

An image analysis system for automatic data acquisition from colored scanned maps

N. Ebi¹, B. Lauterbach², and W. Anheier²

¹ Kohns + Poppenhaeger Ltd., Pfalzbahnstrasse 20, D-66538 Neunkirchen, Germany

² Working Group Digital Systems, University of Bremen, FB1, P.O. Box 330 440, D-28334 Bremen, Germany

Abstract. Many maps [an important source of information for efficient spatial data evaluation using a Geographic Information System (GIS)] must still be digitized manually, a time-consuming and error-prone process. Therefore, we developed the processing of maps (PROMAP) system, which incorporates adequate image analysis. The system can generate a symbolic description of the map that can be imported into a GIS. A color scanner generates a multicolor raster image of the map. This image is split into layers of predefined map colors. Each layer is vectorized and methods like neural network-based symbol and object recognition for the extraction of attributed structure primitives and knowledge-directed image interpretation are applied. The map scene is structured hierarchically. The interface to the GIS is represented by the map objects at the upper levels of hierarchy. These investigations described are part of the interdisciplinary project *Environmental Planning System*. The scope of this project is the combination of data acquisition, the development of an evaluation scheme, and GIS in an integrated concept.

Key words: Map interpretation – Topographic maps – Color scanning – Raster image analysis – Knowledge-directed interpretation

1 Introduction

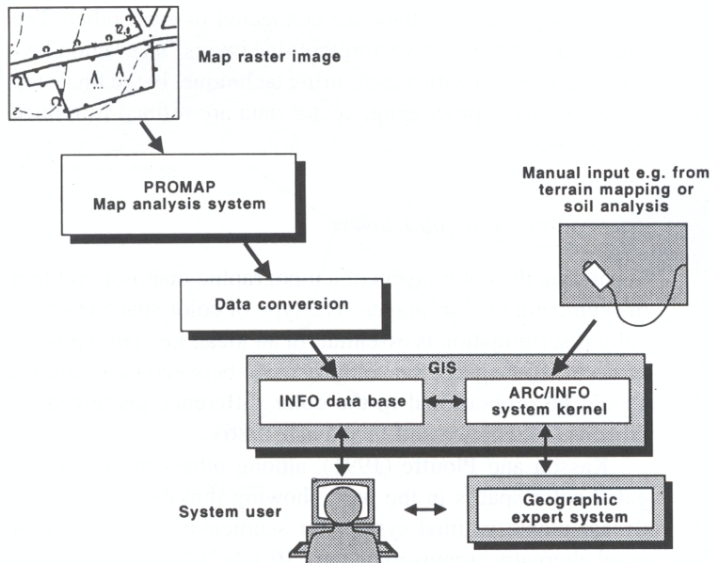
In the past 10 years, Geographic Information Systems (GISs) have become more important for storing and organizing spatial information in computers. The number of applications has kept pace with the advancements in this technology. From high-quality cartography to land planning, natural resource management, environmental assessment and planning, ecological research, etc., the GIS promises to be one of the most extensive computer applications ever to emerge.

For the efficient and flexible use of these systems, it is necessary to combine data acquisition, the development of an evaluation scheme, and GIS in an integrated concept. This system

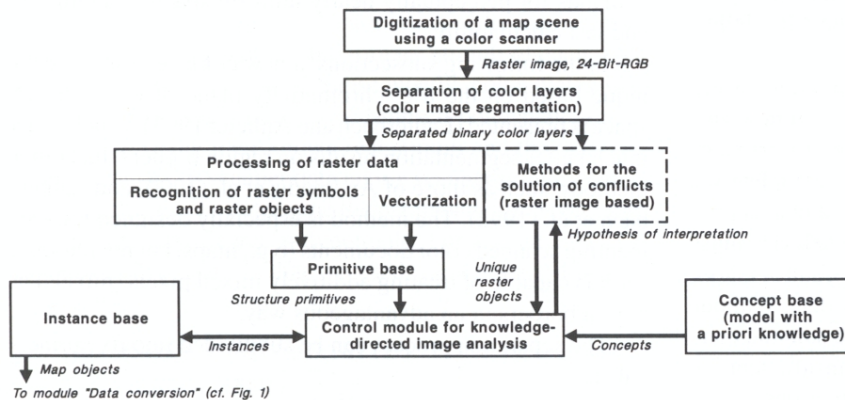
integration is the objective of the interdisciplinary project *Environmental Planning System*. A project group of researchers affiliated with the working groups *Digital Systems* and *Physio-Geography* of the University of Bremen was formed to do the necessary research.

The evaluation scheme to be developed for solving geographic problems of the project (Besslich et al. 1992) refers to determining the output capacity of soils, the protection grade of certain soils, and evaluation of site suitability. This requires the development of an operable, effective catalog of geo-indicators and hierarchical evaluation matrices.

For efficient spatial evaluation, an adequate data base is necessary. Various types of maps are important sources of information. For our application, most of the relevant maps are available only in paper form. The necessary manual digitization requires considerable time and cost because of the large number of manual operations. To increase efficiency, it is useful and challenging to automate the process of extracting information from maps by scanning and applying image analysis methods (Kasturi et al. 1989). Several systems have been developed for solving the task of automatic data extraction for special types of maps. These systems require, as input data, separated map layers (Ilg and Kübler 1989; Yang 1989) or maps of one color (e.g., cadastral maps) (Alemany and Kasturi 1987; Illert 1990; Mulder 1988; Suzuki and Yamada 1990). The majority of current methods aim to automatize the data acquisition from maps on a large scale (1:500 to 1:10 000) with a low density and little variety of information. Solutions for interpretation of maps on a medium scale (1:10 000 to 1:50 000), which have a high density and much variety of information are limited to the acquisition of single layers of information like road nets. The lack of efficient integral solutions for the image analysis of topographic maps on a medium scale and their relevance to our geographic application motivated us to develop the proposed method. We used German topographic maps on a scale of 1:25 000 that are printed with a four-colors printing technique (Schoppmeyer 1991). The printing colors are cyan, magenta, yellow, and black. Therefore, we developed appropriate methods for raster data processing, data organization and knowledge use (Ebi et al. 1992; Lauterbach et al. 1992).



1



2

Fig. 1. System overview

Fig. 2. Principle scheme of the PROMAP system

The cooperation mentioned guarantees the consideration of relevant geographic aspects of map analysis and a continuous specification of a priori knowledge necessary for knowledge-directed image analysis. The main ideas of raster-data processing are scanning with a 24-bit red, green, blue (RGB) scanner, separation of color layers, raster-symbol and raster-object recognition, and vectorization. The principles of knowledge-directed interpretation are those of prototypes (concepts) as the building block of the basic representation, generalization, and aggregation as an abstraction mechanism. The corresponding methods are implemented in the PROMAP system.

This paper describes the details of the PROMAP system. It is divided into sections that focus on the major technical aspects associated with the construction of the map-analysis module. Section 2 provides a system overview. Section 3 describes the raster processing portion of the PROMAP system. The principles and methods of the knowledge-directed image analysis are discussed in the Sects. 4 to 7. Section 8 describes

the environment of the system development and the results obtained by the analysis of a typical map scene.

2 System overview

An overview of the complete environmental planning system is shown in Fig. 1. 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 chose ESRI's ARC/INFO because it is a powerful tool and is in widespread use in Germany.

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

The principle of the proposed knowledge-directed image analysis is shown in Fig. 2. The topographic maps we use are printed in a four-colors printing technique. For this reason, a raster image of the map to be processed is created with a 24-bit RGB color scanner. The smallest objects of the topographic maps are raster dots (e.g., meadow texture) from 0.05 to 0.1 mm in size. Therefore, the minimum scanning resolution is 800 to 1000 dpi. We use an OPTOTECH overhead repro scanner, which is connected to the host computer via an SCSI interface. The scanner allows a resolution of 2000 dpi or more in all scanning modes.

Symbolic image information is extracted by splitting the map image into color layers that are processed by raster-object recognition and vectorization. The extracted symbolic information, represented by so-called *attributed structure primitives*, serves as the data source for the knowledge-directed image analysis. The analysis is implemented 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 must finally be converted (Fig. 1, module *data conversion*) to the ARC/INFO data base format.

An advantage of applying color layer separation to the 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 from which the paper-based map is produced. If binary map layers are used, the complete layers must be digitized. Otherwise, there would not be any control marks available to support spatial registration of raster data. Without control marks, it is usually impossible to define suitable control marks for a spatial registration because of insufficient interrelations between the map layers. The advantage described, together with the analysis methodology incorporated in the PROMAP system, guarantees high flexibility for the analysis of topographic maps.

3 Processing of raster data

The raster data of the map are 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 in the map (Fig. 3).

The separation corresponds to the reverse process of composing binary map layers during printing. In contrast to the map production process, the separation results in layers containing information printed with the same color instead of layers containing information of the same type (e.g., symbols, roads). 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 the separation of layers containing mainly region and line information. These two types of layers must be processed in a different way. A different strategy must be used to separate regions representing map symbols (e.g., houses) from lines

(e.g. road borders) if they are connected to each other. The fourth step consists of vectorizing line layers. The region layers are processed with a contouring technique. In the final step of raster image processing, vector data are refined to reduce redundancy.

3.1 Separation of color layers

Separating the color layers of a topographic map is a problem of segmenting color images. The type of color space used for color discrimination is essential. In an ideal, i.e., perceptually uniform color space, the color distance between two samples should be proportional to the color difference perceived by humans (Mukherjee and Venkatesh 1986).

Kasson and Plouffe (1991), among others, have investigated color spaces in the past, showing that there is no easily calculable optimal system for segmentation purposes. A good alternative seems to be the CIE $L^*u^*v^*$ color space (International Illumination Commission 1986), which meets the demands for perceptually nearly uniform spacing, additivity, and fast calculability.

In the following subsections, a new color separation technique based on the $u'v'$ chromaticity plane of CIE $L^*u^*v^*$ space is presented (Lauterbach and Anheier 1993), which gives much better segmentation results on region borders than other methods, such as those of Andreadis et al. (1990) and Lauterbach et al. (1992). The method is especially designed for segmenting scanned color documents (e.g., maps, but not photos), for it is capable of treating additively mixed pixels (mixels) at region borders in an advantageous way.

The separation process can be subdivided into five principal steps:

- The transformation of RGB data into the $u'v'$ -chromaticity plane of $L^*u^*v^*$ color space and the generation of a $u'v'$ histogram.
 - The detection of peaks (cluster centers) and ridges (mixels) in the histogram.
 - Segmentation with pixel chromaticity as a criterion in considering the additive color mixture laws.
 - Segmentation of the nonchromatic cluster with lightness as a criterion.
 - Region growing of unclassified regions.
- The main advantages of the proposed method are:
- The number and/or position of cluster centers are determined automatically without any presetting.
 - Mixels generated by scanning of region borders are evaluated in a proper way.
 - Small clusters are not suppressed.
 - Scattered regions are not unnecessarily subdivided.

3.1.1 Transformation of RGB data into the $u'v'$ chromaticity plane

Image data produced by the scanner is represented as 3×8 -bit tristimuli in an unstandardized RGB color space. In contrast, the $u'v'$ chromaticity plane is referenced to physical quantities of light. Therefore, the color must be corrected. The first step

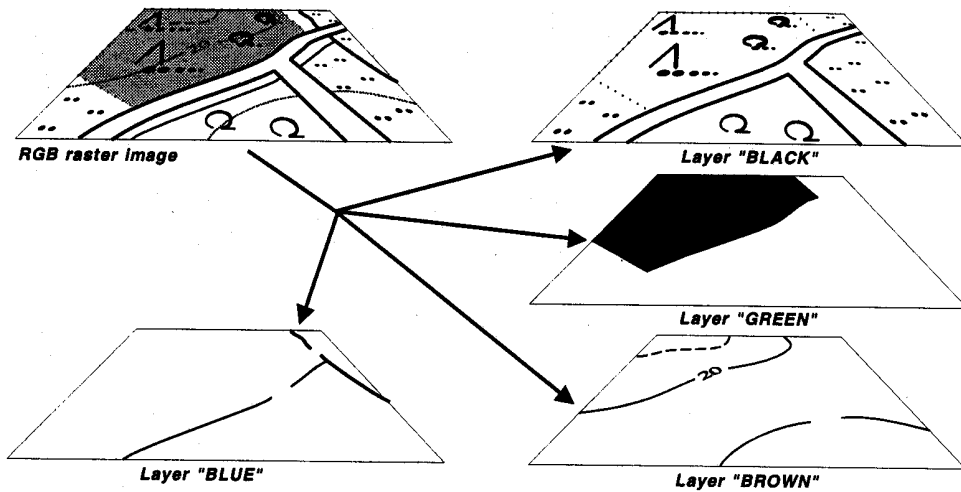


Fig. 3. Separation of color layers

is to determine the colorimetric properties of the scanner. This can be done by calculating the primary stimuli of each scanner channel (Richter 1981). Subsequently, color plates with known colorimetric values (e.g., Munsell color plates) must be scanned. Using their known real stimuli and the ones calculated from RGB scanner data taking scanner primary colors into account, we can determine the parameters for a color correction with first degree polynomials. For the OPTOTECH Optoscan 2000 scanner that we used, the resulting transformation and correction rules are

$$\begin{aligned} u' &= \frac{(0.6964R + 0.462G + 0.1812B)}{(1.3531R + 4.2294G + 1.1052B)} \quad \text{and} \\ v' &= \frac{(0.7308R + 2.4336G - 0.072B)}{((1.3531R + 4.2294G + 1.1052B))} \end{aligned} \quad (1)$$

where R , G , and B are the stimuli produced by the scanner. Based on the chromaticity values u' and v' of all pixels of a scanned map, a two-dimensional histogram is calculated. Since the resulting coordinates u' and v' are real numbers in the range $[0,1]$, they must be discretized (e.g., steps of 0.001) before being used as indices for the histogram.

3.1.2 Detection of peaks and ridges in a $u'v'$ histogram

For detection of peaks in a $u'v'$ histogram, a one-dimensional peak-detection technique proposed by Sezan (1990) was modified for the two-dimensional case. The two-dimensional cumulative distribution function (CDF) $s(u, v)$ is calculated as follows:

$$s(u, v) = \sum_{i=0}^u \sum_{j=0}^v h(i, j) \quad (2)$$

The two-dimensional maxima detection signal (MDS) $r(u, v)$ is the difference of the CDF value $s(u, v)$ and the mean of its square surrounding window of a predefined size. If the signal value $r(u, v)$ is equal to or greater than 0 and $r(u-1, v)$,

$r(u, v-1)$, and $r(u-1, v-1)$ are less than 0, the histogram entry $h(u, v)$ is a maximum.

The detection of ridges corresponds to the detection of maxima in only one direction of the histogram. Therefore, the calculation of CDF and MDS must be modified. First, two one-dimensional CDFs are calculated:

$$s_u(u, j) = \sum_{i=0}^u h(i, j) \quad \text{and} \quad s_v(i, v) = \sum_{j=0}^v h(i, j) \quad (3)$$

Subsequently, the two-dimensional CDF is reckoned as the product of both one-dimensional CDFs:

$$s(u, v) = s_u(u, v) s_v(u, v) \quad (4)$$

MDS computation and evaluation is analogous to the previously described peak detection method. In addition to the detected maxima, more cluster centers are derived from the ridges by fitting a regression line, which also runs through a maximum to sets of ridge points. From the positions of all ridge points belonging to a set, a weighted sum is calculated to determine the coordinates of the additional cluster center. Figure 4 shows the maxima and ridges evaluated from the $u'v'$ histogram of the map image shown in Fig. 5.

3.1.3 Segmentation based on additive color mixture laws

After determining the cluster centers, we segment the scanned map image. The proposed method evaluates primarily the distance of a stimulus from the additive color mixture lines between two cluster centers instead of the distance from the center itself. In this way, segmentation errors of mixels can be reduced significantly, resulting in smoother region contours.

In a first step, all pixels located inside a rejection radius around the cluster centers are classified. The radius is set to 10% of the distance to the nearest neighboring cluster center. Subsequently, for every pixel the distances to all additive color mixture lines are determined. A mixel is assumed to result from an additive mixture of the pair of cluster center colors of which

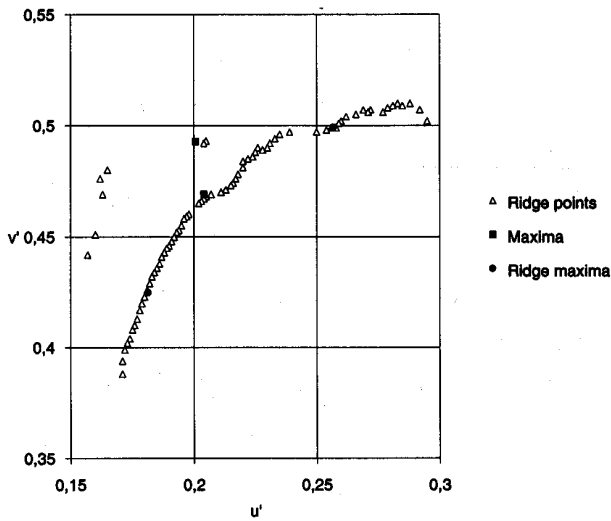


Fig. 4. Maxima and ridges evaluated from the $u'v'$ histogram of the map image shown in Fig. 5

the mixture line is at the minimum distance from the mixel. In the final discrimination step, a mixel is assigned to that cluster of the cluster pair that is nearer to the mixel. Figure 5 shows an example of a mixel classification. With regular clustering techniques, the stimulus to be classified would be assigned to cluster *paper/black* although it probably results from an additive mixture of map colors *forest* and *water* on a region border. Regular clustering techniques may therefore produce a black line along the region border.

After the segmentation step described, one nonchromatic and several chromatic clusters have been extracted from the image histogram. For the maps we use, we must also discriminate for the nonchromatic cluster. Today's segmentation techniques do not allow us to differentiate between light and dark pixels. Therefore, the nonchromatic cluster contains black as well as white pixels. The threshold for distinguishing between the two colors is determined with the histogram peak-detection algorithm as described in (Sezan 1990).

3.1.5 Region growing of classified data

A number of binary color layers corresponding to the number of clusters is separated from the classified image. Some of these layers still include textured regions or have some defects caused by overprinting with other layers. If, for example, a tree symbol (black) is printed over a forest region (green), the assignment of the symbol to the black layer will result in a similarly shaped defect in the green layer. These defects are corrected by 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 the brown layer.

A textured region like a lake area, which is printed with blue raster dots, can be filled with structural texture analysis methods in combination with an algorithm that groups texture

elements (Tomita and Tsuji 1990). With this step the separation of the 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 with a neural network-based technique (Lauterbach and Besslich 1991). We developed the proposed method for the recognition of two-dimensional objects extracted from industrial image scenes (e.g., workpieces), and it proved suitable for map interpretation.

A raster symbol is defined as a connected region of pixels with a predefined shape (e.g., tree symbols, characters). The main algorithm extracts the rotation and fixed size feature vectors based on polar distance measurements. Several types of these measurements can be combined to classify 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 change of first pixel value, and
- the sum of the raster object pixels counted from the CG.

All these measurements are determined for a predefined number of directions, depending on the size of the object. The direction for the polar measurements emerges from the main axis of inertia of the object. Additional contour or diameter measurements are necessary to distinguish between an object rotation from $+\pi$ to $-\pi$.

The feature vectors are evaluated with a hierarchical structure of multilayer 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 by the following equation

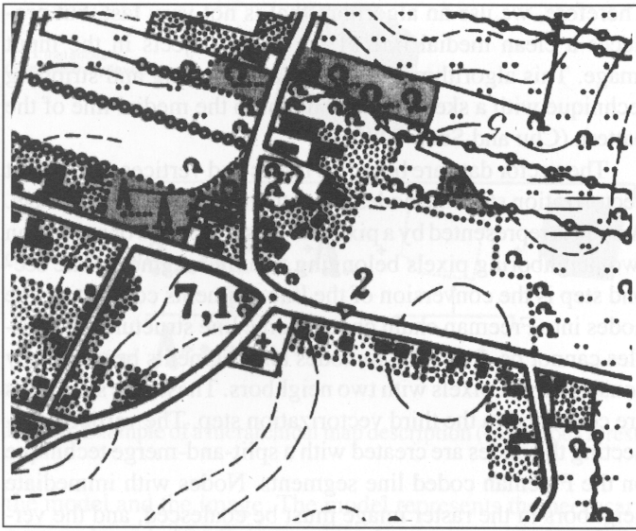
$$o_n = \frac{1}{o_{\max}} \sum_{m=1}^{n_f} i_{mn} w_m, \quad \text{with } w_m = \frac{1 s_{\min}}{l s_m}, \quad (5)$$

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, $l s_m$ is the number of learning steps necessary to train the stage-1 network m , and $1 s_{\min}$ is the minimum number of learning steps that have occurred.

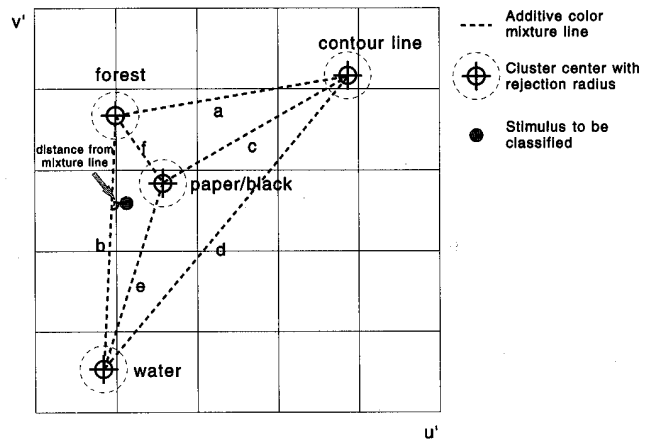
The output of this combination stage is fed into another perceptron, which makes the final decision about the raster object classification. After recognizing a raster object, it is deleted from the layer, and the result of the recognition is put into the primitive base, where it is available for further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

3.3 Recognition of raster objects

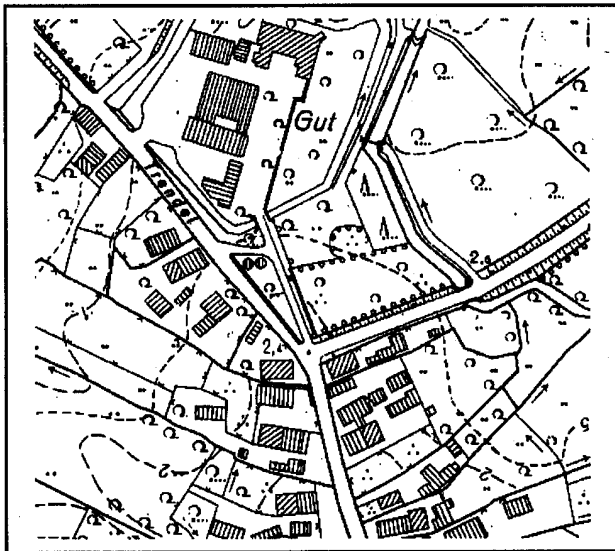
A raster object is defined as a connected region of pixels that has specific properties, but does not have a fixed shape.



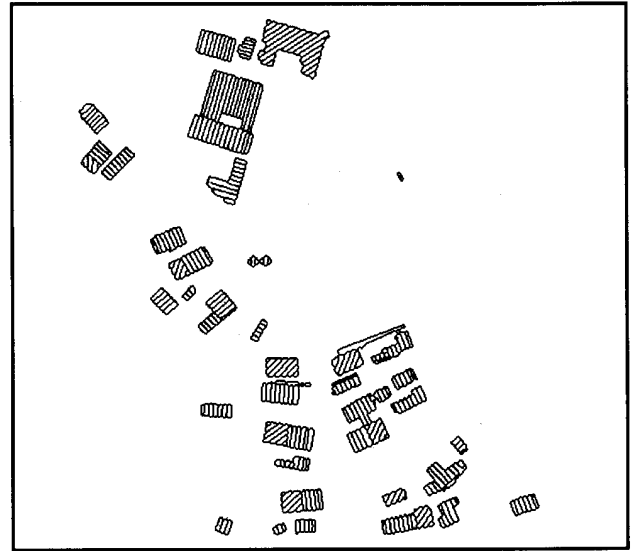
5



6



7a



b

Fig. 5. Example of a topographic map scene (reduced to gray scale; original size is about 27 × 22 mm² or 1070 × 880 pixels; the scanning resolution is about 1010 dpi)

Fig. 6. Segmentation based on additive color mixture laws

Fig. 7. a Original map scene (scale 1:5000); b result of house extraction

The raster objects are recognized with category-specific algorithms. The categories represent classes of objects with similar properties. Examples of object categories are:

- the representation of single houses by hatched areas in the maps of scale 1:5000 (Fig. 7a),
- the representation of single houses by blocks in the maps of scale 1: 25 000 (Fig. 8),
- the representation of urban areas by dense hatching or dots in the maps of scale 1:25 000.

An example for a category-specific algorithm is the recognition of hatched house areas based on a region-oriented technique (Shen et al. 1991). The procedure is applied to a thinned binary image and includes the location, correction, and anal-

ysis of the hatched areas. Immediately neighboring hatched areas can be distinguished by determining the direction of the hatching. The result of the technique applied to the image of Fig. 7a is shown in Fig. 7b. Hatched objects are recognized as houses if their aspect ratio is within a predefined range. These house objects are deleted from the processed layer and the recognition result is stored in the primitive base.

3.4 Separation of region-based and line-based layers

The region data and the line data included in a layer must be processed in different ways. Region data must be contoured, while line data must be vectorized. Thus, it is necessary

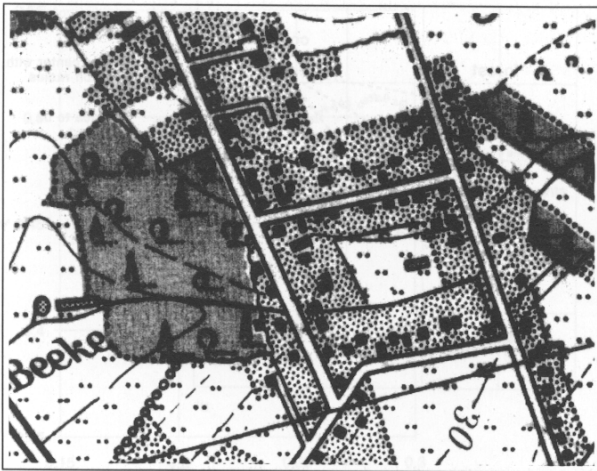


Fig. 8. Example of a topographic map scene (reduced to gray scale; the original size is about 29×23 mm or 1160×900 pixels; the scanning resolution is 1010 dpi)

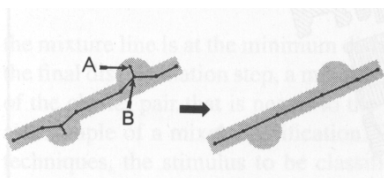


Fig. 9. Illustration of vector refinement

to detect whether each layer contains mainly region or line structures.

This task is performed with a distance histogram based on a medial axis transformation (Pavlidis 1987). The histogram values D_i are calculated using the equation

$$D_i = \sum_{y=1}^n \sum_{x=1}^m d(f_{\text{med}}(p_j(x, y)) - i) \quad (6)$$

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

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 must be contourized. We use a contour-tracing algorithm described in Pavlidis (1987). The line-based layers must be thinned before they are vectorized. Most line-thinning algorithms are critical to use, because they produce a number of short line fragments connected to the skeleton that do not really exist in the line image.

Therefore, we use an algorithm that 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 medial line of the pattern (Chu and Suen 1986).

The vector data are 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 neighboring 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 with a split-and-merge technique on the Freeman coded line segments. Nodes with immediate neighbors in the raster image must be coalesced, and the vertices connected to them must be either corrected or deleted. Finally, attribute data like color, line width, or variance of the line width are extracted from the raster image for each vertex.

3.6 Refinement of vector data

Although the skeleton created by Chu and Suen's (1986) line thinning algorithm is of high quality, there may be some unnecessary lines and nodes in the thinned image. These lines are also vectorized. They can be removed in a step that refines data. We do this with a set of rules. In a first step, short branch vertices (vertex A, Fig. 9) are removed. Then, corresponding V-shaped arrangements of vertices (node B, Fig. 9) are straightened by deleting the center node. After refining the vector data, the nodes and vertices are stored in the primitive base.

4. The basic idea of knowledge-directed image analysis

The attributed structure primitives extracted by the raster image processing methods that we have described are the data source of the 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 the topological type as well as of the thematic type. Associative nets (Quillian 1968; Minsky 1979) based on frames are used as a formalism to represent knowledge that 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 hierarchical 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 map between

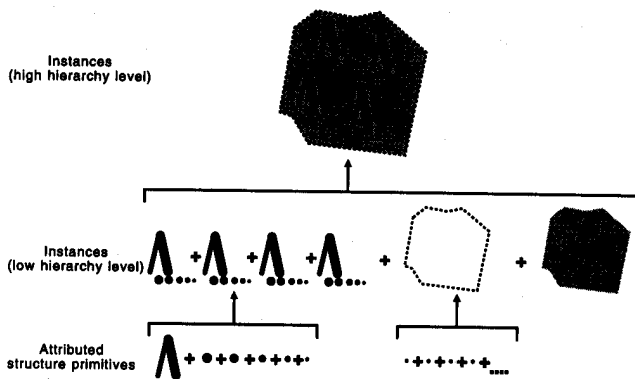


Fig. 10. Example of a hierarchical map description (coniferous forest)

the model and the image. The model represents the necessary a priori knowledge. Since a particular image scene is only an instance of the class of possible image scenes, the model must be in some sense larger than the image. That is, a useful model of the domain of the image typically contains a large amount of information on possible image contents. The mapping processed during analysis normally uses only a small part of the model. This surplus of knowledge guarantees a flexible image interpretation.

Figure 10 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. The area signature, the dots, and the inverted V symbols are the attributed structure primitives.

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 a knowledge representation scheme. The basic structure of the net is the data structure *concept*. A concept contains the intentional description (Sagerer 1985) of a term that is necessary for the model of the given problem. The intention of a term is the abstract definition of its meaning. It includes a characterization of properties that must be satisfied by a concrete fact to be valid for this term. In contrast, the extension encloses the set of all concrete facts of a case that 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. As an example, the concept *virtual continuation of a contour line* can be considered. This clause characterizes connections of contour lines that cannot be derived from the

existing attributed structure primitives in the case of overlapping line segments.

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

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

If both conditions are met, the possible instance is evaluated. This valuation is a measure of the similarity of the instance with the intentional description, i.e., the concept. The valuation represents the certainty factor (cf) for the membership of an instance to the set of instances of the concept and can be a number between 0 and 100. Thus, the valuation depends on the topical problem and is therefore a part of the a priori knowledge given by the model. The valuation of an instance is important mainly for guiding the instantiation process of recursive structures. The procedure of obtaining the valuation of an instance must 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 with 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 the evaluation of the structure relations, attributes, and valuations, the model includes declarative knowledge as well as procedural knowledge, i.e., algorithms. We can quantitatively characterize the qualitative facts of a case represented by the associative net, using the procedural knowledge. For this purpose, the structuring of the instances is analogous to one of the concepts. The procedures of the model correspond to concrete values of the instances.

5.2 Frames

Frames (Harmon and King 1985; Minsky 1979) are used to represent both concepts and instances. The aspects of an instance or a concept are described with a set of *slots*. These slots are separated into facets that can be filled by other frames describing other aspects. An inheritance mechanism is integrated in the frame description of a concept. That is, more concrete concepts summarize their own slots and those of the concepts at higher levels. This inheritance reduces redundancy in defining concepts without influencing the recognition process. In the case of more concrete concepts, only some specific declarations are needed. The inheritance is implemented by the slot *generalization*. This relation *generalization* and the relation *necessary part*, along with their inversions, represent hierarchies of the net. Figure 10 shows the hierarchy with regard to the relation *necessary part*.

A simplified definition of the concept for a coniferous tree is presented on the next page; the procedural components are written in Smalltalk syntax.

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]]
	arguments:	{(SR1),(SR2),(SR3),(SR4),(SR5)}
CF-Threshold	value:	99
T	value:	{Inverted V}
	restriction:	nil
D1	value:	{Dot}
	restriction:	[10 < diameter < 12]
D2	value:	{Dot}
	restriction:	[8 < diameter < 10]
D3	value:	{Dot}
	restriction:	[6 < diameter < 8]
D4	value:	{Dot}
	restriction:	[4 < diameter < 6]
D5	value:	{Dot}
	restriction:	[2 < diameter < 4]
SR1	value:	[Procedure S1]
	arguments:	{(T Position), (D1 Position)}
SR2	value:	[Procedure S2]
	arguments:	{(T Position), (D2 Position)}
SR3	value:	[Procedure S3]
	arguments:	{(T Position), (D3 Position)}
SR4	value:	[Procedure S4]
	arguments:	{(T Position), (D4 Position)}
SR5	value:	[Procedure S5]
	arguments:	{(T Position), (D5 Position)}
Position	value:	[Procedure ConiferousTreePosition]
	arguments:	{(T Position), (D1 Position), (D2 Position), (D3 Position), (D4 Position), (D5 Position)}
Size	value:	[Procedure ConiferousTreeSize]
	arguments:	{(T Size), (D1 Size), (D2 Size), (D3 Size), (D4 Size), (D5 Size)}



Fig. 11. Ideal shape of a coniferous tree

The corresponding ideal shape of the coniferous tree is shown in Fig. 11.

For a successful instantiation of the concept *coniferous tree* (just shown), an instance ($t, d_1, d_2, d_3, d_4, d_5$) of the corresponding concept (*inverted V* and *dot*) must be available for each part of the tree in a defined topology with the necessary attributes. For the intentional 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 other substructures represented by slots. Each substructure owns a *facet value* and a *facet restriction*. The *facet value* contains a reference to the concept and therefore also to the instances of interest. Considering the entry of *facet restriction*, a subset of instances may be determined that is relevant to the instanti-

ation. Thus, T, D1, D2, D3, D4, and D5 characterize lists of relevant instances. The entry of slot *structure relations* defines the structure relations that must be satisfied for a combination of instances ($t, d_1, d_2, d_3, d_4, d_5$) 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 d_1 . For testing the structure relation SR1, *facet value* (of slot SR1) contains the corresponding procedure S1. The arguments are determined by the argument list defined by the *facet arguments*. Each element of the list represents a relational description. The relational description (T position), for example, means that the position of t must be transferred to the procedure S1. If a combination of instances ($t_k, d_m, d_n, d_o, d_p, d_q$) exists that satisfies the structure relations, a valuation using the procedure of *facet 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 *facet arguments* of slot *cf* is used. The instantiation is successful if the result of valuation exceeds the threshold given by the *facet value* of slot *cf-threshold*. Subsequently, the attribute values are evaluated by testing the structure relations. The relevant attributes are

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

defined by the slot *attributes* and specified by the slots *position* and *size*. In case of successful instantiation, an instance I_x (i.e., now x instances of the concept *coniferous tree* exist) is created and stored in the instance base with the data structure shown on the next page. The list of instances located in facet *value* of slot *instances* will be extended by the instance I_x .

The mechanism presented 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 use of map interpretation.

5.3 Extensions of the concept definition

5.3.1 Inclusion of alternative causes of instantiation

The concept definition introduced allows only an interpretation of ideal map scenes so far. With regard to the topology shown in Fig. 11, an instantiation is not possible if dot d2 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. If we use a disjunction of combinations of instances in the slot *necessary parts*, topological alternatives can be included. The corresponding definitions located in the slots *structure relations* and *cf* must also be expanded.

A section of concept definition *coniferous tree* that includes topological alternatives is shown on the next page.

This shows the revised concept definition for the case of a successful instantiation even if dot d2 is absent. In the case of a coniferous tree, a total of 31 choices is defined for consideration of all topological possibilities. If instantiation of several combinations of instances is possible, the one with the largest number of necessary parts will be preferred. The introduction of alternative causes of instantiation is not limited to the representation of the various topological alternatives. In addition, the instantiation of concepts is supported when their necessary parts have no immediate topological relation but only a thematic one, e.g., the concept *tree*, which characterizes a term representing both coniferous and deciduous trees. The inclusion of a corresponding alternative cause of instantiation allows the unification of the instance sets of the concepts *coniferous tree* and *deciduous tree*.

5.3.2 Definition of recursive structures

For the interpretation of real map scenes, the modeling and analysis of recursive structures like contour lines must be included. We have found no satisfactory solution in the literature. Sagerer (1985), for example, who describes the principles of the ERNEST system (Niemann 1990), excludes the possibility of defining recursive structures in the design of the semantic

net language. Our solution uses a special concept definition that supports exactly two necessary parts. The first part describes the nonrecursive basic element used for defining the recursive structure. In the case of a dashed line, this concept describes an individual line segment. The second part refers to the recursive concept itself, and therefore it represents the recursive structure. The principle of the instantiation of a recursive concept c is based on the successive combination of the basic elements given by a concept c_m that is defined as the first necessary part of c . The corresponding principle is shown here:

Recursive structure	Basic elements
r_i	$\{b_p\}$
r_j	$\{b_p, r_i\} = \{b_q, \{b_p\}\}$
\vdots	\vdots
r_n	$\{b_u, r_m\} = \{b_u, \{\dots, \{b_r, \{b_q, \{b_p\}\}\}\}$
b_p, b_q, b_r, b_u :	instances of concept c_m
r_i, r_j, r_m, r_n :	instances of concept c

A first instance r_i of concept c , a so-called primary instance, is created by an instance b_p of the concept c_m . Based on r_i , an instance b_q will be determined, so that the structure relations defined by c are satisfied by the combination consisting of b_p and b_q . The whole process will be repeated until no further instance b_x can be determined, and the demanded relations are satisfied. The instance r_n finally generated represents the maximal recursive structure that was found based on r_i or b_p .

For the instantiation of a recursive concept, not only is the described determination of one solution interesting, but the determination of the set with all possible solutions as well. Not only does one primary instance r_i exist, but a whole set of start instances. The mechanism for the recursive aggregation process is based on a heuristic search in a graph, in which the nodes represent possible combinations of the basic elements. The heuristic rules are based on the map-drawing rules and the knowledge that recursive structures describe linear or two-dimensional map entities. The rules are used for solving contradictions between instance combinations, that is, if one basis element is part of two different recursive structures. The graph analysis is mainly guided by the rules and the individual certainty factors of each instance. Depending on the structures examined, the certainty factors are propagated so that the new evaluation is dependent on the underlying recursive parts.

5.3.3 Introduction of necessary and inadmissible properties

An assumption was made during the instantiation process that instances of concepts acting as necessary parts of the superior concept already exist. To obtain flexibility in image analysis, it is not always necessary or desirable to make this assumption.

Necessary Parts	value:	{(T, D1, D2, D3, D4, D5) OR (T, D1, D3, D4, D5)}
Structure Relations	value:	{(SR1, SR2, SR3, SR4, SR5) OR (SR1, SR3, SR4, SR5)}
CF	value:	[(sr1 & sr3 & sr4 & sr5) ifFalse: [cf=0] ifTrue: [sr2 isNil ifTrue: [cf=100] ifFalse: [sr2 ifTrue: [cf=100] ifFalse: [cf=0]]]]
	arguments:	{(SR1), (SR2), (SR3), (SR4), (SR5)}

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 the 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 cause an unacceptably long processing time. In general, it is not necessary to generate all instances, because only one of them is interesting, and this one normally depends on the other necessary parts. To solve this problem, *necessary* and *inadmissible properties* are introduced supplementarily to the restrictions of necessary parts. For successful instantiation, the corresponding necessary part must have all the *necessary properties* but cannot have any *inadmissible properties*. This means that the properties depend on the other necessary parts, in contrast to the properties forced by the restrictions. If *necessary* and *inadmissible properties* are used, it must be guaranteed that the necessary parts determining the desired properties are independent of the corresponding parts. Otherwise, the consistency of the definition is violated. To distinguish between concepts that may be instantiated directly and those that are instantiated due to *necessary* and *inadmissible properties*, an additional slot has been introduced. This slot defines the type of the concept. The possible entries in the facet *value* of slot *type* are *normal* and *goal driven*. An example for the definition of *necessary* and *inadmissible properties* is shown here; the procedural components are written in Smalltalk syntax:

Necessary Parts	value:	{(NP1, NP2, NP3)}
NP1	value:	Line Segment
	restriction:	nil
NP2	value:	Line Segment
	restriction:	nil
NP3	value:	Connection Line
	restriction:	nil
	necessary properties:	{N1, N2}
	inadmissible properties:	{J1}
N1	variable:	Start
	operator:	=
	method:	[: x x]
	arguments:	{(NP1 End)}
N2	variable:	End
	operator:	=
	method:	[: x x]
	arguments:	{(NP2 Start)}
J1	variable:	Length
	operator:	>
	method:	[: x : y 5 * (x distance : y)]
	arguments:	{(NP1 End), (NP2 Start)}

In this example the instantiation of the concept *connection line* is based on the information given by the necessary parts *NP1* and *NP2*. Therefore, we consider only solutions for instantiation that satisfy the necessary properties *N1* and *N2* and that do not have the admissible property *J1*. The property *N1*, for example, forces the start coordinate *start* given by facet *variable* of slot *N1* to be identical with the end coordinate *end* of *NP1* given by facet *reference value*. The identity of both values is enforced by the entry “=” of the facet *operator* (also possible: <, >, ≥, ≤). Analogously, *N2* specifies the end coordinate of the desired connection line. With the inadmissible property *J1* solutions with a length that exceeds a predefined threshold are prevented. The threshold is defined as five times the distance between the end coordinate of the instance specified via *NP1* and the start coordinate of the instance specified via *NP2*.

5.3.4 Modelling and analysis of conflict and identities

The mechanisms we have described for the instantiation of a concept *c* are based on a monotonic deduction rule:

- If Necessary-Parts-Are-Available: then Instantiation** with the premise
- *Necessary-Parts-Are-Available*: if a combination of the instances defined by the necessary parts exist and the corresponding structure relations are accurate for this combination and the consequence or conclusion
 - *Instantiation*: an instance of *c* exists.

So far, deductions made with this rule have been considered time-invariant. Thereby the number of deductions increases with the number of facts. This feature is not guaranteed during the image analysis of a map. In the sense of nonmonotonic reasoning, new information can result not only in new deductions, but also, because of contrary evidence, in the revision of deductions already established. Therefore, a nonmonotonic relation between the number of facts and the number of deductions is given. For solving this problem a nonmonotonic deduction is introduced:

If Necessary-Parts-Are-Available unless Exception then Instantiation

The truth value of the exception is based on a three-valued logic. That means it can be true, false, or unknown. For the application considered, ignorance about the exception is interpreted as being in favor of deduction.

For the frame-based interpretation of structure primitives, especially two kinds of exceptions are of interest:

```

Conflict-Descriptor-1  condition: [ :oldInstance :newInstance |
                                | sOld sNew eOld eNew |
                                sOld := (oldInstance valueAt: #Startposition);
                                sNew := (newInstance valueAt: #Startposition);
                                eOld := (oldInstance valueAt: #Endposition);
                                eNew := (newInstance valueAt: #Endposition);
                                [((sOld = sNew) & (eOld = eNew)) v ((sOld = eNew) & (eOld = sNew))]
                                solution: [ :oldInstance :newInstance |
                                           (oldInstance valueAt: #Length) > (newInstance valueAt: #Length)]

```

- A currently generated instance of a concept *c* represents the same facts as an instance of *c* already determined.
- A currently generated instance of a concept *c* contradicts an instance of *c* already determined.

These exceptions are characterized either by a conflict or by an identity. In the case of an exception, the deduction of an instance will be revised. Conflict and identity descriptors are introduced for the consideration of exceptions. After every generation of a new instance of concept *c* we check whether a conflict or an identity between the new instance and one already generated exists. In the case of an identity, the deduction of the new instance is usually revised. In the case of a conflict, the corresponding conflict descriptor determines which instance must be revised. For checking whether an exception exists, only unrevised instances are relevant. A new instance can, at most, only be identical to or in conflict with one old instance, because the information characterized by the set of valid instances already established is always consistent. This consistency is guaranteed by continuous checking after every instantiation. In this way, we avoid creating a set that contains more than one instance that satisfies the conditions of an exception. By introducing exceptions into the definition of recursive concepts, the set of structures that must be propagated during instantiation can be additionally limited.

There are several possible causes of conflicting cases, to that various solutions are required. For modeling of this fact, a set was introduced, whose elements refer to individual conflict descriptors. The order of the set represents the sequence for checking whether an exception exists. The structure of an exemplary conflict descriptor is shown later in this section. The slot describing a conflict descriptor contains two facets. The entry of each facet is a procedure with arguments that are known a priori. These are the frame *oldInstance* of the old instance and the frame *newInstance* of the currently generated instance. The procedures use the values of attributes and relations of both instances. The first facet *condition* contains the procedure for checking whether conflict exists or not. Hereby the truth value *true* results if a conflict exists. Then the procedure in the facet *solution* determines which instance must be revised. The truth value *true* indicates the revision of the previously generated instance *oldInstance* and the manifestation of the new instance *newInstance*. An example of a "conflict" descriptor is shown on the top of the page; the procedural components are written in Smalltalk syntax.

This descriptor describes the possible conflict between two line segments. The conflict exists if the start and the end coordinates of both instances correspond. In case of a conflict,

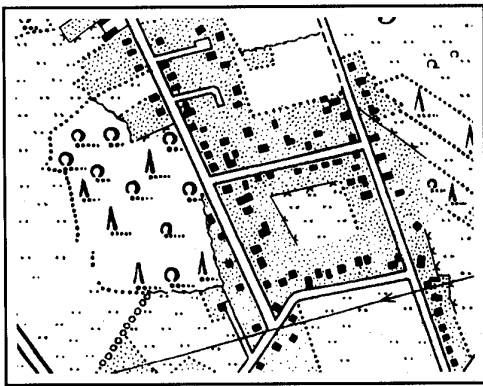
the instance with the greater length will be revised. With this descriptor, the validation of line segments from a point B to a point A is avoided if a line segment from A to B has already been established.

The structure and function of an identity descriptor corresponds to that of a conflict descriptor. The only difference is the absence of the facet *solution*. Instead of a conflict-dependent determination of the instance that must be revised, the default revision is performed.

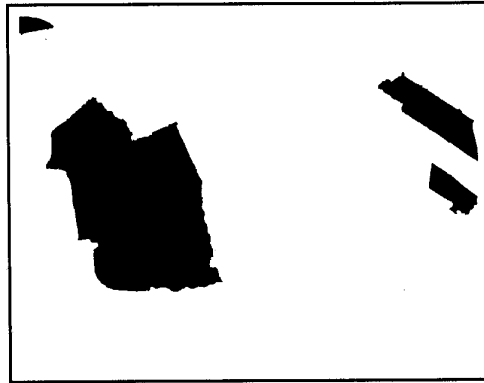
6 Control module for knowledge-directed image analysis

The control module supervises and controls the instantiation of concepts. There are two operation modes. In the *interactive mode* a concept of interest is given as a goal by the user. Then the control module determines the minimal set of necessary concepts at the lowest hierarchical level. Based on this set, superior concepts are instantiated successively until the goal is reached. In the *automatic mode*, instantiation moves up the hierarchy, using all instances at the lowest level. Thus, all concepts of upper layers will be instantiated. In both modes, each concept is instantiated according to the methods described in the previous sections.

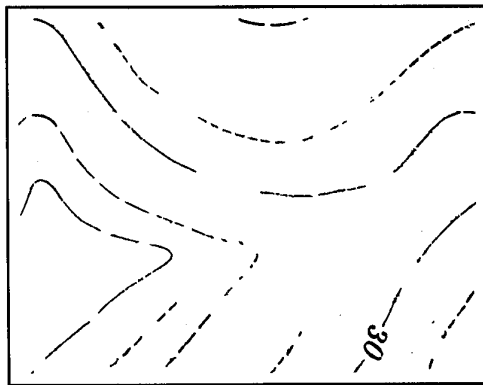
Normally, instantiating a concept results in several instances. Therefore, the control module is able to handle various possibilities during analysis. This feature is important because a definite interpretation of a map scene usually requires us to consider the surroundings. The various possibilities lead to instances that compete with each other. To solve contradictions between the instances of different concepts, the current algorithm requires a solution given implicitly by the mode. When the model is further developed to recognize more complex map entities, this method may not be appropriate. Therefore, our current investigations are directed toward explicitly modeling these conflicts and coping with the consequences. The corresponding algorithm is based on an extension of the conflict descriptors (Sect. 5.3.4) already introduced for dealing with contradiction between instances of different concepts and a reason-maintenance system (de Kleer 1986; Petri 1989). This system supports a belief revision (Puppe 1987) using a dependency network (Reinfrank 1987) by explicitly storing justifications (Doyle 1979).



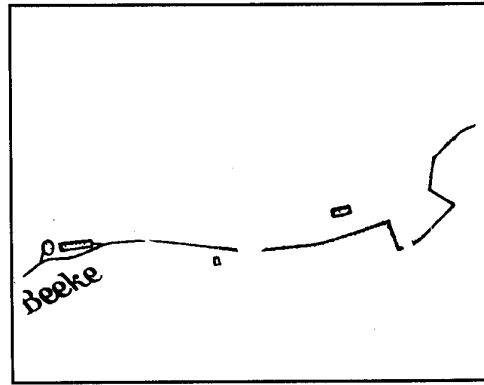
12 a



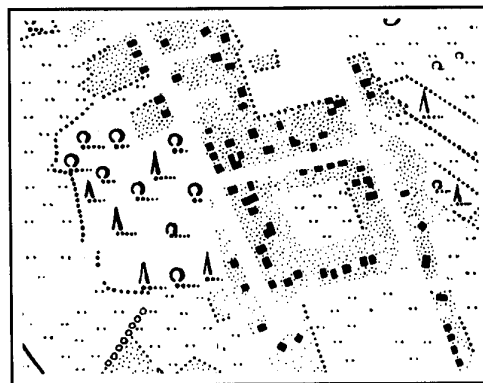
b



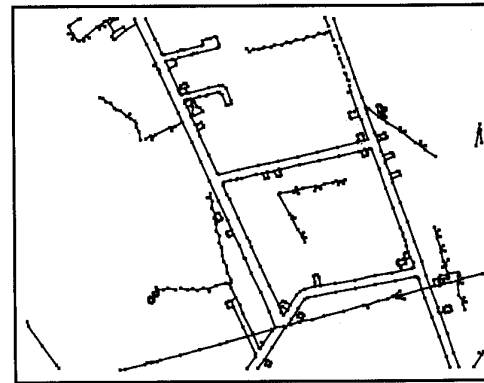
c



d



13 a



b

Fig. 12. a Color layer BLACK; b color layer GREEN; c color layer BROWN; d Color layer BLUE
 Fig. 13. a Recognized symbols of color layer BLACK; b vectors of color layer BLACK

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 conflict, instantiating concepts of the map objects may be impossible because of lack of appropriate structure primitives. A possible reason

for complexity is the overlapping of various map symbols. To solve this problem, we hypothesize which map symbol is expected in the specific image region (refer to concept *virtual continuation of a contour line* in Sect. 5.1). The actual situation of the instantiation is the decisive criterion. Based on the hypothesis, a more specific raster analysis is used to detect the expected symbol in the corresponding color layer and image

region of interest. The instantiation then uses the recognition result from the raster analysis.

8 Implementation and results

The PROMAP system currently contains the basic ideas and methods presented here. At present, the module *conflict solving*, described in Sect. 7, is not yet implemented. However, this will be done in the near future.

The host system is based on a number of 486 PCs with 16-MB RAM that are connected to a local area network. The network server supplies the workstations with a hard disk storage capacity of 1 GB and additional space on a 1 GB rewritable optical disk. To implement raster data processing and knowledge-directed analysis, the development environment must satisfy various requirements. Since the processing of raster data consists mainly of pixel-based image-processing, the corresponding methods are implemented with High C (Metaware). This compiler generates 386 code with the possibility of accessing an address space of 4 GB supported by a virtual memory manager. This feature is necessary for handling the large number of data. The knowledge-directed system is implemented with the object-oriented programming language and development environment Smalltalk-80. Both the properties of object-oriented programming (e.g., data abstraction, encapsulation, inheritance, polymorphism) and the incremental working Smalltalk-80 compiler allow rapid prototyping. Considering Model-View-Controller paradigm (Krasner and Pope 1988) a modular system was developed that also supports interactive commands given by the user for additional control of the analysis. Although Smalltalk-80 is an ideal development environment for the knowledge-directed analysis system, its principle based on a virtual machine causes a nonoptimal system performance. Concerning time-consuming operations (e.g., testing of structure relations), the conversion of the corresponding Smalltalk methods to C functions is in progress. The C functions can be integrated with the existing analysis system as so-called *user primitives*.

Figure 8 shows a test scene scanned from a topographic map of scale 1:25 000. Figures 12a–d show the binary color layers BLACK, GREEN, BROWN, and BLUE that have been separated from the original RGB image of Fig. 8 with the methods described in Sect. 3.1. The layers are shown after removing the defects caused by overprinting. As an example for the subsequent raster processing, Fig. 13a and b show the results of raster symbol recognition and vectorization on the binary color layer BLACK (Fig. 12a). Problems occur in some cases when map symbols are connected to each other. Currently, we are developing methods for separating house symbols from road lines.

The current prototype system includes a model representing concepts for the interpretation of *meadow*, *coniferous forest*, *deciduous forest*, *mixed forest*, *bush*, *single tree*, *building*, *road segment*, *crossing*, *contour line (1.25 m, 2.5 m, 5 m, 10 m intervals, and without a specific interval)*, and *water*. The corresponding interpretation results for this example are shown in

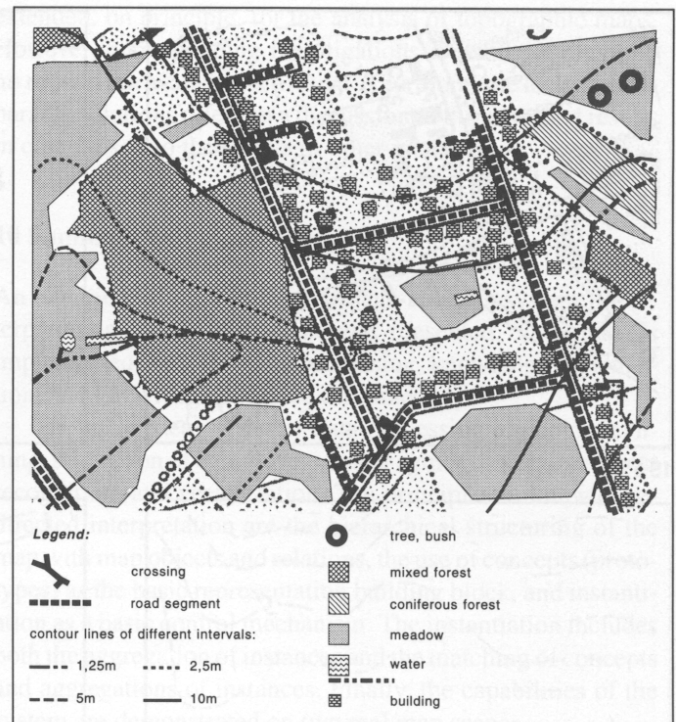
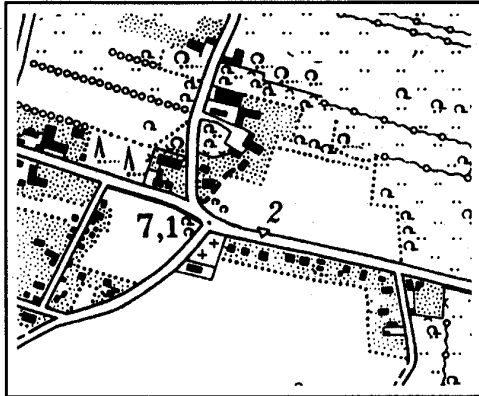


Fig. 14. Interpretation results for the map shown in Fig. 8

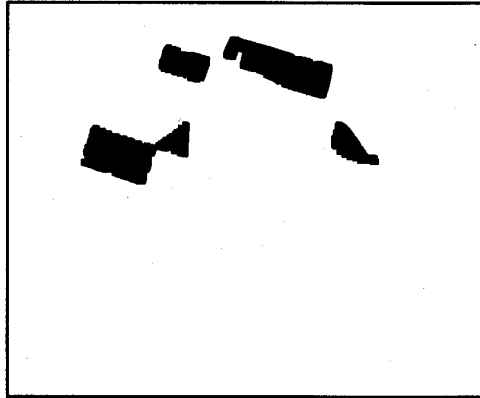
Fig. 14. The gray-level image of the original map scene underlies the marks for the recognized map objects. For the analysis of further details, additional concepts may be defined.

The results of recognizing meadows (Fig. 14) shows two differences compared to a human map reader's results. First, the connections of individual meadows are missing, caused by missing triangular subareas used for meadow recognition based on triangulation. Second, the extracted meadow boundaries do not correspond exactly to the real ones, as expected from a cartographer, but describe the enclosing polygon given by the outer meadow symbols. This difference is caused by the missing explicit boundary of meadow signs used in German topographic maps. The real boundary is only given by an implicit minimal covering polygon that is defined by various neighboring map signs like roads and hedges. The detection and definition of meadow boundaries is not obvious. The ambiguities make determining real boundaries very difficult, even for experienced human map readers. For this reason, we intended to determine minimal and correct meadows, including their boundaries, in the system being developed. Currently, an adaptation to the geographically relevant borders must be done manually with the GIS. To solve this problem, we are developing a graph-based algorithm [compare Yang (1989)], which searches for the minimal enclosing polygon out of all possible border elements (e.g., roads, hedges) for every meadow.

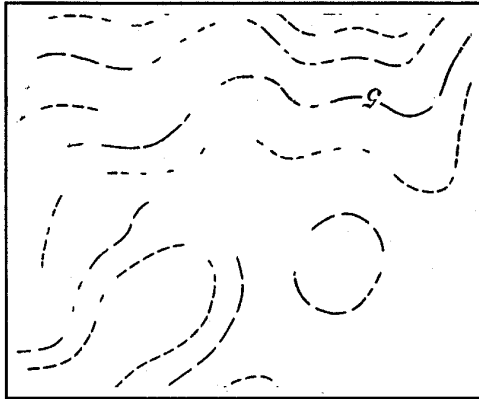
The results of the analysis of another example is shown in Figs. 5a and 15–17. For this example, the determination of contour lines of specific intervals was not possible. This was caused by an insufficient data basis generated by the raster image preprocessing of the brown map layer. Reasons for the poor quality might be calibration errors of the scanner, weak con-



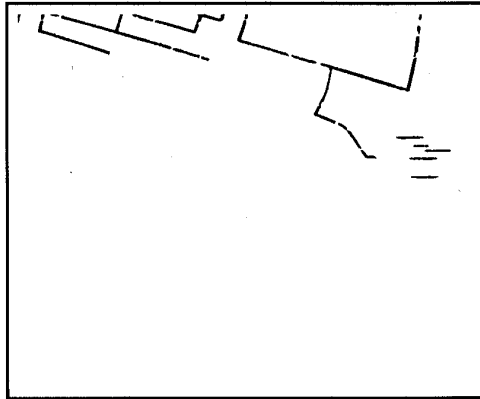
15a



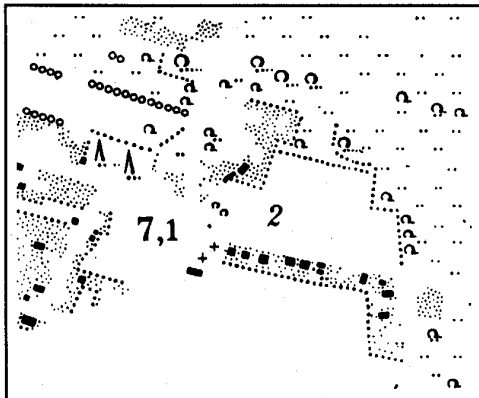
b



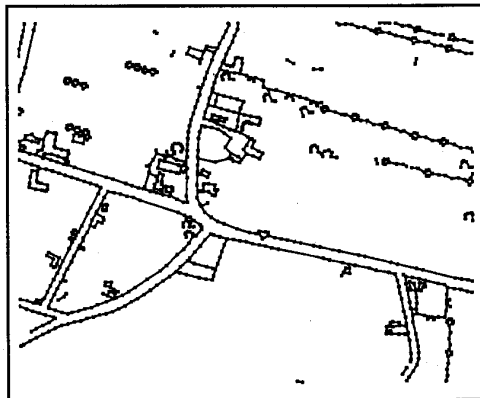
c



d



16a



b

Fig. 15. a Color layer BLACK; b color layer GREEN; c color layer BROWN; d color layer BLUE
 Fig. 16. a Recognized symbols of color layer BLACK; b vectors of color layer BLACK

trast of the map, and problems resulting from overlapping map symbols. Two problems occur in recognizing contour lines. First, the contour lines could not be merged because of missing segments. Furthermore, breaks in four places could not be closed in consequence of their extension and the misalignment of the corresponding contour line endings. The breaks can be detected automatically with an abstract concept *Unexpected-End-Of-A-Contour-Line*, but must be closed interactively.

Problems caused by the separation of the color layers indicate that a map interpretation based on the corresponding structure primitives is a compromise. If possible, the use of individual separate map layers is preferred. In this case, each map layer must be image processed, but, of course, separating the color layers is not necessary. Then the resulting data can be interpreted in a knowledge-directed way. Knowing the meanings of individual layers beforehand allows a simplification of

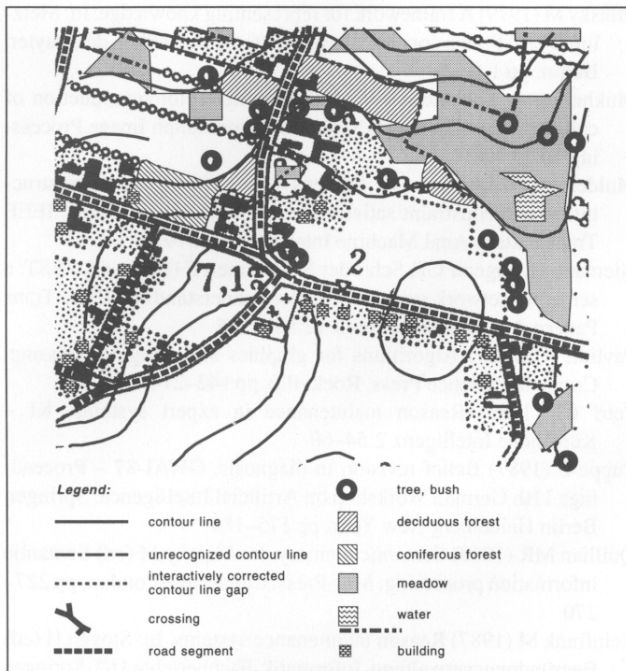


Fig. 17. Interpretation results for the map shown in Fig. 5

the model. The use of individual layers needs additional registration of input image data, which may make defining control points difficult.

The results prove that the frame-based method described is suitable for image-interpretating German topographic maps on a scale 1:25 000. Raster preprocessing together with an integral representation scheme in the form of a semantic net based on frames was achieved. The corresponding analysis strategy supports the high information density and variety of this kind of maps and the acquisition of the spatial relationship of geographically relevant map entities. The definition of the hierarchical map model allows simple acquisition and structuring of the information as well as of the complex relations between the information units via individual concepts of the various abstraction levels. The efficiency of the proposed method is based mainly on the combination of raster preprocessing and new mechanisms for modeling and analysis of recursive structures, of conflicts and identities, of alternative reasons for an instantiation, and of the enormous number of data caused by the high information density of German topographic maps. A uniform representation scheme and a uniform strategy of analysis support modularity and flexibility. They guarantee the extensibility and adaptability of the method developed for interpreting various kinds of map signs.

The systems described in the literature focus on automatic data acquisition from maps on a large scale that have a lower density and little variety of information. Methods for analyzing maps on a medium scale are limited to interpreting single information layers like road nets with meanings that are partially known a priori. Innovative knowledge-based solutions for the analysis of cadastral maps (Mayer et al. 1992) can be

extended, on principle, for the analysis of topographic maps. However, corresponding investigations are still pending and no reports are available at present. Therefore, the lack of comparable solutions rules out a discussion of the proposed results in comparison to the results of other methods.

10 Summary and conclusions

An overview of the PROMAP image analysis system for interpreting topographic maps was presented. The methods implemented for raster-data processing, knowledge organization, and knowledge use were discussed.

The main ideas of raster data processing are color scanning, separation of color layers, raster symbol and raster object recognition, and vectorization. The principles of knowledge-directed interpretation are the hierarchical structuring of the map with map objects and relations, the use of concepts (prototypes) as the basic representation building block, and instantiation as a basic control mechanism. The instantiation includes both the aggregation of instances and the matching of concepts and aggregations of instances. Finally, the capabilities of the system are demonstrated on two real map scenes.

Further work will be directed towards the conflict-solving module and the improvement of the existing methods. Eventually, the system will be tested with more map scenes. Furthermore, how more complex raster processing steps for extraction of structure primitives (Ebi and Besslich 1991) can improve the overall system performance must be investigated.

Acknowledgements. This work was supported by the German Research Association (DFG) under grant Stäblein/Besslich: Environmental Planning System (Sta 126/19-1).

References

- Alemanly J, Kasturi R (1987) A computer-vision system for interpretation of paper-based maps. *Proceedings SPIE Conference on Applications of Digital Image Processing X* 829:125–137
- Andreadis I, Browne M, Swift JA (1990) Image pixel classification by chromaticity analysis. *Patt Recogn Letters* 11:51–58
- Besslich P, Delius I, Ebi N, Lauterbach B, Stäblein G (1992) Umweltplanungssystem: Integration wissensbasierter Bildinterpretation von Karten zur Bewertung von Standorteignung und Bodenschutz. In: Peschel GJ (ed) *Wissensbasierte Systeme in den Geowissenschaften. Beiträge zur Mathematischen Geologie und Geoinformatik* (Original in German, pp 18–25)
- Commission Internationale de l'Eclairage (International Illumination Commission) (1986) *Colorimetry* (Original in German). CIE Publication Nr. 15.2, Vienna
- Doyle J (1979) A truth maintenance system. *Artif Intell* 12:231–272
- Ebi N, Besslich P (1991) Extraktion paralleler Linienstrukturen am Beispiel topographischer Karten (Original in German). In: Radig B (ed) *Mustererkennung 1991, Proceedings 13. DAGM-Symposium, Informatik-Fachberichte 190*. Springer, Berlin Heidelberg New York, pp 312–319
- Ebi N, Lauterbach B, Besslich P (1992) Automatic data acquisition from topographic maps using a knowledge-based image analysis system. In: Fritz LW, Lucas JR (eds) *International Archives*

- of Photogrammetry & Remote Sensing, Volume XXIX, Part B4, Commission IV, International Society for Photogrammetry and Remote Sensing (ISPRS), American Society for Photogrammetry and Remote Sensing, pp 655–663
- Chu YK, Suen CY (1986) An alternate smoothing and stripping algorithm for thinning digital binary patterns. *Signal Processing* 11:207–222
- de Kleer J (1986) An assumption-based TMS. *Artif Intell* 28:127–162
- Harmon P, King D (1985) *Expert systems – artificial intelligence in business*. Wiley & Sons, New York, pp 44–46
- Ilg M, Kübler O (1989) *Wissensbasiertes Verstehen von Straßenkarten. Mustererkennung 1989. Proceedings 11. DAGM-Symposium, Informatik-Fachberichte 219 (Original in German)*. Springer, Berlin Heidelberg New York, pp 354–358
- Illert A (1990) *Automatische Erfassung von Kartenschriften, Symbolen und Grundrißobjekten aus der Deutschen Grundkarte 1:5000 (Original in German)*. *Wissenschaftliche Arbeiten der Fachrichtung Vermessung an der Universität Hannover, Nr. 166*, Hannover
- Kasson JM, Plouffe W (1991) *Subsampled device-independent interchange color spaces*. In: Bender W, Plouffe W (eds) *Image Handling and Reproduction Systems Integration. Proceedings of SPIE 1460:11–19*
- Kasturi R, Fernandez R, Amlani ML, Feng W (1989) *Map data processing in geographic information systems*. *IEEE Comput Soc* 12:10–21
- Krasner GE, Pope ST (1988) *A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. ParcPlace Systems*, Palo Alto, Calif
- Lauterbach B, Anheier W (1993) *Segmentierung farbiger kartographischer Vorlagen in empfindungsgemäßen Farbräumen (Original in German)*. In: Poepl S, Haudels H (eds) *Mustererkennung 1993, Proceedings 15. DAGM-Symposium*, Springer, Berlin Heidelberg New York, pp 733–740
- Lauterbach B, Besslich P (1991) *A neural network based recognition of complex two-dimensional objects*. *Proceedings of the International Conference on Industrial Electronics, Control and Instrumentation, Kobe, Japan*. *IEEE Publishing Services, New York* 3:2504–2509
- Lauterbach B, Ebi N, Besslich P (1992) *PROMAP – a system for analysis of topographic maps*. *Proceedings of IEEE Workshop on Applications of Computer Vision, Palm Springs, Calif, IEEE Computer Society Press*, pp 46–55
- Mayer H, Heipke C, Maderlechner G (1992) *Knowledge-based interpretation of scanned large-scale maps using multi-level modelling*. In: Fritz LW, Lucas JR (1992) *International Archives of Photogrammetry & Remote Sensing, Volumen XXIX, Part B3, Commission IV, International Society for Photogrammetry and Remote Sensing (ISPRS)*. American Society for Photogrammetry and Remote Sensing, pp 578–585
- Minsky M (1979) *A framework for representing knowledge*. In: Metzger D (ed) *Frame conceptions and text understanding*. de Gruyter, Berlin, pp 1–25
- Mukherjee A, Venkatesh YV (1986) *Digital color reproduction of color television monitors*. *Comput Vision Graph Image Processing* 36:114–132
- Mulder JA, Mackworth AK, Havens WS (1988) *Knowledge structuring and constraint satisfaction: the map see approach*. *IEEE Trans Pattern Anal Machine Intell* 10:866–879
- Niemann H, Sagerer GF, Schröder S, Kummert F (1990) *ERNEST: a semantic network system for pattern understanding*. *IEEE Trans Pattern Anal Machine Intell* 12:883–905
- Pavlidis T (1987) *Algorithms for graphics and image processing*. Computer Science Press, Rockville, pp 142–214
- Petri CJ (1989) *Reason maintenance in expert systems*. *KI – Künstliche Intelligenz* 2:54–60
- Puppe F (1987) *Belief revision in diagnosis*. *GWAI-87 – Proceedings 11th German Workshop on Artificial Intelligence*. Springer, Berlin Heidelberg New York, pp 175–184
- Quillian MR (1968) *Semantic memory*. In: Minsky M (ed) *Semantic information processing*, MIT Press, Cambridge London, pp 227–270
- Reinfrank M (1987) *Reason maintenance systems*. In: Stoyan H (ed) *Begründungsverwaltung, Informatik-Fachberichte 162*. Springer, Berlin Heidelberg New York, pp 1–26
- Richter M (1981) *Einführung in die Farbmatrik*. de Gruyter, Berlin New York
- Sagerer G (1985) *Darstellung und Nutzung von Expertenwissen für ein Bildanalyse-system (Original in German)*. Springer, Berlin Heidelberg New York
- Schoppmeyer J (1991) *Farbreproduktion in der Kartographie und ihre theoretischen Grundlagen (Original in German)*. Polygraph Verlag, Frankfurt, pp 42–74
- Sezan MI (1990) *A peak detection algorithm and its application to histogram-based image data reduction*. *Comput Vision Graph Image Processing* 49:36–51
- Shen W, Ebi N, Besslich P (1991) *Schraffurerkennung in graphischen Vorlagen (Original in German)*. In: Radig B (ed) *Mustererkennung 1991, Proceedings 13. DAGM-Symposium, Informatik-Fachberichte 290*. Springer, Berlin Heidelberg New York, pp 312–319
- Suzuki S, Yamada T (1990) *MARIS: map recognition input system*. In: Pau LF (ed) *Mapping and spatial modelling for navigation*. Springer, Berlin Heidelberg New York, pp 95–116
- Tomita F, Tsuji S (1990) *Computer analysis of visual textures*. Kluwer Academic, Norwell, pp 71–92
- Yang J (1989) *Automatische Digitalisierung von Deckfolien der Deutschen Grundkarte 1:5000 – Bodenkarte (Original in German)*. *Wissenschaftliche Arbeiten der Fachrichtung Vermessung an der Universität Hannover, No. 161*