Photogrammetrische Woche 1991
Schriftenreihe, Heft 15, 1991
Institut für Photogrammetrie

KNOWLEDGE BASED IMAGE PROCESSING

H. Niemann, Erlangen

1. INTRODUCTION

There are simple images — like the *Necker cube* shown in Figure 1 — where the *same* gray level distribution allows *two* different interpretations because both of them are consistent with a priori knowledge. There are examples of more complex images or image sequences — like the one shown in Figure 2 — where the non-expert in the field cannot give an interpretation because he does not have the relevant task-specific knowledge.

From these examples it is evident that processing of an image up to a reasonable level of abstraction not only considers the input image pixels, but also relies on a priori knowledge. In general, a system for *knowledge based image processing* (or image analysis, or image understanding) tries to obtain a symbolic description $\mathcal{B}$ (or an interpretation) of an image f which

- optimally fits to the observed data (or pixels)
- and is maximally consistent with internally represented task-specific knowledge.

A more formal definition of the symbolic description is given at the end of Section 3.

Of course, any type of automatic processing requires specialized knowledge on the side of the *designer* to write some type of program, for example, for linear filtering or camera calibration; this is not termed knowledge-based processing in this paper. The relevant point here is that at least some part of the task-specific knowledge is represented explicitly within the *processing system* and is used by it.

There are quite a few publications on knowledge based processing, for example the papers [4, 8, 29, 34, 32, 49] or the books [2, 17, 28, 33, 31].

In this paper we will not give an overview of proposed and implemented techniques, but rather will outline a particular approach in some detail (as allowed by the length of the paper) and refer to alternative or related approaches by means of references. The general *modules* of a system for knowledge based image processing are shown in Figure 3. The four essential modules are the methods for performing an initial segmentation, the representation of intermediate results, the knowledge representation and utilization, and the control of an analysis process (or the determination of a processing strategy). In addition the three modules for knowledge acquisition, explanation, and user interface may be desirable depending on the intended use of the system.

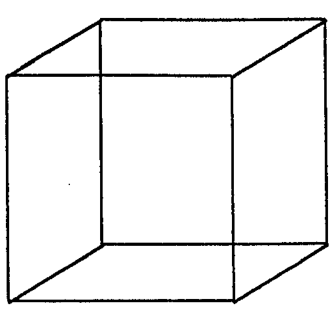

Figure 1: The Necker cube is a simple example of an image allowing two interpretations

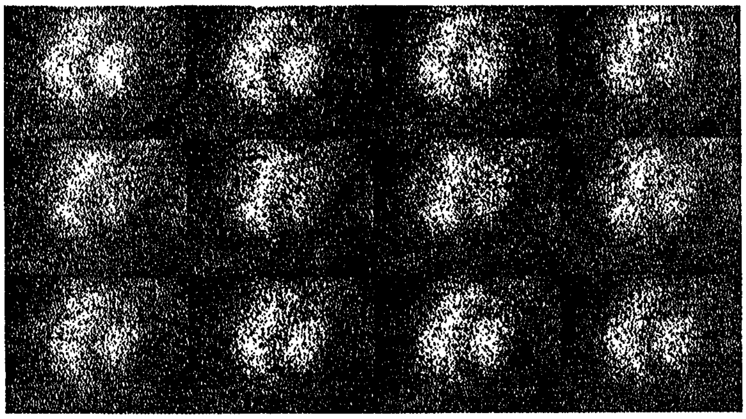

Figure 2: An example of an image whose interpretation requires specialized knowledge (it is a scintigraphic image sequence covering one cycle of the beating heart; it shows normal motility of the left ventricle)

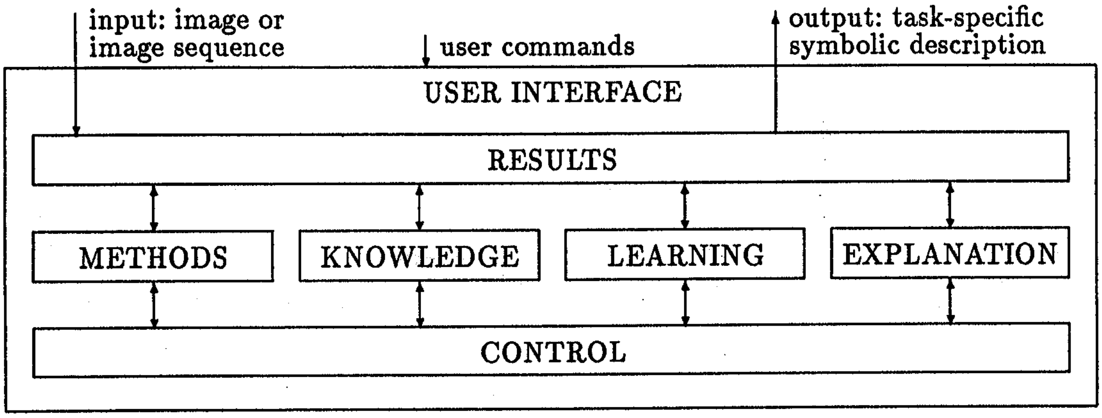

Figure 3: The main modules of a system for knowledge based image analysis

The modules in Figure 3 do not imply a certain order for performing the processing steps; rather these steps are determined depending on the stored knowledge and on the intermediate results by the control module. On the other hand, certain typical processing steps can be stated as shown in Figure 4. Processing starts with an initial phase of mainly data-driven (or bottom-up) processing involving no or little (explicitly represented) task-specific knowledge. The result is an initial segmentation of the image, and this phase is considered only very briefly in Section 2. It follows a phase of mainly model-driven (or bottom-up) processing incorporating task-specific knowledge; this phase is treated in Section 3. In Section 4. we consider the use of a knowledge base to interprete an image. Finally, a conclusion is given in Section 5.

Examples of applications of knowledge based image analysis are autonomous vehicles [1, 12], document understanding [20, 27], industrial applications including robotics [7, 22], medical image understanding [19, 47], and remote sensing [21, 40].

Photogrammetrische Woche 1991
Schriftenreihe, Heft 15, 1991
Institut für Photogrammetrie

KNOWLEDGE BASED IMAGE UNDERSTANDING

mainly model-driven processing using explicit task specific knowledge

4. INFERENCE, ACTION

state description	actions, inferences
mapping to knowledge base	inferences, interpretations
search space	plan or answer generation
present state, goal state, difference	planned action

3. RECOGNITION OF OBJECTS

objects	attributes
volumes, surfaces	symbolic names, attrib., relations
physical objects	for objects, facts, and so on,
facts, events, situations	in space and time, quality

mainly data-driven processing using no task-specific knowledge

2. INITIAL SEGMENTATION

segmentation objects	attributes
line elements, lines	gray level, color, texture
regions, surfaces, volumes	position, depth, motion
vertices, recognition elements	symmetry, shape, quality

1. PREPROCESSING

coding, thresholding, filtering, morphology, normalization, restauration

Figure 4: Typical processing levels in a system for knowledge based image analysis

2. INITIAL SEGMENTATION

The goal of *initial segmentation* is to decompose the image into segmentation objects O *without* the use of explicitly represented task-specific knowledge. The ideal case is that segmentation objects correspond to meaningful objects and parts of objects as percieved by a human observer. However, it seems that this ideal cannot be achieved without the use of task-specific knowledge and hence can only be approximated during initial segmentation. A consequence of this is that segmentation errors may occur in general and must be accounted for in subsequent processing.

According to Figure 4 we assume that this phase works mainly data-driven. The initial segmentation $\mathcal{A}$ is a network of *segmentation objects*

$$\mathcal{A} = \langle O \rangle \,. \tag{1}$$

Examples of segmentation objects are vertices, lines, regions, or surfaces. They have attributes like location, color, depth, or motion. Since, for example, a surface or a region is surrounded by a contour line and the contour may be broken into smaller lines, we consider a segmentation object as a recursive structure

$$\begin{aligned} O \;=\; & (D : T_O, (A : (T_A, \mathcal{R} \cup V_T))^*, (P : O)^*, (K : O)^*, (V : O)^*, \\ & (S(A_O, A_P, A_K) : \mathcal{R})^*, \mathrm{G} : \mathcal{R}^n) \end{aligned} \tag{2}$$

having attributes A, parts P, concretes K, specializations V, structural relations S, and a judgment $\mathbf{G}$. This definition has been chosen to account for the representation of complicated segmentation results and to provide a suitable *interface* to knowledge-based processing, in particular to the concept C in (3). An object-oriented specification and implementation of O is described in [38]. Text books treating various approaches to segmentation are, for example, [2, 31, 39, 41].

Segmentation may treat only the two-dimensional (2D) image or try to recover, for example, lines and surfaces in a 3D scene; it may work on one static image or on a time sequence of images. Hence, there is a large variety of problems and an even larger variety of approaches which cannot be covered in this paper.

3. KNOWLEDGE REPRESENTATION

In a knowledge-based system a suitable formalism has to be specified and implemented for storing the relevant task-specific knowledge. In addition, it is very useful if at least parts of this knowledge can be obtained automatically. Various approaches to knowledge representation have been suggested and are in use, for example, predicate logic [37, 48], fuzzy logic [3, 50], and non-monotonic logic [5], frames [25] and semantic networks [14, 24, 36], production systems and expert systems [9, 28, 30], relational data bases [13], syntactic and structural methods [10, 15, 26], and (artificial) neural networks [42]; not all of them have been tried for image analysis.

Basically, a formalism for knowledge representation in an image analysis system should allow one to represent in the computer a certain *conception* denoting an entity in the real (physical) world, for example, a 'highway', a 'car', an 'accident on a highway', and so on; (in this paper we ignore the representation of abstract conceptions like 'honor' or 'happiness'). The first idea might be to just attach a symbol, like A, to a conception. Obviously, this is insufficient in general because one also would like to represent for a conception its attributes (see above), its relations to other conceptions (like 'behind', 'after', 'parallel to'), its parts (like 'wheels of a car', 'head of a person'), its specializations (like 'sportscar', 'convertible'), and perhaps additional elements. Therefore, we represent a conception by a recursive structure C and call this internal representation a *concept*

$$\begin{aligned} C &= \Big(D : T_C, (A : (T_A \mapsto F))^*, [H_{OBL}, H_{OPT}, H_{INH}]^*, (V : C)^*, \\ &\qquad (M : C)^*, (L : I)^*, (S(A_C, A_P, A_K) \mapsto F)^*, (\mathbf{G} \mapsto F)\Big), \\ H &= ((P_{ci} : C^+)^*, (P_{cd} : C^+)^*, (K : C^+)^*) . \end{aligned} \tag{3}$$

A detailed description of a concept in this sense is given in [36]; here we outline only some basic properties by means of the example in Figure 5

According to the above definition a concept C has a set of *attributes* (or features, properties) A each one referencing a function F which can compute the value of the corresponding attribute. The notation $(A : \ldots)^*$ indicates that there may be an arbitrary number (including zero) of attributes in a concept.

A concept has obligatory, optional, and inherent components denoted by H_{OBL}, H_{OPT}, and H_{INH}, respectively, which in turn consist of context-independent and context-dependent *parts* P_{ci} and P_{cd} as well as of *concretes* K. In the figure only one obligatory, context-independent part is shown. The notation $(P_{ci} : C^+)^*$ indicates that there may be an arbitrary number (including zero) of context-independent

Photogrammetrische Woche 1991
Schriftenreihe, Heft 15, 1991
Institut für Photogrammetrie

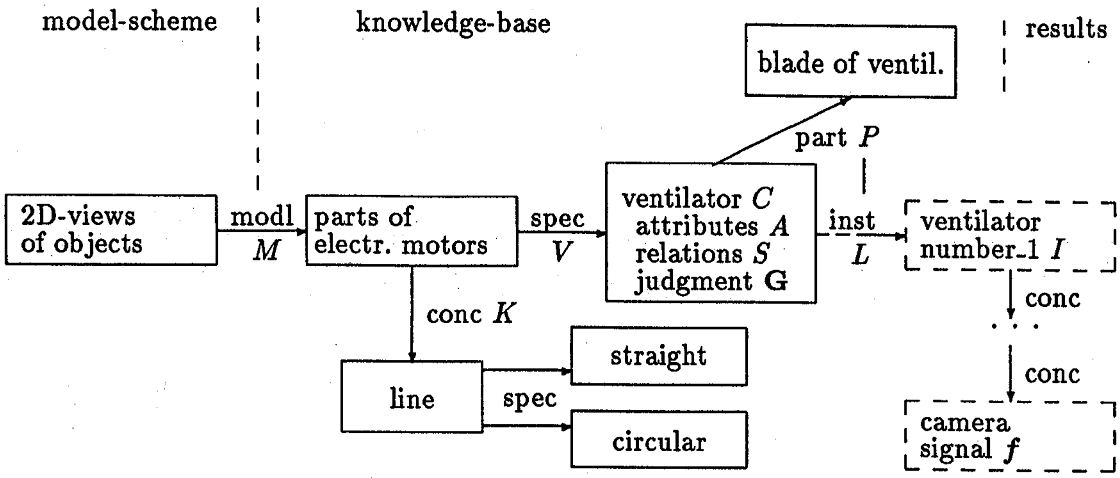

Figure 5: A pictorial example of the concept 'ventilator' and its links to other concepts

parts; each part is related to the concept by a *link*; each link in turn references an arbitrary number of concepts (but at least one).

A concept may have concretes K which are on another level of abstraction or in another conceptual system. In our example the 'parts of electric motors' are determined from 'lines' which are more concrete (or closer to the level of pixels).

In general there may be *structural relations* S between the attributes A_C of a concept or of attributes A_P and A_K of its parts and concretes, respectively. For example, there are certain constraints on the angles of ventilator blades. Since in image processing noise and processing errors are inevitable, relations usually can only be established with limited precision. Therefore, each relation references a function F computing a measure of the degree of fulfilment of this relation, for example, in the sense of a fuzzy relation.

In order to define hierarchies of conceptions it is possible to introduce a concept as the *specialization* V of some other (more general) concept. It is assumed that all attributes, relations, parts, and concretes of the more general concept are *inherited* by the specialized one, unless this is explicitly excluded. This enhances the compactness of a knowledge base. For example, the 'line' is inherited to all concepts which are specializations of 'parts of electric motors'.

In order to support the automatic acquisition of knowledge it is possible to state certain a priori knowledge in a *model-scheme* and reference this by the link *model* M. For example, the model-scheme may state that a view of an object consists of surfaces and holes.

In knowledge-based image analysis one tries to determine which objects and so on are actually present in the image. The occurrence of an object or so on is represented by an *instance* $I(C)$ of the corresponding concept C. The relation between the concept and its instance is represented by a link L from C to $I(C)$. An instance is represented by a structure identical to (3) except that references to functions are replaced by the actual values computed by those functions from the image.

Since due to noise and processing errors the occurrence of an instane can only be infered with limited certainty and precision, a *judgment* G is attached to each concept. The judgment is a vector of numbers measuring the degree of confidence in the actual occurrence of the instance and its expected contribution to the success of analysis.

There may be the situation that some instances have been computed and allow the restriction of attribute values of a concept C which cannot yet be instantiated. In this case a so called *modified concept* $Q(C)$ is created. This way *constraints* can be propagated bottom-up and top-down.

The implementation results in a special type of a semantic network [36, 43]. Related work is found, for example, in [6, 23, 24, 46].

The available task-specific knowledge is represented in a *model* $\mathcal{M}$ which is network of concepts

$$\mathcal{M} = \langle C \rangle . \tag{4}$$

Hence, knowledge is represented homogeneously by concepts related to each other by the various links.

In particular we assume that the goal of image analysis is itself represented by one or more concepts which we denote as the *goal concepts* C_g. The goal concepts might be 'recognition and localization of industrial parts' or 'interpretation of an aerial image'. The description of an image then is represented by an instance $I(C_g)$ of the goal concept. Since every concept has an attached judgment $\mathbf{G}$, there is also a judgment $\mathbf{G}(C_g)$. A formalization of knowledge-based image analysis as stated in the introduction is to request the computation of an optimal instance I^* and define

$$\mathbf{G}(I^*(C_g)|\mathcal{M},\mathcal{A}) = \max_{\{I(C_g)\}} \{\mathbf{G}(I(C_g)|\mathcal{M},\mathcal{A})\} , \tag{5}$$

$$\mathcal{B}(f) = I^*(C_g) . \tag{6}$$

The dependence on $\mathcal{A}$ expresses the fit to image data, and the dependence on $\mathcal{M}$ expresses the consistency with task-specific knowledge required in the introduction.

The essential assumptions here are that it is possible to

- compute a sufficiently reliable initial segmentation $\mathcal{A}$,
- acquire the relevant task-specific knowledge in the model $\mathcal{M}$,
- specify adequate judgments $\mathbf{G}$.

It has been demonstrated in several applications that this can be done, see for example, [34, 35, 32, 43].

An important task is the automatic construction at least of parts of the model $\mathcal{M}$. However, this point has to be omitted due to space limitations. Examples of work in this direction are [11, 44].

4. KNOWLEDGE UTILIZATION

The success of knowledge-based processing depends on the efficient and robust utilization of the available knowledge taking into account the observed image data. In general, *inferences* have to be made depending on $\mathcal{A}$ and $\mathcal{M}$. This can be done in any of the approaches mentioned in the first paragraph of the last section.

With respect to the above discussion and the equations in (5,6) it remains to demonstrate that it is feasible to compute an optimal instance $I^*(C_g)$. The idea of optimization is natural to semantic networks using graph search [31], to neural networks using the iterative optimization of an error or energy function [42], or to synergetic computers using the optimization of a potential function [16]. We outline the optimization in a semantic network.

The fundamental processing activity in a semantic network is the computation of instances. An instance can be computed either if the concept does not have parts and concretes or if there are already

Photogrammetrische Woche 1991
Schriftenreihe, Heft 15, 1991
Institut für Photogrammetrie

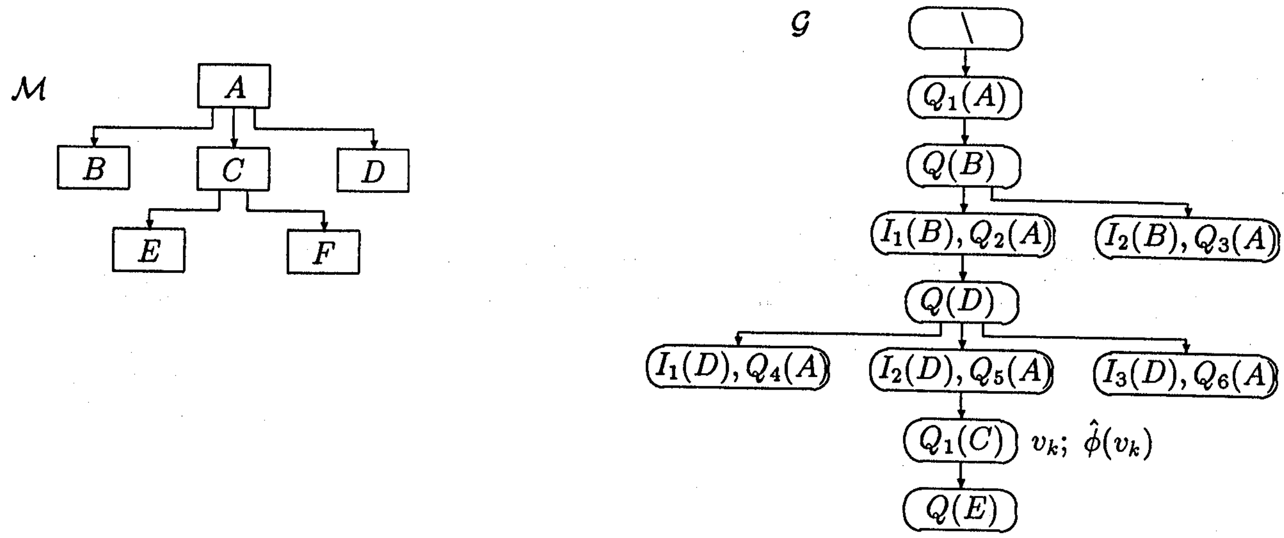

Figure 6: An example of a model $\mathcal{M}$ and a section of the search tree $\mathcal{G}$ generated by modification of concepts and computation of instances

instances of all of its parts and concretes. In the former case we have a so called *atomic concept* and we assume that it can be instantiated using results from the initial segmentation $\mathcal{A}$. In the latter case an instantiation process proceeds from more concrete instances to more abstract ones. A set of rules defining precisely the conditions for instantiation and modification of concepts is given in [36].

The basic idea is to adapt the A^* graph search algorithm to this problem. Figure 6 shows that the computation of instances of concepts in a model generates a search tree $\mathcal{G}$ because for each concept there may be several competing instances due to processing ambiguities. In the figure it is assumed that the goal is to instantiate the concept A. Initially the search tree only contains the empty root node. Since A is not atomic, it is modified to $Q(A)$ (this modification may be emtpy if no intermediate results are available) and appended to the root. Next the model is expanded top-down and the concept B is modified and appended to the search tree. Since B is atomic it can be instantiated which is done next. It is assumed that two instances $I_1(B)$, $I_2(B)$ can be computed which are appended to the search tree. In addition new modifications $Q_2(A)$, $Q_3(A)$ are computed and added to the corresponding search tree nodes. This way the instantiation and modification of concepts proceeds until A is instantiated. Every node v_k in the search tree has a judgment $\phi(v_k)$. An estimate $\hat{\phi}(v_k)$ is computed using the judgments of available instances and modified concepts in this node. At every step of analysis the judgment of the goal node C_g is assumed to be equal to the current estimate of the judgment of the corresponding search tree node.

Since there may be multiple instances, the problem is to find a path in the search tree leading to an optimal instance $I^*(C_g)$ with minimal expenditure. The main steps of an algorithm for doing this are shown in Figure 7. It has a strictly modular representation. The two main classes of modules are task-independent **shell** functions and task-dependent **application** functions. The former can be written once and provided in a system shell. The latter have to be adapted to the task domain and must be written by the system designer. However, some useful default functions can often be defined. At first an **application** function provides a list of goal concepts; possible default functions are to provide a set of atomic concepts (enforcing an initial data-driven phase of analysis), a set of concepts in the top level of concretes (enforcing an initial model-driven phase of analysis), or a set of concepts on a medium

Input: application function to provide a list of goal concepts $C_{gi},\ i = 1, \ldots, n$					
Initialize: application function and shell function to provide list *OPEN* with nodes $v_{gi},\ i = 1, \ldots, n$ and their judgments $\phi(v_{gi})$					
WHILE *OPEN* is not empty DO:					
	select from *OPEN* the best scoring node v_k by application function *select*(*OPEN*) , remove v_k from *OPEN*				
	IF	user function $end\_analysis(v_k)$ decides that an analysis goal has been achieved			
	THEN	STOP - successful end of search or end of resource			
	IF	application function $goal\_conc(v_k)$ defines nonempty set S of new goal concepts			
	THEN	shell function $gen\_goal(v_k, S)$ generates new goals and corresponding nodes on *OPEN*			
	ELSE	IF	one object in v_k can be instantiated		
		THEN	shell function $instant(v_k)$ to instantiate v_k		
		ELSE	IF	there is one object in v_k with an unfulfilled premise	
			THEN	shell function $expand(v_k)$ to expand v_k	
			ELSE	shell function $opt\_spec(v_k)$ to consider optional parts and specializations	
STOP - unsuccessful end of search					

Figure 7: An outline of the main steps of a general control algorithm; it distinguishes actions which are task-independent and can be supplied in a system shell (shell-functions) and actions which are task-dependent and have to be specified by the system designer (application functions)

level of concretes (e. g. the level of individual objects in image understanding). Then an application and an shell function initialize the search space. In particular the judgment of nodes is task-dependent, the initialization of the search tree nodes is task-independent.

After the initialization phase the algorithm starts processing nodes on the list *OPEN*. A application function selects the best scoring node v_k from this list. Processing stops if either the list *OPEN* becomes empty, that is a *failure* of search occurs, or an application function detects that a successful end of analysis occurred. A possible default function is to stop if a concept on a specified level of abstraction was instantiated *and* at least $p\%$ of the image are interpreted. Alternatively, one may require that at most m regions in the image, each one not larger than p pixels, are left uninterpreted. The function may include a limit on computing time and/or number of nodes inspected.

If the best node selected from *OPEN* is not a goal node, the option is offered to provide a set S of new goal concepts by an application function. This is useful to focus the search in the case that the initial list of goal concepts is on a lower level. A purely bottom-up approach in this case might unnecessarily expand large portions of the model. Possible default functions are to define superior concepts (superior with respect to the hierarchy of concretes) in a look-up table or to go s steps in the inverse direction of part and/or concrete links. If new goal concepts are defined, a shell function generates the list of concepts on an instantiation path from the new goal concept to the current goal of the search tree node v_k.

Finally the algorithm takes one of three actions. At first it tries to instantiate some object on node v_k. An object as used here is a concept, a modified concept, or an instance. It is mentioned that a node in the search tree conceptually contains all instances and modified concepts generated so far and also the uninstantiated concepts. If possible, the instantiation is done by a shell function. If no object can be

Photogrammetrische Woche 1991
Schriftenreihe, Heft 15, 1991
Institut für Photogrammetrie

instantiated, the node is expanded by a **shell** function. If no more expansions can be done, it is checked whether optional parts and concretes can be added or whether more special concepts can be included. This is done by a **shell** function.

5. REMARKS

Knowledge based image processing takes into account the observed pixels as well as the available task-specific knowledge in order to obtain a symbolic description (or an interpretation) on a level of abstraction suited to the requirements of an application or a user. A general definition has been given in the form of an optimization problem. It was mentioned that optimization can be done, for example, in semantic networks and in neural networks.

Computations in a semantic network are symbolic and sequential; however, by means of the attached functions F in (3) also numerical procedures can be integrated and various steps of analysis can be carried out in parallel. Computations in a neural network are numerical and parallel; however, it has been demonstrated that also symbolic computations can be carried out (see, for example, [18]) and presently simulations of neural networks usually are carried out on a sequential machine.

A semantic network allows the specification of a general model (e.g., a model of 'images with streets and cars') specifying only the type of objects (e.g., 'street', 'car') and their relations (e.g. 'cars are on streets', 'streets may intersect'), but leaving open their number, relative position, and actual location. Two different images then may have two different descriptions with respect to the represented streets and cars, but nevertheless they can be obtained using the same model and the same control algorithm of the type shown in Figure 7. This requires, among others, the dynamic allocation of new storage space to hold results of a size depending on the content of the image (not on the number of pixels or on the size of the model alone). We do not see this possibility in the neural networks of today.

The A^*-algorithm used above is known to have exponential complexity. Hence, the above control algorithm cannot be just extended to very complex situations, but requires special modifications like pruning or beam-search. It seems that the iterative optimization in a neural network offers advantages here.

Learning from observations is common in neural networks through training algorithms, but requires fairly involved procedures in semantic networks. On the other hand, representing 'textbook knowledge' in the concepts of a semantic network is easily possible, although it has to be done manually since no machine so far can read and understand a textbook; it requires the careful preparation of a training set in neural networks.

The above remarks outline some of the strengths and weaknesses of semantic networks as used in our present work. Of course, in future work we hope to overcome at least some of the weaknesses and add some additional strengths. The work in [45] shows that a special type of a semantic net may be transformed to a neural net, indicating that there is no fundamental difference.

REFERENCES

[1] *Proc. Conf. on Intelligent Autonomous Systems*, 1986.
[2] D.H. Ballard. *Computer Vision.* Prentice Hall, Englewood Cliffs N. J., 1982.

[3] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[4] T. Binford. Survey of model-based image analysis systems. *Int. Journal of Robotic Research*, 1, 1982.

[5] D.E. Bobrow (ed.). Special issue on non-monotonic logic. *Artificial Intelligence*, 13, 1980.

[6] R. J. Brachman and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[7] M. Brady, editor. *Robotics Science.* MIT Press, Cambridge, Mass., 1989.

[8] R. Brooks. Symbolic reasoning among three-dimensional models and two-dimensional images. *Artificial Intelligence*, 17:285–348, 1981.

[9] B. Buchanan and E. Shortliffe. *Rule-Based Expert Systems.* Addison-Wesley, Reading, MA, 1984.

[10] H. Bunke and A. Sanfeliu, editors. *Syntactic and Structural Pattern Recognition: Theory and Applications.* Volume 7 of *Computer Science*, World Scientific Publishing Corp., Singapore, 1990.

[11] J.H. Connell and M. Brady. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31:159–183, 1987.

[12] I.J. Cox, editor. *Autonomous Robot Vehicles.* Springer, Berlin, New York, 1990.

[13] J.C. Date. *An Introduction to Database Systems.* Addison-Wesley, Reading, Mass., 5 edition, 1990.

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

[15] K.S. Fu, editor. *Syntactic Pattern Recognition, Applications.* Springer, Berlin, Heidelberg, New York, 1977.

[16] H. Haken. *Synergetic Computers and Cognition.* Springer, Berlin, Heidelberg, 1990.

[17] A. R. Hanson and E. M. Riseman, editors. *Computer Vision Systems.* Academic Press, New York, 1978.

[18] G.E. Hinton. Preface to the special issue on connectionist symbol processing. *Artificial Intelligence*, 46(1-2), 1990.

[19] K.H. Höhne, editor. *Digital Image Processing in Medicine.* Springer, Berlin, 1981.

[20] H.S. Hou. *Digital Document Processing.* J. Wiley, New York, 1983.

[21] J.R. Jensen. *Introductory Digital Image Processing: A Remote Sensing Perspective.* Prentice Hall, Englewood Cliffs, N.J., 1986.

[22] T. Kanade, editor. *Three-Dimensional Machine Vision.* Kluwer, Boston, 1987.

[23] A. Kobsa. *The SB-ONE knowledge representation workbench.* SFB 314 (XTRA), Memo Nr. 31, Univ. des Saarlandes, FB 10, Saarbrücken, F. R. of Germany, 1989.

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

[25] M. Minksky. A framework for representing knowledge. In P.H. Winston, editor, *The Psychology of Computer Vioson*, pages 211–277, McGraw-Hill, New York, 1975.

[26] R. Mohr, T. Pavlidis, and A: Sanfeliu, editors. *Structural Pattern Analysis.* Volume 19 of *Computer Science*, World Scientific Publishers, Singapore, 1989.

[27] M. Nadler. Survey on document coding and segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 28:240, 1984.

[28] M. Nagao and T. Matsuyama. *A Structural Analysis of Complex Aerial Photographs.* Plenum Press, New York, 1980.

[29] H.-H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing*, 6:59–74, 1988.

[30] A.M. Nazif and M.D. Levine. Low level image segmentation: an expert system. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:555–577, 1984.

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

[32] 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.

[33] H. Niemann and H. Bunke. *Künstliche Intelligenz in Bild- und Sprachanalyse.* B. G. Teubner,

Photogrammetrische Woche 1991
Schriftenreihe, Heft 15, 1991
Institut für Photogrammetrie

Stuttgart, 1987.

[34] H. Niemann, H. Bunke, I. Hofmann, G. Sagerer, F. Wolf, and H. Feistel. A knowledge based system for analysis of gated blood pool studies. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7:246–259, 1985.

[35] H. Niemann and G. Sagerer. *Semantic Networks for Scene Interpretation*. Plenum Press, New York, in preparation.

[36] 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.

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

[38] D. Paulus. *Objektorientierte Bildsegmentierung*. PhD thesis, Universität Erlangen-Nürnberg, Technische Fakultät, Erlangen, 1990.

[39] W. K. Pratt. *Digital Image Processing*. Wiley-Interscience, New York, 1978.

[40] J.A. Richards. *Remote Sensing Digital Image Analysis*. Springer, Berlin, 1986.

[41] A. Rosenfeld and A.C. Kak. *Digital Picture Processing Vol. 1 and 2*. Academic, New York, 1982.

[42] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing*. Volume 1: Foundations, The MIT Press, Cambridge, Mass., eighth edition, 1988.

[43] G. Sagerer. *Automatisches Verstehen gesprochener Sprache*. Volume 74 of *Reihe Informatik*, BI Wissenschaftsverlag, Mannheim, 1990.

[44] S. Schröder. *Integration einer Wissenserwerbskomponente in eine Systemumgebung für die Musteranalyse. VDI Fortschrittberichte Reihe 10: Informatik/ Kommunikationstechnik, Nr. 136*, VDI Verlag, Düsseldorf, 1990.

[45] L. Shastri. *Semantic Networks: An Evidential Formalization and its Connectionist Realization*. Pitman, London, 1988.

[46] J.F. Sowa, editor. *Principles of Semantic Networks*. Morgan Kaufmann, San Mateo, Calif., 1991.

[47] P. Suetens. *Medical Image Processing*. Bellingham, Washington, 1986.

[48] L. Wos, R. Overbeek, E. Lusk, and J. Boyle. *Automated Reasoning, Introduction and Applications*. Prentice Hall, Englewood Cliffs, N.J., 1984.

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

[50] L.A. Zadeh. Knowledge representation in fuzzy logic. *IEEE Trans. on Knowledge and Data Engineering*, 1:89–100, 1989.

ABSTRACT

The goal of knowledge-based image analysis is to determine a symbolic description of an image which is consistent with available a priori knowledge and optimally fits to the observed data. An approach is outlined which consists of the computation of an initial segmentation and of a symbolic description. It is suggested that knowledge is represented in concepts. Knowledge-based processing then amounts to the computation of instances of concepts. The symbolic description is defined to be the optimal (or best scoring) instance of a goal concept. It is outlined how such an optimal instance can be computed.

Photogrammetrische Woche 1991
Schriftenreihe, Heft 15, 1991
Institut für Photogrammetrie

WISSENSBASIERTE BILDVERARBEITUNG

ZUSAMMENFASSUNG

Das Ziel der wissensbasierten Bildverarbeitung ist die Berechnung einer symbolischen Beschreibung, die konsistent mit dem vorhandenen a priori Wissen ist und optimal zu den Bilddaten paßt. Es wird ein Ansatz beschrieben, in dem zunächst eine initiale Segmentierung berechnet wird und dann die symbolische Beschreibung. Wissen wird in Konzepten repräsentiert. Die wissensbasierte Verarbeitung läuft dann auf die Berechnung von Instanzen der Konzepte hinaus. Die symbolische Beschreibung wird als die bestbewertete Instanz eines Zielkonzepts definiert. Es wird dargelegt, wie eine solche Instanz berechnet werden kann.

Prof. Dr.-Ing. H. Niemann
Lehrstuhl für Informatik 5 (Mustererkennung)
Universität Erlangen-Nürnberg
Martensstraße 3
D-8520 Erlangen