PROMAP - A System for Analysis of Topographic Maps *

B. Lauterbach, N. Ebi, Ph. Besslich
Working Group Digital Systems,
University of Bremen, FB1,
P.O. Box 330440, D-2800 Bremen 33, Germany

Abstract

A system for automatic data acquisition from topographic maps (PROMAP- Processing of Maps) is presented. Maps are an important source of information for efficient spatial data evaluation using Geographic Information Systems (GIS). At present a lot of relevant maps have still to be digitized manually, which is a time-consuming and error-prone process. To improve the situation, we developed the PROMAP-system which incorporates adequate image analyzing methods. The system is capable of generating a symbolic description of the map contents that may be imported into a GIS (e.g. ARC/INFO).

1: Introduction

In the past 10 years, Geographic Information Systems (GIS) are gaining importance for storing and organizing spatial information in a computer. In parallel to the advancements in this technology, the quantity of applications has increased. From high-quality cartography to land planning, natural resource management, environmental assessment and planning, ecological research etc., GIS promises to be one of the most extensive computer applications ever to emerge.

For an efficient and flexible use of these systems it is necessary to combine data acquisition, development of an evaluation scheme and GIS in an integrated concept. This system integration is the objective of the interdisciplinary project *Environmental Planning System*. For the corresponding investigations a project group was constituted by researchers affiliated with the working groups *Digital Systems* and *Physio-Geography* of the University of Bremen.

The evaluation scheme to be developed for solving geographic problems of the project [1] refers to the determination of output capacity of soils, protection-grade of certain soils and evaluation of site-suitability. For that, it is necessary to develop an operationable, effective catalog of geo-indicators and hierarchical evaluation matrices.

For an efficient spatial evaluation it is necessary to have an adequate data base. Different types of maps are an important source of information for this. For our application, most of the relevant maps are only available in paper-based form. The necessary manual digitization requires considerable time and cost because of the large number of manual operations. For increasing efficiency it is useful and challenging to automate the process of extracting information from maps by use of scanning and applying image analysis methods [2]. Several systems have been developed for solving the task of automatic data extraction for special types of maps. These systems require as input data already separated map layers [3] or maps of one color (e.g. cadastral maps) [4], [5], [6]. For our application, we are using German topographic maps of scale 1:25 000 and 1:5 000 which are printed in four colors printing technique. Fig. 1 shows a test scene scanned from a topographic map of scale 1:25 000 containing the colors black, light green, brown and blue (for printing purposes reduced to a grey level image).

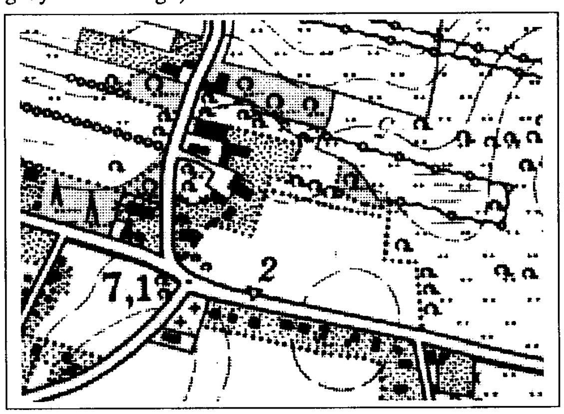

Fig. 1: Example of a topographic map scene (original size is 25x18mm resp. 1039x759 pels, reduced to grey level).

We developed appropriate methods for raster data processing, knowledge organization and knowledge use [7]. The main ideas of raster data processing are scanning using a 24-Bit-RGB-scanner, separation of color layers, raster symbol and raster object recognition and vectorization. The principles of knowledge-directed interpretation are those of prototypes (con-

* This work is supported in part by DFG (German Research Association) under contract Stäblein/Besslich: Environmental Planning System (Sta 126/19-1).

0-8186-2840-5/92 $03.00 © 1992 IEEE

cepts) as the basic representation building block, generalization and aggregation as abstraction mechanism. The corresponding methods are realized in the PROMAP-system.

2: System overview

An overview of the complete environmental planning system is shown in Fig. 2. The system kernel contains modules for storage, modification, manual digitization, graphic representation and evaluation of spatial data. Considering the complexity of the system kernel we preferred to use a commercial GIS. We decided for ARC/INFO from ESRI, because it is a powerful tool and is in wide-spread use over Germany.

The acquisition of spatial data is done by means of knowledge-directed topographic map analysis (PROMAP-system) and interactive input of additional information obtained from soil analysis, remote sensing techniques, terrain mapping or other sources. The system user will have the opportunity to evaluate the spatial information via an expert system having access to the data base.

The principle of the proposed knowledge-directed image analysis is shown in Fig. 3. The topographic maps we use are printed in four colors printing technique. For this reason, a raster image of the map to be processed is created using a 24-Bit-RGB color scanner. The smallest objects contained in the topographic maps are raster dots (e.g. meadow texture) of size 0.05 to 0.1 mm. Therefore, the minimum scanning resolution is 800 to 1000 dpi.

Symbolic image information is extracted by splitting the map image into color layers which are processed using raster object recognition methods and vectorization. The extracted symbolic information, represented by so-called *attributed structure primitives* serves as data source for the knowledge-directed image analysis. The analysis is realized by the control module, the image model (concept base), the instance base and the image based conflict solving module. The instances represent the extracted map information and have finally to be converted to the ARC/INFO data base format (module *data conversion*, cf. Fig. 2).

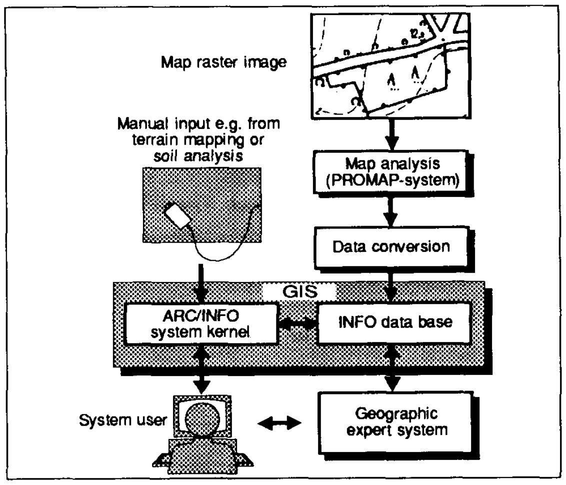

Fig. 2: System overview.

An advantage of applying color layer separation to RGB raster images of maps is the possibility of scanning and processing only the area of interest of the paper-based map. It is not necessary to scan a whole map or the complete set of binary map layers of which the paper-based map is produced. If binary map layers are used, the complete layers have to be digitized. Otherwise there would not be any control marks available to support spatial registration of ras-

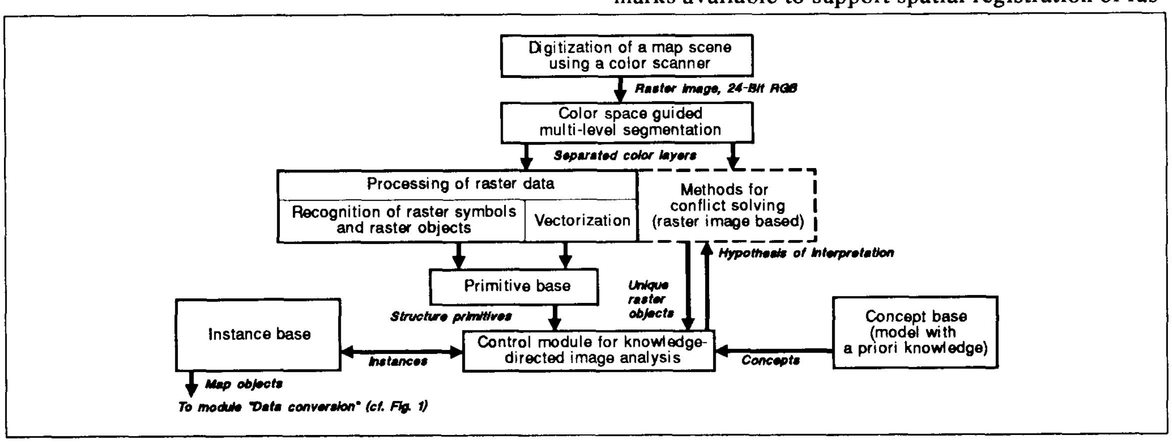

Fig. 3: Principle scheme of the PROMAP-system

ter data. In case of absence of control marks, it is in general not possible to define suitable control marks for a spatial registration because of insufficient interrelations between the map layers. The described advantage together with the analysis methodology incorporated in the PROMAP-system guarantees a high flexibility for analysis of topographic maps.

3: Processing of raster data

The raster data of the map is processed to create a symbolic attributed description (*attributed structure primitives*) of all basic elements contained in the map (vectors, symbols, regions). This is done in five steps. The first step is the separation of the color layers contained in the map. The separation corresponds to the reverse process of the composition of binary map layers during printing. In opposition to map production process, the separation will result in layers containing information printed using the same color instead of layers containing information of the same type (e.g. symbols, roads). Fig. 4 shows the color layers BLACK, LIGHT GREEN, BROWN and BLUE that have been separated from the original RGB image of Fig. 1 using the methods described in section 3.1.

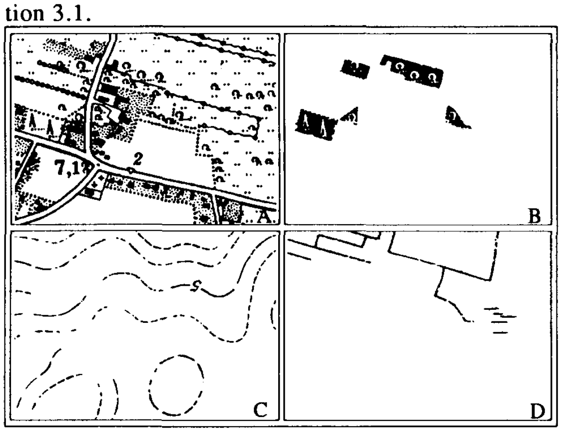

Fig. 4: Color layers BLACK (A), LIGHT GREEN (B), BROWN (C) and BLUE (D) separated from image shown in Fig. 1.

The second step is the recognition of raster symbols and objects in the color layers. Recognized raster symbols and objects are then removed from the layers. The third step is separation of layers containing mainly region and line information. These two types of layers have to be processed in a different way. A different strategy has to be used to cut off symbols (e.g. houses) from lines (e.g. road borders) if they are connected to each other. The fourth step consists of vectorization of the line layers. The region layers are processed using a contouring technique. In the final step of raster image processing, vector data is refined to reduce redundancy.

3.1: Separation of color layers

The color layer separation is based on an unsupervised classification technique. The location of clusters in color space is different for every map. It depends on the type of the map, the production process, the condition of the map, the type of illumination and the spectral characteristics of the scanner unit. The number of clusters to be created, i.e. the number of colors contained in the map has to be defined manually. Typically there are five or six colors contained in a topographic map including background. The separation process may be subdivided into four principal steps:

- Preliminary determination of color class centers using a 3-D histogram,
- Improvement of the cluster center positions using a topological colormap technique based on Kohonen's self organizing feature map,
- Classification of map raster data and
- Region growing of classified data to remove unclassified regions.

For a preliminary determination of centers of color clusters a 3-D histogram is calculated and its maxima are determined using an algorithm described in [8] which we expanded for 3-D arrays. The idea in the algorithm is to find values in the 3-D histogram that are not maxima, find connected components of same value and delete them. The undeleted values constitute the local maxima.

The number of histogram maxima has to equal the number of predefined map colors. Therefore, the existing maxima have to be coalesced iteratively until this condition is satisfied. This is done using the following procedure:

1) Calculation of a distance measure of the maxima to all the others using the Euclidian distance.
2) Marking of couples consisting of m maxima that have to be coalesced in a coalition table. m is determined by

$$m = m_a - m_p \ , \qquad (1)$$

where m_a is the present number of maxima and m_p is the predefined number of colors including the background. If the distance measure of a couple of maxima is less than a predefined threshold, this couple is a privileged candidate for coalition.
3) The table with marked couples is processed recursively to find tuples of maxima that have to be coalesced.

4) A new maximum is created out of the previously grouped maxima using the following equation:

$$c = \frac{1}{H_s} \sum_n c_n H_n \quad \text{with } H_s = \sum_n H_n , \qquad (2)$$

where H_n is the histogram value of the maximum n of the processed tuple, c_n is the R, G or B value of the maximum n and c is the resulting new R, G or B value.

5) If the present number of maxima after the coalition is above the number of predefined colors, the algorithm is continued with step 1.

The next step in classification is improvement of initial estimates of cluster centers. This is done using a iterative topological color map algorithm [9] which is based on Kohonen's self organizing feature map. The color map is initialized with the predetermined cluster center positions and the map size is kept constant. After finishing the iterations, the cluster center positions are determined.

The classification is done using a minimum distance classificator with a fixed rejection threshold [10]. For each cluster the rejection radius is set to a value that guarantees that cluster spheres are not overlapping. Thus, the radius r_i of cluster i is set to

$$r_i = 0.45 \min(d_{ij}) \quad \text{for all } j \neq i , \qquad (3)$$

where d_{ij} is the distance between the centers of clusters i and j. This kind of classification will result in unclassified pixels mainly on the borders between two regions of different colors. If a classification of these border pixels is done using a larger rejection threshold, this will result in a high misclassification rate and in the creation of corroded regions of different colors along the object contours. In this case the following image analysis algorithms would extract a lot of wrong information that prevents the knowledge-directed system from a sensible data interpretation. Therefore, it is better to fill unclassified regions by a region growing technique.

From the classified image m_p-1 binary color layers may be separated. Some of these layers still include textured regions or they have some defects caused by overprinting with other layers. If for example a tree symbol (black) is printed over a forest region (light green), the assignment of the symbol to the black layer will result in an equally shaped defect in the light green layer. These defects may be corrected using region growing techniques with a defined set of rules, as for example,

Set a 0-pixel in the light green layer to 1 if it belongs to a closed 0-region and there is a 1-pixel either in the black or brown layer.

A textured region like a lake area, which is printed using blue raster dots, may be filled using structural texture analysis methods in combination with a texture element grouping algorithm [11]. With this step the separation of color layers is completed.

3.2: Recognition of raster symbols

A rotation and size invariant recognition of separate, not overlapping raster symbols can be obtained using a neural network based technique [12]. A raster symbol is defined as a connected region of pixels with a predefined shape (e.g. tree symbols, characters). The major algorithm extracts rotation and size invariant feature vectors based on polar distance measures. Several types of these measures may be combined for the classification of a single raster symbol or object, for example

- the distance from the center of gravity (CG) of the raster object to its outmost border,
- the distance from the CG to the first change of pixel value,
- the sum of the raster object pixels counted from the CG.

All these measures are determined for a predefined number of directions depending on the object size. The direction for the polar measurements starts from the main axis of inertia of the object, using additional contour or diameter measurements that are necessary to distinguish between an object rotation of $\pm\pi$.

The feature vectors are evaluated using a hierarchical structure of multi-layer perceptrons. There is one perceptron for the direct evaluation of each feature vector (stage-1 network). The number of network inputs corresponds to the vector size, the number of outputs corresponds to the size of the object set. The outputs of the stage-1 networks are combined using the following equation

$$o_n = \frac{1}{o_{max}} \sum_{m=1}^{n_f} i_{mn} w_m , \quad \text{with } w_m = \frac{ls_{min}}{ls_m} , \qquad (4)$$

where n is the index of the output or input unit, m is the index of the stage-1 network and n_f is the overall number of the stage-1 networks. o_{max} is the output with the maximum activity. w_m is a weight factor, ls_m is the number of learning steps necessary to train the stage-1 network m and ls_{min} is the minimum number of learning steps that has occured.

The output of this combination stage is fed into a further perceptron, which makes the final decision about the raster object classification. After recognizing a raster object, it is deleted from the layer and the recognition result is put into the primitive base, where it is available for further interpretation.

3.3: Recognition of raster objects

A raster object is defined as a connected region of pixels that has specific properties but does not have a fixed shape. The raster objects are recognized using category-specific algorithms. Hereby, categories are representing classes of objects with similar properties (e.g. representation of single houses by hatched areas in the maps of scale 1:5 000).

An example for a category-specific algorithm is the recognition of hatched house areas based on a region-oriented technique [13]. The procedure is applied to a thinned binary image and includes a location, correction and analysis of the hatched areas (cf. Fig. 5). Directly neighbored hatched areas may be distinguished by determination of the direction of hatching. Hatched objects are recognized as houses if their aspect ratio is in a predefined range. House objects are deleted from the processed layer and the recognition result is stored in the primitive base.

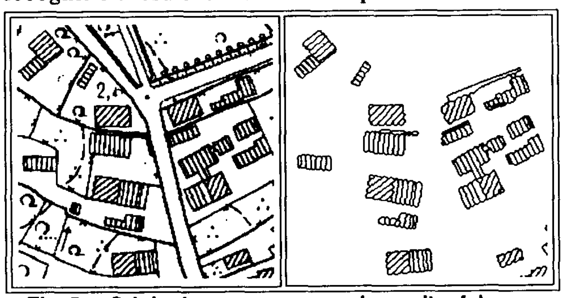

Fig. 5: Original map scene and result of house extraction.

3.4: Separation of region-based and line-based layers

The region data and the line data included in a layer has to be processed in different ways. Region data hast to be contourized while line data has to be vectorized. Thus, for every layer it is necessary to detect whether it contains mainly regions or lines.

This task is performed using a distance histogram based on a medial axis transformation. The histogram values D_i are calculated using the equation

$$D_i = \sum_{y=1}^{n} \sum_{x=1}^{m} d\left(f_{med}(p_j(x,y)) - i\right) \qquad (5)$$

with $d(x) = \begin{cases} 1 & \text{for } x = 0 \\ 0 & \text{otherwise} \end{cases}$,

where m and n are the image dimensions and $p_j(x,y)$ is the value of the pixel represented by the coordinates x and y in the layer j. Function f_{med} yields the minimum distance of the pixel at position (x,y) from the raster object border. The histogram of a line-based layer has a tall shape whereas a region-based layer yields a wide histogram.

3.5: Vectorization

Vectorization is performed on one pixel wide line structured images. Therefore, the region-based layers have to be contourized. This is done using a contour tracing algorithm. The line-based layers have to be thinned before vectorizing them. Most line thinning algorithms are critical to use, because they produce a number of short line fragments connected to the skeleton which do not really exist in the line image. Therefore, we use an algorithm which is not very fast but produces a clean medial line of the raster objects in the input image. This algorithm is based on a smoothing and stripping technique with a skeleton adjustment to the pattern medial line [14].

The vector data is based on nodes and vertices. In the first vectorization step the nodes are extracted from the line image. A node is represented by a pixel that has either less or more than two neighborhood pixels belonging to a line segment. The second step is the conversion of the line segments connecting the nodes into Freeman chain codes. Some line structures like circles cannot be converted to nodes and segments because they consist only of pixels with two neighbors. These line structures are converted in the third vectorization step. The vertices connecting the nodes are created using a split and merge technique on the Freeman coded line segments. Nodes that are directly neighbored in the raster image have to be coalesced and the vertices connected to them have either to be corrected or deleted. Finally, attribute data like color or line width is extracted from the raster image for each vertex.

3.6: Refinement of vector data

Although the skeleton created by the line thinning algorithm of [14] is of high quality, there may be some unnecessary lines and nodes in the thinned image. These lines will also be vectorized. They may be removed in a vector data refinement step. This can be obtained applying a set of rules. In a first step short branch vertices are removed. Subsequently corresponding V-shaped arrangements of vertices are straightened by deleting the center node. After refining the vector data, the nodes and vertices will be stored in the primitive base.

4: Basic idea of knowledge-directed image analysis

The attributed structure primitives extracted by the raster image processing methods described above are the data source for subsequent analysis strategy. The description level of the analysis is based on a hierarchical structuring of the map with map objects and relations between these objects. These relations may be of topological type as well as of thematical type. Associative nets [15], [16], [17] based on frames are used as a formalism for representation of knowledge which is characterized by these objects and their relations.

For the following description of the system it is important to distinguish between map objects representing more or less complex cartographic facts (e.g. terrain area) and simple raster objects. The structure primitives are defined at the lowest hierarchy level. At higher levels a map object represents the composition of one or more map objects of the lower levels of abstraction.

Knowledge-directed image analysis tries to attach a meaning to an image scene. One way of doing so is to use an explicit *model* of what the image can contain and then construct a mapping between the model and the image. Hereby the model represents the necessary *a priori* knowledge.

Fig. 6 shows the principle of the hierarchical map description by example of a coniferous forest. The map object *coniferous forest* consists of a combination of coniferous tree objects, forest border and the forest area signature. Each coniferous tree object is again described by a composition of an inverted V symbol and several dots in a defined topology. The forest border is composed of a sequence of dots. Hereby, the area signature, the dots and the inverted V symbols are the attributed structure primitives.

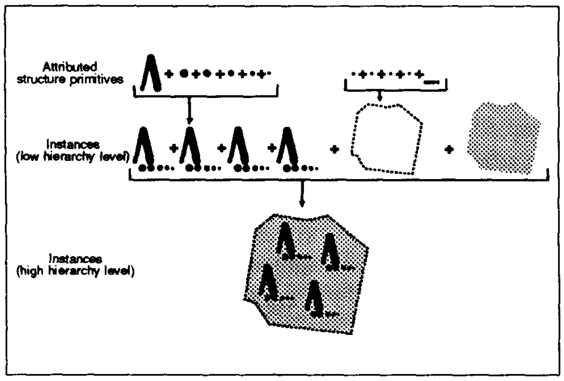

Fig. 6: Example of a hierarchical map description (coniferous forest).

5: Knowledge representation

5.1: Concepts and instances

The *a priori* knowledge necessary for map interpretation is provided by a model acting as long term memory. As mentioned previously, an associative net serves as knowledge representation scheme. The basic structure of the net is the data structure *concept*. A concept contains the intensional description [18] of a term which is necessary for the model of the given problem. The intension of a term is the abstract definition of its meaning. It includes a characterization of properties which must be satisfied by a concrete fact to be valid for this term. On the other hand the extension encloses the set of all concrete facts of a case which satisfy the definition of meaning. The elements of the extensional set of a term are called *instances* of the corresponding concept. For applications of map interpretation, the concepts represent cartographic objects as well as abstract notions necessary for solving conflicts in interpretation.

In the present state of our system the intensional description of a concept is given completely by *necessary parts*, *structure relations* and *attributes*. For generation of an instance of a concept the following conditions have to be considered:

- Instances of concepts have to be made available. These instances are related to the concept to be instantiated by the relation *necessary part*.
- The defined structure relations have to be satisfied.

If both conditions are met, the valuation of the possible instance is performed. This valuation is a measure for the similarity of the instance with the intensional description, i.e., the concept. The valuation represents the certainty factor (cf) for the membership of an instance to the set of realizations of the concept. Thus, the valuation depends on the actual problem and is therefore a part of the *a priori* knowledge given by the model. The procedure to obtain the valuation of an instance has to be defined within the concept. The instantiation is successful if the valuation is above a threshold cf_{th} also defined within the concept. In this case, the attributes of the instance will be evaluated using information in the concept. The instance is then stored in the instance base, which acts as a short term memory. A reference to the instance is also made available within the concept.

Concerning evaluation of the structure relations, attributes and valuations, the model encloses declarative knowledge as well as procedural knowledge, i.e., algorithms. The structuring of the instances is analogous to the one of the concepts. The procedures of the model correspond to concrete values of the instances.

5.2: Frames

Frames [17], [19] are used for representation of both concepts as well as instances. The aspects of an instance or a concept are described with a set of *slots*. These slots may be filled by other frames describing different aspects. An inheritance mechanism is integrated in the frame description of a concept. The inheritance is realized by the slot *generalization*. This relation *generalization* and the relation *necessary part* along with their inversions represent hierarchies of the net. Fig. 6 shows the hierarchy with regard to the relation *necessary part*.

CONCEPT *Coniferous Tree*		
Generalization	value:	{Tree}
Necessary Parts	value:	{T, D1, D2, D3, D4, D5}
Structure Relations	value:	{SR1, SR2, SR3, SR4, SR5}
Attributes	value:	{Position, Size}
CF	value:	[(sr1 & sr2 & sr3 &sr4 & sr5) ifTrue: [cf = 100] ifFalse: [cf = 0]]
	args:	{(SR1), (SR2), (SR3), (SR4), (SR5)}
CF-Threshold	value:	99
T	value:	{Inverted V}
	restriction:	nil
D1	value:	{Dot}
	restriction:	[10 < diameter < 12]
... D2, D3, D4, ...		
D5	value:	{Dot}
	restriction:	[2 < diameter < 4]
SR1	value:	[Procedure S1]
	args:	{(T Position), (D1 Position)}
... SR2, SR3, SR4, ...		
SR5	value:	[Procedure S5]
	args:	{(T Position), (D5 Position)}
Position	value:	[Procedure ConiferousTreePosition]
	args:	{(T Position), (D1 Position), (D2 Position), (D3 Position), (D4 Position), (D5 Position)}
Size	value:	[Procedure ConiferousTreeSize]
	args:	{(T Size), (D1 Size), (D2 Size), (D3 Size), (D4 Size), (D5 Size)}

Fig. 7: Simplified definition of concept *coniferous tree* (args = facette *arguments*).

A simplified definition of the concept for a coniferous tree is presented in Fig. 7. The corresponding ideal shape of the coniferous tree is shown in Fig. 8.

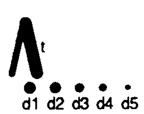

Fig. 8: Ideal shape of a coniferous tree.

For a successful instantiation of the concept *coniferous tree* shown in Fig. 7, an instance (t, d1, d2, d3, d4, d5) of the corresponding concept (*inverted V* and *dot*) has to be available for each part of the tree in a defined topology with necessary attributes. For the intensional description, the necessary parts (T, D1, D2, D3, D4, D5) are defined in the slot *necessary parts*. The single elements of the list are references to further substructures represented by slots. Each substructure owns a facette *value* and a facette *restriction*. The facette *value* contains a reference to the concept and therefore also to the instances of interest. Considering the entry of facette *restriction*, a subset of instances may be determined that is relevant for the instantiation. Thus, T, D1, D2, D3, D4 and D5 characterize lists of relevant instances. The entry of slot *structure relations* defines the structure relations that have to be satisfied for a combination of instances (t, d1, d2, d3, d4, d5) to execute a successful instantiation. The combination of instances is determined from the lists T, D1,..., D5. With regard to the example, SR1 defines the necessary topology of t and d1. For testing of structure relation SR1, facette *value* (of slot SR1) contains the corresponding procedure S1. The arguments are determined by the argument list defined by the facette *arguments*. Each element of the list represents a relational description. The relational description (T Position) for instance means that the position of t has to be transfered to the procedure S1. If a combination of instances (t_k, d_m, d_n, d_o, d_p, d_q) exists, which satisfies the structure relations, a valuation using the procedure of facette *value* located in slot *cf* is executed. The necessary arguments are determined analogously to the testing of the structure relations. For that, the argument list located in the facette *arguments* of slot *cf* is used. The instantiation is successful, if the result of valuation exceeds the threshold given by the facette *value* of slot *cf-threshold*. Subsequently, the attribute values are evaluated according to the testing of structure relations. The relevant attributes are defined by the slot *attributes* and specified by the slots *position* and *size*. In case of successful instantiation, an instance *Ix* (i.e. now x instances of the concept *coniferous tree* are existing) will be created and stored in the instance base using the data structure shown in Fig. 9. The list of instances located in facette *value* of slot *instances* will be extended by the instance *Ix*.

INSTANCE I_x	
Instance Of	value: Coniferous Tree
Necessary Parts	value: {(T t_k), (D1 d_m), (D2 d_n), (D3 d_o), (D4 d_p), (D5 d_q)}
Structure Relations	value: {(SR1 true), (SR2 true), (SR3 true), (SR4 true), (SR5 true)}
CF	value: 100

Fig. 9: Example for an instance I_x of the concept *coniferous tree*.

The presented mechanism for knowledge representation is a simplified description. The following system extensions give an idea of the additional features integrated in the present system. They are necessary for professional utilization of map interpretation.

5.3: Extensions of concept definition

5.3.1: Inclusion of topological alternatives: The concept definition introduced so far allows only an interpretation of ideal map scenes. With regard to the topology shown in Fig. 8, an instantiation is not possible if dot d3 is not present. This is contradictory to the flexible and fault tolerant human capability of reading and interpreting a map. To increase flexibility of analysis, the concept definition has been expanded. Using a disjunction of combinations of instances in the slot *necessary parts*, topological alternatives can be included. The corresponding definitions located in slots *structure relations* and *cf* have also to be expanded.

5.3.2: Definition of recursive structures: For analysis of recursive structures like contour lines, a special concept definition exists, which is based on exactly two necessary parts. The first part describes the non-recursive basic element. With its help, the recursive structure is defined. In the case of a dashed line, this concept describes an individual line segment. The second part contains a reference to the concept itself and therefore represents the recursive structure.

5.3.3: Introduction of constraints: So far an assumption was made during instantiation process that instances of concepts acting as necessary parts of the superior concept are already existing. To obtain flexibility in image analysis it is not always necessary and desirable to make this assumption. If, for example, an instance of the concept *connection line* is necessary to process the instantiation of a concept, it is normally not possible to generate all instances of *connection line* in advance for the current map scene. Trying to do this would result in an overflow of the instance base and an unacceptable long processing time. But in general, it is not necessary to generate all instances, because only one of them is of interest and this one depends normally on the other necessary parts concerned. To solve this problem *constraints* are introduced supplementary to the *restrictions* of necessary parts. These constraints describe properties which the necessary part has to possess for successful instantiation. These properties therefore depend on the other necessary parts concerned in contrast to the properties forced by the restrictions.

6: Control module for knowledge-directed image analysis

The control module supervises and controls instantiation of concepts. Two operation modes exist. In the *interactive mode* a concept of interest is given by the user as a goal. Thereupon, the control module determines the minimal set of necessary concepts at the lowest hierarchy level. Based on this set the instantiation of superior concepts is performed successively until the goal concept is reached. In the *automatic mode* the instantiation is obtained in a bottom-up manner using all instances at the lowest hierarchy level. Thus, all concepts of superior layers will be instantiated. In both modes the instantiation of a single concept is performed in accordance to the methods described in the previous sections.

Normally instantiation of a concept results in several instances. Therefore, the control module is able to handle different alternatives during analysis. This feature is important because normally a definite interpretation of a map scene requires consideration of the context of surroundings. Existence of different alternatives leads to instances that are in competition with each other. For management of the interpretation hypotheses a graph controlled by a belief-revision-algorithm is used. This algorithm is based on aspects of truth-maintenance-systems [20], [21], [22], [23], [24]. The instance that is relevant for further instantiations is selected by an evaluation algorithm. The selection depends on the type of concept. For all types the certainty factor *cf* is used. For recursive concepts the number of non-recursive basic elements is considered. In case of recursive and simultaneous goal driven concepts the constraints satisfied by the actual instance are compared to those of the underlying preceded hypothesis.

7: Knowledge-directed interpretation support for high complexity map scenes

Problems in map interpretation may occur if the raster image is too complex for the context-independent raster processing methods presented so far. In such cases of conflicts, the instantiation of concepts of map objects may not be possible because of lack of appropriate structure primitives. A possible reason for complexity may be the overlapping of different map symbols. For the solution of this problem a hypothesis is generated, that states which map symbol is expected in the specific image region.

Hereby, the actual situation of instantiation is the decisive criterion. Based on this hypothesis a more specific raster analysis method is used to detect the expected symbol in the corresponding color layer and image region of interest. Depending on this analysis the subsequent instantiation uses the recognition result obtained in the previous step.

8: Implementation and results

The PROMAP-system contains currently the basic ideas and methods presented here. So far the module *conflict solving*, described in section 7, is not yet implemented, however this will be done in near future.

The host system is based on 486-PCs with 16 MByte RAM which are connected to a local area network. The network server supplies the workstations with a harddisk storage capacity of 1 GByte and additional space on a 1GByte rewritable optical disk. We use an OPTOTECH overhead repro scanner which is connected to the host via SCSI interface. For the implementation of raster data processing and knowledge-directed analysis the development environment has to satisfy different requirements. Since processing of raster data mainly consists of pixel-based image processing actions, the corresponding methods are implemented using High-C (from Metaware). The knowledge-directed system is realized by the object-oriented programming language and development environment Smalltalk-80. Both, the properties of object-oriented programming as well as the incremental working Smalltalk-80-compiler allow a rapid prototyping. In consideration of the Model-View-Controller-paradigm [24] a modular system was developed that also supports interactive commands given by the user for additional control of analysis.

The current prototype system includes a model representing concepts for interpretation of *coniferous forest, deciduous forest, bush, meadow, hedge, mound with hedge, urban area, road segment, crossing, contour line (1m interval), contour line (5m interval), swamp (moor) and draining ditch.* The corresponding interpretation results for the example shown in Fig. 1 resp. Fig. 4 are shown in Figs. 10 and 11. Hereby the marks for the recognized map objects are underlayed by the grey level image of the original map scene. For the analysis of further details, additional concepts may be defined.

Although the PROMAP-system allows a flexible map analysis, some critical problems have to be solved interactively by the system user. An example for this type of problems is shown in Fig. 11, where the contour line in the lower left corner had to be closed manually (see mark *interactive concatenation* in Fig. 11). An automatic concatenation of the ends of the contour lines was not possible because of missing contour line signs in the paper-based map.

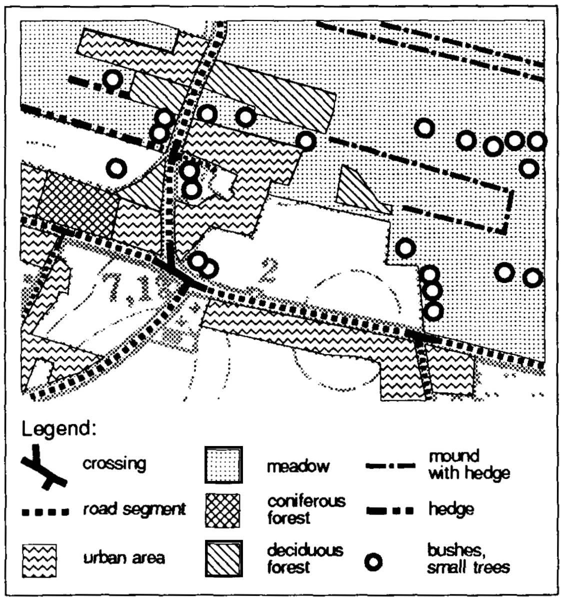

Fig. 10: Interpretation result for recognition of *crossing, road segment, urban area, bush, meadow, coniferous forest, deciduous forest, mound with hedge* and *hedge.*

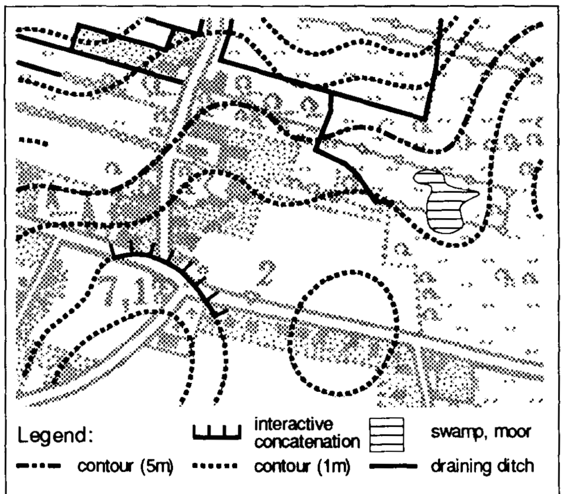

Fig. 11: Interpretation result for recognition of *contour line* (5m and 1m interval), *draining ditch* and *swamp (moor).*

9: Summary and conclusions

An overview of the PROMAP image analysis system for interpretation of topographic maps was presented. The realized methods for raster data processing, knowledge organization and knowledge use were discussed. Finally the capabilities of the system were demonstrated on a map scene.

Further work will be directed towards the conflict solving module and the improvement of the already existing methods. Eventually, the system will be tested for more map scenes. Furthermore, it must be investigated how more complex raster processing steps for extraction of structure primitives [25] may improve the overall system performance.

References

[1] P. Besslich, I. Delius, N. Ebi, B. Lauterbach and G. Stäblein, "Umweltplanungssystem: Integration wissensbasierter Bildinterpretation von Karten zur Bewertung von Standorteignung und Bodenschutz", in *Wissensbasierte Systeme in den Geowissenschaften*, Beiträge zur Mathematischen Geologie und Geoinformatik, G.J. Peschel, ed., Sven von Loga Verlag, to be published in 1992.

[2] R. Kasturi, R. Fernandez, M.L. Amlani and W. Feng, "Map Data Processing in Geographic Information Systems", in *Computer*, IEEE Computer Society, No. 12, December 1989, pp. 10-21.

[3] M. Ilg and O. Kübler, "Wissensbasiertes Verstehen von Straßenkarten", in *Mustererkennung 1989*, Proc. 11. DAGM-Symposium, Informatik-Fachberichte 219, H. Burkhardt, K.H. Höhne and B. Neumann, eds., Springer-Verlag, Berlin, 1989, pp. 354-358.

[4] J. Alemany and R. Kasturi, "A Computer-Vision System for Interpretation of Paper-based Maps", in *Proc. SPIE Conf. Applications of Digital Image Processing X (1987)*, Vol. 829, 1987, pp. 125-137.

[5] J.A. Mulder, A.K. Mackworth and W.S. Havens, "Knowledge Structuring and Constraint Satisfaction: The Map See Approach", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, November 1988, pp. 866-879.

[6] S. Suzuki and T. Yamada, "MARIS: Map Recognition Input System", in *Mapping and Spatial Modelling for Navigation*, L.F. Pau, ed., Springer-Verlag, Berlin, 1990, pp. 95-116.

[7] N. Ebi, B. Lauterbach and P. Besslich, "Automatic Data Acquisition from Topographic Maps Using a Knowledge-Based Image Analysis System", in *International Archives of Photogrammetry & Remote Sensing*, Volume XXIX, Part B4, Commission IV, International Society for Photogrammetry and Remote Sensing (ISPRS), 1992, pp. 655-663.

[8] B.B. Chaudhuri and B.U. Shankar, "An Efficient Algorithm for Extrema Detection in Digital Images", in *Pattern Recognition Letters*, Vol. 10, No. 8, August 1988, pp. 81-85.

[9] A. Springub, D. Scheppelmann, U. Engelmann and H.P. Meinzner, "Generierung von optimalen Falschfarben aus Echtfarbbildern mit Hilfe der Topologischen Karte", in *Mustererkennung 1990*, Proc. 12. DAGM-Symposium, Informatik-Fachberichte 254, R.E. Großkopf, ed., Springer-Verlag, Berlin, 1990, pp. 267-275.

[10] J.A. Richards, *Remote Sensing - Digital Image Analysis*, Springer-Verlag, Berlin, 1986, pp. 190-205.

[11] F. Tomita and S. Tsuji, *Computer Analysis of Visual Textures*, Kluwer Academic Publishers, Norwell, Massachusetts, 1990, pp. 71-92.

[12] B. Lauterbach and P. Besslich, "A Neural Network Based Recognition of Complex Two-Dimensional Objects", in *Proc. of the Internat. Conf. on Industrial Electronics, Control and Instrumentation*, Kobe (Japan), IEEE Publishing Services, New York, Vol. 3, 1991, pp. 2504-2509.

[13] W. Shen, N. Ebi and P. Besslich, "Schraffurerkennung in graphischen Vorlagen", in *Mustererkennung 1991*, Proc. 13. DAGM-Symposium, Informatik-Fachberichte 290, B. Radig, ed., Springer-Verlag, Berlin, 1991, pp. 312-319.

[14] Y.K. Chu and C.Y. Suen, "An Alternate Smoothing and Stripping Algorithm for Thinning Digital Binary Patterns", in *Signal Processing*, Elsevier Science Publishers, North-Holland, No. 11, November 1986, pp. 207-222.

[15] M.R. Quillian, "Semantic Memory", in *Semantic Information Processing*, M. Minsky, ed., MIT Press, Cambridge and London, 1968, pp. 227-270.

[16] J.R. Brachman, "On the Epistemological Status of Semantic Networks", in *Associative Networks - Representation and Use of Knowledge by Computers*, N.V. Findler, ed., Academic Press, New York, 1979, pp. 3-50.

[17] M. Minsky, "A Framework for Representing Knowledge", in *Frame Conceptions and Text Understanding*, D. Metzing, ed., de Gruyter, Berlin, 1979, pp. 1-25.

[18] G. Sagerer, *Darstellung und Nutzung von Expertenwissen für ein Bildanalysesystem*, Springer-Verlag, Berlin, 1985.

[19] E. Rich, *Artificial Intelligence*, McGraw-Hill, Berkeley, 1983, pp. 229-233.

[20] J. Doyle, "A Truth Maintenance System", in *Artificial Intelligence*, Elsevier Science Publishers, North-Holland, No. 12, December 1979, pp. 231-272.

[21] J. deKleer, "An Assumption-Based TMS", in *Artificial Intelligence*, Elsevier Science Publishers, North-Holland, No. 28, 1986, pp. 127-162.

[22] F. Puppe, "Belief Revision in Diagnosis", in *GWAI-87 - Proc. of. 11th German Workshop on Artificial Intelligence*, K. Morik, ed., Springer-Verlag, Berlin, 1987, pp. 175-184.

[23] O. Dressler and H. Freitag, "Truth Maintenance Systeme", in *KI - Künstliche Intelligenz*, Oldenbourg-Verlag, München, No. 2, 1989, pp. 13-19.

[24] G.E. Krasner and S.T. Pope, *A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80*, ParcPlace Systems, Palo Alto, 1988.

[25] N. Ebi and P. Besslich, "Extraktion paralleler Linienstrukturen am Beispiel topographischer Karten", in *Mustererkennung 1991*, Proc. 13. DAGM-Symposium, Informatik-Fachberichte 290, B. Radig, ed., Springer-Verlag, Berlin, 1991, pp. 312-319.