MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

INTERPRETATION OF INDUSTRIAL SCENES BY SEMANTIC NETWORKS

Heinrich Niemann, Heinz Brünig, Richard Salzbrunn, Stefan Schröder
Lehrstuhl für Informatik 5 (Mustererkennung)
Universität Erlangen-Nürnberg

Martensstraße 3
8520 Erlangen, F.R. of Germany

ABSTRACT

An industrial vision system is presented which is designed general enough not only to recognize parts but also to evaluate their quality, inspect the environment, and control a sequence of actions for the placement of the part. The paper concentrates on the three points of segmentation, automatic construction of object models, and identification and localization of parts. It points out that due to the general approach for knowledge-based processing an expansion of the system capabilities is possible.

1. INTRODUCTION

Flexible production systems are important to increase the productivity of labour which in turn is of relevance to the competitiveness of an industrial society [3]. Hence there is a strong interest in the development of expert systems for industrial purposes and robots equipped with sensory capabilities [11,13,15,12]. Among the requirements to a sensory system for industrial purposes are speed, adaptability to new tasks, reliability, and economic efficiency; the applications of such systems are in the areas of assembly of industrial parts, of quality control by visual inspection, and of supervision of the working range of machines.

This paper presents an industrial vision system which is based on a set of general and powerful tools for knowledge-based interpretation of sensor signals using semantic networks (the so-called system shell) and task-dependent algorithms for the efficient execution of a particular task (the so-called application functions). In Sect.2 the main properties of the system shell are described. The task domain of our system is the assembly of small electric motors of the type used in cars. The paper is limited to the recognition and localization of parts of these motors. In Sect.3 we give a brief account of segmentation, in Sect.4 we describe the recognition and localization, and in Sect.5 we consider the automatic construction of object models from camera views and from CAD data. A summary and conclusion is given in Sect.6.

2. SYSTEM SHELL

Image understanding is viewed as a sequence of processing steps transforming the input via several levels of intermediate results to the desired output. In particular we employ an initial phase of mainly data-driven processing to obtain an initial segmentation of an image, and then a mainly model-driven phase of processing to obtain the final result. The relevant data structures for representing results are the initial sensor data $\boldsymbol{f}$ which are a sequence of integer sample values, the initial symbolic description $\mathcal{A}$ which is a network of segmentation objects O, and the task specific symbolic description $\mathcal{B}$, which is a network of instances I. They are computed from concepts C of a semantic network representing a model $\mathcal{M}$ of the task domain. An instance should be consistent with the model and optimally fit to the sensor data.

The central task-independent component of a system is the framework for representation and use of knowledge. We use a *semantic network* employing three types of nodes, that is the concept C, the modified concept Q, and the instance I, as well as six types of links, that is the specialization V, part P, concrete K, instance I, reference R, and model M link. Details of this definition are given in [7,9], some examples of related work are found in [4,5]. A *concept* is a computer representation of some conception, for example, a class of 3D-objects, a motion, a floor plan, or an assembly plan. In addition, a concept may have attributes A, structural relations S, and a judgment vector $\mathbf{G}$. It is a structure

$$\begin{aligned} C &= \Big(D : T_C, (A : (T_A \mapsto F))^*, (M : C)^*, (L : I)^*, \\ &\quad [H_{OBL}, H_{OPT}, H_{INH}]^*, (S(A_C, A_P, A_K) \mapsto F)^*, \\ &\quad (V : C)^*, (R : C^+)^*, (\mathbf{G} \mapsto F)\Big) \qquad (1) \\ H &= \Big((P_{ci} : C^+)^*, (P_{cd} : C^+)^*, (K : C^+)^*\Big) . \end{aligned}$$

An illustration showing a concept, an instance, and some of the links is given in Fig.1. The concept is the only data strucure for knowledge representation. Since it may reference other concepts via its links, a network of concepts may be constructed representing a model of a task domain. In order to increase the efficiency of a representation, obligatory, optional, and inherent components $[H_{OBL}, H_{OPT}, H_{INH}]$ of a concept are distinguished, and in order to also represent context-sensitive relations, context-*ind*ependent and context-*d*ependent parts P_{ci}, P_{cd} are distinguished.

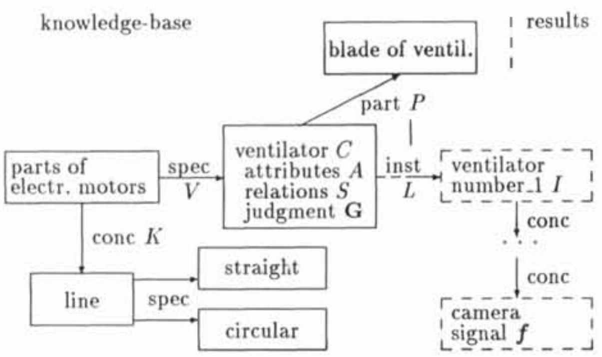

Figure 1: An example of a concept C showing also four out of the six link types

The modified concept has the same data structure as a concept, but its attributes or relations are more restricted due to available instances of other concepts. This allows an adaptation of a knowledge base to intermediate results and increases the efficiency of processing.

The activity of a system based on a semantic network is the instantiation and modification of concepts. There a three rules defining the conditions for the computation of instances, and three more rules defining the computation of modified concepts [9]. Since one of the last three rules defines the computation of modified concepts from results of initial segmentation, there is also a natural and well-defined interface between initial segmentation and knowledge-based processing.

In a system having a large knowledge base and noisy input data, there will be competing and alternative results. Therefore, several of the rules may be applicable to many concepts and modified concepts using different subsets of data. A *control algorithm* of the type shown in Fig.2 selects from among the available alternatives the most promising one. Basically this algorithm is an adaptation of the well-known A*-algorithm [10] to the peculiarities of the above definition of a semantic network. Additional details of the control algorithm are given in [7,9].

Software facilities for the definition of concepts, the implementation of the rules for instantiation and modification of concepts, the control algorithm, and utilities for the creation of an operational system are implemented in the system shell ERNEST [9]. This is used to implement the above mentioned industrial vision system.

3. SEGMENATION

Initial segmentation is a phase of mainly data-driven processing using no task-specific knowledge. It yields an initial description $\mathcal{A}$ of an input image $\boldsymbol{f}$ consisting of segmentation objects O (e.g. lines, regions, vertices) and their relations (e.g. parallelism of two lines or inclusion of two regions). In order to get a simple interface to knowledge-based processing, the definition of a segmentation object is just a simplified version of (1) [8].

provide a list of goal concepts C_{gi}, $i = 1, \ldots, n$ and a list $OPEN$ with nodes v_{gi} and their judgment					
WHILE $OPEN$ is not empty DO:					
	select from $OPEN$ the best scoring node v_k				
	IF	an analysis goal has been achieved			
	THEN	STOP - successful end of search			
	IF	set S of new goal concepts can be provided			
	THEN	new goals and corresponding nodes on $OPEN$			
	ELSE	IF	one object in v_k can be instantiated		
		THEN	instantiate and modify v_k		
		ELSE	IF	there is an object in v_k with an unfulfilled premise	
			THEN	expand and modify v_k, consider inherent links	
			ELSE	consider optional parts and specializations	
STOP - unsuccessful end of search					

Figure 2: The main steps of a control algorithm

The vision system is intended to be flexible and adaptable to new tasks. Hence a variety of segmentation procedures was developed providing contours, lines, regions, and surfaces (by shape from shading). For simple objects already the contour may be sufficient for recognition, for more involved objects additional information from lines, regions, or surfaces may be requested. Fig.3 gives an example of the segmentation of a bearing shield showing the information obtained from a line based segmentation. The reliability of the segmentation has been tested using a variety of images and imaging conditions. The details of the algorithms and data structures are given in [2], the software system for designing algorithms in [1].

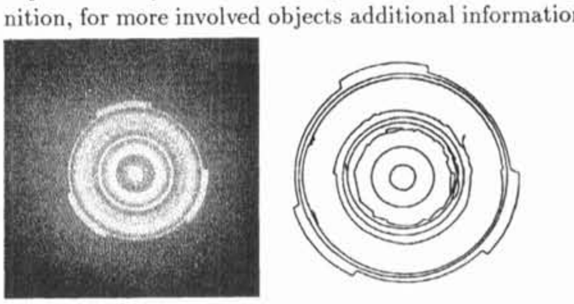

Figure 3: A gray level image and its segmentation

4. RECOGNITION AND LOCALIZATION

In the following we assume that from the set of parts to be recognized a model

$$\begin{aligned} \mathcal{M} &= \langle \mathcal{M}_\kappa = \langle C_{\kappa j}\,,\, j = 1, \ldots, n_\kappa \rangle,\, \kappa = 1, \ldots, k \rangle \\ &= \langle C_{\kappa j}\,,\, j = 1, \ldots, n \rangle. \end{aligned} \tag{2}$$

is available. It consists of a network of concepts $C_{\kappa j}$ represented in a semantic network according to Sect.2. An individual model $\mathcal{M}_\kappa$ is constructed for every object, and then the individual models are attached as optional parts to one model $\mathcal{M}$. The automatic generation of such a model will be considered in Sect.5. Recognition and localization of an object amounts to an instantiation of its corresponding concept $\mathcal{M}_\kappa$, and this is done by the control algorithm of Sect.2. The requirements of matching a segmentation result to a model component and of computing localization parameters are treated as particular attributes A having attached procedures F as shown in (1); during instantiation the computation of the requested attribute values is initiated by the control algorithm.

The recognition strategy is based on a 'hypothesize-and-test' approach. For brevity it is assumed here that only three location parameters (for rotation and translation in a plane) are to be estimated. Hence, one initial correspondence between two lines (or also two initial correspondences between two circular holes) is sufficient for the hypothesize phase. Correspondences are preferably hypothesized between long lines or arcs. Segmentation distinguishes the line types '(straight) line', '(circular) arc', and 'circle'. A correspondence is allowed either between (line OR arc) and (line OR arc) or between circle and circle. Allowing a correspondence between line and arc accounts for the fact that arcs with a large radius may be segmented as lines and vice versa. The initial correspondences are augmented using the 'view-point-consistency' constraint [6]. An object is assumed to be present if a sufficient number of its model elements can be matched. Having found one object the control algorithm automatically looks for further objects since they are modeled as optional components in (2).

The control algorithm uses a judgment function to select among competing nodes in the search space. Every instance $I(C)$ of a concept C has an associated judgment vector $\mathbf{G} = (G_{mod}, G_{seg}, G_{ver})$. Its components are the *model judgment*, the *segmentation judgment*, and the *verification judgment*. The selection of the best hypothesis is done by a quantized lexicographical ordering of $\mathbf{G}$ [8].

If two corresponding lines (or two pairs of corresponding circular holes) have been found, initial estimates for the location parameters are computed. Let $\mathbf{x}_m$ be the vector of a line in the model and $\mathbf{x}_s$ be the vector of a line in the scene. The vectors are scaled to equal lengths. The rotation angle is computed from the scalar product of the vectors, the translation from the difference of their centers. The obtained values may be used to iteratively compute parameters achieving a globally optimal estimate.

Using models from 7 objects and a sample of 100 images each containing up to 3 objects is has been demonstrated experimentally that good reliability and precision are achieved.

5. OBJECT MODELS

The automatic construction of object models is an important point for a flexible vision system. Two input types are available, that is camera images and/or CAD files. The former are useful to also model certain types of segmentation errors and to obtain information about color and texture, the latter provide precise geometrical data without the problems and errors of image segmentation. Hence both types of input are treated.

A model-scheme $\mathcal{S}$ contains available a prior knowledge about the construction of an object model $\mathcal{M}_\kappa$, for example, that a two-dimensional model consists of 2D-lines and -regions, or that a three-dimensional model consists of 3D-lines and -surfaces. The number and geometrical properties of the lines and so on are unknown a priori and are obtained from camera images or CAD data. The model-scheme contains declarative knowledge; a part of the procedural knowledge is task-independent and available in the system shell described in Sect.2, the task-dependent part of the procedural knowledge is contained in functions F attached to concepts in the model-scheme. This reduces the amount of modifications when switching to another type of model, for example, from a 2D to a 3D model, since mainly the model-scheme has to be changed. As an example Fig.4 shows the model scheme used to construct 2D models from gray level images.

An object model is constructed by the following steps:

1) Compute an initial segmentation

$$\mathcal{A}_\kappa = \Phi_1(\boldsymbol{f}_\kappa) \tag{3}$$

as outlined in Sect.3. The segmentation has to correspond to the model-scheme, for example, color, texture, or 3D-surfaces can only be included in the model if the raw data are provided by segmentation.

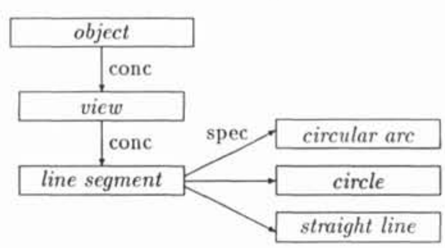

Figure 4: A model scheme for the automatic construction of models from gray level images

2) Transform the initial segmentation to a semantic network giving the *observation description*

$$\mathcal{N}_\kappa = \Phi_2(\mathcal{A}_\kappa)\,. \tag{4}$$

3) Merge the old model ${}^{\rho-1}\mathcal{M}_\kappa$ with the new observation to get the new model containing the *sample description*

$${}^{\rho}\mathcal{M}_\kappa = \Phi_3({}^{\rho-1}\mathcal{M}_\kappa, \mathcal{N}_\kappa). \tag{5}$$

Since the new observation may have arbitrary location, this step requires a localization operation as outlined in Sect.4. But since we assume supervised learning, it does not require a recognition operation.

4) After observing N images of the same object, gener-

alize the model ${}^N\mathcal{M}_\kappa$ to get the final *object model* of an industrial part

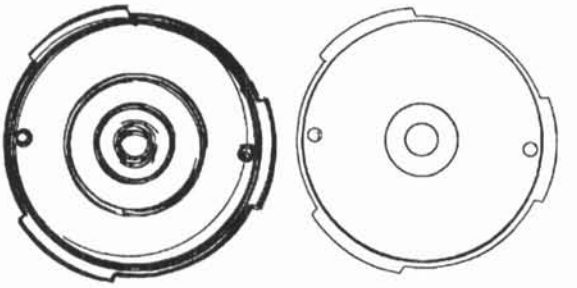

Figure 5: A sample description (left), and the object model derived from it (right)

$$\mathcal{M}_\kappa = \Phi_4({}^N\mathcal{M}_\kappa) . \tag{6}$$

This step mainly eliminates infrequent parts of the model since they are assumed to be segmentation errors, but it also may retain a line and a circular arc as competing model elements.
5) Repeat the above steps for $\kappa = 1, \ldots, k$, that is for all industrial parts to be recognized and attach them as optional parts to a *model* $\mathcal{M}$ as given in (2). The details of the above steps are described in [14].

Fig.5 shows in pictorial form an observation description ${}^N\mathcal{M}$ obtained from $N = 10$ images of the same object in different locations and the object model $\mathcal{M}$ derived from it. It has been verified experimentally that models constructed this way can be used to reliably recognize and locate objects by the approach described in Sect.4.

Constructing models from CAD data does not require a segmentation step. In fact, the symbolic description of the object is already available from the CAD data. Therefore, in this case model construction mainly amounts to extracting the relevant information from the CAD data file and converting it to the semantic network representation. In order to increase the efficiency of the recognition and localization phase, this conversion is not done exactly one-to-one. Rather, it is tried to minimize the number of concepts in the semantic network since this number influences the time of computation.

6. SUMMARY AND OUTLOOK

The paper presented a general framework using semantic networks for implementing a knowledge-based industrial vision system. It illustrated the approach on the example of recognition and localization of parts of electric motors. The three steps of segmentation, object recognition and localization, and automatic model construction were described. An operational system is available. Further work is directed towards integrating the system into an assembly robot and including additional information like color or 3D-segmentation results. Due to the generality of knowledge representation it is possible to also include a complete assembly plan into the knowledge base.

Acknowledgment: The work reported here has been done in cooperation with the Siemens AG, Erlangen, in the project 'PAP'.

References

[1] L. Azzarelli, M. Chimenti, O. Salvetti, H. Brünig, and H. Niemann. Interactive processing and archiving of images. *Image and Vision Computing*, 8:241–250, 1990.

[2] H. Brünig. *Konzeption und Realisierung einer flexiblen Bilsegmentierung für ein wissensbasiertes Bildanalysesystem.* PhD thesis, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, 1990.

[3] K. Feldmann, M. Geiger, U. Herzog, H. Niemann, B. Schmidt, and H. Wedekind, editors. *Fachtagung Rechnerintegrierte Produktionssysteme*, Erlangen, 1987.

[4] N.V. Findler, editor. *Associative Networks.* Academic Press, New York, 1979.

[5] D. Kumar, editor. *Current Trends in SNePS — Semantic Network Processing Systems. Lecture Notes of AI Vol. 437*, Springer, Berlin, Heidelberg, 1990.

[6] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[7] H. Niemann. *Pattern Analysis and Understanding, 2. ed. Springer Series in Information Sciences 4.* Springer, Berlin, Heidelberg, 1990.

[8] H. Niemann, H. Brünig, R. Salzbrunn, and S. Schröder. Interpretation industrieller Szenen mit Methoden der künstlichen Intelligenz. *it — Informationstechnik, in preparation*, 1991.

[9] H. Niemann, G. Sagerer, S. Schröder, and F. Kummert. ERNEST: A semantic network system for pattern understanding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9:883–905, 1990.

[10] N.J. Nilsson. *Principles of Artificial Intelligence.* Springer, Berlin, Heidelberg, New York, 1982.

[11] W.A. Perkins. Inspector: a computer vision system that learns to inspect parts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5:584–592, 1983.

[12] U. Rembold, editor. *Robot Technology and Applications.* Marcel Dekker, Inc., New York, 1990.

[13] P. Rummel and W. Beutel. Workpiece recognition and inspection by a model based scene analysis system. *Pattern Recognition*, 17:141–148, 1984.

[14] S. Schröder. *Integration einer Wissenserwerbskomponente in eine Systemumgebung für die Musteranalyse.* PhD thesis, Technische Fakultät, Universität Erlangen-Nürnberg, Erlangen, 1990.

[15] M. Yachida and S. Tsuji. A versatile machine vision system for complex industrial parts. *IEEE Trans. Computers*, 26:882–894, 1977.