N91-15625

A Computer Vision System for the Recognition of Trees in Aerial Photographs

Axel J. Pinz
Institute of Surveying and Remote Sensing
University of Agriculture
Peter Jordan Str. 82
A-1190 Wien, Austria

Abstract: Increasing problems of forest damage in Central Europe set the demand for an appropriate forest damage assessment tool. In this paper the *V*ision *E*xpert *S*ystem *VES* is presented. VES is capable of finding trees in color infrared aerial photographs - this is the first step towards an automatic forest damage interpretation system. Concept and architecture of VES are discussed briefly. The system is applied to a multisource test data set. The processing of this multisource data set leads to a multiple interpretation result for one scene. An integration of these results will provide a better scene description by the vision system. This is achieved by an implementation of Steven's correlation algorithm.

Key words: Aerial image understanding, image understanding, knowledge-based image analysis, frames, object representation for computer vision systems, dot pattern correlation

1 INTRODUCTION

1.1 Forest damage interpretation

During the past years research concerning the assessment of forest damage using color infrared aerial photographs was done at IVF. *IVF* stands for "*I*nstitut für *V*ermessungswesen und *F*ernerkundung" - the Institute of Surveying and Remote Sensing at the University of Agriculture in Vienna. The benefits of color infrared aerial photographs for the interpretation of vegetation are discussed in detail in [Sch89]. However, to be able to understand the method described in this paper, the reader should be familiar with a few details.

The condition of a tree is evaluated by interpreting the color of its crown in a color infrared aerial photograph. Since, compared to damaged vegetation, healthy vegetation tends to reflect more light in the infrared band and less in the red one (see Fig. 1.1), *healthy trees look red* in a color infrared photograph, while bad trees will have less red and more green color, thus appearing pale. But the color of a tree will depend on both the tree's vitality and the *tree species*. For example, a healthy pine will show a color similar to the one of a damaged spruce.

In many parts of Central Europe a very intensive and heterogeneous kind of landuse takes place. From the forest damage interpretation point of view this means, that normally many different kinds of trees will be found within one forest stand. Also, the condition of the trees in a stand may vary significantly. In a typical Austrian forest it is quite common to find a pine by the side of a spruce and to find a healthy tree close to a very bad one. As a consequence, to get correct results of a "forest-condition-inventory", as it is called in Austria, it is necessary, to interpret *the species and the color of the single tree*. Trying to use remote sensing methods for this forest-inventory, data from satellites like LANDSAT or SPOT are not convenient, only aerial photographs will provide sufficient spatial resolution.

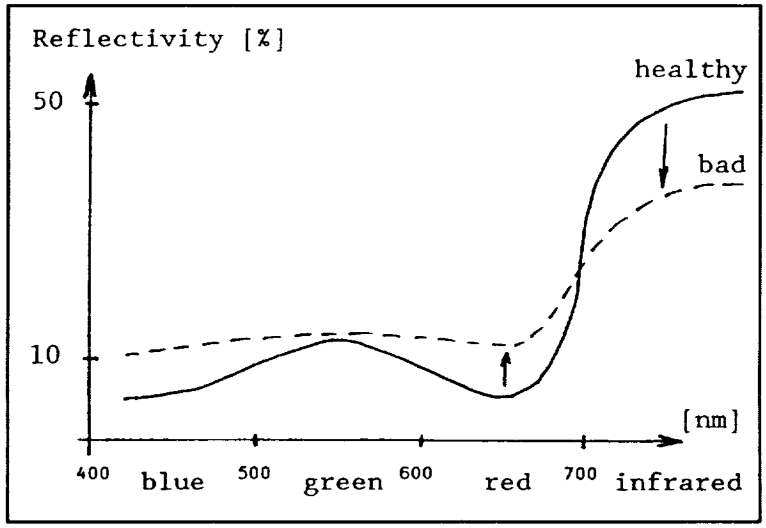


Fig. 1.1 Reflectivity of vegetation (in principle)

Interpreting color infrared aerial photographs for forest inventory purposes therefore calls for the following procedure:

1. Find a tree in the aerial photograph.
2. Determine the tree species.
3. Determine the tree vitality by interpretation of the color (and the texture) of the tree.

In this paper we discuss the problem of finding trees in aerial photographs (1.) by means of computer vision. While the color information is required for the determination of species and condition of a tree (2. and 3.), tree-finding can be done using a monochrome image. Therefore in this paper only monochrome images are shown. They were produced by averaging the three color channels of a color infrared image.

1.2 A tree finding computer vision system

In addition to the task of finding trees the application of a computer vision system will be extended to serve for several remote sensing tasks at IVF. For this purpose an image understanding system - the *V*ision *E*xpert *S*ystem *VES* - was built. The architecture of VES has already been presented in detail in [Pin88] and [Pin89]. The system therefore will be discussed very briefly in chapter 2. Figures 1.2 and 1.3 show the result of VES processing a typical test-image. The scale of the image was 1:4000 and it was digitized with a pixel size of 25μm. The digital image was 512x512 pixels (Fig. 1.2) and VES found 169 circular image objects from which 70 scene objects were derived (Fig. 1.3).

Fig. 1.2 Original image

Fig. 1.3 VES result

There were several problems encountered in the course of this first application of VES. First of all, the pixel scale was unrealistic - 1 pixel represented a square of $10cm^2$ in the scene. Second, the system was very slow due to an inadequate hardware component. Third, the experience with the system led to more sophisticated ideas about representation and about the evaluation of the interpretation result.

As a consequence, a successor system of VES - the *Vision Station VS* - is currently under development at IVF. In a first step the VES functionality was ported to VS. Due to the better performance of VS most of the "VES-results" presented in this paper were done on the VS simulating a VES-behaviour.

At this point the evaluation problem should be discussed in more detail. A computer vision system starts with a given image and a problem specification (e.g. "find trees"). As the process of automatic image interpretation proceeds, a *scene description* begins to emerge. In the case of VES this is a two-stage process. At first *image objects* are found. Then some of them are put into relation to a certain *scene object*. There are several control strategies for vision systems: top-down, bottom-up and bidirectional (Fig. 1.4). The features of each of these strategies were discussed by Matsuyama [Mat87]. He and many others (e.g. [Hav83], [Keo85], [Pin89], [Nag80]) tried to avoid the problem of combinatorial explosion of the search size in a bidirectional system by using search space limiting control structures (either top-down/bottom-up or other limiting techniques in a bidirectional system). Besides these "conventional" approaches there have been more recent efforts to find other control mechanisms (e.g. Matsuyama's hypergraph [Mat88] or Burt's pattern tree [Bur88]). However, for a conventional system it is crucial to be able to *evaluate the interpretation results*. In VES and VS we try to calculate a quality value for each object. This helps in discarding of very uncertain objects. But these quality value calculations sometimes are imprecise themselves and the crucial questions still remain: Is the result correct? Is the result complete? Are there still objects missing? Can the interpretation process be terminated? As a conclusion, any additional source helping to improve the quality assessment should be used. In this paper we will investigate the *use of multisource data to gain a more robust scene-description*.

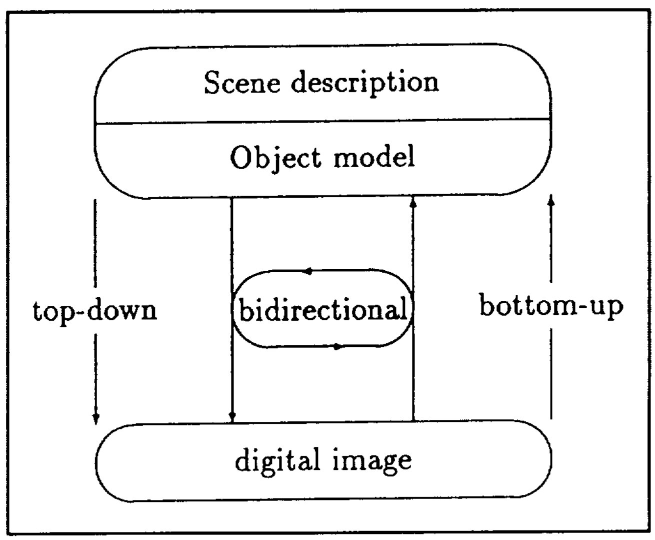
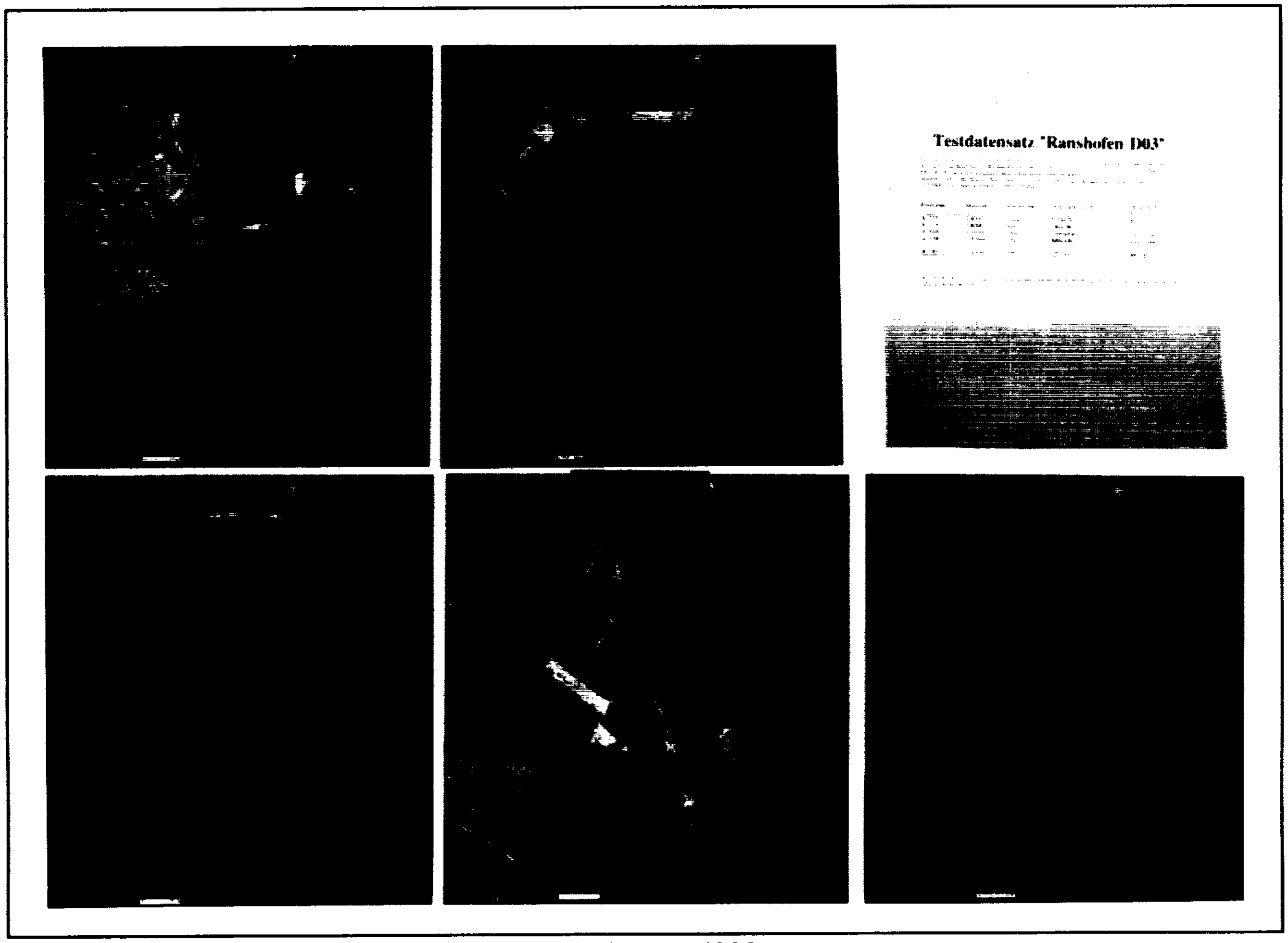


Fig. 1.4 Control strategies

1.3 The test data set

The test data set is shown in Fig. 1.5. It consists of five aerial images taken at April 15, 1984 (images a. - d.) and August 23, 1984 (image e.). There are four different scales: 1:32000 (a.), 1:16000 (b.), 1:8000 (c.) and 1:4000 (d. and e.). These aerial images originally were taken to investigate the abilities of human interpreters. It turned out that while it is still possible to locate a tree in the 1:32000 image, the correct determination of tree species and tree vitality calls for a scale of about 1:12000 - 1:15000 (this will also depend on the selected film type and on the exposure and development conditions) [Sch89].



a. spring 1:32000 b. spring 1:16000
c. spring 1:8000 d. spring 1:4000 e. summer 1:4000

Fig. 1.5 The test data set "Ranshofen D03"

Small portions of these five images, *each showing the same part of the scene*, were digitized with 25μm (a. and b.), 50μm (c.) and 100μm (d. and e.) pixel size. This lead to a pixel scale of approximately 40cm in the scene (b. - e.) and 80cm in the case of a.. We plan to use this data set for several purposes. We want to investigate resolution-dependent performance variations in automatic tree detection and species interpretation [Bis89], [Pin90]. The data set also supplies *different views* (in space and time) of the same objects. It is therefore expected to get a *more robust scene description* by proper combination of results from several images.

1.4 Related work

Aerial image analysis has always been a major field of application for model based vision systems. Most of them were concerned with finding *artificial, man-made objects*. McKeown et al. present a rule-based approach in the system SPAM [Keo85]. Several systems were developed by Matsuyama (e.g. ACRONYM, SIGMA, LLVE) [Mat87]. He used frames and he examined the three "classical" control strategies bottom-up, top-down and bidirectional. VES also uses frames, which were introduced by Minsky as a proper form of representation for vision tasks [Min75]. In the Mapsee2 system the similar concept of schemas was used for knowledge representation [Hav83]. In our Vision Station the representation of objects is based on the Common Lisp Object System CLOS [Bob88]. More recent work (e.g. Burt's pattern tree [Bur88], Matsuyama's multilayered hypergraph [Mat88]) deals with hierarchical (pyramid) control structures, trying to avoid the drawbacks of top-down, bottom-up or bidirectional. Earlier work includes the VISIONS-System [Han78a],[Han78b] and a system by Nagao and Matsuyama [Nag80].

Most computer vision systems use a kind of modeling mechanism. There are object models in the scene domain (3D) and image objects (2D). Image objects are found during the interpretation process, thus being individual (vs. generic) objects. One can distinguish between the four object classes discussed in detail below (see 2.3: scene/image, generic/individual). In comparison to other systems, where a border between two classes may be missing or implicitly defined (see e.g.: discussion of the importance of discriminating between image level and scene level information [Mat87], short vs. long term memory in VISIONS [Han78b]), there is an exact definition of all four classes in VES. This object representation scheme is in fact controlling most of the VES-processes.

A complete computational model is given by Marr [Mar82]. Viewing our results as "place-tokens" in the sense of Marr, we found a structure similar to Glass patterns [Gla69] and we tried to correlate the results from different images using Steven's algorithm [Ste78]. Several mathematical models were developed to describe the phenomenon of orientation perception in random dot patterns [Mat90].

Dealing with the problem of the *interpretation of natural (vs. man-made) scenes*, the effort is often directed towards a complete segmentation of the image (e.g. [Oht85], [Naz84]). Related work concerning the application of finding trees in aerial photographs was done by Haenel et al. [Hae87]. While he developed very specific algorithms for this task, we try to establish a more universal vision system. Supplied with proper knowledge, VES and VS will be able to solve many other perceptual tasks in remote sensing.

2 THE VISION EXPERT SYSTEM VES

There were several major goals in the development of VES. The system architecture should be open and flexible. VES should be appropriate for a broad field of applications and experiments. The resulting complicated framework was then filled with knowledge and methods for the specific problem domain of finding trees. This was the first application test of VES.

2.1 Architecture and implementation

The claimed universality of the system together with the available hard- and software at IVF led to a hybrid architecture. The system consists of a host computer and an image processing system. While under VES both the image processing software and the LISP-system is run on the same host, in the VS-environment the LISP-part is done on a seperate workstation. This is shown by the

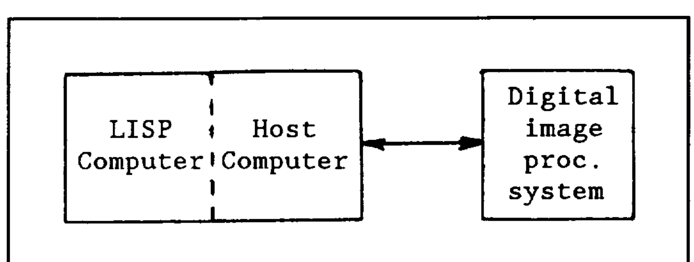


Fig. 2.1 Hardware components

dashed line in Fig. 2.1. The interaction between the software components is illustrated by Fig. 2.2.

VES is organized as a top-down strategy vision system with the possibility of being extended to a bidirectional system in the future. Core part of the system is the object representation in frames. VES is implemented in INTERLISP. The frame representation language FRL was used as a basis for the VES frames [Rob77]. Most of the digital image processing modules are written in PASCAL.

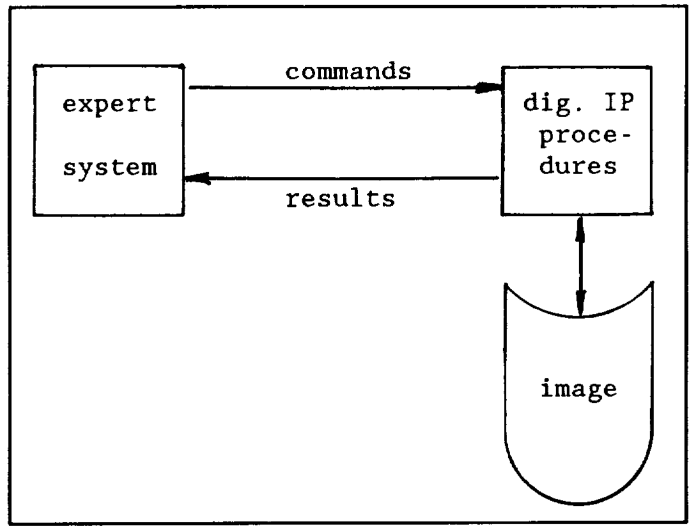


Fig. 2.2 Interaction of software components

2.2 The VES frames

With the exception of two rules all the explicit knowledge is stored in frames. There are *object-*, *method-* and *procedure-frames*. The frames are interconnected by various relations (e.g. ako/instance, part/whole, represents/rep-by) thus forming groups of several semantic networks.

If there is knowledge about how to find a certain object, then the slot METHLIST of this object's object-frame contains a list of applicable methods, each element pointing at a method-frame. When a method is selected and applied the result usually is a sequence of processes. Some of them will be LISP-functions, others are image processing modules. The interface between LISP and the image processing modules is handled by the procedure-frames. They contain information about the calling sequence, parameters and resulting effects of an image processing module.

2.3 Object representation

We distinguish between *scene objects* (OBSC) and *image objects* (OBIM) on the one hand and between *generic objects* and *individual objects* on the other hand. While the latter (CLASSIFICATION GENERIC or INDIVIDUAL) are a standard feature of FRL to separate models from instances, the distinction between scene- and image-objects is quite common for a computer vision system. In Fig. 2.3 the regions A and B represent the system's initial knowledge before an interpretation is started ("static knowledge") - the models for scene objects and models for image objects. Regions C and D constitute the "dynamic knowledge" about the interpreted scene. During the process of image interpretation, at first individual image objects are found (region D), later instances for corresponding individual scene objects are established (region C). From the VES point of view, region C is the result of a successful image interpretation: it contains all scene objects which the system has found in an image taken from a certain scene. This is a description by objects, not a segmentation of the image. Normally the objects don't cover all of the area of the image. During the course of an interpretation process, the system will try possible relations between hypotheses for scene objects and already-found image objects. It will end up with the best relation which finally constitutes the correct interpretation for the image object.

Fig. 2.4 gives an example of an interpretation situation. The world is divided into scene- and image-objects. An individual scene object (pine0) was found - pine0 is a pine, a tree and a scene object. It is represented in the image by circle8. Circle8 is an individual circle, an area (vs. point or line) and an image object. It currently represents the scene object pine0.

A **scene objects OBSC**	**image objects OBIM** B
(AKO ($VALUE (OBSC))) (CLASSIFICATION ($VALUE (GENERIC))) **generic objects**	(AKO ($VALUE (OBIM))) (CLASSIFICATION ($VALUE (INDIVIDUAL)))
individual objects (AKO ($VALUE (OBSC))) (CLASSIFICATION ($VALUE (INDIVIDUAL))) C	(AKO ($VALUE (OBIM))) (CLASSIFICATION ($VALUE (INDIVIDUAL))) D

Fig. 2.3 The four different object classes of VES

2.4 Control of the interpretation process

The interpretation process is always invoked by the search for an object. A valid object must be represented in a generic frame. Correct search commands might be:

(FIND '(TREE))	...	find trees,
(FIND '(TREE ROAD))	...	find trees and roads,
(FIND '(CIRCLE))	...	find circles (image objects).

After an initialization phase (loading and establishing of global parameters like name of the image, scale, etc.) the system grasps the frame representing the object being searched for and the top-down search process begins. The methods found in the slot METHLIST are evaluated and the best method is chosen. While the search for image objects yields individual image objects, the search for scene objects forces the search for corresponding image objects. For example, "find tree" or "find road" might invoke "find circle" or "find line". If image objects are found they must survive object-specific tests which are also stored in the method frame. Next, a scene object is generated and the corresponding relations between scene- and image-object are set. A method may also contain tests for scene objects. If a test fails, the scene object will be removed while the image object remains. This completes a top-down process. A list of individual objects which are all instances of the generic object that had been searched for was produced.

Two rules extend this pure top-down strategy. VES is trying to improve the interpretation by applying these rules again and again, until no rule fires any more, thus finishing the complete interpretation process.

Rule 1: If there are "tunable" parameters for an object being searched for, try to vary one parameter and repeat the search.

Rule 2: If an object being searched for is known to have "contrary" objects, then extend the search to these objects and check if a conflict occurs.

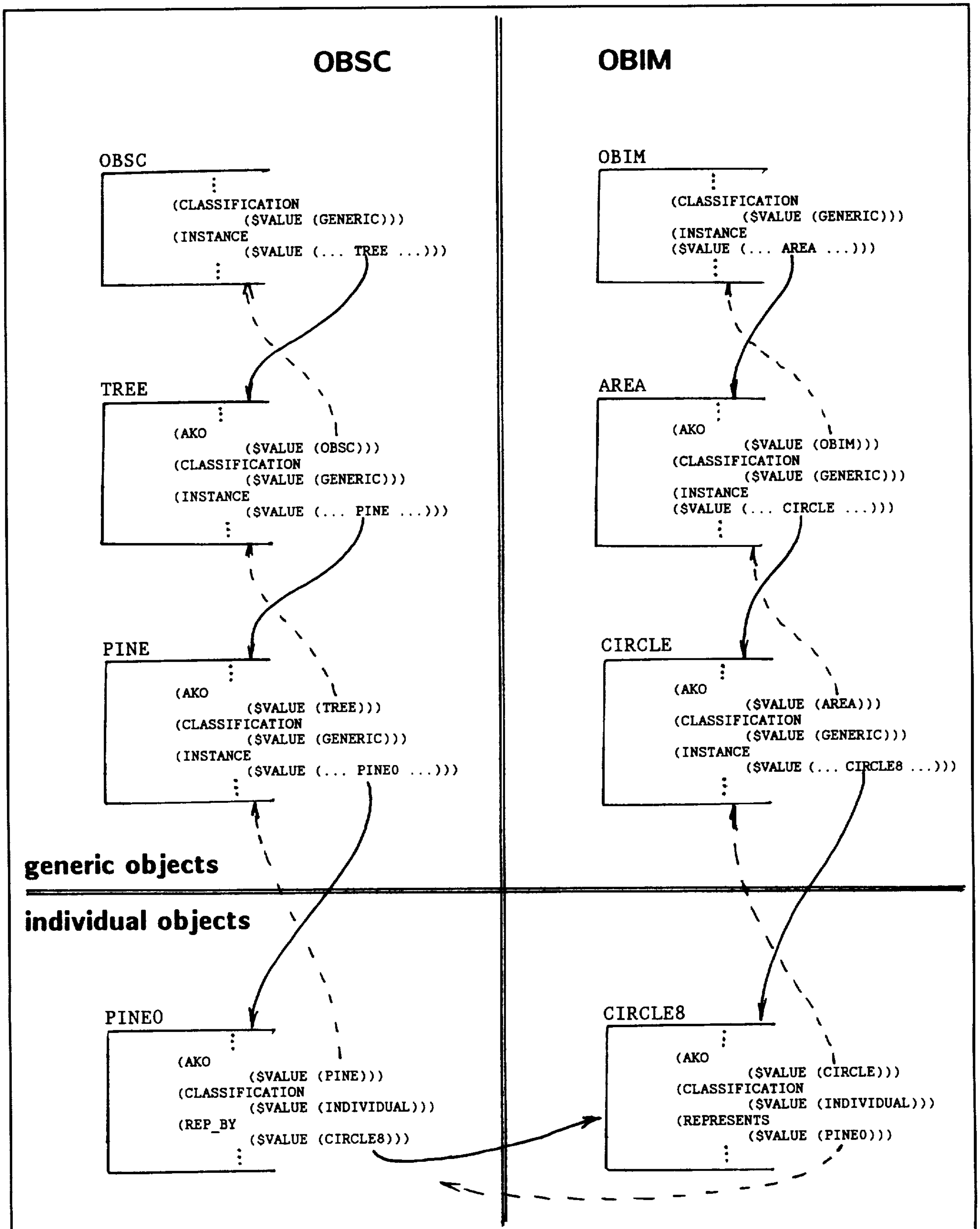


Fig. 2.4 An example of scene and image objects

3 DISCUSSING VES PROCESSES

In this chapter the processes and methods which were implemented to recognize trees in aerial photos are discussed. Fig. 3.1 displays a very simplified scheme of the processes in VES. Starting with the task (usually entered by the user) of finding a certain scene object OBSC, the search for a corresponding image object OBIM is initiated. Image objects are found and connected with scene objects, thus finishing one top-down process. Application of global rules leads to several repetitions until no rule is applicable any more. The corresponding up-arrow in Fig. 3.1 is marked with a dashed line because it is also possible to request one single top-down process without application of global rules.

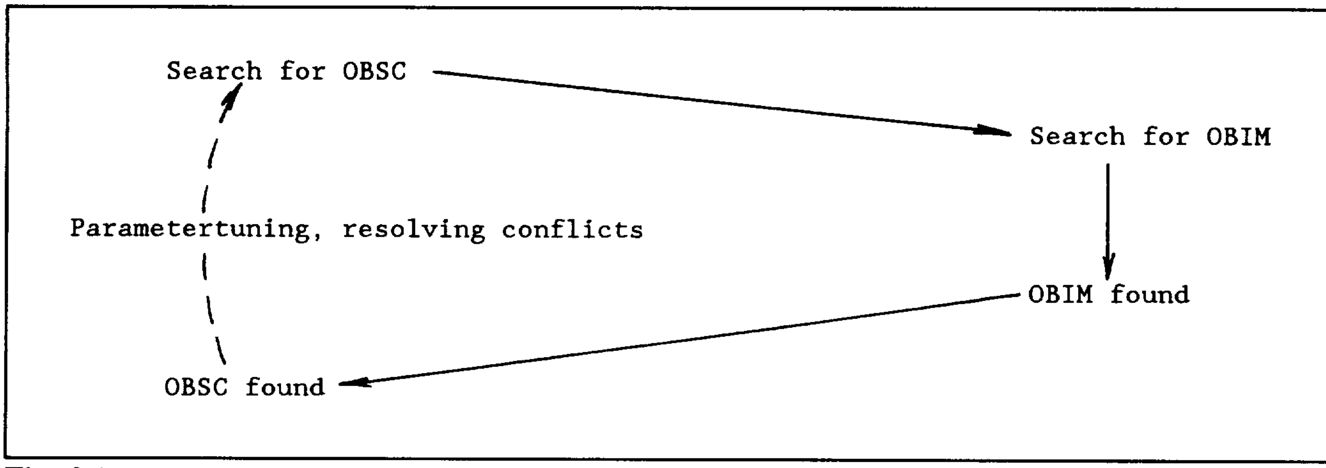


Fig. 3.1 VES processes

After the initialization phase, VES is ready to accept search commands. One top-down process is started by

(FIND '(TREE)) ... find trees.

A complete process, including multiple repetitions by application of the global rules, is invoked by

(START '(TREE)).

VES finds the method METH0 in the slot METHLIST of the frame TREE. METH0 assumes trees to appear as bright circularly shaped image objects. This assumption holds for trees inside a forest and is a very good assumption to make in the central parts of an aerial photo where objects are viewed from above. Towards the edges of the photograph, the direction of view is changing, e.g. a spruce appearing not circular but triangular in shape. At first METH0 is searching for bright circular image objects, next, every circle is assigned to an individual scene object "tree". This is followed by a test. If two trees are standing too close to one another, the tree with the larger radius is removed.

The application of METH0 automatically invokes the new task of

(FIND '(CIRCLE)) ... find circles.

The structure of the frame CIRCLE is similar to the one of TREE. The method METH1, searching for bright circular image-objects in a stepwise process, is found in the slot METHLIST.

A bright circular object may be viewed as a local maximum of brightness in the image. Usually there will be a lot of texture information found within a tree's crown. This would lead to many local maxima within one crown. Therefore, a lowpass filter must be applied before the search for local maxima can take place.

The original black and white image (it was produced by averaging the 3 channels of a color infrared image) is the input to METH1. Lowpass filtering is achieved by a local window operation using the image processing system. The size of the window (the "size" of the lowpass) is calculated from the image's scale and the expected size of the searched object (radius of the tree's crown = radius of circle). Next to the lowpass filtering the local maxima are searched for. Because of the preceding lowpass filtering, a local maximum usually covers an area of pixels of equal brightness. The center of gravity of each area is taken as the exact location of the local maximum.

In the final step METH1 checks the found object for circular shape by inspecting the "radial brightness distribution". This distribution is obtained by drawing concentric circles around the maximum's position, summing up all pixels lying on a circle and taking the average (see Fig. 3.2). For a circular object the resulting diagram (mean brightness / radius) should show a distribution as in Fig. 3.2. The module which is computing the radial brightness distribution to decide whether the object is circular needs the following three input parameters: smallest radius, largest radius and minimum brightness decrease (the mean brightness has to be n% lower at the edge of the object than at its center). It turned out, that the necessary brightness decrease n is scale-dependent. In images of a scale of 1:4000 a good value for n was 35 - 40 %, while n had to be reduced to 30 % for scales of about 1:8000. The module returns either the radius of the found circular object at which this minimum decrease is reached or NIL, if any of the above three conditions do not hold.

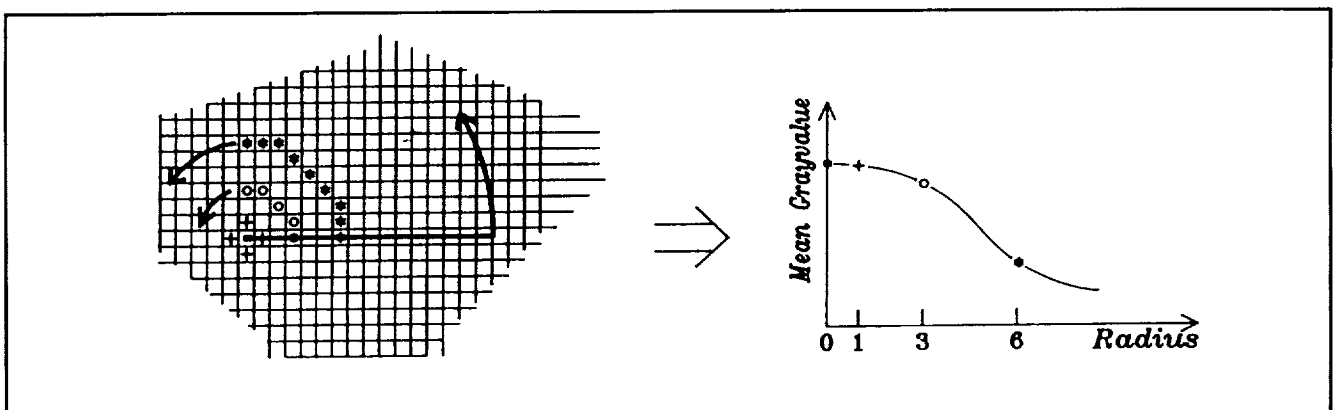


Fig. 3.2 Radial brightness distribution

This completes one top-down shot. The two main stages are shown in Fig. 3.3 and Fig. 3.4 (the original image is Fig. 1.2). Fig. 3.3 shows the lowpass filter (in this case a 25x25 window lowpass was selected by VES) together with the local maxima. Fig. 3.4 shows the corresponding circles that survived the "radial brightness distribution" test. Each of these circles is assigned to a scene object (tree). Some of the trees are removed by the final test in METH0 (if standing too close).

If the interpretation process is started by (START '(TREE)), the global rules will be applied. The parameter variation will produce two more lowpass filters and this will result in new local maxima, circles and trees. The search for contrary objects (in this test case a road was entered manually) leads to the elimination of trees that would grow in the middle of a road. The final result shown in Fig. 1.3 was obtained after two parameter variations (19x19 and 31x31 lowpass window).

Fig. 3.3 Local maxima

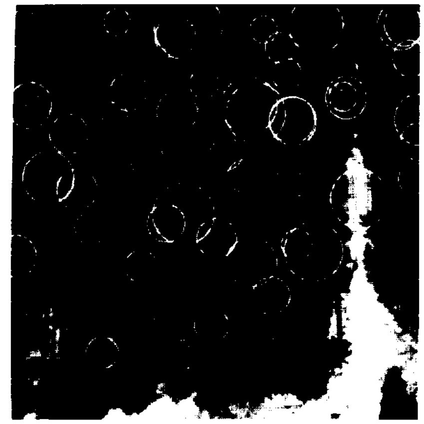


Fig. 3.4 Circles from Fig. 3.3

4 PROCESSING THE TEST DATA SET

We took a small portion of each of the five images Fig. 1.5 a. - e. each showing approximately the same part of the scene. The size of these portions is 512x512 pixels (b. - e.) and 256x256 pixels (a.). All five images were processed with the standard VES tree-search (search for a default crown radius of 2,5m followed by two parameter variations (1,25m and 5m)). The original 512x512

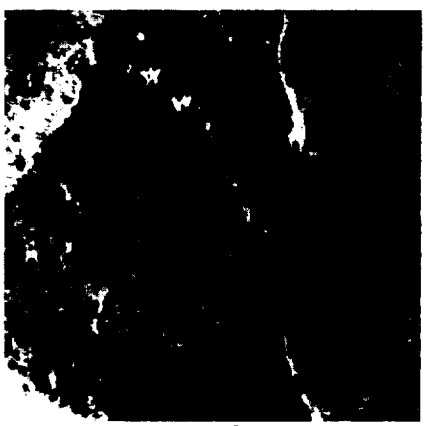


Fig 4.1 512^2 portion of d.

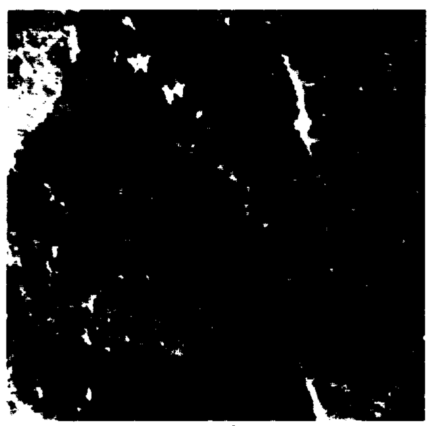


Fig. 4.2 512^2 portion of c.

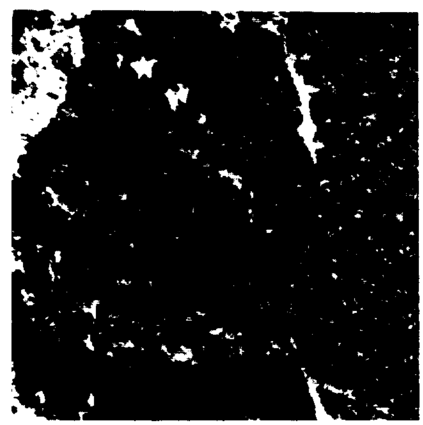


Fig. 4.3 Circles from Fig. 4.2

Fig. 4.4 128^2 portion of d.

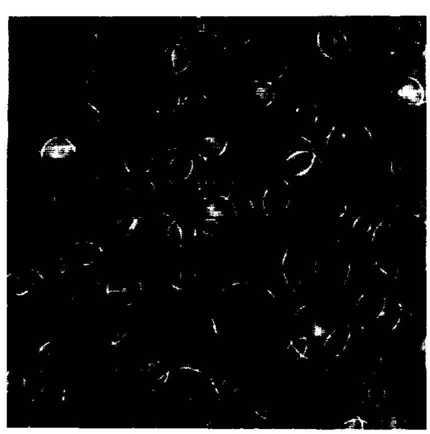


Fig. 4.5 128^2 portion of c.

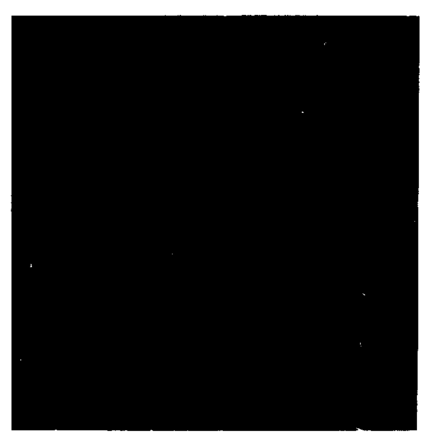


Fig 4.6 A correlation result

images are shown for d. (Fig. 4.1) and c. (Fig. 4.2). Fig. 4.3 shows the circles found in Fig. 4.2 after the first top-down process. The final results (trees found) are shown in detail for 128x128 portions of d. (Fig. 4.4) and c. (Fig. 4.5).

The results of this experiment were very interesting: While even in the worst case (1:32000, image a.) many of the large crowns were detected, there was no "perfect" interpretation in any of the five cases a. - e.. Of course, the best results were obtained for the larger scales (c. - e.). But in each result there were several trees missing that were found in another case. The same is true for erroneous artifacts, which don't show up in more than one result at the same location. As a conclusion - the desired result of the interpretation of the whole data set (a. - e.) would be a careful *combination of the several results*. And, working with "intelligent" vision systems, we would favour a robust solution that doesn't require too precise and detailed instructions, similar to the ability of a person to identify the same tree in two different images.

As a first step towards this goal we tried the following procedure. We generated dot images of the five results. For each tree a dot mark located at the center of the circle representing the tree was produced. When two different results were overlayed and displayed in different colors, the resulting image was very similar to the dot patterns described by Glass [Gla69] and Stevens [Ste78]. In our case the patterns of one result may be converted to another one by assuming a superimposition of translation, rotation and a small change of scale. The remaining "noise" is caused by the individual height of each tree, and by the different position of the sun and viewing position for each image. In addition, due to the imperfect interpretation, some points are missing or added in the other image. Stevens called his patterns "Glass patterns" and he developed a *local algorithm* for the correct correlation of associated points. We implemented Steven's algorithm and tested it on the dot images generated from the interpretation results of a. - e.. One result of a correlation between the two images shown in Fig. 4.4 and Fig. 4.5 is shown in Fig. 4.6.

The results of this experiment were imperfect but very promising. Taken alone, Steven's algorithm is not effective enough for our patterns. This is due to the noise effects discussed above and due to the occurence of rather large point displacements. The algorithm will have to be adapted for our purposes - there are already several ideas for improvements. When viewed as one component of a larger vision system, even the actual performance of the algorithm is valuable. The correlation results will be processed by VES. Several heuristics may be applied, e.g. the fact that correlated trees should be of similar size. The correlation should also hold for more than two of the results (a. - e.). If there is a component in the system, that is able to determine the tree species [Bis89,Pin90], then correlated trees must have the same species. Current research at IVF is addressing these topics.

5 CONCLUDING REMARKS

It has been shown, that the use of multisource data can improve the quality and robustness of the interpretation result of a computer vision system. While synergic effects of this kind are well known, the proposed approach is also robust from another point of view. We do not need the geometric rectification of our multisource data to compare them. We also don't need complete or very accurate correlation results. The system is able of comparing two objects from two scenes just like a human interpreter looking at the two images. In a way the knowledge of a system like VES may be viewed as an alternate data source itself.

Many problems were discussed only very briefly or not at all. The ideas about the representation of objects, processes and methods in VES are improved in the VS environment. This representation problem is closely coupled with the problem of control of the interpretation process. Methods like the one described above can help in getting a better assessment of the current interpretation result. Dealing with multisource data, the representation problem becomes even more difficult: While there is one individual object, there can be several scenes (several scene objects) and many images (many image objects). Furthermore we believe that a good approach for a vision system in a natural environment should be rather different from the one in a man-made environment. Fuzziness in shape and morphology of natural objects has to be reflected in fuzzy and robust models and methods.

ACKNOWLEDGEMENTS

The author wishes to thank Renate Bartl and Horst Bischof for fruitful discussions and their help in producing the experimental results presented in this paper. We thank IMKAI (Inst.f. Med.Kybernetik u.AI., Univ.Wien) for making their implementation of FRL available to us. This research was funded by the following contracts: "Rechnerunterstützte objektivierte Luftbild-interpretation" (Fonds zur Förderung der wissenschaftlichen Forschung Proj. Nr. 4489 und Jubiläumsfonds d. Österr. Nationalbank) und "Schwerpunkt Fernerkundung" (Projekt S38/2 des Fonds zur Förderung der wissenschaftlichen Forschung)

REFERENCES

[Bob88] Bobrow D., et al., "Common Lisp Object System Specification", Draft submitted to X3J13, 1988

[Bis89] Bischof, H. and Pinz, A., "The Use of Neural Networks for the Recognition of Tree Species in Digital Images", in Pinz, A. (ed.): "Knowledge-based Pattern Recognition", OCG Schriftenreihe 49, Oldenbourg 1989, (in German)

[Bur88] Burt, P.J., "Attention Mechanisms for Vision in a Dynamic World", Proc. 9.ICPR, Rome, 1988, pp.977-987

[Gla69] Glass, L., "Moiré Effect from Random Dots", Nature, Vol.223, pp.578-580, Aug. 1969

[Han78a] Hanson A.R. and Riseman E.M. (eds.), "Computer Vision Systems", Academic Press, Orlando, 1978

[Han78b] Hanson A.R. and Riseman E.M., "VISIONS: A Computer System for Interpreting Scenes", in Hanson, Riseman (eds.), "Computer Vision Systems", Academic Press, Orlando, 1978

[Hae87] Haenel S., Tränkner H. and Eckstein W., "Automatic Detection of Tree Crowns in Aerial Photographs: The Path through the Bottleneck", in 2.DFVLR-Statusseminar, Oberpfaffenhofen, 1987, (in German)

[Hav83] Havens W. and Macksworth A., "Representing Knowledge of the Visual World", IEEE Computer, Aug. 1983, pp.90-96

[Mar82] Marr, D., "VISION", Freeman, 1982

[Mat87] Matsuyama, T., "Knowledge-Based Aerial Image Understanding Systems and Expert Systems for Image Processing", IEEE Trans. Geoscience and Remote Sensing, Vol. GE-25, No.3, May 1987, pp.305-316

[Mat88] Matsuyama, T., "Expert Systems for Image Processing - Knowledge-Based Composition of Image Analysis Processes -", Proc. 9.ICPR, Rome, 1988, pp.125-133

[Mat90] Mates J., Lánsky P. and Yakimoff N., "Models for the Perception of Orientation in Random Dot Patterns", Biol. Cybern. 63, pp.71-80, Springer 1990

[Keo85] McKeown D.M., Wilson A.H. and McDermott J., "Rule-Based Interpretation of Aerial Imagery", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No.5, Sept. 1985, pp.570-585

[Min75] Minsky, M., "A framework for representing knowledge", in Winston, P.H. (ed), "The Psychology of Computer Vision", Mc Graw Hill, 1975

[Nag80] Nagao M. and Matsuyama T., "A Structural Analysis of Complex Aerial Photographs", Plenum Press, New York, 1980

[Naz84] Nazif A.M. and Levine M.D., "Low Level Image Segmentation: An Expert System", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No.5, Sept. 1984, pp.555-577

[Oht85] Ohta Y., "Knowledge-Based Interpretation of Outdoor Natural Color Scenes", Pitman publishing, London, 1985

[Pin88] Pinz, A., "An Image-Understanding Expert System for the Recognition of Trees in Color-Infrared Aerial Photographs", Ph.D. Thesis, TU Wien, Okt.1988, (in German)

[Pin89] Pinz, A., "Final Results of the Vision Expert System VES", in Pinz, A. (ed.): "Knowledge-Based Pattern Recognition", OCG Schriftenreihe, Oldenbourg 1989

[Pin90] Pinz A. and Bischof H., "Constructing a Neural Network for the Interpretation of the Species of Trees in Aerial Photographs", Proc. 10.ICPR, Atlantic City, IEEE 1990

[Rob77] Roberts, Goldstein, "The FRL Primer, The FRL Manual", MIT Memos 408 and 409, MIT, 1977

[Sch89] Schneider, W., "Using Remote Sensing for Forest-Inventory; Methods, Possibilities and Limits", FBVA-Berichte, Sonderheft, Wien, 1989, (in German)

[Ste78] Stevens, K.A., "Computation of Locally Parallel Structure", Biol. Cybern. 29, pp.19-28, Spiringer 1978