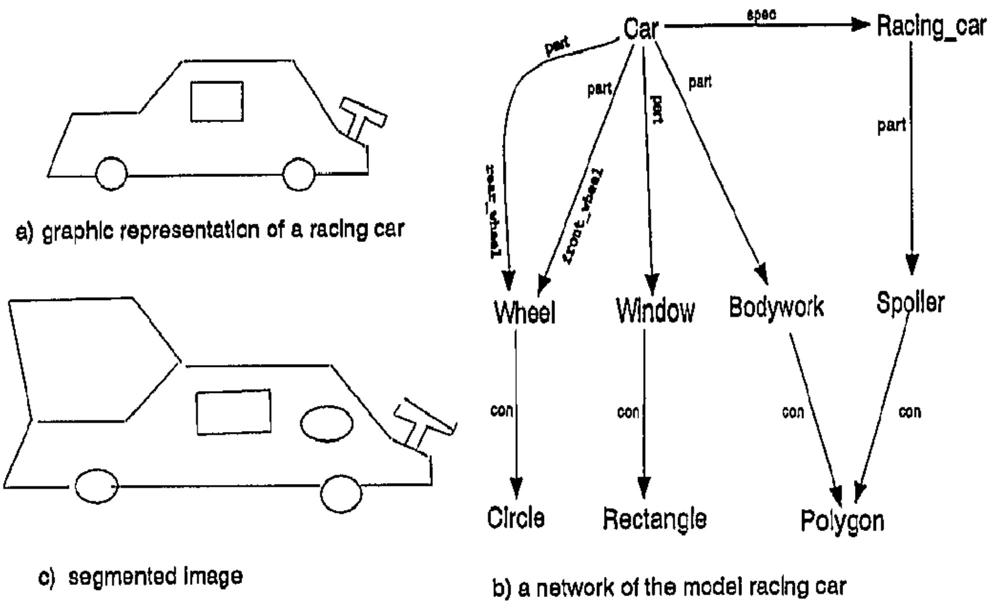

Figure 1:

natural components (i.e. *Car* $\xrightarrow{part}$ *Wheel*). For a clear distinction of knowledge of different levels of abstraction the link type **concrete** is introduced. For example, the concepts *Wheel* and *Circle* represent terms of different levels because wheel belongs to the level "named object" while circle belongs to the level "geometric objects". According to the fact that circle is more concrete to the signal than wheel we introduce the link *Wheel* $\xrightarrow{con}$ *Circle*. Beside the type a link has a particular name expressing the functional role of this link. For example, a link from *Car* to *Wheel* can be characterized by the roles "frontwheel" or "rearwheel".

Additionally, a concept can be described by attributes which represent numerical or symbolic features of a concept. For example, possible attributes for *Car* are height, length, or speed. Furthermore, one can specify relations defining a relationship between different attributes, i.e. "height < length". As the results of the initial segmentation are often not perfect, an instance of a concept may be more or less erroneous. For that reason, the definition of a concept is completed by a judgment function which estimates the correspondence of an area of the sensor signal to the term defined by the related concept.

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 if you have all parts of an object which can be taken apart then you can put it together. In the network language, this idea is expanded to the existance of instances for all parts and concretes of a concept. That means that a concept with no parts and no concretes can be instantiated at once on the basis of the sensor signal. These instances can be used to instantiate more complex concepts and so on. For the creation of a modified concept M(A), the existance of a new modified concept M(B) or a new instance I(B) is sufficient if one of the following conditions is valid: $A \xrightarrow{part} B$, $A \xrightarrow{con} B$, $B \xrightarrow{part} A$, or $B \xrightarrow{con} A$. For a detailed description of the network language see [4, 7].

Figure 1-a shows a graphic representation of a racing car. A simple model of this racing car can be represented in the semantic network in the following manner (see Figure 1-b). *Racing_car* is a specialization of the concept *Car*. *Car* has four parts, namely *Bodywork*, *Window* and two *Wheels* with the functional roles front_wheel and rear_wheel. (For simplicity, the roles of all other links are not shown in Figure 1-b). *Wheel* is concretized by the concept *Circle*, *Window* by *Rectangle*, and *Bodywork* and *Spoiler* by *Polygon*. Due to the fact that a racing car is a special car, the related concept inherits all parts of *Car* and has the additional part *Spoiler*. Figure 1-c shows a segmented image containing hypotheses for circles, polygons, and rectangles. Beside the correct hypotheses, one incorrect polygon and one incorrect circle were detected. In the next section, this image will be analyzed with the problem-independent con-

trol algorithm.

3 A problem–independent control algorithm

Figure 2 shows an outline of a general control algorithm which offers both data-driven and model-driven control features. The algorithm is demonstrated with the example of Figure 1.

initialize the search space by the node $n0$; OPEN := $\emptyset$		
select initial goal concepts C_k		
FOR all C_k		
	generate one successor node nk of the root node $n0$ and insert C_k in nk	
	judge nk due to an optimistic estimation for the costs of a complete interpretation and bring nk to OPEN	
WHILE OPEN $\neq \emptyset$		
	remove the best-judged node n from OPEN	
	IF	(1) the analysis goal is reached for node n
	THEN	stop analysis as n contains the optimal interpretation of the sensor signal
	ELSIF	(2) a new instance can be created
	THEN	create a new instance and generate data-driven modifications up to the goal concept
	ELSIF	(3) all objects of n are instances but the level of interpretation is not sufficiently abstract
	THEN	estimate bottom-up new goals due to the paths in the knowledge base
	ELSIF	(4) a model-driven modification is possible
	THEN	create top-down a new modified concept
	ELSIF	(5) there exist not interpreted signal areas, although n contains a complete interpretation of an appropriate level
	THEN	due to the model, create a modified concept of a specialization

Figure 2: An outline of a problem-independent control algorithm

As image and speech signals are ambiguous, competing instances and thus competing interpretations are calculated. To focus on the most promising interpretation the A*-algorithm is used to direct the analysis. Every node in the search space represents one consistent (partial) interpretation of the sensor signal. Therefore, the search space is initialized by the root node $n0$, and the set OPEN containing the active nodes is the empty set. Then, as starting points of the analysis initial goal concepts have to be selected. According to the level of abstraction a more data-driven or a more model-based strategy is first performed. For example, with the concept *Polygon* (most concrete level) the sensor data can be immediately incorporated into the analysis. This is done by the instantiation of *Polygon*. Such an instance I(*Polygon*) represents a concrete polygon found in the sensor data. On the other hand, an initial goal concept *Car* causes a model–driven strategy, as the expectations of the model determine the further processing. Each goal concept is regarded as a competing hypothesis and therefore is inserted in one successor node of $n0$. To process only promising interpretations each search tree node n is judged on the basis of the concepts, modified concepts and instances collected in n. The initial nodes nk only contain one concept so the judgment is an optimistic estimation of the costs of a complete interpretation. Otherwise, the judgments of the modified concepts and instances are taken into account. An example for a judgment function in the task–domain "speech understanding" is given in [8] and for "object recognition" in [2]. In our small example we select *Polygon* as the single goal concept, so that only one node $n1$ is in OPEN.

After that initialization phase, the A*-algorithm begins to work. While OPEN is not the empty set the best–judged node n is selected for further processing. If the analysis goal is reached, i.e. the entire sensor signal is interpreted on a sufficiently abstract level, the analysis is finished and n contains the optimal interpretation. This is not fulfilled for node $n1$, therefore the next condition is tested. As *Polygon* has no parts and no concretes, the instantiation process is activated. Depending on the results of preprocessing for each detected polygon in the image one instance and one successor node is created (see Figure 3). They represent competing partial interpretations of the image. In the following, every new search tree node is judged and inserted into OPEN. Generally, the instantiation is a data–driven process which creates a new interpretation of a signal area.

For the next iteration we assume that $n2$ is the best–judged node. For this node the third condition is fulfilled and new goals are estimated bottom–up according to the network. By this data–driven process the high-level knowledge is incorporated into the analysis whereby the search process can be reduced. Figure 4 shows two competing search tree nodes after the estimation of new goals. In node $n5$ I_1(*Polygon*) is interpreted as a body-work of a car, while for $n6$ I_1(*Polygon*) is interpreted as a spoiler of a racing car. These hypotheses are expressed by the two modified concepts M_1(*Car*) and M_1(*Racing_car*) and by the two instances I_1(*Bodywork*) and I_1(*Spoiler*). After the first estimation, the new goals usually do not belong to the

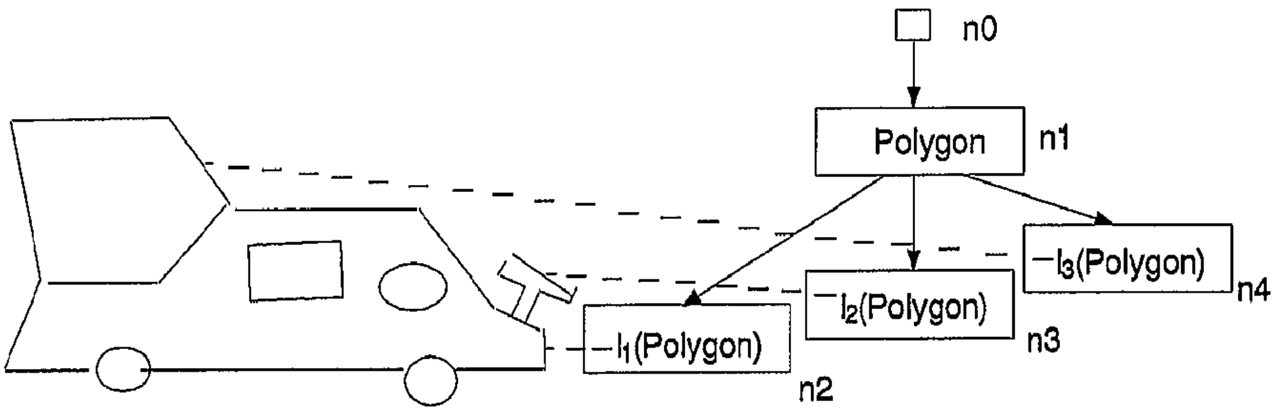

Figure 3: Search tree after the instantiation of *Polygon*

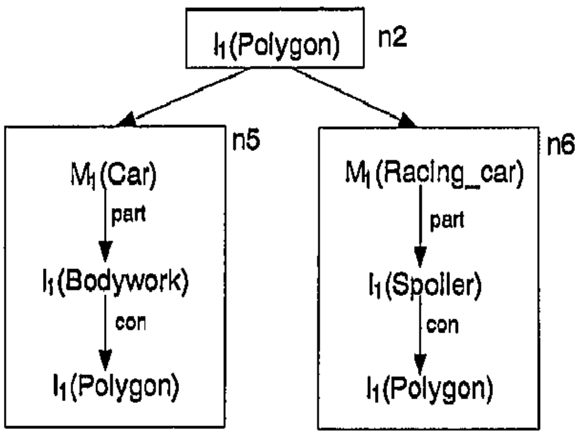

Figure 4: Content of search tree nodes after the estimation of new goals

most abstract level. They only represent intermediate goals which are verified in a model–driven manner. After the verification, new goals in higher levels are estimated. This alternating process is repeated until the desired level of abstraction is achieved. Depending on the length of the estimated path a varying amount of knowledge of the model is used for further processing. Correspondingly, a more or less model-driven strategy is designed.

In the next iteration, for the (best-judged) node *n5* a model–driven modification is feasible. On the basis of M_1(*Car*) a new modified concept for *Wheel* (i.e. with the functional role front_wheel) is generated. With this process all restrictions of M_1(*Car*) are propagated into M_1(*Wheel*). Figure 5-a shows the new generated search tree node and the restriction of M_1(*Wheel*) resulting from the position of I_1(*Polygon*) in the image. In the next step, these restrictions are propagated into M_1(*Circle*), whereby the admissible instances are reduced to the circle in the restricted area. Therefore, the number of competing search tree nodes is reduced, too. Figure 5-b shows the content of a node after this process. By a model–driven modification of *Wheel* (role rear_wheel) and *Circle* the concept *Car* can be instantiated in an efficient way (see Figure 6-a). If the analysis goal is reached for that node the control algorithm stops. Otherwise, the last condition is fulfilled and a modified concept M_1(*Racing_car*) is created (see Figure 6-b). After a top–down modification of *Spoiler* and *Polygon* the corresponding modified concepts and M_1(*Racing_car*) can be instantiated.

In this example the basic properties of the control algorithm are demonstrated. These are the data-driven interpretation in terms of the knowledge base (instantiation, goal estimation) and the model–driven generation of predictions out of the knowledge base (top–down modification, specialization). Depending on the selection of initial goals and on the intermediate estimation of new goals almost any strategy can be achieved. A detailed description of the control algorithm may be found in [1].

4 Realized applications

The successful utilization for different task-domains indicates the quality of the presented control algorithm. The applications cover the interpretation of industrial scenes [2], the diagnostic interpretation of image sequences of the heart [6], the automatic diagnosis of arthrosis of the knee joint [5], and the understanding of spoken language [3]. The obtained results show that the problem-independent control algorithm is able to handle totally different applications in an efficient manner.

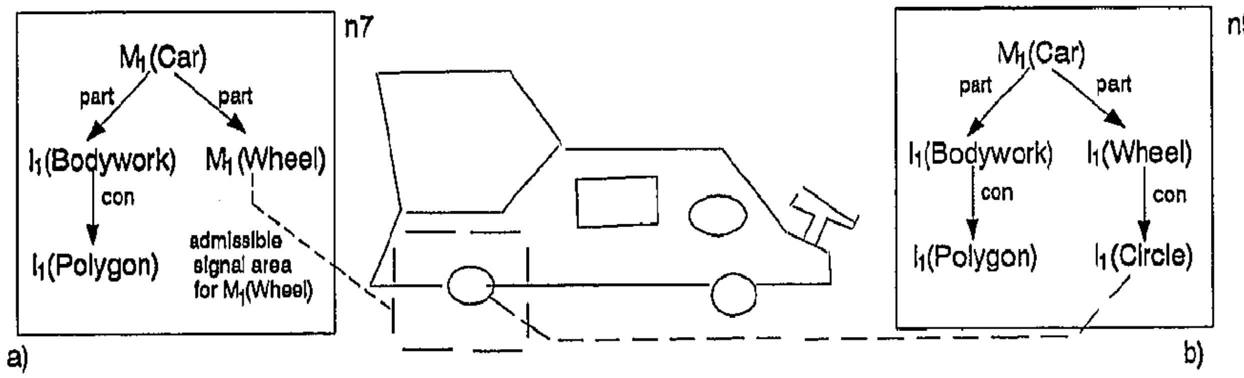

Figure 5: Content of search tree nodes during the analysis

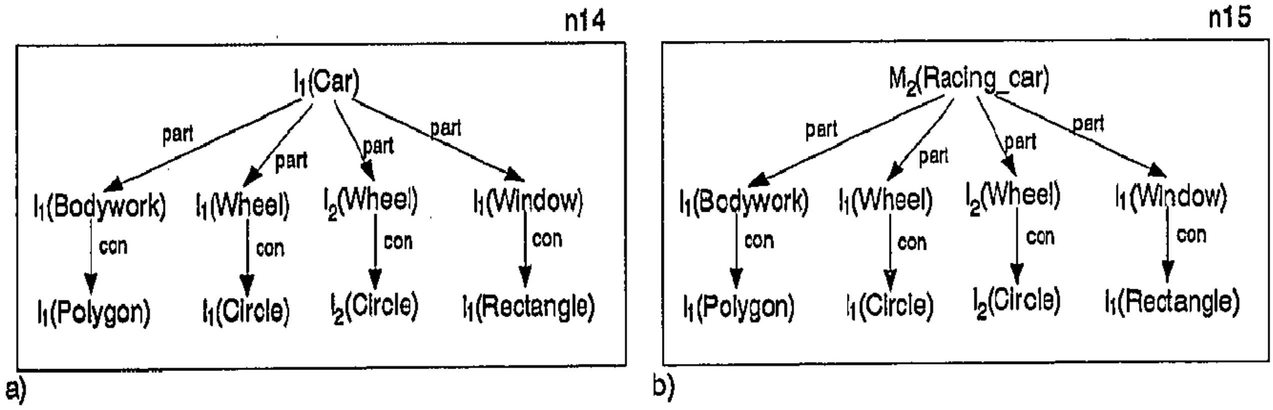

Figure 6: Content of search tree nodes during the analysis

References

[1] F. Kummert. *Flexible Steuerung eines sprachverstehenden Systems mit homogener Wissensbasis.* PhD thesis, Technische Fakultät der Universität Erlangen-Nürnberg, 1991.

[2] H. Niemann, H. Brünig, R. Salzbrunn, and S. Schröder. A knowledge-based vision system for industrial applications. *Machine Vision and Applications*, 3:201–229, 1990.

[3] H. Niemann, G. Sagerer, U. Ehrlich, G. Schukat-Talamazzini, and F. Kummert. The interaction of word recognition and linguistic processing in speech understanding. In P. Laface and R. DeMori, editors, *Speech Recognition and Understanding*, NATO ASI Series F 75, pages 425–453. Springer, Berlin, Heidelberg, 1992.

[4] 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*, 12:883–905, 1990.

[5] H. Niemann, D. Wetzel, P. Weierich, G. Sagerer, and K. Glückert. Methods of artificial intelligence in medical imaging. In *7. convegno internazionale e mostra sulle applicazioni della computer graphics nella produzione, progettazione e gestione (I.CO.GRAPHICS 1992)*, pages 253–260, Milano, 1992. Mondadori Informatica SPS – AICOGRAPHICS.

[6] G. Sagerer. Automatic interpretation of medical image sequences. *Pattern Recognition Letters*, 8:87–102, 1988. Special Issue on Expert Systems in Medical Imaging, Elsevier Science Publisher, Amsterdam.

[7] G. Sagerer. *Automatisches Verstehen gesprochener Sprache*, volume 74 of *Reihe Informatik*. Bibliographisches Institut, Mannheim, 1990.

[8] G. Sagerer, U. Ehrlich, F. Kummert, H. Niemann, and E. G. Schukat-Talamazzini. A Flexible Control Strategy with Multilevel Judgements for a Knowledge Based Speech Understanding System. In *9th International Conference on Pattern Recognition*, pages 788–790, Rome, 1988.