

DTIC COPY

R&D 5679-EN-01

(1)

AD-A222 675

Automatic Line Network Extraction from Aerial Imagery of Urban Areas through Knowledge Based Image Analysis

Final Technical Report

December 1989

United States Army
Research, Development and Standardization Group UK
London England

DTIC
ELECTE
JUN 12 1990
S D D

Contract Number: DAJA 45-86-C-0049

FGAN-FIM, Eisenstockstr. 12
D-7505 Ettlingen 6, W. Germany

"Original contains color"
plates: All DTIC reproductions
will be in black and
white

Approved for Public Release; distribution unlimited

90 06 11 145

Automatic Line Network Extraction from Aerial Imagery of Urban Areas through Knowledge Based Image Analysis

Final Technical Report

H. Fuger, K. Jurkiewicz, H. Kazmierczak,
B. Nicolin, W. Ott

December 1989

United States Army
Research, Development and Standardization Group UK
London England

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Contract Number: **DAJA 45-86-C-0049**

Contractor: Forschungsgesellschaft fur Angewandte Naturwissenschaften
(FGAN)
Wachtberg, W. Germany

Principal Investigator: Prof. Dr.-Ing. H. Kazmierczak
Forschungsinstitut fur Informationsverarbeitung und Mustererkennung (FIM)
Eisenstockstr. 12, D-7505 Ettlingen, W. Germany

FIM Report No. 209

The Research reported in this document has been made possible through the support and sponsorship of the U.S. Government through its United States Army Research, Development and Standardization Group UK. This report is intended only for the internal management use of the Contractor and the U.S. Government

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188 Exp. Date: Jun 30, 1986	
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC DISTRIBUTION UNLIMITED		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) FIM-No. 209			5. MONITORING ORGANIZATION REPORT NUMBER(S) R&D 5679-EN-01		
6a. NAME OF PERFORMING ORGANIZATION Forschungsinstitut für Informationsverarbeitung/Mustererkennung		6b. OFFICE SYMBOL (If applicable) FGAN-FIM	7a. NAME OF MONITORING ORGANIZATION European Research Office, USARDSG-UK		
6c. ADDRESS (City, State, and ZIP Code) FGAN-FIM, D-7505 Ettlingen 6, Eisenstockstr. 12, West Germany			7b. ADDRESS (City, State, and ZIP Code) Box 65, FPO NY 09510-1500		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION EUR. R&D USARDSG-UK		8b. OFFICE SYMBOL (If applicable) AMXSN-UK-RE	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAJA45-86-C-0049		
8c. ADDRESS (City, State, and ZIP Code) Box 65, FPO NY 095101500			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO 61102A	PROJECT NO 1L1611- 02BH57	TASK NO. 01
			WORK UNIT ACCESSION NO		
11. TITLE (Include Security Classification) (U) Automatic Line Network Extraction from Aerial Imagery of Urban Areas through Knowledge Based image analysis					
12. PERSONAL AUTHOR(S) Prof. Dr. H. Kazmierczak					
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM Nov 86 TO Aug 89		14. DATE OF REPORT Aug 89	
				15. PAGE COUNT 152	
16. SUPPLEMENTARY NOTATION Image processing, structural pattern recognition, blackboard oriented symbolic processing, knowledge based image analysis, image understanding, aerial imagery, urban areas					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
0805	Geodesy				
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Different methods for automatic detection of line objects applied to aerial images to extract streets from urban scenes are investigated. First, test results achieved from two existing methods of low level iconic image processing by stream following (line tracking) and structured parallel operations (image filtering, feature extraction) are given. Second, a medium level iconic image processing method developed for edge and area segmentation is de- scribed and results from image segmentation are presented symbolically. Then two preliminary approaches of high level symbolic processing by knowledge based blackboard oriented structure analysis are tested. One is originating with preprocessing by low level edge filtering, the other by medium level area segmentation. First results from the image understanding method for street network extraction are presented,					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL J. C. COMATI			22b. TELEPHONE (Include Area Code) 01-402 7331		22c. OFFICE SYMBOL AMXSN-UK-RE

DD FORM 1473, 81 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Contents

Report Documentation Page with Abstract	i
Contents	ii
List of Figures	iv
Summary	vii
1 Introduction	1
1.1 Problems	1
1.2 New Approach	2
1.3 Test Imagery	4
1.4 Implementations	5
1.5 Investigations	9
2 Extraction of Line Objects by Stream Following	11
2.1 The Line Extraction Method	11
2.2 Line Extraction from Urban Scenes	12
2.2.1 Test Imagery	12
2.2.2 Parameter Adjustment	14
2.2.3 Results of Line Extraction	14
3 Extraction of Line Objects by Structured Parallel Operations	34
3.1 Structured Image Operations	34
3.2 Method and Results of Line Extraction	35
4 Structure Oriented Image Analysis with Blackboard System	41
4.1 Situation Driven and Model Controlled Image Analysis	42
4.1.1 Image Preprocessing	42
4.1.2 Symbolic Image Analysis	42
4.2 The Blackboard Production System	47
4.2.1 Blackboard	47
4.2.2 Production Cycle	47
4.2.3 General Assessment and Context Spaces	49
4.2.4 Final Assessment and Stop Criterion	51
4.3 Edge Oriented Detection of Intersections	51

4.3.1	Parallel Rewrite System	51
4.3.2	Set of Productions	52
4.3.3	Results from Detection of Intersections	57
5	Iconic Image Filtering and Segmentation	76
5.1	Gray Level Pyramid	76
5.1.1	Generation of the Gray Level Pyramid	77
5.1.2	Projection of an Image Point	80
5.2	Contrast Pyramids	80
5.2.1	Detection of Edge Elements	80
5.2.2	Detection of Spots	81
5.3	Edge Oriented Segmentation	84
5.3.1	Starting Point List for Edge Segmentation	87
5.3.2	Tracking of Edge Elements and Feature Extraction	87
5.3.3	Assessment of Edge Elements and Segments	89
5.3.4	Approximation of Edge Segments by Polygons	91
5.3.5	Symbolic Description of Edge Segments	93
5.4	Area Oriented Segmentation	93
5.4.1	Starting Point List for Area Segments	95
5.4.2	Spot Projection into Levels of Higher Resolution	95
5.4.3	Area Segmentation by Binarization	97
5.4.4	Tracking of Contour Elements and Feature Extraction	98
5.4.5	Assessment of Area Segments	98
5.4.6	Approximation of Area Segments by Polygons	100
5.4.7	Symbolic Description of Area Segments	107
5.5	Results from Edge and Area Segmentation	107
6	Extended Production Model for Street Network Extraction	109
6.1	Area Oriented Detection of Intersections	111
6.2	Extended Set of Productions	112
6.3	Results from Area Oriented Extraction of Street Networks	115
6.3.1	PHOENIX_2A1	117
6.3.2	PHOENIX_2B1	125
6.3.3	BIETIGHEIM1_31	125
6.3.4	BIETIGHEIM2_31	125
7	Conclusions	147
	Acknowledgement	150
	Bibliography	151

List of Figures

1.1	Test image PHOENIX	6
1.2	Test image BIETIGHEIM1	7
1.3	Test image BIETIGHEIM2	8
2.1	Line extraction from PHOENIX_1A/11/2/4/2/10/20/0	18
2.2	Line extraction from PHOENIX_1A/22/2/4/2/10/20/0	19
2.3	Line extraction from PHOENIX_2A/11/2/4/2/10/20/0	20
2.4	Line extraction from PHOENIX_2A/22/2/4/2/10/20/0	21
2.5	Line extraction from BIETIGHEIM1.1/11/2/4/2/10/20/0	22
2.6	Line extraction from BIETIGHEIM1.1/11/2/4/2/10/20/20	23
2.7	Line extraction from BIETIGHEIM1.2/11/2/4/2/10/20/0	24
2.8	Line extraction from BIETIGHEIM1.3/11/1/5/3/5/10/0	25
2.9	Line extraction from BIETIGHEIM2.1/11/2/4/2/10/20/0	26
2.10	Line extraction from BIETIGHEIM2.1/11/2/4/2/5/20/0	27
2.11	Line extraction from BIETIGHEIM2.1/11/4/4/2/10/20/0	28
2.12	Line extraction from BIETIGHEIM2.2/11/4/5/3/10/20/0	29
2.13	Line extraction from BIETIGHEIM2.3/11/1/5/3/10/20/0	30
2.14	Line extraction from PHOENIX_1B/22/2/4/2/10/20/0	31
2.15	Line extraction from PHOENIX_2B/22/2/4/2/10/20/0	32
2.16	Superimposed dark line extraction from PHOENIX_2B (I)	33
3.1	Image processing with structured parallel operations	35
3.2	Estimates of sequential and parallel image processing	36
3.3	Superimposed dark line extraction from PHOENIX_2B (II)	38
3.4	Superimposed dark line extraction from PHOENIX_2B (III)	39
3.5	Superimposed bright line extraction from PHOENIX_2B (IV)	40
4.1	Preprocessing	43
4.2	Derivation tree: <i>bridge</i>	44
4.3	Structure analysis (I)	45
4.4	Structure analysis (II)	46
4.5	Blackboard system	48
4.6	Principle of a production cycle	49
4.7	Example of context space	50
4.8	Production net	53
4.9	Principle of line prolongation	54

LIST OF FIGURES

v

4.10	Test image PHOENIX_2A1	60
4.11	Preprocessing of PHOENIX_2A1/base objects	61
4.12	Analysis of PHOENIX_2A1/triggering base objects	62
4.13	Analysis of PHOENIX_2A1/extracted objects	63
4.14	Test image PHOENIX_2B1	64
4.15	Preprocessing of PHOENIX_2B1/base objects	65
4.16	Analysis of PHOENIX_2B1/triggering base objects	66
4.17	Analysis of PHOENIX_2B1/extracted objects	67
4.18	Test image BIETIGHEIM1_31	68
4.19	Preprocessing of BIETIGHEIM1_31/base objects	69
4.20	Analysis of BIETIGHEIM1_31/triggering base objects	70
4.21	Analysis of BIETIGHEIM1_31/extracted objects	71
4.22	Test image BIETIGHEIM2_31	72
4.23	Preprocessing of BIETIGHEIM2_31/base objects	73
4.24	Analysis of BIETIGHEIM2_31/triggering base objects	74
4.25	Analysis of BIETIGHEIM2_31/extracted objects	75
5.1	Pyramidal image structure	78
5.2	Generation of a pyramidal image point	78
5.3	Gray level pyramid of PHOENIX_2A1	79
5.4	Absolute value of gradients of PHOENIX_2A1	82
5.5	Direction of gradients of PHOENIX_2A1	83
5.6	Spot curvature pyramid of PHOENIX_2A1	85
5.7	Spot surroundedness pyramid of PHOENIX_2A1	86
5.8	Coding of direction	88
5.9	Tracking of edge elements	88
5.10	Ranking of successors	90
5.11	Ranking of predecessor	90
5.12	Edge oriented segmentation of PHOENIX_2A1	92
5.13	Symbolic description of edge segments	94
5.14	Projection of spots into levels of higher resolution	96
5.15	Contour tracking	99
5.16	Area oriented segmentation of PHOENIX_2A1 (bright areas I) . .	101
5.17	Area oriented segmentation of PHOENIX_2A1 (bright areas II) . .	102
5.18	Area oriented segmentation of PHOENIX_2A1 (dark areas I) . .	103
5.19	Area oriented segmentation of PHOENIX_2A1 (dark areas II) . .	104
5.20	Area oriented segmentation of PHOENIX_2A1 (bright areas III) .	105
5.21	Area oriented segmentation of PHOENIX_2A1 (dark areas III) . .	106
5.22	Symbolic description of area segments	108
6.1	Edge oriented analysis of 256 × 256 section of PHOENIX_2A1 . .	110
6.2	Area oriented production net	111
6.3	Preprocessing of PHOENIX_2A1/small areas	118
6.4	Analysis of PHOENIX_2A1/big areas	119
6.5	Analysis of PHOENIX_2A1/long areas	120

6.6	Analysis of PHOENIX_2A1/streets	121
6.7	Analysis of PHOENIX_2A1/intersections	122
6.8	Analysis of PHOENIX_2A1/street networks	123
6.9	Analysis of PHOENIX_2A1/extraction results	124
6.10	Preprocessing of PHOENIX_2B1/small areas	126
6.11	Analysis of PHOENIX_2B1/big areas	127
6.12	Analysis of PHOENIX_2B1/long areas	128
6.13	Analysis of PHOENIX_2B1/streets	129
6.14	Analysis of PHOENIX_2B1/intersections	130
6.15	Analysis of PHOENIX_2B1/street networks	131
6.16	Analysis of PHOENIX_2B1/extraction results	132
6.17	Preprocessing of BIETIGHEIM1_31/small areas	133
6.18	Analysis of BIETIGHEIM1_31/big areas	134
6.19	Analysis of BIETIGHEIM1_31/long areas	135
6.20	Analysis of BIETIGHEIM1_31/streets	136
6.21	Analysis of BIETIGHEIM1_31/intersections	137
6.22	Analysis of BIETIGHEIM1_31/street networks	138
6.23	Analysis of BIETIGHEIM1_31/extraction results	139
6.24	Preprocessing of BIETIGHEIM2_31/small areas	140
6.25	Analysis of BIETIGHEIM2_31/big areas	141
6.26	Analysis of BIETIGHEIM2_31/long areas	142
6.27	Analysis of BIETIGHEIM2_31/streets	143
6.28	Analysis of BIETIGHEIM2_31/intersections	144
6.29	Analysis of BIETIGHEIM2_31/street networks	145
6.30	Analysis of BIETIGHEIM2_31/extraction results	146

Summary

The work can be summarized as follows:

- Application of a sequential iconic image processing method using stream following techniques and a parallel iconic image processing method using structured parallel operations (SIO), previously developed for *rural* scenes, to *urban* areas demonstrating the need for more powerful image analyzing methods.
- Application of an existing blackboard oriented image analysis (BPI) method to *urban* areas to extract intersections with streets composed of line primitives generated from existing preprocessing.
- Development of a method for segmenting an image into edge and area elements as primitives in order to improve the preprocessing for blackboard oriented analysis.
- Development of a preliminary model for the extraction of the traffic network of streets and intersections within *urban* areas by blackboard oriented image analysis (BPI) and development of an interface accepting area segments as primitives received from the improved preprocessing.
- Implementation of methods in FORTRAN and PASCAL programming languages in a VMS operating system for DEC VAX computers.
- Documentation of test results from processing suburban imagery of Phoenix, Arizona (USA) and Bietigheim, Southern Germany (FRG) projected with a scale of 1:4,000 to 1:56,000 and digitized with an aperture of $400\mu m$ to $50\mu m$ representing a pixel size of approximately $2.8m$ to $0.4m$ on earth.

Chapter 1

Introduction

Several different methods for the automatic extraction of line objects from aerial imagery of *rural* scenes, based on differing principles, had been developed and implemented at FIM partly in charge of the U.S. Government [2,8,12]. The methods perform the extraction task *conservatively*, e.g. line segments which are distorted by a considerable amount of noise or occlusion, are not accepted. Thus the extraction results, e.g. the polygons representing the network of line objects, tend to be correct, reliable, but not necessarily complete.

The methods [2] had been extended to perform the extraction task in two images of a stereo pair [8]. The extraction results of this process have a higher degree of correctness and reliability, because they are produced by coincident extraction from two images of the same area. Procedures for automatic extraction of line shaped objects from aerial images had been improved and completed by artificial intelligence aspects [12]. A general model of road network had been used to complete road extraction from images. Digital elevation data had been used to guide the process of river and creek extraction from images.

1.1 Problems

The basic idea for the automatic extraction of line objects is not to process and analyze the complete image matrix systematically by applying a variety of special operations. The resulting multitude of preliminary results would have to undergo complicated compatibility tests which will - in most cases - be anything else but trivial. On the contrary, the process is focussed only on promising sections of the image matrix by applying locally selective hypothesis and verification techniques.

Concerning *rural* scenes the solution of the problem had been attacked in two different functional steps [2]:

- To initialize any automatic image analysis process, some sort of a systematic *screening* of the image matrix is necessary. As roads and streets are to be extracted the initial screening of the image matrix consists of a

search for line shaped segments which serve as starting segments or cues for an extraction process (parallel approach).

- It is supposed that the search for starting segments of lines yields results of high confidence. This is the case, if the area covered by the image is *rural*, or extends from *urban* regions into neighboring *rural* country or at some locations in *urbanized* regions where undistorted segments of roads are clearly detectable. Beginning at those starting segments, the line extraction process is applied, which consists of an alternating locally or regionally adaptive gray level analysis by stream following techniques (sequential approach).

Transparencies at a scale of approximately 1:70,000 were chosen as test imagery digitized by a scanning apparatus of $25\mu m$ corresponding to a circular spot size of 1.75m diameter on earth. Thus the line extraction procedure had been adapted to a line width of about 3 to 8 pixels only corresponding with the width of roads at that scale. Later on a new problem solution of the second step of that line extraction method had been investigated and implemented by a parallel approach [3,9]. The parallel approach avoids the first functional step and shows nearly equal performance as the sequential approach. The extraction of line objects is accomplished by iconic structured parallel operations (SIO). In addition, the execution time of different parallel processing schemes had been estimated.

When applied to imagery of *rural* scenes, the performance of such an automatic line extraction method can be considered as successful. However, when applied to imagery of *urban* areas, the success and reliability of that method decrease remarkably due to the complex mixture of natural and man made objects at these locations. Because of the complex gray level situation in *urban* areas the locally or regionally restricted gray level analysis, using cues of arterial roads at the boundaries from rural country to urban areas, e.g., will soon fail to yield reliable detection results of roads and streets.

The improvement of the signal processing algorithms without incorporation of semantics does not produce evident better results. To enhance the performance of the extraction procedure, highly sophisticated knowledge based image analysis is necessary to allow progress in this research field.

1.2 New Approach

This research project has concentrated on the detection and extraction of roads and streets in the context of surrounding objects in images of *urban* areas. The work was started applying existing image processing methods for segmentation and feature extraction and an existing knowledge based method for structure oriented analysis of image segments. The investigations proceeded by improving the preprocessing and adapting the structural analysis. Hence, segmentation procedures for edge like and area like objects were developed and a first

simplified model for the dominant structures of an urbanized area and a control strategy for the whole procedure were considered using a blackboard oriented approach [5,6,9,10,11]. Thereby, only area like objects has been used for structural analysis so far.

To solve at least part of automatic line network extraction problem from imagery of *urban* areas the application of a compound of higher sophisticated processing methods is necessary. The basic idea of the new approach is as follows:

- First, the gray level signal analysis has to be extended from the restricted locality sufficient for the sequential stream following technique to a larger area, typically 512×512 pixels as for the parallel approach. Within this area the gray level signal analysis is extended to detect any elementary segment, e.g. line segments, small blobs, small areas, etc. Thereby a distinct initialization step as used for the stream following technique is avoided.
- Next, subsets of these sets of elementary segments have to be selected on the basis of similarity of properties (features), e.g. line segments of similar length, orientation and distance. It has been proved that among those subsets of elementary segments (primitives) the dominant structures of objects can be found.
- Last, a comparison to a detailed model of the relevant structures of an urbanized area allows the classification of the structures found in the images and, hence, is able to distinguish roads and streets from other dominant objects shown in the image. The central problem consists in the development of a suitable model of an urbanized area. The model is organized in the form of a blackboard based syntactical production system for image analysis (BPI) [11]. The contents of the model is taken from some test images. Attention has been given to the problem of a control strategy for a suitable cooperation between the pool of (active) image analyzing methods and the (passive) model data.

Thus, feature and context information of image segmentation received from preprocessing have to be converted from raster presentation of vectors to blackboard presentation of basic elements (primitives or base objects) for structural analysis. Attributes are associated with each element organized in sets and stored in an associative memory. The structural analysis or so-called symbolic processing is carried out successively on symbols (elements) starting with basic ones (primitives) received from preprocessing.

After having generated new symbols according to a production net, symbolic processing proceeds with deduced symbols (deduced elements or partial objects). For effective processing of interrelations, the symbol (element or object) space with its attributes can be matched against different context spaces containing the same attributes. From the context spaces expectation criteria

are derived. In addition, a quality measure is generated for basic symbols from preprocessing and for each deduced symbol from symbolic processing. The expectation criteria and the quality measures representing a confidence evaluation (symbol or object assessment) are assigned to the corresponding symbols (elements or objects).

The process of symbolic processing (structural analysis) is accomplished situation driven and model controlled. This means that the next step of symbolic processing is chosen according to the highest evaluation of a situation compatible with the model. The process (recognition) is terminated when all hypotheses are tested. The image description can be achieved by analyzing the nodes of the production net from bottom to top. The top nodes reached, e.g. represent detected objects. The blackboard based syntactical production system for symbolic processing (BPI) performs processing of noisy symbolic image data (elements or objects), deduces symbolic data from vaguely declared knowledge, assigns confidence to symbolic data, and controls execution by opportunistic scheduling from symbolic data situations.

The BPI model consists of a blackboard for entering symbolic image data (elements or objects), a number of modules for symbolic processing (structural analysis) representing knowledge of image understanding and symbolic processing based on a syntactical production system, and a control module. The control module performs recognition of the best data situation from symbol (element or object) space, assessment and assignment of appropriate hypotheses for symbolic processing (situation driven and model controlled), and selection of the processing modules concerned with testing the assigned hypotheses (structural analysis).

The advantage of BPI as compared with other concepts is its structural flexibility with respect to the extension of symbol (element or object) space, modularity of processing (structural analysis), and disposition of knowledge, as well as its capability of retaining all intermediate steps of symbolic processing and avoiding conclusive decisions.

1.3 Test Imagery

It was decided to select test imagery of two levels of complexity for the investigations carried out during this research project:

- Test image of Phoenix, Arizona (USA), where housing areas follow simple rules and, hence, display very clear structures (low level of complexity), and
- Test images of Bietigheim, Southern Germany (FRG), where the urban structures are more complicated (medium level of complexity).

The test images have been digitized using a $50\mu m$ or $100\mu m$ raster, which results in a pixel size of approximately $0.4m$ to $1.0m$ on earth. The chosen

original image matrices have a size of 4096×4096 respectively 2048×2048 pixels. In some cases a signal processing was necessary in order to correct scan errors as in scenes from Bietigheim.

In order to test the line extraction methods the following test images were used:

- PHOENIX (black white positive, see Figure 1.1)
 - 4096 \times 4096 pixels
 - 1 : 20,000 scale
 - 50 μm pixel size (on image)
 - 1 m pixel size (on earth)
- BIETIGHEIM1 (black white positive, see Figure 1.2)
 - 4096 \times 4096 pixels
 - 1 : 14,000 scale
 - 50 μm pixel size (on image)
 - 0.7 m pixel size (on earth)
- BIETIGHEIM2 (black white positive, see Figure 1.3)
 - 2048 \times 2048 pixels
 - 1 : 4,000 scale
 - 100 μm pixel size (on image)
 - 0.4 m pixel size (on earth)

As image analysis requires access to different resolution levels of a test image because of the variable size of the segments contained in every image, multiresolution images were generated by subsequent averaging of the gray values of four adjacent pixels. This results in a series of image matrices (image pyramids) with a resolution (or size) decreasing by a factor of 2 from image to image.

1.4 Implementations

All methods have been implemented by experimental software for the purpose of analyzing images, assessing performance and limitations of algorithms, and documenting results. FORTRAN and PASCAL programming languages in a VMS operating system for DEC VAX computers have been used. Software for image segmentation for edge and area like objects has been developed as follows:

- iconic image filtering (preprocessing):

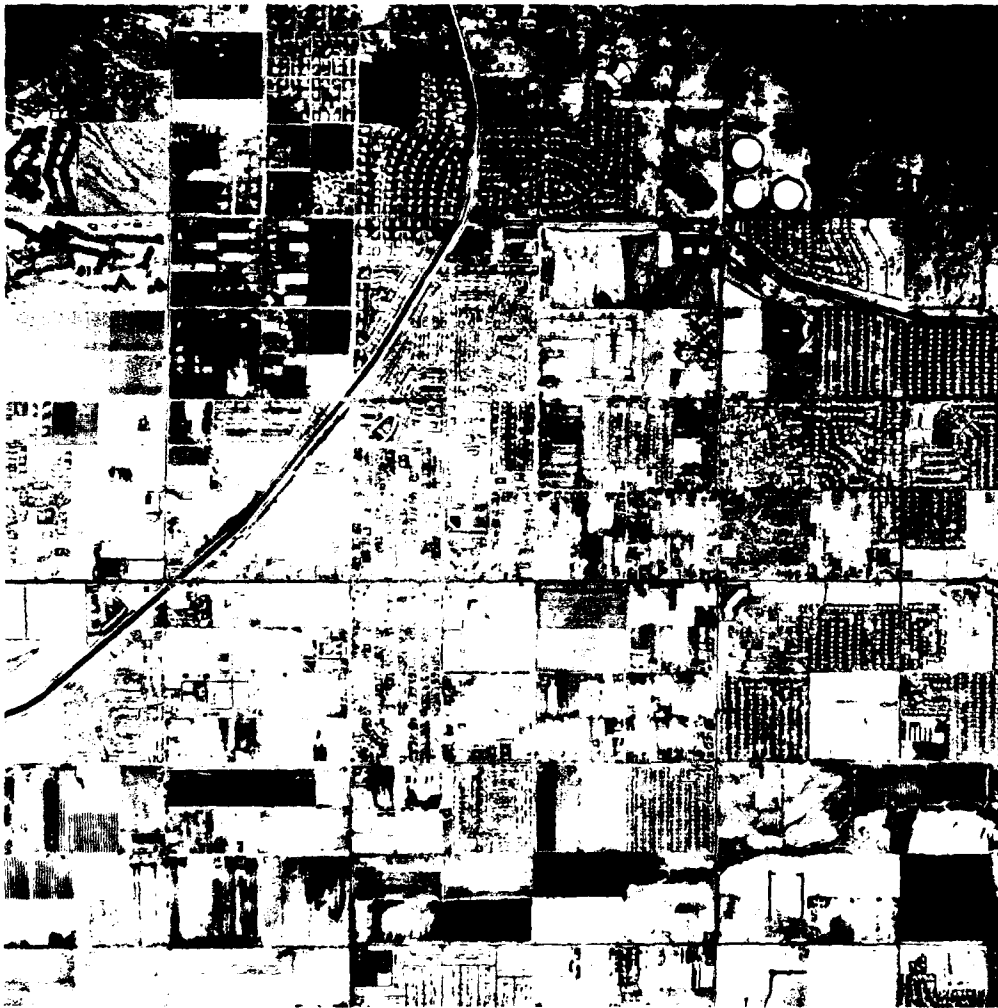


Figure 1.1: Test image PHOENIX
(original matrix size 4096×4096 pixels, pixel size and scale $50\mu m : 1m$)



Figure 1.2: Test image BIETIGHEIM1
(original matrix size 4096×4096 pixels, pixel size and scale $50\mu\text{m} : 0.7\text{m}$)

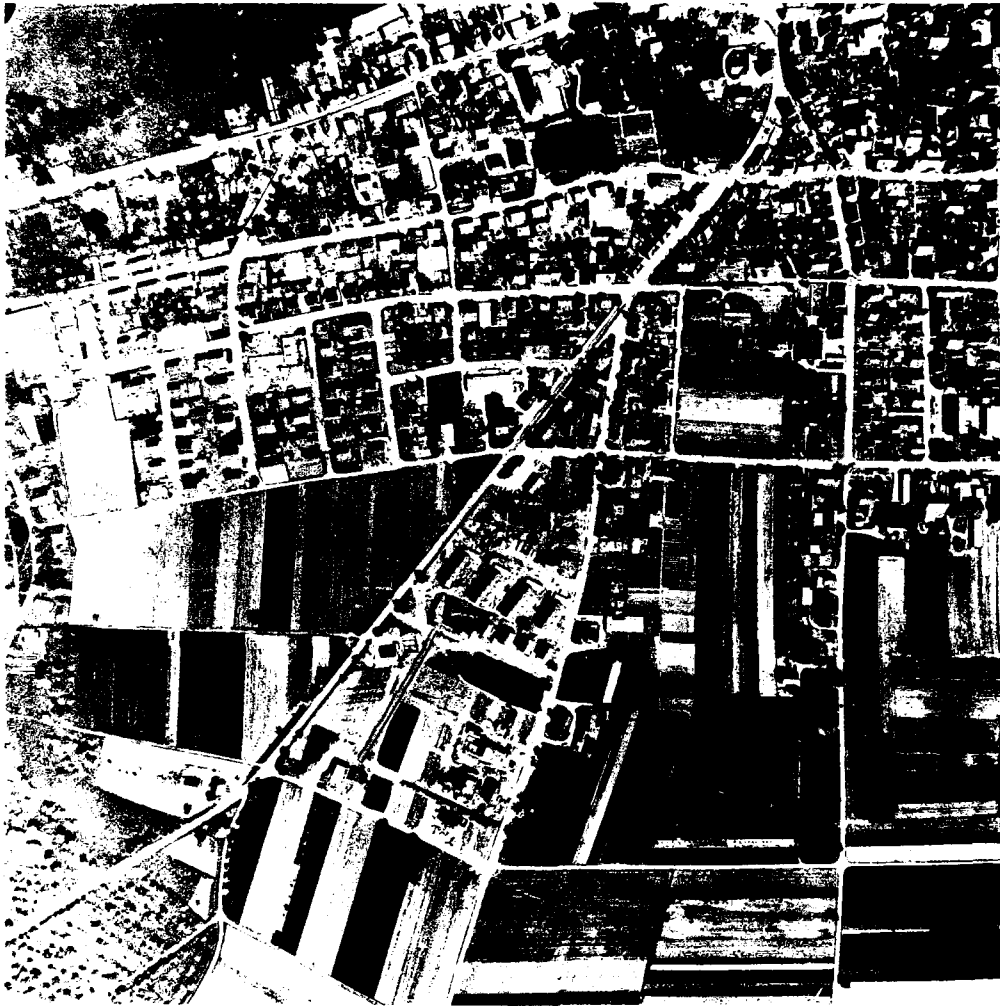


Figure 1.3: Test image BIETIGHEIM2
(original matrix size 2048×2048 pixels, pixel size and scale $100\mu m : 0.4m$)

- generation of a gray level multiresolution image pyramid
- application of a gradient operator to detect edge elements
- application of a spot operator to cue dark and bright contrasting areas in pyramidal levels of different resolution
- iconic image segmentation (preprocessing):
 - tracking of edge elements and combination to edge like objects
 - thresholding of dark and bright contrasting areas and determination of area like objects by contour following
 - feature extraction and output of symbolic description of edge like objects
 - feature extraction and output of symbolic description of area like objects

The following BPI software was applied to symbolic structure oriented image analysis:

- BPI precompiler converting BPI source code of knowledge sources to standard PASCAL code, calling BPI runtime system
- BPI runtime system
 - queue manager for priority controlled activation of knowledge sources
 - associative memory simulator for blackboard access by knowledge sources
 - support routines for handling of workstation interactions and files

That BPI software has not been developed under this contract.

1.5 Investigations

The investigations being conducted during the period of the contract are described in the following chapters of this report. The work can be summarized as follows:

- Application of a previously developed sequential iconic image processing method for *rural* scenes to *urban* areas demonstrating the need for more powerful image analyzing methods. That method could be used, however, for the cueing of streets by extracting arterial roads at the boundary from rural country to urban areas. Some extraction results and the problems arising from misapplication are demonstrated in chapter 2.

- Application of a previously developed parallel iconic image processing method for *rural* scenes to *urban* areas. After parameter adaption that line extraction method using structured parallel operations (SIO) could be used for the extraction of roads and streets from both rural country and urban areas. Some extraction results are demonstrated in chapter 3.
- Application of an existing blackboard oriented image analysis method to *urban* areas to extract intersections with streets composed of line primitives generated from existing preprocessing. That method could be used for the cueing of streets in the urban area for the post processing of streets. The principles of the method together with extraction results are explained in chapter 4.
- Development of a method for segmenting an image into edge and area elements as primitives in order to improve the preprocessing for blackboard oriented analysis. The method and the results obtained from preprocessing are described in chapter 5.
- Development of a preliminary model for the extraction of the traffic network of streets and intersections within *urban* areas by blackboard oriented image analysis and development of an interface accepting area segments as primitives received from the improved preprocessing. The used model and the results obtained with this new approach are demonstrated in chapter 6.

Concluding remarks about overall performance, remaining problems and requirements of further improvement are given in chapter 7.

Chapter 2

Extraction of Line Objects by Stream Following

A method for the automatic extraction of roads and rivers from large aerial images covering rural scenes had been developed and demonstrated successfully during previous research projects for USAETL in 1981, 1983 and 1986 [2,8,12]. When starting this project that method had been applied to urban scenes first in order to get a basis for comparison of the old method with the new approach.

2.1 The Line Extraction Method

The line extraction method using stream following techniques is to be explained briefly. The method consists of the three procedures:

- starting point search
- local extraction of lines
- regional extraction of lines

The procedure for starting point search analyzes sample lines or columns of the image to detect short segments of line objects, the features of which (width, contrast, straightness, etc.) correspond with predefined model segments.

The local extraction procedure works on a one dimensional, semicircular sample line taken from the image at the very neighborhood of the actual processing position; the gray values of the sample line are analyzed to detect a next cross section of the line object which agrees with the cross section of the actual processing position.

The regional extraction procedure works on an *area of interest* the location of which is predicted from the actual processing position. The gray values of that area are analyzed to detect several collinearly located cross sections of the line object which can be accepted as a continuing segment of this object.

The combined application of these three procedures, starting with the first one and alternating the second and third one, automatically extracts lines from aerial imagery. Some considerable sophistication had been integrated into the extraction methods to tolerate local distortions and noise. However, the extraction will stop in all cases, where the decision in favor of the line continuation is not reliable. At these locations, a variety of adjusting steps could be considered to eliminate extraction stops including but not confined to the following:

- changing the starting and processing locations, and
- changing the acceptance thresholds of contrast and shape features.

More details about the line extraction methods can be found in the quoted references [2,8,12].

2.2 Line Extraction from Urban Scenes

The procedures for the detection of starting points and for the subsequent extraction of lines from rural scenes were applied to selected 1024×1024 sections of the test images PHOENIX, BIETIGHEIM1, and BIETIGHEIM2 with sub-urban scenes consisting of mixed rural and urban areas.

2.2.1 Test Imagery

The following images were used for test purposes:

- PHOENIX-1A (1/4 section of PHOENIX)
 - 1024 \times 1024 pixels
 - 1 : 40,000 virtual scale
 - 100 μm pixel size (on image)
 - 2 m pixel size (on earth)
- PHOENIX-2A (1/16 section of PHOENIX)
 - 1024 \times 1024 pixels
 - 1 : 20,000 real scale
 - 50 μm pixel size (on image)
 - 1 m pixel size (on earth)
- BIETIGHEIM1.1 (1/1 image of BIETIGHEIM1)
 - 1024 \times 1024 pixels
 - 1 : 56,000 virtual scale
 - 200 μm pixel size (on image)

2.8 *m* pixel size (on earth)

- BIETIGHEIM1.2 (1/4 section of BIETIGHEIM1)

1024 × 1024 pixels

1 : 28,000 virtual scale

100 μ *m* pixel size (on image)

1.4 *m* pixel size (on earth)

- BIETIGHEIM1.3 (1/16 section of BIETIGHEIM1)

1024 × 1024 pixels

1 : 14,000 real scale

50 μ *m* pixel size (on image)

0.7 *m* pixel size (on earth)

- BIETIGHEIM2.1 (2/1 composition of BIETIGHEIM2)

1024 × 1024 virtual pixels (512 × 512 real pixels)

1 : 16,000 virtual scale

400 μ *m* pixel size (on image)

1.6 *m* pixel size (on earth)

- BIETIGHEIM2.2 (1/1 image of BIETIGHEIM2)

1024 × 1024 pixels

1 : 8,000 virtual scale

200 μ *m* pixel size (on image)

0.8 *m* pixel size (on earth)

- BIETIGHEIM2.3 (1/4 section of BIETIGHEIM2)

1024 × 1024 pixels

1 : 4,000 real scale

100 μ *m* pixel size (on image)

0.4 *m* pixel size (on earth)

- PHOENIX.1B (1/4 section of PHOENIX)

1024 × 1024 pixels

1 : 40,000 virtual scale

100 μ *m* pixel size (on image)

2 *m* pixel size (on earth)

- PHOENIX_2B (1/16 section of PHOENIX)

1024 × 1024 pixels
 1 : 20,000 real scale
 50 μm pixel size (on image)
 1 m pixel size (on earth)

2.2.2 Parameter Adjustment

The road extraction procedure can be adapted to a scene of an aerial image in accordance with scale and contrast by the following parameters p_1 to p_7 :

- p_1) kind of contrast (bright line objects coded by 11 or dark line objects coded by 22)
- p_2) minimum width of the line object (default: 2 pixels)
- p_3) increment for minimum width representing the maximum width of the line object (default: 4 pixels)
- p_4) increment for maximum width representing the width of crossings of roads (default: 2 pixels)
- p_5) gray value difference (default: 20), fixing the gray level threshold with respect to the gray value of the reference pixel characterizing the gray value contrast between line object and its surrounding. The line width must fall into the given variation width of minimum (p_2) and maximum line width ($p_2 + p_3$)
- p_6) absolute variation of gray value for the next standard sample line characterizing the constancy of the gray values within the line object (default: 20)
- p_7) and grid distance fixed to 40 pixels for the starting point search variable only by shifting the processing area (default: 0 pixels)

The parameters p_1 to p_7 are coded by their number of pixels or their gray value, respectively.

2.2.3 Results of Line Extraction

The results received from applying the line extraction method to the test imagery are demonstrated by Figure 2.1 to Figure 2.16. The extracted lines are overlaid on the input images in red color. The parameter settings for adjustment are assigned by appending their code words to the name of the test image as $name/p_1/p_2/ \dots /p_7/$. The test results in detail are:

- PHOENIX_1A/11/2/4/2/10/20/0 (see Figure 2.1)

Figure 2.1 shows a section of the test image PHOENIX covering a suburban area. Only few bright access roads and other line objects, e.g. along the water way, are found because of noise from the urban structure of the scene and misadjustment of parameters. Within the housing area no streets are extracted.

- PHOENIX_1A/22/2/4/2/10/20/0 (see Figure 2.2)

Figure 2.2 shows the same test image as in Figure 2.1, however, the line extraction method is adjusted to dark line objects. The line extraction results are equally poor as from Figure 2.1.

- PHOENIX_2A/11/2/4/2/10/20/0 (see Figure 2.3)

Figure 2.3 shows a smaller section of the test image PHOENIX than in Figure 2.1, whereby the line extraction method is adapted to smaller bright line objects. Here, lines are also extracted within the housing area, however, not the dark streets themselves but the bright side walks are found.

- PHOENIX_2A/22/2/4/2/10/20/0 (see Figure 2.4)

Figure 2.4 shows the same test image as in Figure 2.3, however, the line extraction method is adjusted to dark line objects. The line extraction result within the housing area is better than compared with that from Figure 2.2, however, street extraction is yet incomplete because of noise from the urban structure of the scene and missing flexibility of parameter adjustment.

- BIETIGHEIM1_1/11/2/4/2/10/20/0 (see Figure 2.5)

Figure 2.5 shows the test image BIETIGHEIM1 covering rural as well as suburban areas with bright roads and streets. Depending on parameter adjustment arterial roads can be extracted from the rural areas. Within the housing areas the streets are partly found.

- BIETIGHEIM1_1/11/2/4/2/10/20/20 (see Figure 2.6)

Figure 2.6 shows the same test image as in Figure 2.5, however, a shift of 20 pixels is accomplished with respect to the processing area with fixed search grid in order to see the influence of different starting points. The line extraction result is not identical compared with that from Figure 2.5 but the quality of road and street extraction is comparable.

- BIETIGHEIM1_2/11/2/4/2/10/20/0 (see Figure 2.7)

Figure 2.7 shows a section of the test image BIETIGHEIM1, whereby the line extraction method is adapted to smaller bright line objects than in Figure 2.5. Therefore smaller streets, roads, and other line objects are partly found.

- BIETIGHEIM1.3/11/1/5/3/5/10/0 (see Figure 2.8)

Figure 2.8 shows a smaller section of the test image BIETIGHEIM1 than in Figure 2.7, whereby the line extraction method is adapted to even more variable and smaller bright line objects than in Figure 2.5. Therefore only some borders of streets and other line objects are partly found.

- BIETIGHEIM2.1/11/2/4/2/10/20/0 (see Figure 2.9)

Figure 2.9 shows the test image BIETIGHEIM2 covering another scene with rural and suburban areas with bright roads and streets. If the parameters are adjusted well the arterial roads can be found in most cases. Starting with the arterial roads some streets in the urban area are found in the close prolongation of the arterial roads. However, they are lost when penetrating into the urban area.

- BIETIGHEIM2.1/11/2/4/2/5/20/0 (see Figure 2.10)

Figure 2.10 shows the same test image as in Figure 2.9. Because of lower adjustment of parameters, other and more line objects are produced.

- BIETIGHEIM2.1/11/4/4/2/10/20/0 (see Figure 2.11)

Figure 2.11 shows the same test image as in Figure 2.9. Because of altered adjustment of parameters, other line objects are favored.

- BIETIGHEIM2.2/11/4/5/3/10/20/0 (see Figure 2.12)

Figure 2.12 shows the test image BIETIGHEIM2, where the line extraction method is adapted to smaller bright line objects than in Figure 2.9. In order to recover the streets of Figure 2.9 the parameters p_1, p_2, p_3 have to be increased. Because of misadjustment of parameters fewer line objects are found than in Figure 2.9.

- BIETIGHEIM2.3/11/1/5/3/10/20/0 (see Figure 2.13)

Figure 2.13 shows a section of the test image BIETIGHEIM2, where the line extraction method is adapted to even smaller bright line objects as in Figure 2.9. Therefore only some borders of streets and other line objects are partly found. Because of misadjustment of parameters fewer line objects are found than in Figure 2.9.

- PHOENIX_1B/22/2/4/2/10/20/0 (see Figure 2.14)

Figure 2.14 shows another section of the test image PHOENIX covering a suburban area. Only few broad dark roads and other line objects are found because of noise from the urban structure of the scene and misadjustment of parameters. Within the housing area streets are hardly extracted.

- PHOENIX_2B/22/2/4/2/10/20/0 (see Figure 2.15)

Figure 2.15 shows a smaller section of the test image PHOENIX than in Figure 2.14, whereby the line extraction method is adapted to smaller dark line objects. Here, lines are also extracted within the housing area, however, street extraction is yet incomplete because of noise from the urban structure of the scene and missing flexibility of parameter adjustment.

In general, the results complied to what was expected: The reliability of the line extraction method developed for *rural* scenes decreases remarkably when applied to imagery of *urban* areas due to the complex mixture of natural and man made objects. Outside the housing areas most of the roads can be found. However, the series of tests show that the method of line extraction is not robust when applied to suburban scenes. The results are very sensitive against the parameter settings because shape and contrast of streets vary strongly.

Within housing areas the method fails, when performing the line extraction with fixed and not adapted parameter settings. However, when superimposing many results extracted with different parameter settings adapted to different resolution levels of an image and to different radiometric and geometric features of lines, then the line extraction result becomes acceptable as demonstrated by Figure 2.16.

- PHOENIX_2B/varying parameters (see Figure 2.16) with superposition of results from
 PHOENIX_1B/22/3/5/3/10/10/0
 PHOENIX_1B/22/3/5/3/10/10/10
 PHOENIX_2B/22/3/5/3/10/10/0
 PHOENIX_2B/11/3/5/3/10/10/0 (inverted gray values)
 PHOENIX_2B/22/4/4/2/10/20/0
 PHOENIX_2B/22/4/5/3/10/10/0
 PHOENIX_2B/22/4/4/2/10/10/0

Figure 2.16 shows a section of the test image PHOENIX as in Figure 2.15 with the line extraction method adapted to dark line objects. In case of superposition of the seven extraction results stated above, the dark lines within the housing area representing most of the streets are extracted nearly completely.

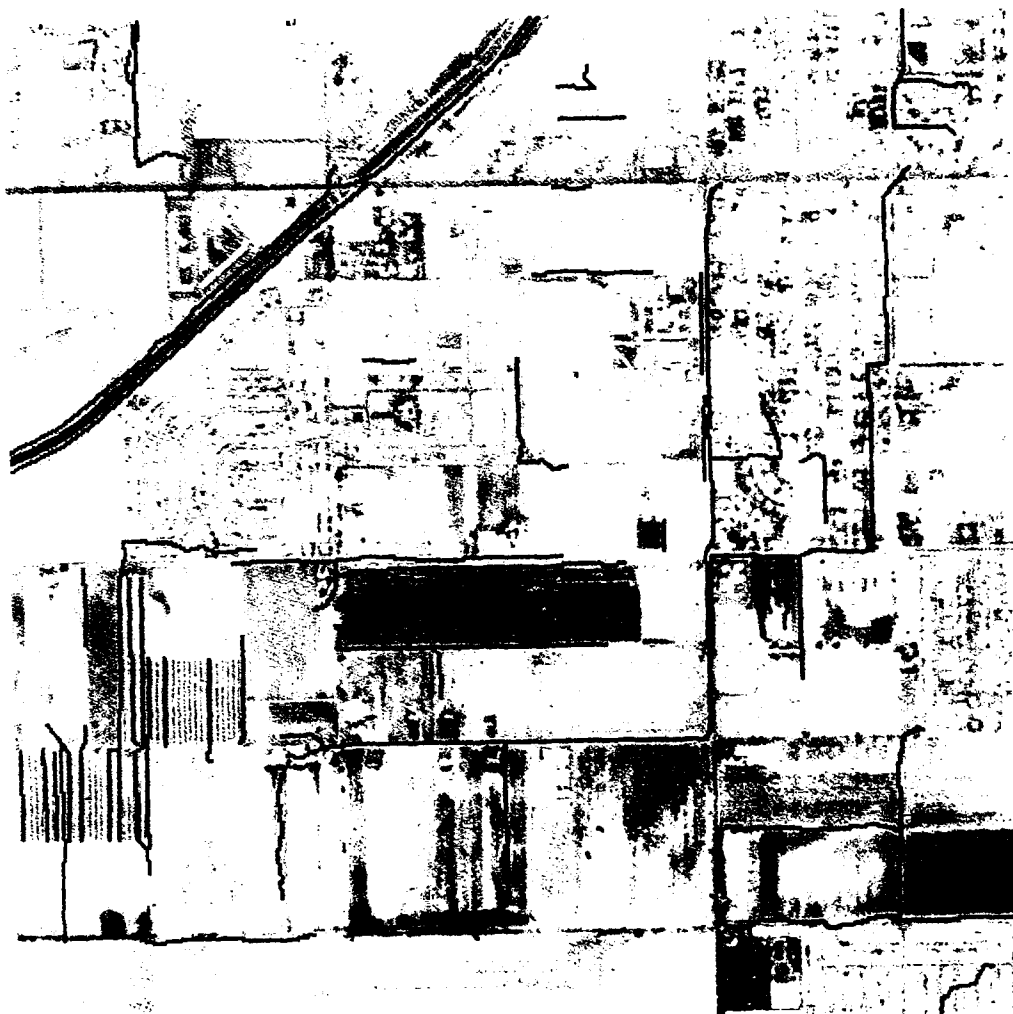


Figure 2.1: Line extraction from PHOENIX_1A/11/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $100\mu\text{m} : 2\text{m}$)

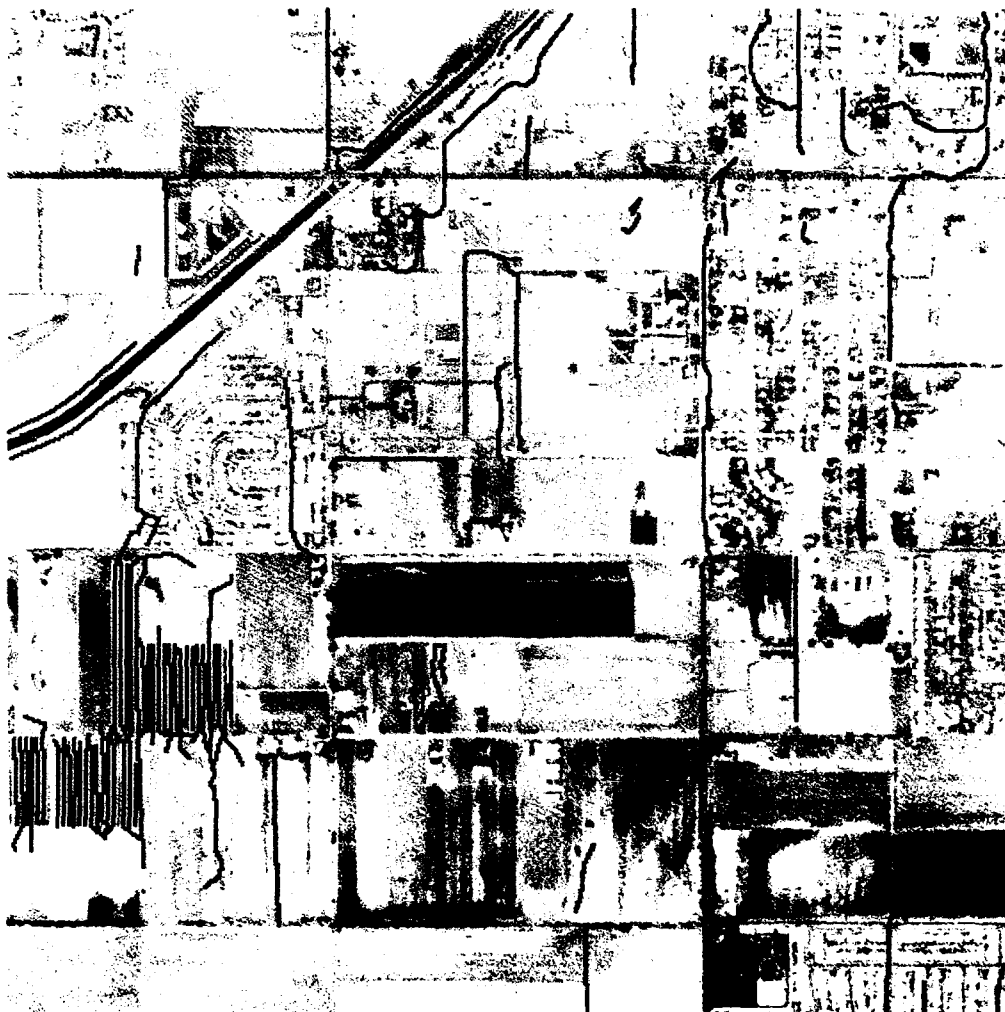


Figure 2.2: Line extraction from PHOENIX.1A/22/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $100\mu m : 2m$)

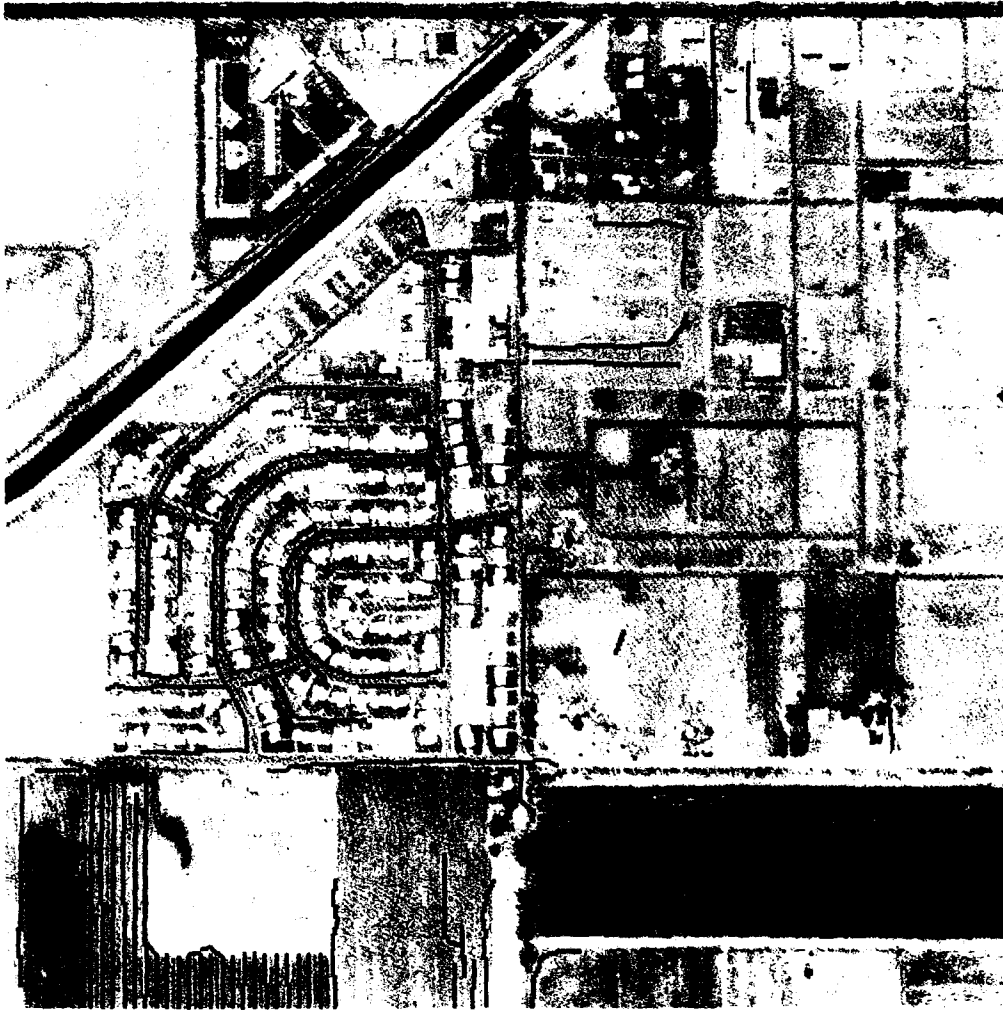


Figure 2.3: Line extraction from PHOENIX.2A/11/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 1m$)

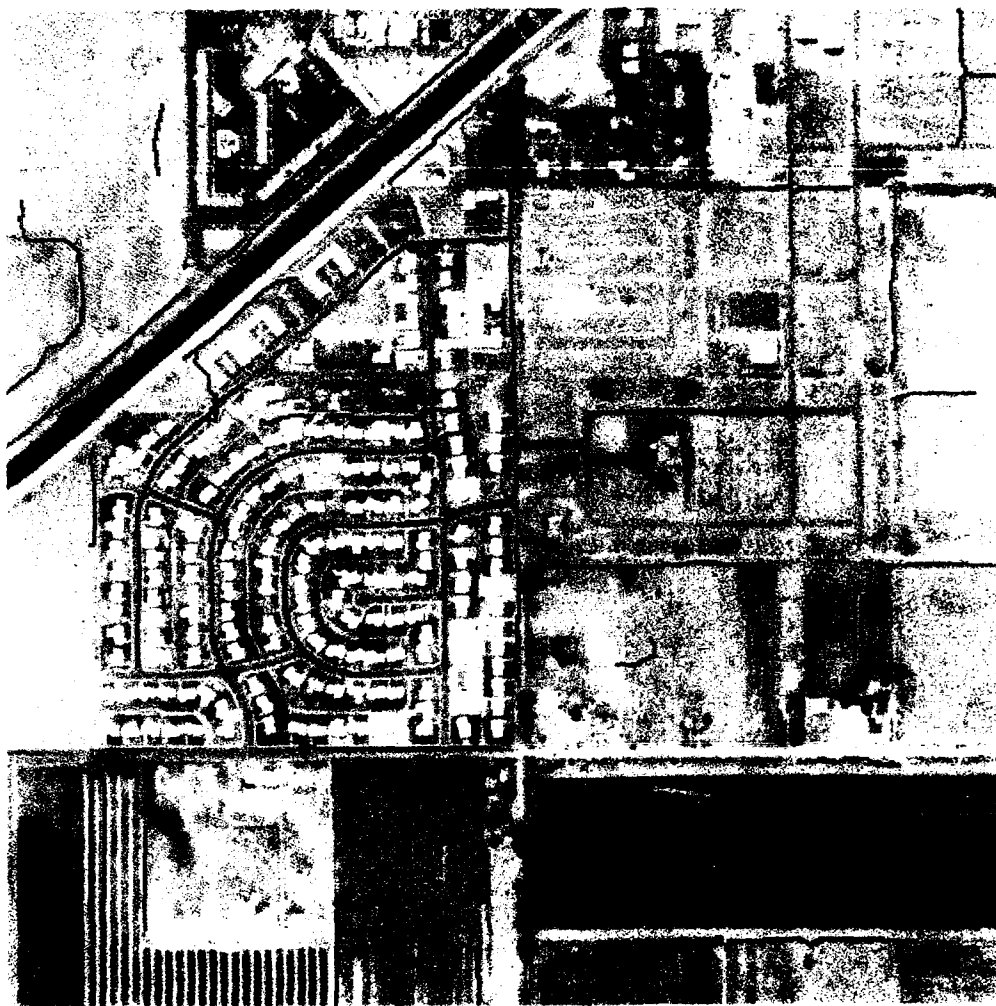


Figure 2.4: Line extraction from PHOENIX_2A/22/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 1m$)



Figure 2.5: Line extraction from BIETIGHEIM1.1/11/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $200\mu m : 2.8m$)



Figure 2.6: Line extraction from BIETIGHEIM1.1/11/2/4/2/10/20/20
(matrix size 1024×1024 pixels, pixel size and scale $200\mu m : 2.8m$)



Figure 2.7: Line extraction from BIETIGHEIM1_2/11/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $100\mu m : 1.4m$)

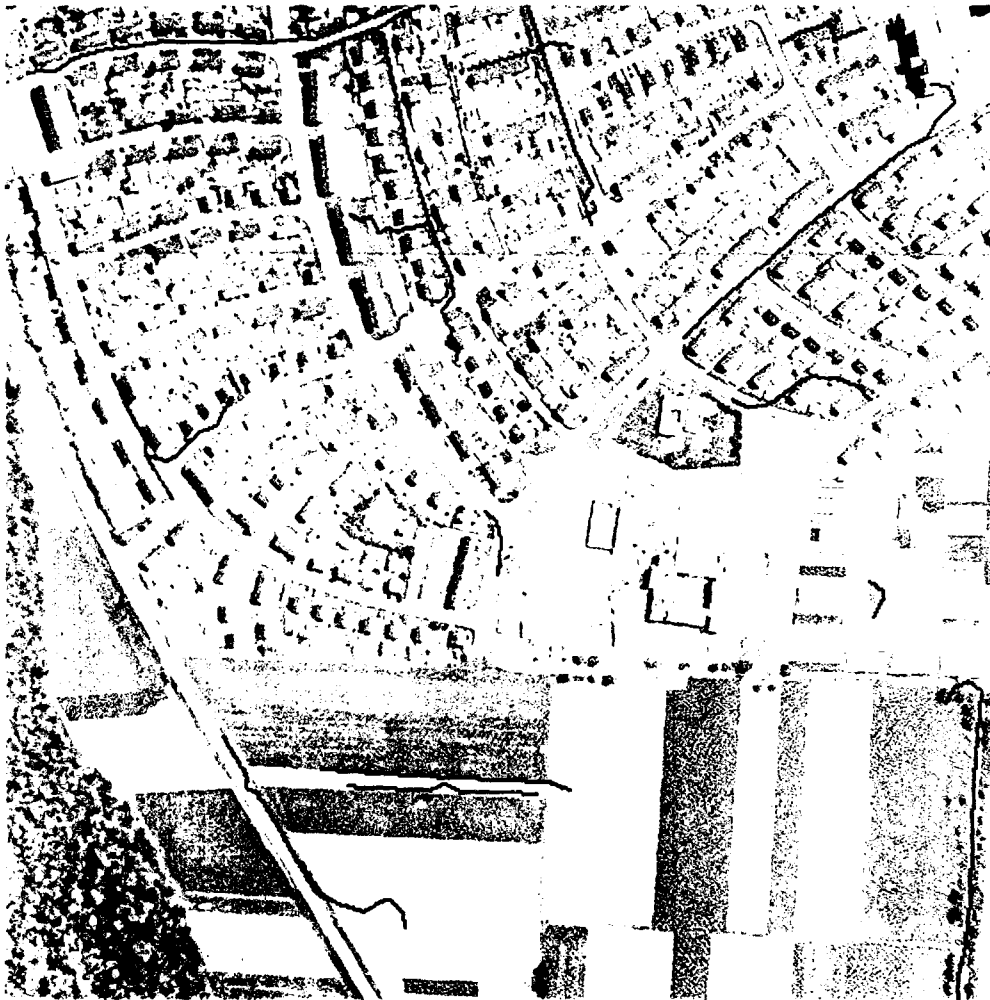


Figure 2.8: Line extraction from BIETIGHEIM1.3/11/1/5/3/5/10/0
(matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 0.7m$)



Figure 2.9: Line extraction from BIETIGHEIM2.1/11/2/4/2/10/20/0
(matrix size 512×512 pixels, pixel size and scale $400\mu m : 1.6m$)

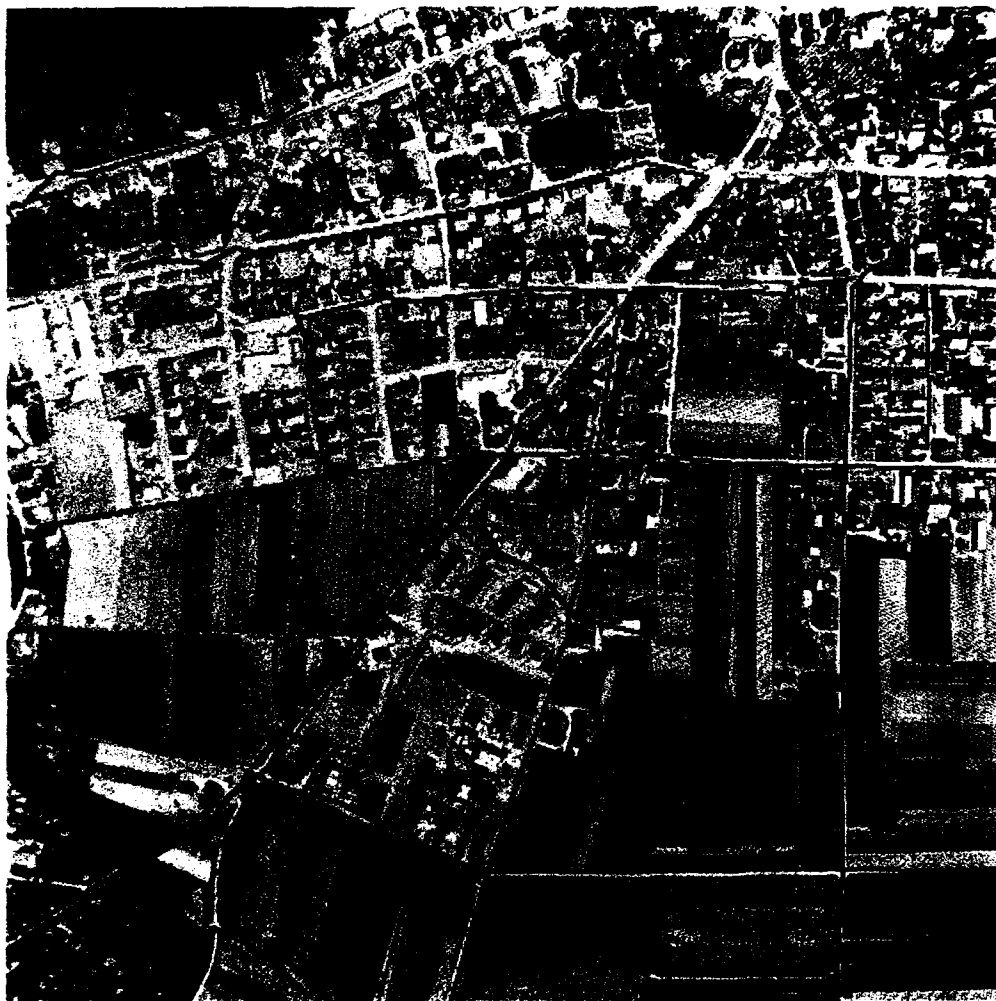


Figure 2.10: Line extraction from BIETIGHEIM2.1/11/2/4/2/5/20/0
(matrix size 512×512 pixels, pixel size and scale $400\mu m : 1.6m$)



Figure 2.11: Line extraction from BIETIGHEIM2.1/11/4/4/2/10/20/0
(matrix size 512×512 pixels, pixel size and scale $400\mu m : 1.6m$)



Figure 2.12: Line extraction from BIETIGHEIM2.2/11/4/5/3/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $200\mu m : 0.8m$)



Figure 2.13: Line extraction from BIETIGHEIM2.3/11/1/5/3/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $100\mu m : 0.4m$)

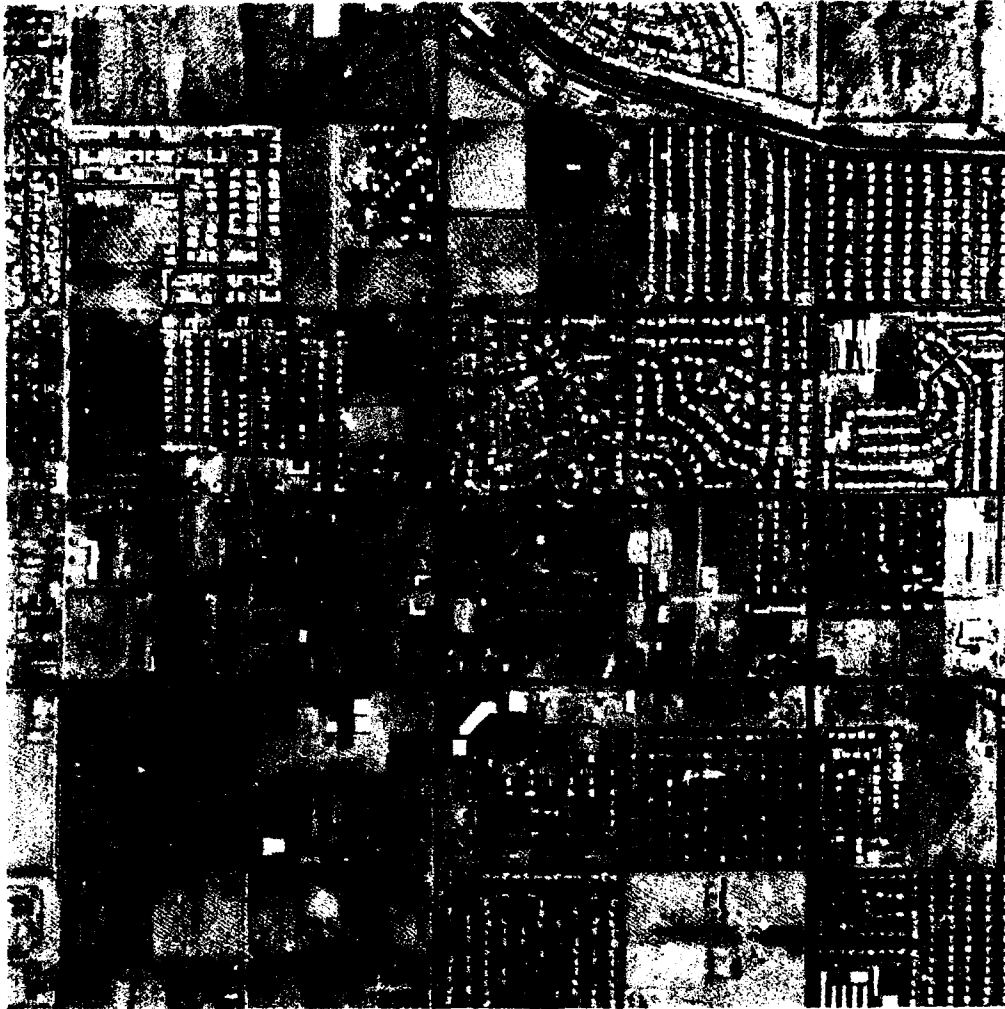


Figure 2.14: Line extraction from PHOENIX.1B/22/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $100\mu m : 2m$)

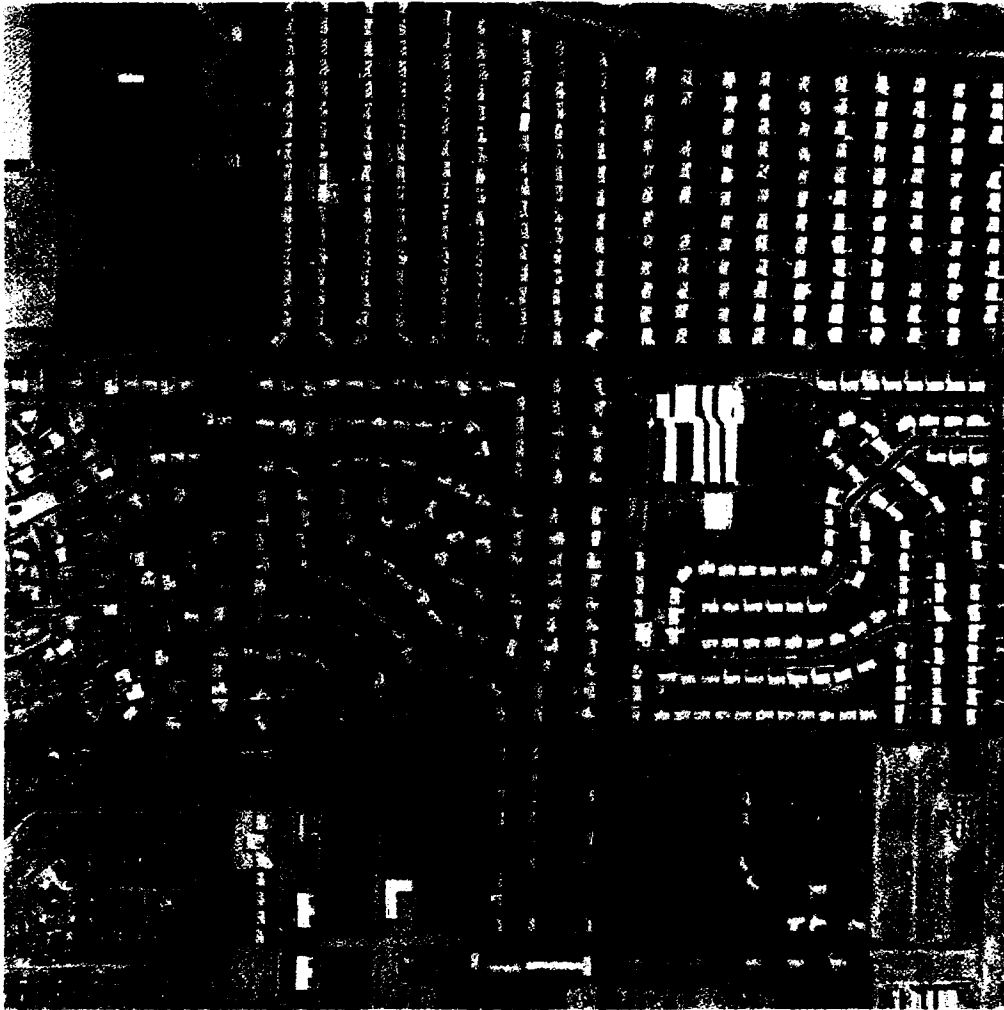


Figure 2.15: Line extraction from PHOENIX.2B/22/2/4/2/10/20/0
(matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 1m$)

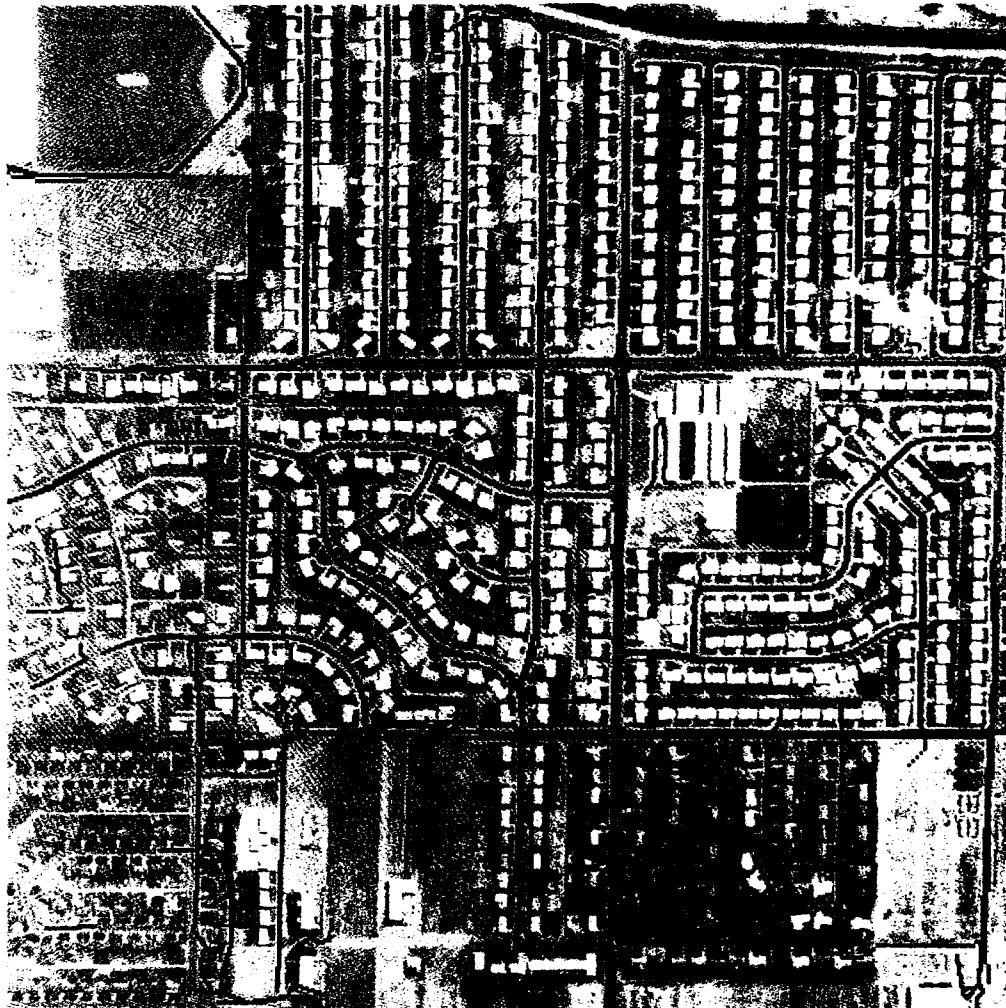


Figure 2.16: Superimposed dark line extraction from PHOENIX.2B/varying parameters (matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 1m$)

Chapter 3

Extraction of Line Objects by Structured Parallel Operations

In chapter 2. a sequential method for line extraction is described which had been developed prior to this project. Later on a new problem solution of that method had been investigated and implemented by a parallel approach. In [3] it is suggested to formulate algorithms for iconic processing by structured parallel operations SIO. Image analysis based on SIO can easily be transferred to every parallel processing architecture.

3.1 Structured Image Operations

The model for context dependent parallel processing with SIO is shown in Figure 3.1. A SIO is performed by applying a function f on image data i to produce output data o . The processing P of the function f is controlled by 4 descriptors for the specification of functional execution, operational area, parameteric adjustment, and operational context. Together with input and output a SIO is defined by 6 descriptors:

$$o := f_{SIO}(i)$$

$$\text{with SIO} = |DI|DF|DN|DP|DC|DO|$$

and descriptors	DI	input data list
	DF	function list/selection
	DN	neighborhood list/selection
	DP	parameter list/selection
	DC	context list/selection
	DO	output data list

If structured parallel processing is involved in methods, it is easy to transfer the algorithms and the software, e.g. developed on general purpose comput-

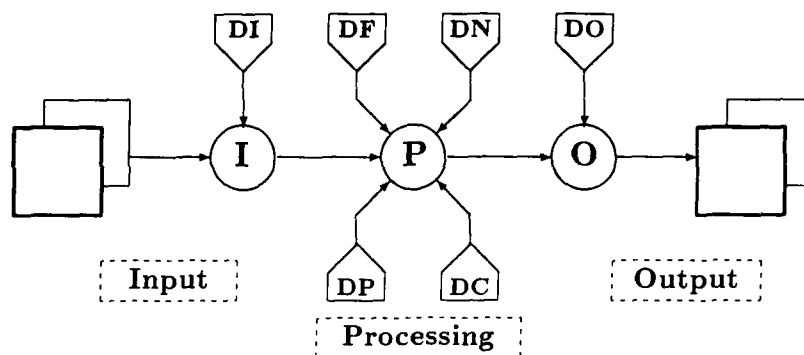


Figure 3.1: Image processing with structured parallel operations SIO ([3])

ers, to special parallel computer systems. As compared with computer simulation (VAX: sequential or parallel/SIO method) speed up factors of 5 orders of magnitude are within the scope (Figure 3.2), when using special systems as the flexible image processor FLIP (16 processor cascade see [4]), a multiple instruction (MI)/single data (SD) pipeline processor, a single instruction (SI)/multiple data (MD) cellular processor, or a multiple instruction (MI)/multiple data (MD) multiprocessor system.

3.2 Method and Results of Line Extraction

The SIO extraction of lines from rural scenes was applied to a selected section of the test image PHOENIX with suburban scenes consisting of mixed rural and urban areas. The following image was used for test purposes:

- PHOENIX_2B (1/16 section of PHOENIX)

1024 × 1024 pixels

1 : 20,000 real scale

50 μm pixel size (on image)

1 m pixel size (on earth)

In [3] the line extraction required for detecting and tracking roads in aerial photos had been performed by using a sequence of 16 SIOs with the following 9 functions: LOCAL ADAPTIVE THRESHOLD, EDGE DIRECTION, DOUBLE EDGE, LINE LENGTH, LINE ENDINGS, EDGE AGGLOMERATION, REDUCE LINE, ENLARGE LINE, CONNECT LINES. Without further adaption that method was used for line extraction from urban imagery. The line

	VAX sequential	VAX SIO	FLIP cascade	MI/SD pipeline	SI/MD 128 × 128	MI/MD 64 × 64
processing power	1 MIPS	1 MIPS	45 MIPS	50 MIPS	10 MAOPS (array operations)	10 MIPS (per PU)
processed data	2,500 (object points)	1 M bytes	1 M bytes	1 M bytes	8 M bytes	3.2 M bytes
total instructions or array operations	$75 \cdot 10^6$ (30 ms per object point)	$100 \cdot 10^6$	$75 \cdot 10^6$	$75 \cdot 10^6$	$0.4 \cdot 10^3$	$0.07 \cdot 10^6$
processing time	75 s	100 s	1.7 s	1.5 s	0.04 s	0.007 s
speed up factor	1	0.75	44	48	1,800	10,700

Figure 3.2: Estimates of sequential and parallel image processing [3]

extraction procedure can be adjusted to a scene of an aerial image by several parameters. The method is adjusted to the detection of bright lines only. In order to detect dark lines the gray values of the input image must be inverted prior to application of the method. Different parameter settings were applied using only the four parameters p_1, p_2, p_3, p_4 from several other parameters:

- p_1) kind of contrast (bright line objects coded by 11 or dark line objects via image inversion coded by 22)
- p_2) image scaling by low pass filtering of the image of given format (specification by number of pixels per edge of filtered image)
- p_3) number of pixels for the distance of the center of the search area for anti parallel edges
- p_4) number of pixels for the width of the search area for anti parallel edges

The results received from applying the SIO line extraction method to the test imagery are demonstrated by Figure 3.3 to Figure 3.5. The extracted lines are overlaid on the input image in red color. The parameter settings for adjustment are assigned by appending their code words to the name of the test image as $name/p_1/p_2/p_3/p_4$. The test results in detail are:

- PHOENIX_2B/varying parameters (see Figure 3.3) with superposition of results from
PHOENIX_2B/22/512/3/4
PHOENIX_2B/22/512/4/4

Figure 3.3 shows a section of the test image PHOENIX as in Figure 2.16 with the line extraction method adapted to dark line objects. In case of superposition of the two extraction results described above,

the dark lines within the housing area representing most of the streets are extracted nearly completely.

- PHOENIX_2B/varying parameters (see Figure 3.4) with superposition of results from
PHOENIX_2B/22/512/4/4
PHOENIX_2B/22/1024/5.5/5

Figure 3.4 shows a section of the test image PHOENIX as in Figure 3.3 with the line extraction method adapted to dark line objects with greater variation of line width. In case of superposition of the two extraction results described above, the dark lines within the housing area representing most of the streets are extracted more completely.

- PHOENIX_2B/varying parameters (see Figure 3.5) with superposition of results from
PHOENIX_2B/11/512/4.5/5
PHOENIX_2B/11/1024/3/4

Figure 3.5 shows a section of the test image PHOENIX as in Figure 2.16 with the line extraction method, however, adapted to bright line objects. In case of superposition of the two extraction results described above, the bright lines within the housing area representing border lines of the streets, ways, and other line objects are extracted nearly completely.

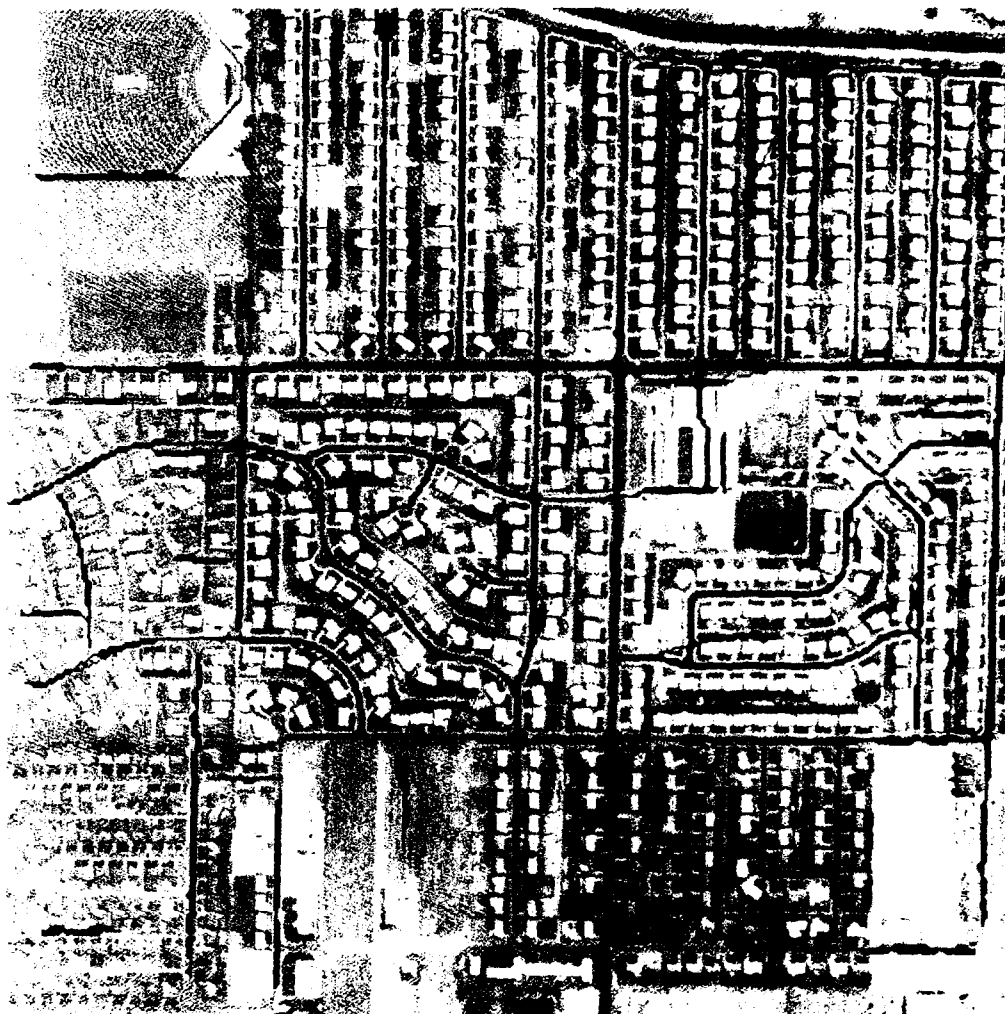


Figure 3.3: Superimposed SIO dark line extraction from PHOENIX.2B/
varying parameters /22/512/3/4 and /22/512/4/4
(matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 1m$)

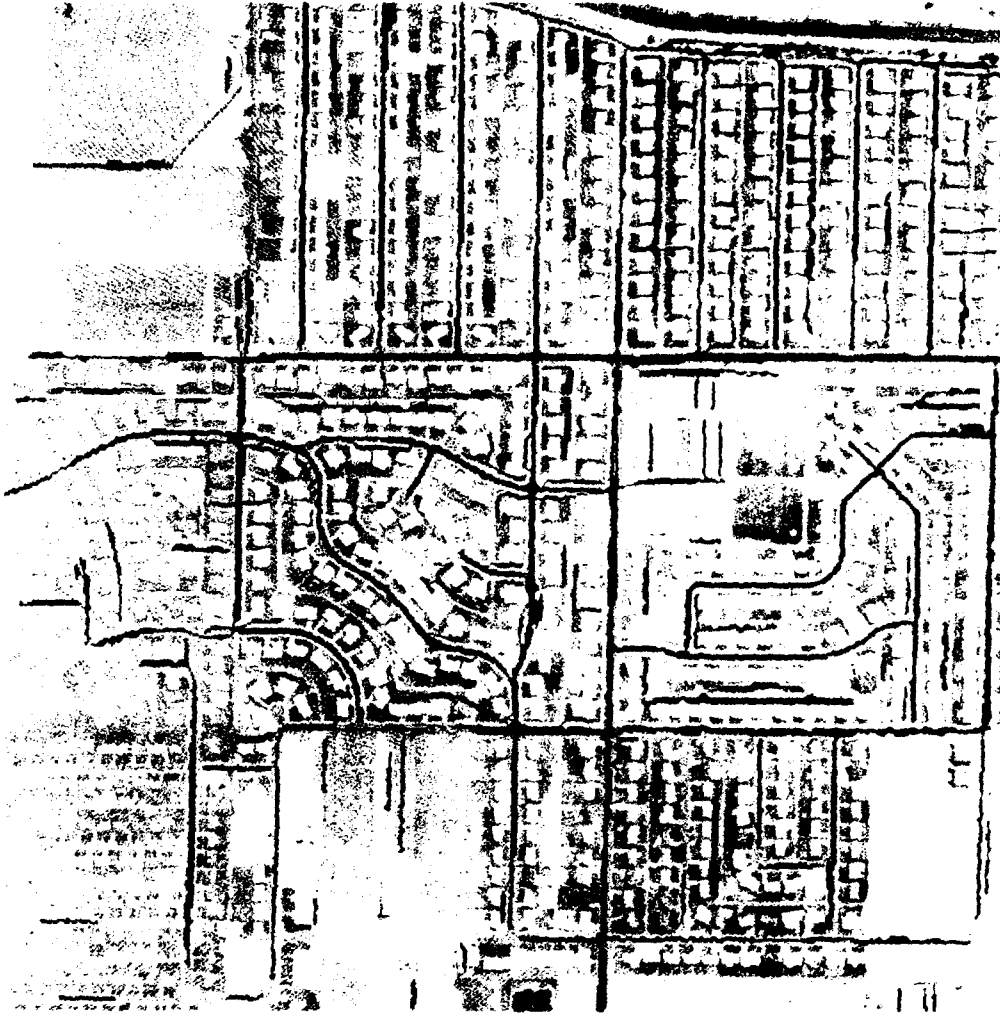


Figure 3.4: Superimposed SIO dark line extraction from PHOENIX.2B/
varying parameters /22/512/4/4 and /22/1024/5.5/5
(matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 1m$)

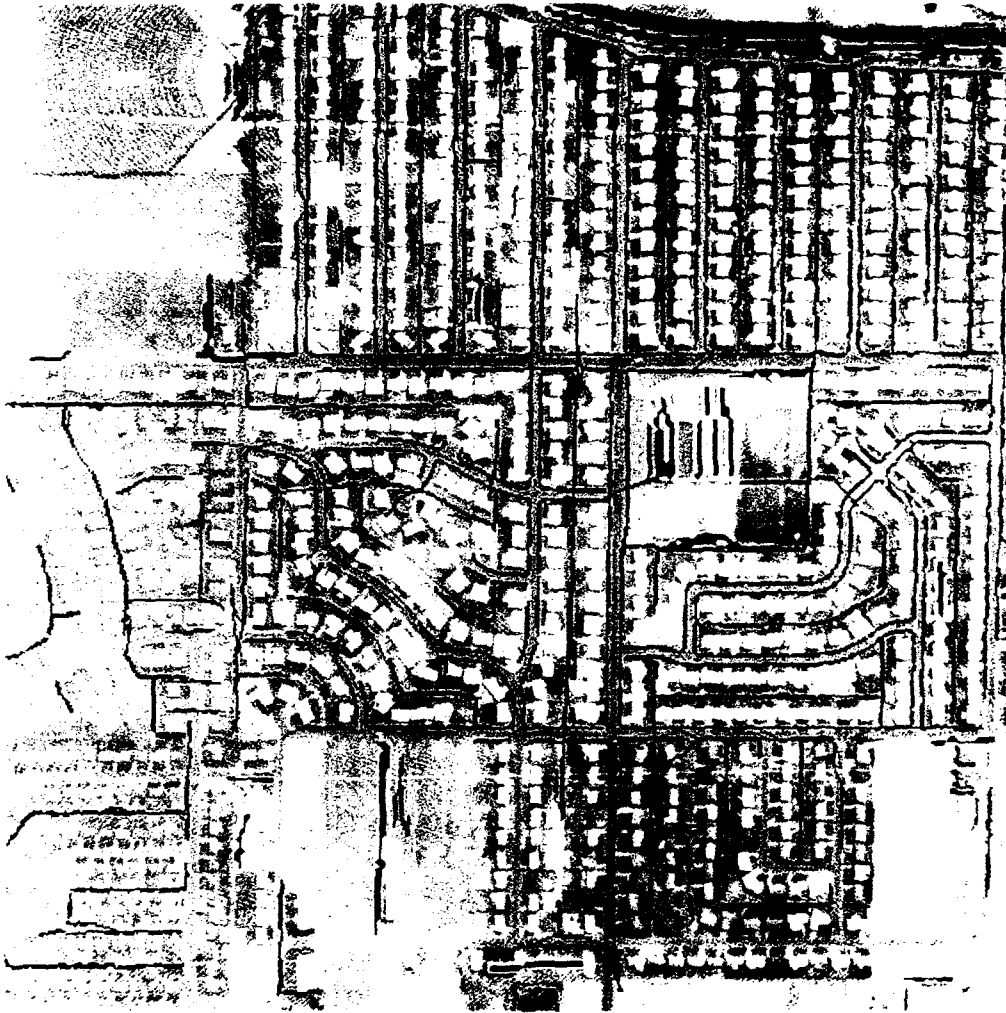


Figure 3.5: Superimposed SIO bright line extraction from PHOENIX 2B/
varying parameters /11/512/4.5/5 and /11/1024/3/4
(matrix size 1024×1024 pixels, pixel size and scale $50\mu m : 1m$)

Chapter 4

Structure Oriented Image Analysis with Blackboard System

For the analysis of objects in images a syntactical classification scheme is used which associates each object with a reference object or object class by analyzing the structure of the image according to the available hypotheses. A special type of such classifier had been realized at FIM [9,11] as the blackboard oriented production system for image understanding (BPI). This chapter describes the principle of the structural analysis demonstrated by a bridge identification task. In addition a formerly developed model for the detection of intersections adapted to a street network is applied to sections of 512×512 pixels from a subset of images used for the extraction of line objects by stream following:

- PHOENIX_2A (1/16 subimage of PHOENIX)
 - 1024 \times 1024 pixels
 - 1 : 20,000 real scale
 - 50 μm pixel size (on image)
 - 1 m pixel size (on earth)
- PHOENIX_2B (1/16 subimage of PHOENIX)
 - 1024 \times 1024 pixels
 - 1 : 20,000 real scale
 - 50 μm pixel size (on image)
 - 1 m pixel size (on earth)
- BIETIGHEIM1.3 (1/16 subimage of BIETIGHEIM1)
 - 1024 \times 1024 pixels

1 : 14,000 real scale
50 μm pixel size (on image)
0.7 m pixel size (on earth)

- BIETIGHEIM2.3 (1/4 subimage of BIETIGHEIM2)

1024 \times 1024 pixels
1 : 4,000 real scale
100 μm pixel size (on image)
0.4 m pixel size (on earth)

4.1 Situation Driven and Model Controlled Image Analysis

Starting with basic elements (terminals, base objects) as primitives generated by image preprocessing, the BPI system produces deduced elements (nonterminals, partial or target objects) consisting of more and more complex objects up to the target object. To build more complex objects from less complex ones, the system has to test hypotheses about the objects. Each hypothesis is tested by a knowledge source of the blackboard system. The generated objects are stored in the blackboard memory for associative access. All knowledge sources are used in parallel to get all possible interpretations of the image. One object can be part of several more complex objects.

4.1.1 Image Preprocessing

In the first step of the preprocessing a multilevel binarization controlled by histogram analysis of the gray value image is accomplished. Next the contour lines received by this method are approximated by straight lines, circle lines, or corners as primitives (base objects). The approximated primitives are assessed according to their quality of approximation and stored in the blackboard. They are the input data for the structural analysis by symbolic image processing.

In Figure 4.1 a gray value image a) is compared with the reconstructed image b) built up by the contour lines which are approximated by straight short lines received from preprocessing. The picture b) presents the results of the preprocessing and gives an impression of the input data used for symbolic image analysis: The number of the gray values is reduced and the shape of the contours of areas which have the same gray value is simplified.

4.1.2 Symbolic Image Analysis

The BPI system used in this project analyzes the structure of an image composed of the primitives (base objects) received from preprocessing. Starting

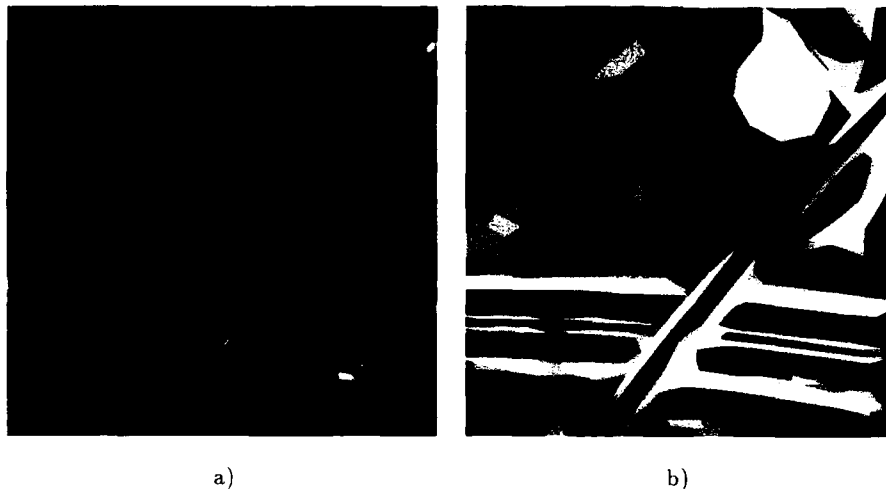
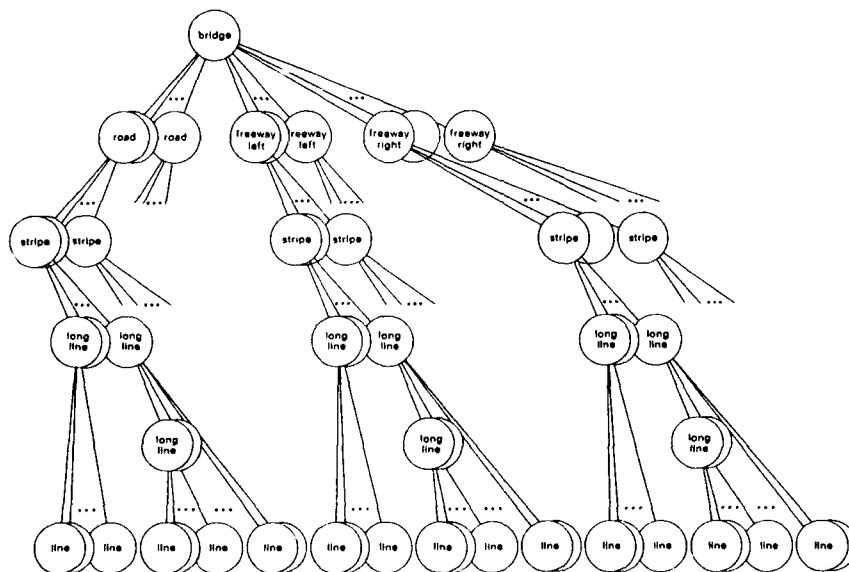


Figure 4.1: Preprocessing of a gray value image, a) original gray value image, b) image reconstructed by contour lines approximated by pieces of lines

with these base objects more and more complex partial objects (elements) are deduced according to the available hypotheses and the presence of objects in the image until a reference object or an object (target object) of a given class is found or all elements are processed. A derivation tree exists for each generated reference, resp. target object with the root of it being the reference, resp. target object itself, the branches being the deduced objects, and the leaves being the primitive objects. In Figure 4.2 a derivation tree is shown for the reference object *bridge*. Starting with the base objects *line* received from pre-processing and proceeding with the partial objects *long line*, *stripe*, and *road*, resp. *freeway* (*left side* and *right side* from the bridge), the target object *bridge* is constructed. In this BPI system, for example, the hypotheses *to be part of a road* and *to be part of a freeway left or right of the bridge* are assigned to *stripes*. The knowledge about the models used to construct the objects is stored as knowledge sources in the processing modules.

In order to give an example, the results of structural analysis by symbolic image processing (identification of a bridge) are shown in Figure 4.3 and Figure 4.4. In the parts a) and c) of the figures all elements (partial objects) generated before completing the identification of the reference object *bridge* are shown. The parts b) and d) show only those elements (partial objects) which belong to the derivation tree of the reference object *bridge*.

Figure 4.2: Derivation tree of the reference object *bridge*

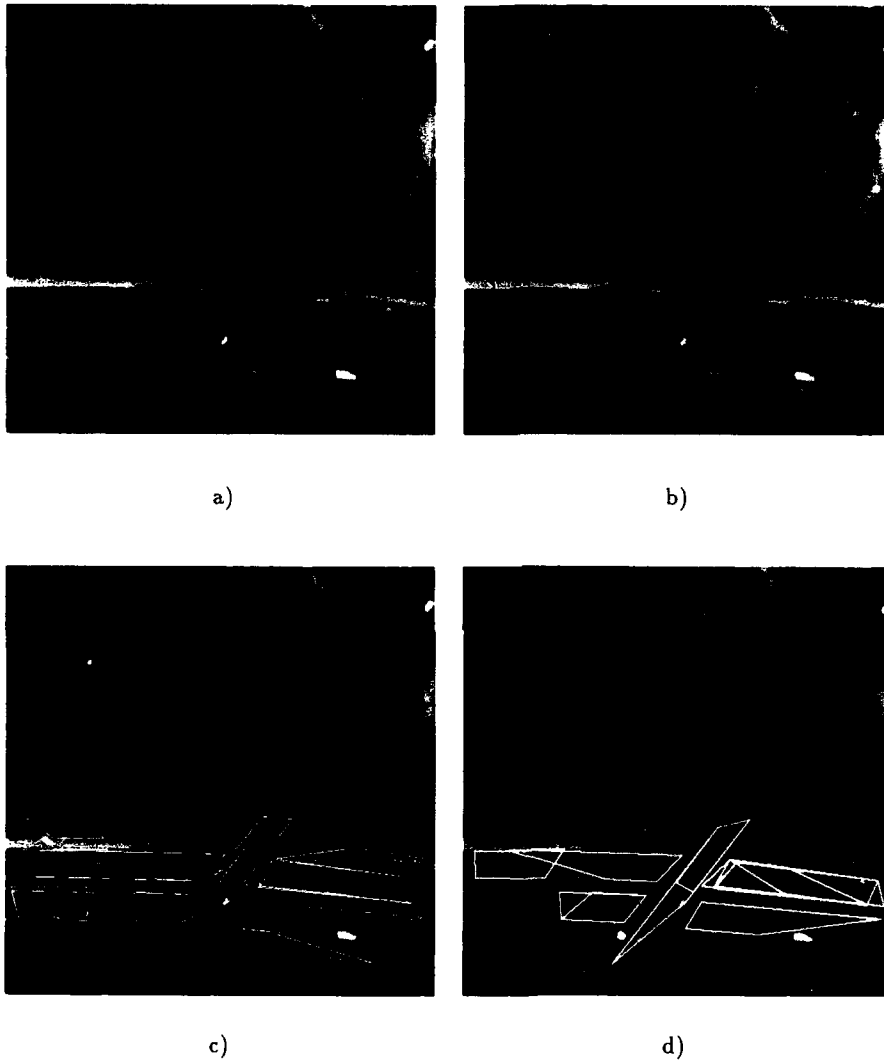


Figure 4.3: Result of structure analysis *bridge (I)*, a), c) generated *long lines* and *stripes*, b), d) *long lines* and *stripes* belonging to the reference object

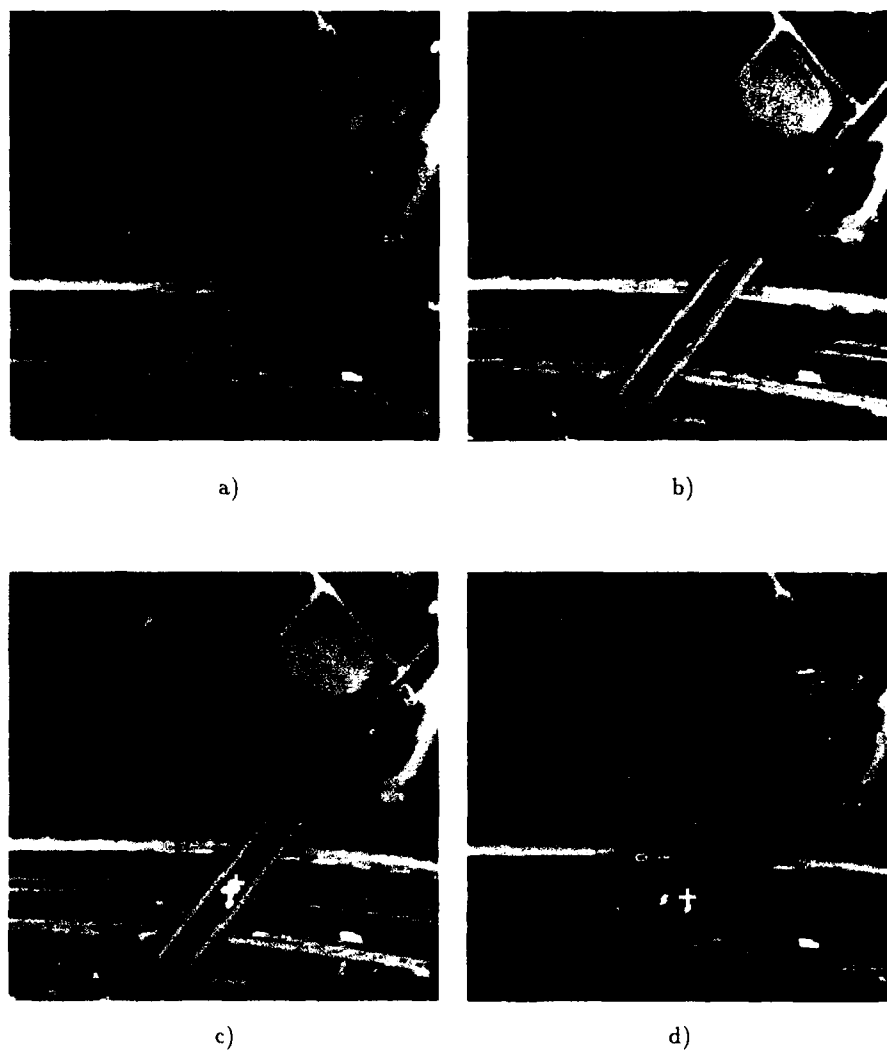


Figure 4.4: Result of structure analysis *bridge (II)*, a) generated parts of *roads* and *freeways*, b) parts of *roads* and *freeways* belonging to the reference object, c) generated *bridge* positions, d) position of identified *bridge*

This BPI system had been used successfully to identify reference objects like a *bridge* or an *intersection*. More details are explained in [10]. For another project the BPI system was extended by new knowledge modules for the classification of objects especially for the detection of *intersections* (see section 4.3 [5,6]).

4.2 The Blackboard Production System

The principle structure of the blackboard system used for identification or classification purposes is shown in Figure 4.5. The system contains a memory (blackboard), in which all data such as base objects (e.g. *line*), partial objects (e.g. *road*) and target objects (e.g. *bridge*) are stored. The data are processed by verification programs (processing modules with knowledge sources) working independently from each other. The data transfer (exchange) may occur only via the blackboard.

Every blackboard system uses a supervisor that decides which data (base or partial objects) stored in the blackboard are to be processed next and which verification program must become active. By means of a queue and by using an assessment strategy the *best* object is selected for every processing step.

An example for symbolic image processing with a BPI system is given in the Figures 4.3 and 4.4 shown above.

4.2.1 Blackboard

The blackboard is the central database of the system realized as associative memory accessed by the queue manager of the supervisor and by all programs of the processing modules for testing the hypotheses. Base objects (e.g. *line*), partial objects (e.g. parts of roads and freeways as *long line*, *stripe*, *road*, *free way left* and *freeway right*), and target objects (e.g. *bridge*) called *objects* in the following are entered into the blackboard. Objects are described by their attributes. Examples for attributes are: *type*, *length*, *orientation*, or *position*. Therefore an object of the database could be characterized by e.g. *type* \Leftarrow *line*, *length* \Leftarrow 40, *orientation in degrees* \Leftarrow 30, and *coordinates of the gravity point* \Leftarrow (*x* = 10, *y* = 50).

4.2.2 Production Cycle

The base objects, e.g. short lines from image preprocessing are entered as assessed elements in a priority driven queue for symbolic processing. Furthermore they are stored in the blackboard together with their description by attributes (Figure 4.6). During each production cycle it is tried to combine base objects to more complex partial objects, e.g. long lines, or to compose much more complex partial objects, e.g. roads, of earlier generated partial objects, e.g.

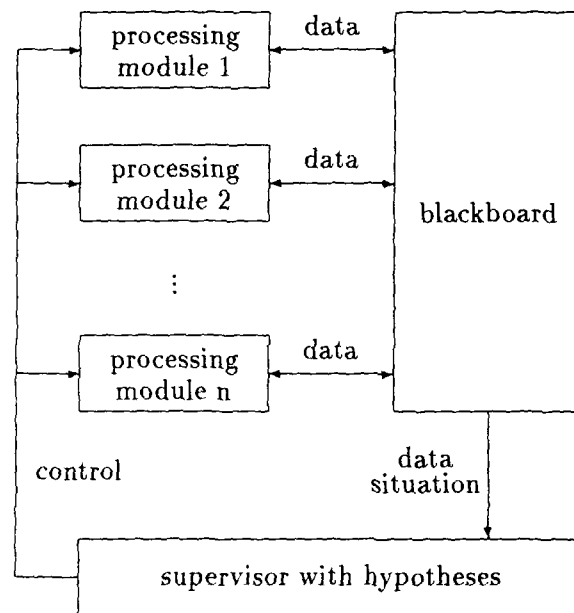


Figure 4.5: Blackboard system

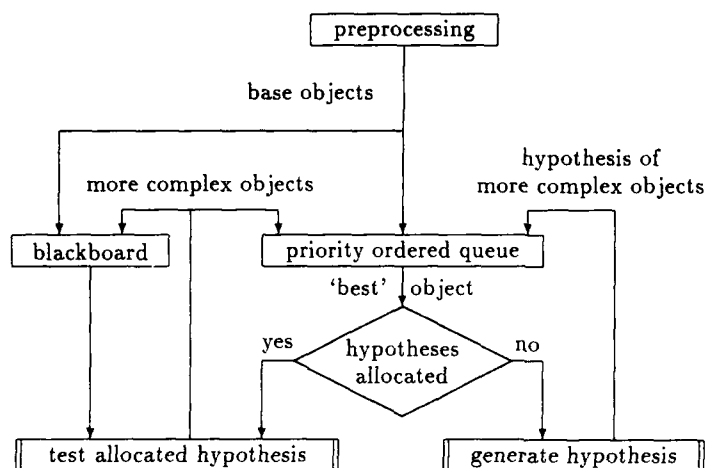


Figure 4.6: Principle of a production cycle

stripes. The production of deduced elements proceeds until the set of nonprocessed elements is empty, or the target object, e.g. bridge is constructed.

The synthesis of a more complex partial object from a less complex partial object during each production cycle proceeds in two steps. Initially, hypotheses are assigned to objects available in the queue. The objects are priority ordered according to their assessment. Finally, the hypotheses assigned to the objects are tested, and, if they are confirmed, those objects allocated to a confirmed hypothesis are generated, assessed and put into the blackboard and into the queue.

4.2.3 General Assessment and Context Spaces

All objects not yet being processed are put into the queue. It is the main task of the supervisor to select from the probably great number of nonprocessed objects those objects, which have a good assessment with respect to as many criteria as possible, and which, therefore, can be combined properly to more complex objects.

Base objects are assessed during the preprocessing cycle resulting from the quality of the image filtering operation. For each object allocated to a specific processing module (knowledge source), an adequate set of assessment quantities has to be defined for their attributes. First, during execution time of the processing module when generating an object an assessment value is allocated to each assessment quantity. The assessment can be chosen object oriented ac-

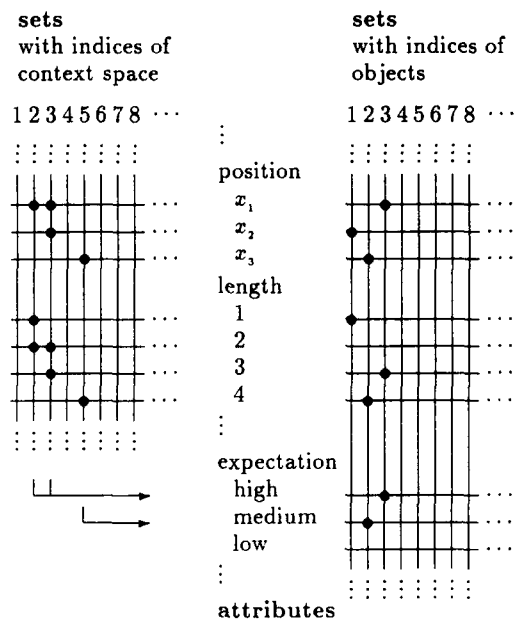


Figure 4.7: Example of context space consisting of index sets for objects and context each addressable by attributes

cording to the quality of the generation or application oriented according to certain search criteria for object identification or classification. Second, during the control cycle of the supervisor a general assessment is derived from the values of the separately assessed attributes of the objects. This result leads to the decision which object is to be selected next for symbolic processing.

For effective processing and selection of objects, the object space with its attributes can be matched against different context spaces containing the same attributes. For associative access the objects are stored in a relational data base using index lists resident in memory. The index sets addressable by attributes are formatted by binary coding or by bit masking as shown in Figure 4.7. Competitive and concurrent knowledge for symbol interpretation is provided by context being at the disposal of the supervisor. The context space is associated to the object space as shown in Figure 4.7. In [15] a proposal for a BPI hardware is made.

4.2.4 Final Assessment and Stop Criterion

An additional task of the assessment procedure is to specify the quality of the target object with which a reference object is identified or an object of a class is detected. This final assessment depends on the degree of matching between the reference model and the target object (see Figure 4.4c and 4.4d). Therefore the stop criterion for an identification run is the defined degree of precision with which the target object is detected. Generally, in case of no reference object can be identified or in case of object classification, a run is terminated after all objects have been processed.

4.3 Edge Oriented Detection of Intersections

In order to detect intersections a formerly developed model for the identification of crossing roads was applied. First a formal description of the BPI system and detailed information about the applied productions are given. Then an example of object classification by blackboard oriented structural image analysis is presented.

4.3.1 Parallel Rewrite System

Starting with primitives received from preprocessing more and more complex partial objects are generated by adequate productions (verification programs) until, if present in the image to be analyzed, the desired reference object or an object of that class is generated. The entire process can be described as a parallel rewrite system R . Referring to formal languages the system R can be expressed as a 4 *tupel* of sets, however, its structural complexity of elements (objects) is much higher than known from string generating grammars and its deduced elements specified by productions are tested inversely for recognition:

$$R = (V_t, V_n, S, P) \quad \text{with}$$

V_t = set of base objects
 V_n = set of partial objects
 S = set of target objects
 P = set of productions

The objects produced by preprocessing can be considered as the base objects (e.g. *lines*). Partial objects are all objects generated by the productions (e.g. *long lines, stripes, fields, streets, intersections*). The productions specify the transformation of a triggering base object or partial object together with other specified objects into another partial object or target object. One or more productions exist for the generation of each object. In the BPI system the productions are realized as procedural verification programs. The target objects (e.g. *street network*) and a small subset of those objects which can be derived

from the targets by *contain-the-part* relations (e.g. *intersections*) are the image objects to be detected or extracted, respectively.

4.3.2 Set of Productions

The following definitions are applied for the extraction of street networks with intersections:

$$\begin{aligned}
 V_t &= \{ \text{line} \} \\
 V_n &= \{ \text{long line, stripe, street, field, intersection} \} \\
 S &= \{ \text{field, street network} \} \\
 P &= \{ \begin{aligned}
 &P01: \text{line (line)} \longrightarrow \text{long line;} \\
 &P02: \text{long line (line, long line)} \longrightarrow \text{long line;} \\
 &P03: \text{long line (line, long line)} \longrightarrow \text{stripe;} \\
 &P04: \text{long line (line, long line)} \longrightarrow \text{stripe, stripe;} \\
 &P05: \text{stripe (stripe)} \longrightarrow \text{street;} \\
 &P06: \text{stripe (stripe)} \longrightarrow \text{field;} \\
 &P07: \text{street (stripe, street)} \longrightarrow \text{street;} \\
 &P08: \text{street (street)} \longrightarrow \text{intersection, ..., intersection;} \\
 &P09: \text{intersection (street, intersection)} \longrightarrow \text{street network;} \\
 &P10: \text{street network (street network)} \longrightarrow \text{street network} \}
 \end{aligned}
 \end{aligned}$$

The BPI system had been originally applied to identification tasks with reference objects strongly described by their attributes. When applying the BPI system to the classification of objects, resp. detection of the target object, the classes must be described weakly by the attributes of the objects concerned. Therefore additional classes below and above the class to be recognized have to be introduced to achieve a high performance.

In order to reduce misrecognition of streets, for example, the competing partial object *field* below the object class *intersection* is introduced because of confusion between streets and fields. And above the object class *intersection* the integrating target object *street network* is introduced to guarantee that only intersections are recognized which can be connected to a street network. All detected isolated intersections get a worse assessment than that intersections which lie within the street network.

Starting from a base object the *is-part-of* relations of the production set which generally are not free of context can be described graphically as a production net. The applied production net for the object class *street network* with the contextfree productions *P01* to *P10* is shown in Figure 4.8. Here an arrow corresponds to the contextfree relation *is-part-of* and the production net can be structured hierarchically.

The productions *P01* to *P10* are described in detail in the following subsections. Thereby the so-called *triggering* partial object denotes that object which

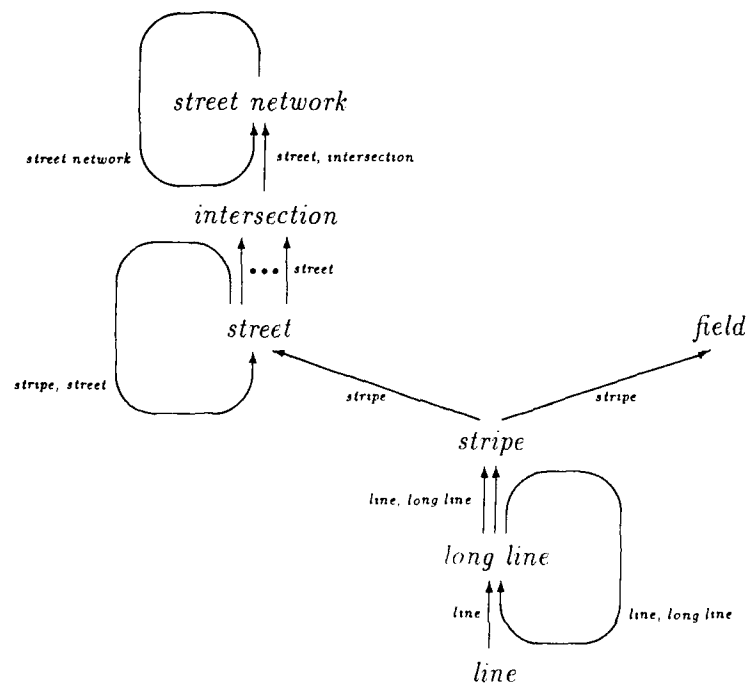


Figure 4.8: Production net for the edge oriented detection of intersections

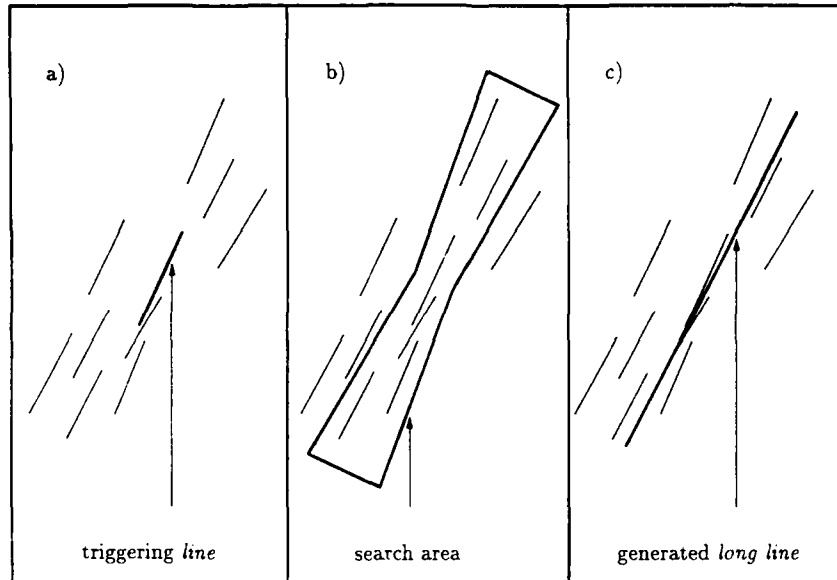


Figure 4.9: Principle of line prolongation

is chosen from the supervisor as the (at present) *best* object for further processing (see Figure 4.6).

P01: Line \rightarrow Long Line

If a base object *line* is selected as best object, then the hypothesis for the partial object *long line* has to be tested. The associated verification program for testing that hypothesis is triggered by the base object *line* selected as best object. For testing purpose a narrow search area around the selected line is examined in order to detect other lines with a similar orientation (parallelism). In case of a successful examination the partial object *long line* is built up by the selected line together with those lines found. After composition of the long line an assessment is accomplished according to the overlapping of lines and the length of the long line. Figure 4.9 shows the principle of line prolongation.

P02: Long Line \rightarrow Long Line

If a partial object *long line* is selected as best object, then the hypothesis for the partial object *long line* has to be tested in addition to the hypotheses P03 and P04. The associated verification program for testing that hypothesis is triggered by the partial object *long line* selected as best object. For testing purpose a narrow search area around the selected long line is examined in order to detect other lines or long lines with a similar orientation (parallelism). In case of a successful examination the partial object *long line* is built up by the selected long line together with those lines and long lines found. After composition of the long line an assessment is accomplished according to the overlapping of lines and long lines, and the length of the generated long line.

P03: Long Line \rightarrow Stripe

If a partial object *long line* is selected as best object, then the hypothesis for the partial object *stripe* has to be tested in addition to the hypotheses P02 and P04. The associated verification program for testing that hypothesis is triggered by the partial object *long line* selected as best object. For testing purpose a narrow search area on both sides of this long line is examined in order to detect other long lines with a similar orientation. The width of the search area can vary within given limits. In case of a successful examination for one deduced object the partial object *stripe* is built up by the selected long line together with those lines and long lines found. After composition of the stripe an assessment is accomplished according to the length of the generated stripe and the accuracy of parallelism of long lines.

P04: Long Line \rightarrow Stripe, Stripe

If a partial object *long line* is selected as best object, then the hypothesis for the partial object *stripe* has to be tested in addition to the hypotheses P02 and P03. The associated verification program for testing that hypothesis is triggered by the partial object *long line* selected as best object. For testing purpose a narrow search area on both sides of this long line is examined in order to detect other long lines with a similar orientation. In case of a successful examination for a pair of deduced neighboring objects two partial objects *stripe* are built up by the selected long line together with those lines and long lines found. After composition of the two stripes tangent to each other an assessment is accomplished according to the length of the generated stripes and the accuracy of parallelism of long lines.

P05: Stripe \rightarrow Street

If a partial object *stripe* is selected as best object, then the hypothesis for the partial object *street* has to be tested in addition to the hypothesis P06. The associated verification program for testing that hypothesis is triggered by

the partial object *stripe* selected as best object. For testing purpose a narrow search area around this element is examined in order to detect other stripes with a similar orientation for stripe prolongation. In case of a successful examination the partial object *street* is built up by the selected stripe together with those stripes found. After composition of the street an assessment is accomplished according to the length of the generated street and its accuracy from stripe composition.

P06: Stripe → Field

If a partial object *stripe* is selected as best object, then the hypothesis for the partial object *field* has to be tested in addition to the hypothesis P05. The associated verification program for testing that hypothesis is triggered by the partial object *stripe* selected as best object. For testing purpose a narrow search area around this element is examined in order to detect other stripes with a similar orientation for stripe prolongation. In case of a successful examination a street like stripe prolongation is built up by the selected stripe together with those stripes found. If the found stripe prolongation has many parallel fictitious "roadsides" then it is implied that the street like object is a field. Therefore the partial object *field* is generated instead of the partial object *street*. The composition and assessment of the field may be suppressed.

P07: Street → Street

If a partial object *street* is selected as best object, then the hypothesis for the partial object *street* has to be tested in addition to the hypothesis P08. The associated verification program for testing that hypothesis is triggered by the partial object *street* selected as best object. For testing purpose a narrow search area around this element is examined in order to detect other stripes or streets with a similar orientation for street prolongation. In case of a successful examination the partial object *street* is built up by the selected street together with those stripes and streets found. After composition of the street an assessment is accomplished according to the length of the generated street and its accuracy from stripe and street composition.

P08: Street → Intersection, ..., Intersection

If a partial object *street* is selected as best object, then the hypothesis for the partial object *intersection* has to be tested in addition to the hypothesis P07. The associated verification program for testing that hypothesis is triggered by the partial object *street* selected as best object. For testing purpose many search areas originating from equidistant positions along the selected street are oriented at different angles with respect to the direction of the street. The selected street is used to find other streets which have a crossing with this street within the image. The streets found have to meet the criterion *closeness to the*

selected street. In case of a successful examination the partial objects *intersection* are built up by the selected street together with those streets found. After composition of the intersections an assessment is accomplished according to the above mentioned criterion.

P09: Intersection \rightarrow Street Network

If a partial object *intersection* is selected as best object, then the hypothesis for the target object *street network* has to be tested in addition to the hypothesis P10. The associated verification program for testing that hypothesis is triggered by the partial object *intersection* selected as best object. For testing purpose all streets belonging to the selected intersection are examined with regard to other intersections assigned to those streets or to edges of the image. In case of a successful examination the target object *street network* is built up by the selected intersection together with those sections of the streets which are neighboring to the selected intersection. If no neighboring intersection exists that section of the street which leads to the edge of the image is added to the street network. After composition of the street network the assessment of all intersections involved are increased above a given threshold. However, in cases that the selected intersection cannot be integrated into the street network its assessment is reduced. If the intersection can be integrated into the street network later on then its assessment is increased again.

P10: Street Network \rightarrow Street Network

All partial objects *intersection* generated by testing the hypothesis P09 are assembled. The intersections with their associated streets can be accessed by searching for assessment values greater than the threshold given for successful examination of intersections.

4.3.3 Results from Detection of Intersections

In order to test the model for the edge oriented detection of intersections described in the previous section the following images were used:

- PHOENIX.2A1 (1/4 section of PHOENIX.2, see Figure 4.10)

512 \times 512 pixels

1 : 20,000 real scale

50 μ m pixel size (on image)

1 m pixel size (on earth)

- PHOENIX.2B1 (1/4 section of PHOENIX.2, see Figure 4.14)

512 \times 512 pixels

1 : 20,000 real scale

50 μm pixel size (on image)

1 m pixel size (on earth)

- BIETIGHEIM1.31 (1/4 section of BIETIGHEIM1.3, see Figure 4.18)

512 \times 512 pixels

1 : 14,000 real scale

50 μm pixel size (on image)

0.7 m pixel size (on earth)

- BIETIGHEIM2.31 (1/4 section of BIETIGHEIM2.3, see Figure 4.22)

512 \times 512 pixels

1 : 4,000 real scale

100 μm pixel size (on image)

0.4 m pixel size (on earth)

Figure 4.11, 4.15, 4.19, and 4.23 show the gray value images after preprocessing. The overlaid yellow lines are the base objects *line* from preprocessing. In the first step of the preprocessing a binarization by 10 levels controlled by histogram analysis of the gray value image is accomplished. Next the contour lines received by this method are approximated by straight lines. This is done by fixing a start point and counting 40 pixels (stick length) on the contour line to yield the end point. Start point and end point are connected by a straight line and both the area between straight line and contour line and the distance between start point and end point are determined. From both measures the quality of the detected line primitive of given stick length is classified. For the assessment of the line primitive 6 quality classes are provided. Thereby the sixth class represents no qualification for a straight line and the first class an excellent straight line with respect to the given stick length. Depending on the line quality the stick length may be varied and another trial may be accomplished. The next step of line approximation may proceed starting from a new contour point placed back a given number of pixels, e.g. 20 pixels on the contour line.

In Figure 4.12, 4.16, 4.20, and 4.24 the triggering base objects *line* are presented. The triggering lines are displayed in yellow color denoting those objects which at any given time are chosen from the supervisor as the best object for further processing with respect to the production *P01*.

In Figure 4.13, 4.17, 4.21, and 4.25 the results of the detection of intersections are presented. The object class *intersection* is marked in green color. Intersections are connected by straight thin green lines in order to indicate the target object *street network*. The partial objects *street* are displayed in red color, whereas the noncomposing partial objects *stripe* and *long line* are displayed in yellow or blue, respectively.

Comparing the results of street extraction achieved from structure oriented image analysis with those from the line object extraction by stream following a considerable gain in recognition performance can be observed. The used model works quite well if the streets have no curves. This is due to the fact that the model uses only straight lines. With a better parameter setting one can suppose that shorter streets are found. Obviously a more detailed image preprocessing together with a more sophisticated symbolic description of objects for structural image analysis are required.

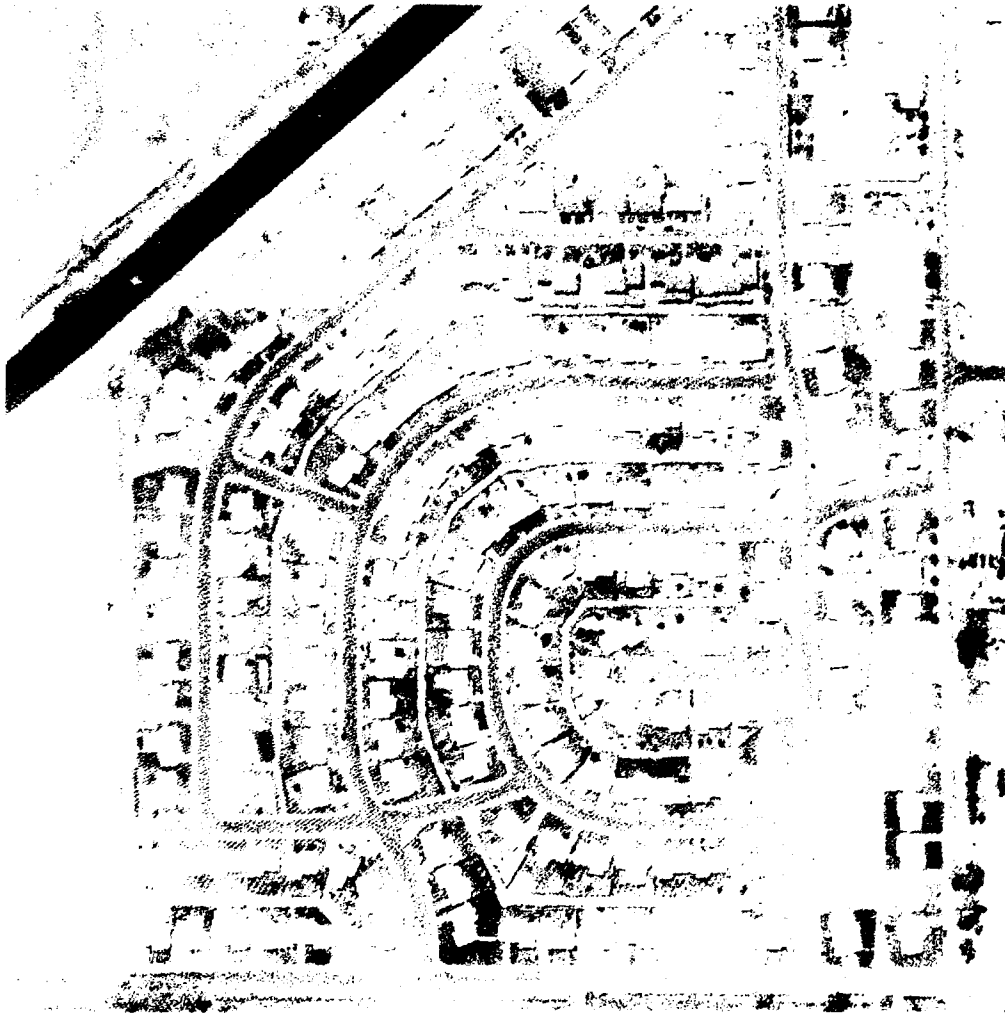


Figure 4.10: Test image PHOENIX.2A1
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 4.11: Preprocessing of PHOENIX_2A1/base objects: *line* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 4.12: Analysis of PHOENIX.2A1/triggering base objects: *line* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 4.13: Analysis of PHOENIX_2A1/target object: street network (green), partial objects: intersection (green), street (red), noncomposing partial objects: stripe (yellow), long line (blue) (matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 4.14: Test image PHOENIX.2B1
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)

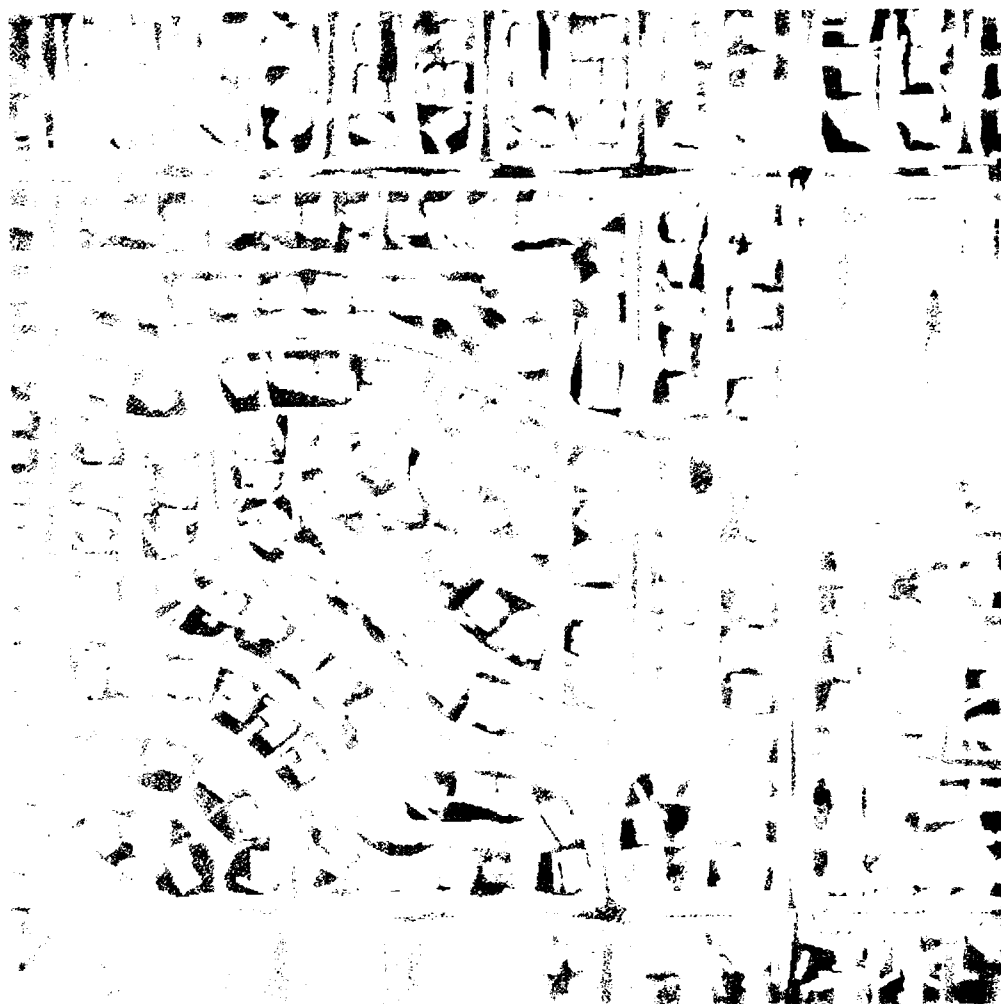


Figure 4.15: Preprocessing of PHOENIX.2B1/base objects: *line* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 1\text{m}$)



Figure 4.16: Analysis of PHOENIX_2B1/triggering base objects: *line* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 4.17: Analysis of PHOENIX_2B1/target object: street network (green), partial objects: *intersection* (green), *street* (red), noncomposing partial objects: *stripe* (yellow), *long line* (blue) (matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 4.18: Test image BIETIGHEIM1.31
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 0.7m$)

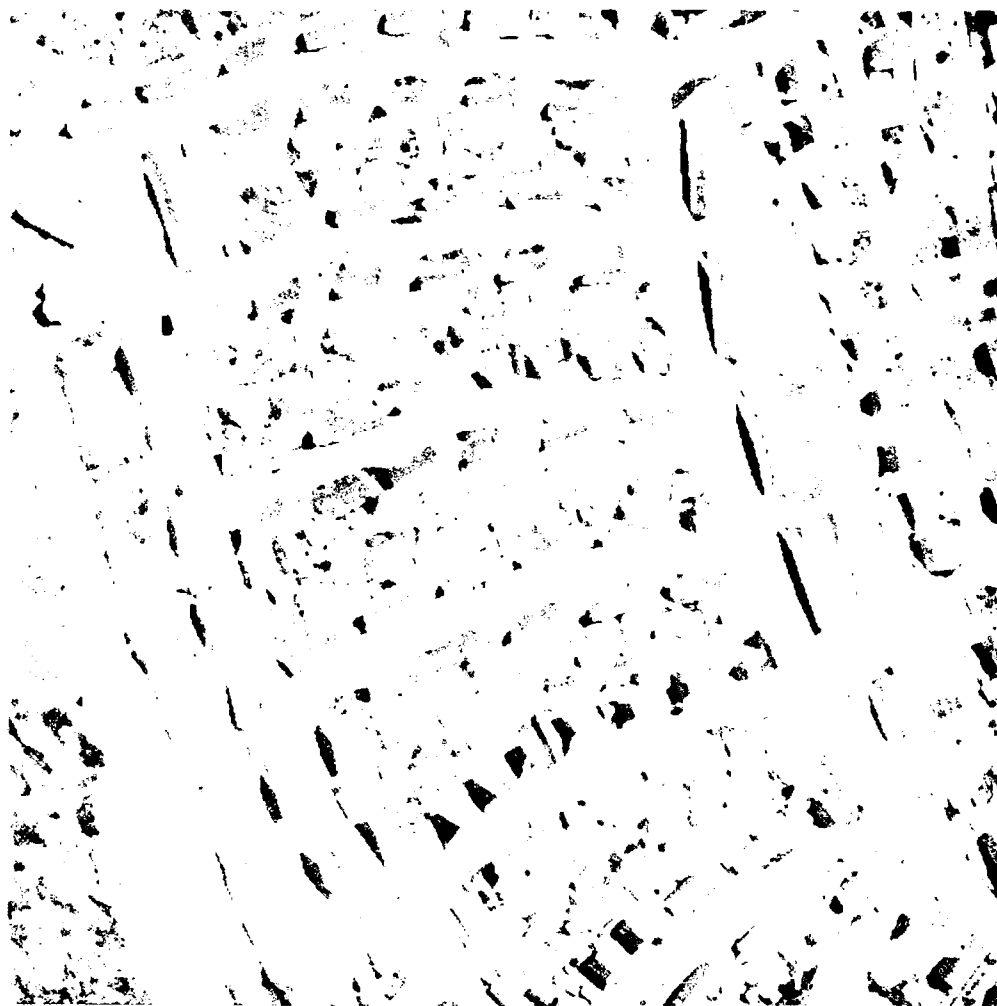


Figure 4.19: Preprocessing of BIETIGHEIM1.31/base objects: *line* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 0.7m$)

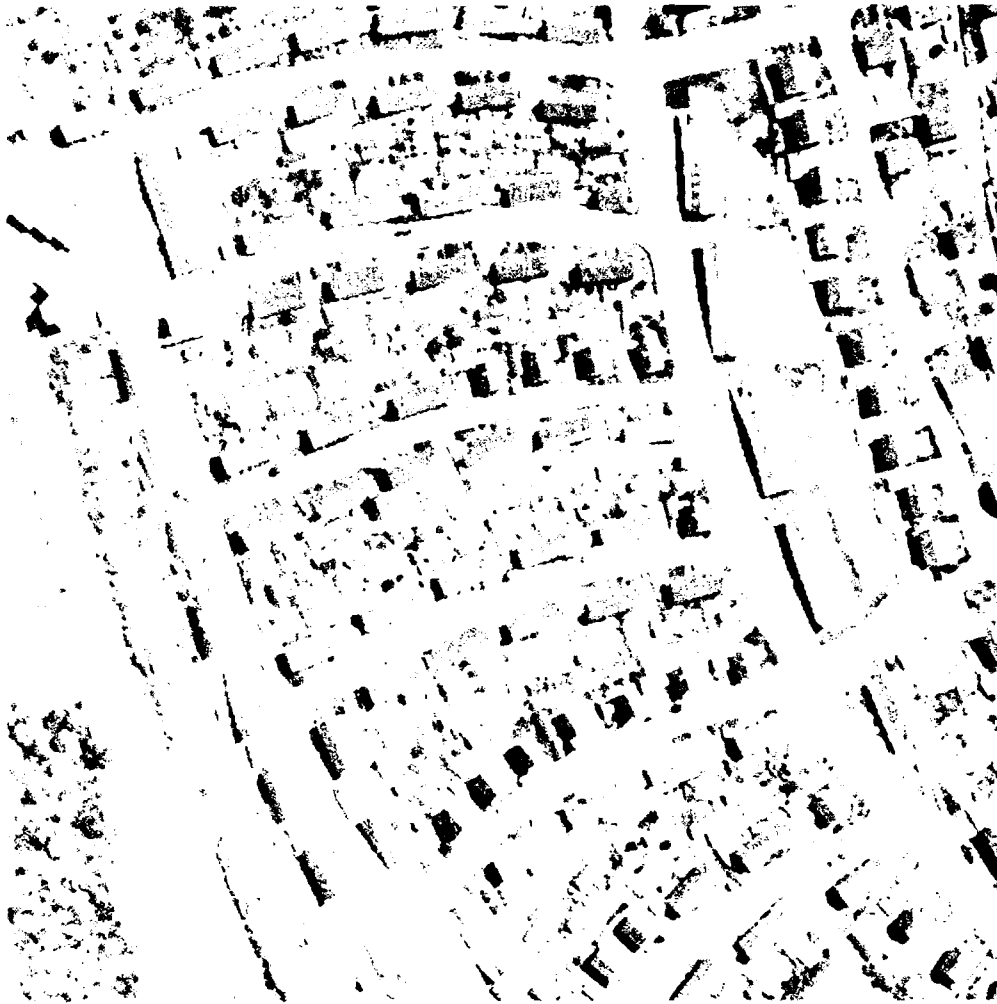


Figure 4.20: Analysis of BIETIGHEIM1.31/triggering base objects:
line (yellow) (matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 0.7\text{m}$)

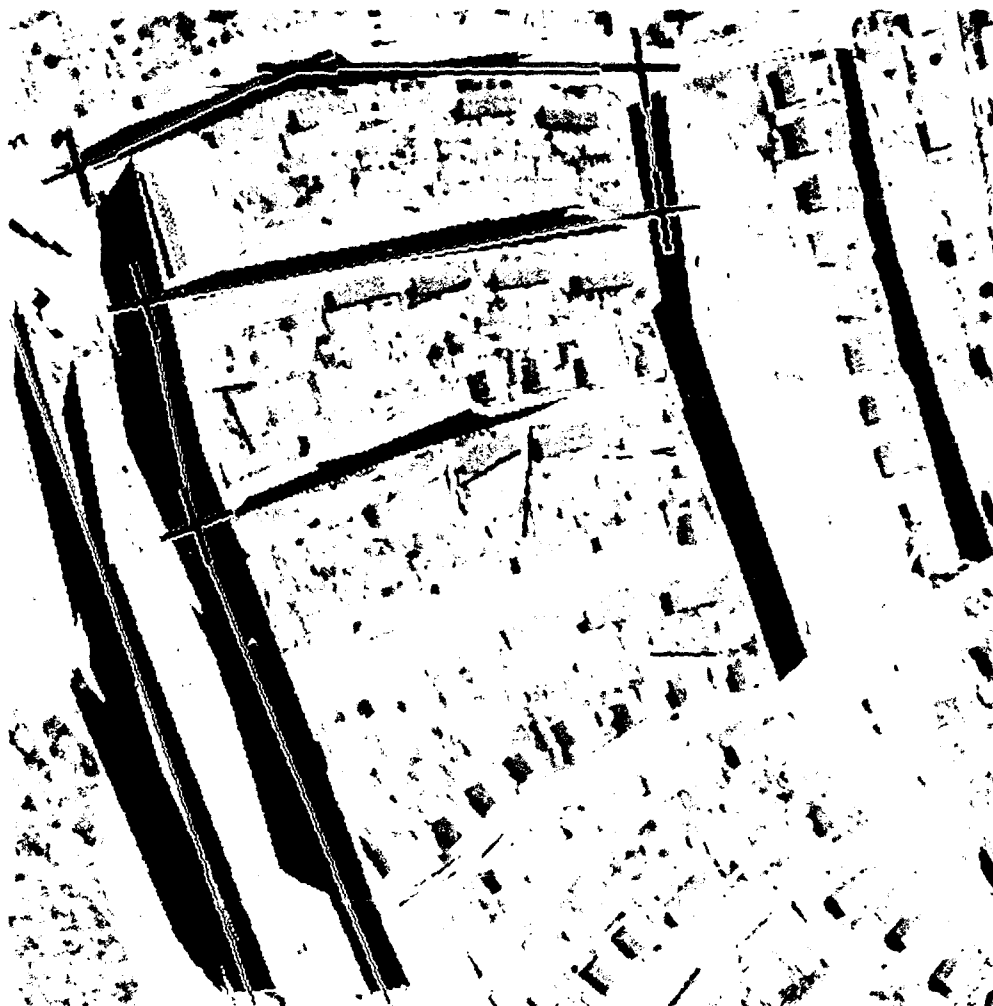


Figure 4.21: Analysis of BIETIGHEIM1.31/target object:
street network (green), partial objects: *intersection* (green), *street* (red),
noncomposing partial objects: *stripe* (yellow), *long line* (blue)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 0.7\text{m}$)



Figure 4.22: Test image BIETIGHEIM2.31
(matrix size 512×512 pixels, pixel size and scale $100\mu m : 0.4m$)



Figure 4.23: Preprocessing of BIETIGHEIM2.31/base objects: *line* (yellow)
(matrix size 512×512 pixels, pixel size and scale $100\mu m : 0.4m$)



Figure 4.24: Analysis of BIETIGHEIM2 31/triggering base objects:
line (yellow) (matrix size 512×512 pixels, pixel size and scale $100\mu\text{m} : 0.4\text{m}$)



Figure 4.25: Analysis of BIETIGHEIM2.31/target object:
street network (green), partial objects: *intersection* (green), *street* (red),
noncomposing partial objects: *stripe* (yellow), *long line* (blue)
(matrix size 512×512 pixels, pixel size and scale $100\mu m : 0.4m$)

Chapter 5

Iconic Image Filtering and Segmentation

The implemented improvement of image preprocessing required for street network analysis by means of an extended structural model is described in this chapter.

First, a multiresolution gray level pyramid by successive reduction of the digitized image is computed. Then two image filtering operators, which transform the gray values into gray value differences representing contrast information of the image, are applied to the gray level pyramid. The first operator generates edge elements which are combined to edge segments in a next step. The second operator is a spot detector which indicates bright or dark contrasting spots in the gray level pyramid. These spots are indicators used to extract area elements from the high resolution level of the pyramid. These two procedures are utilized for iconic image preprocessing to compute area, resp. edge like image segments. The results of these segmentation procedures are symbolically described and stored in a blackboard for further processing by structural image analysis.

The preprocessing is applied to a 512×512 section of the image PHOENIX_2A for demonstration of the implemented methods:

- PHOENIX_2A1 (1/4 section of PHOENIX_2A)

512 \times 512 pixels

1 : 20,000 real scale

50 μm pixel size (on image)

1 m pixel size (on earth)

5.1 Gray Level Pyramid

With respect to the observation that objects often can be more easily recognized in images of low resolution (this is due to the fact that confusing details

in the high resolution images are not present in the low resolution ones), a gray level pyramid of the image has to be generated before filtering [13].

5.1.1 Generation of the Gray Level Pyramid

The pyramidal image structure consists of a series of images representing different resolution levels of the original image. In order to generate these different resolved images the image scanned with the best resolution is filtered by a two dimensional low pass of variable smoothing strength. This procedure is equivalent to a repeated scanning with different sampling rates. Another possibility to generate such a pyramid is to combine nonoverlapping cells of $m \times m$ pixels to a new image point in the next resolution level of the pyramid.

Figure 5.1 shows the pyramidal image structure realized by the procedure described as follows. The original image is divided into squares with the size of 2×2 pixels. Each square in a level n is transformed into one image point in the level $n - 1$. This process is repeated for each level of the pyramid until the top of it is reached. Therefore a pyramid consists of $N + 1$ levels with the dimension $2^n \times 2^n$, with $n = 0, 1, \dots, N$. The level with the dimension $2^N \times 2^N$ is the basis of the pyramid, whereas the top of the pyramid is built by the single point ($2^0 \times 2^0$) of the level 0. Consequently each level of the pyramid consists of a matrix $(s_{i,j})^{(n)}$, with $i, j = 1, 2, \dots, 2^n$, and $n = 0, 1, \dots, N$. When transferring from level n with an image size of $n \times n$ pixels to level $n - 1$, the image size is reduced to $n/2 \times n/2$ pixels, that is a quarter of the original image size.

The computation of the matrix $(s_{i,j})^{(k-1)}$ from the matrix $(s_{i,j})^{(k)}$ in any level k is done by a suitable operator \mathcal{O} . Figure 5.2 shows the generation of a new image point p_0 in level $k - 1$ from the quadrupel of the image points in the level k by using \mathcal{O} . This mapping is defined by the operator \mathcal{O} , such that:

$$s_{p_0}^{(k-1)} = \mathcal{O}(s_{p_1}^{(k)}, s_{p_2}^{(k)}, s_{p_3}^{(k)}, s_{p_4}^{(k)})$$

Appropriate operators \mathcal{O} , for example, are:

$$\begin{aligned} \mathcal{O}_{\text{avg}} &= 1/4 \sum_{i=1}^4 s_{p_i}^{(k)} \\ \mathcal{O}_{\text{min}} &= \min_{i=1}^4 s_{p_i}^{(k)} \\ \mathcal{O}_{\text{max}} &= \max_{i=1}^4 s_{p_i}^{(k)} \end{aligned}$$

The generation of multiresolution gray level pyramids can be done by any smoothing algorithm. The structural levels of the pyramid show a growing disappearance of the finer details in the levels of worse resolution. The gray level pyramid of structural levels 8 to 0 resulting from applying an unweighted averaging operator to the original image is shown in Figure 5.3.

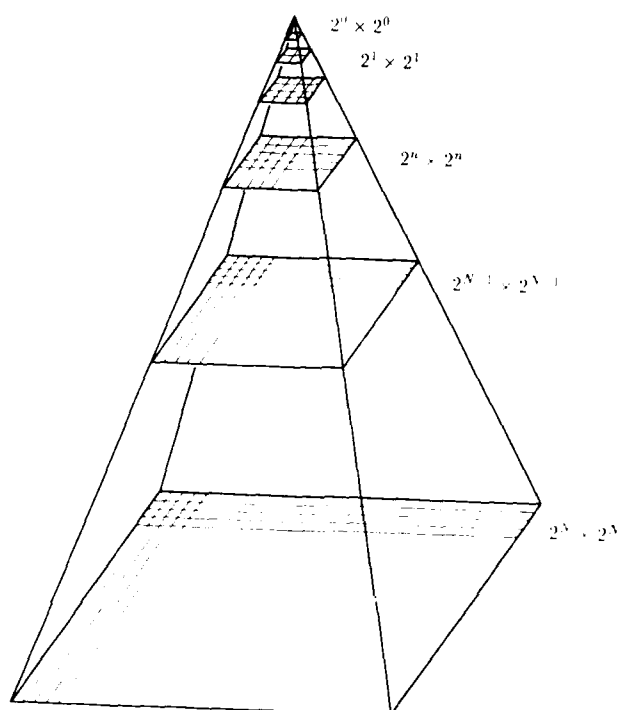


Figure 5.1: Pyramidal image structure

$$\begin{pmatrix} p_1 & p_2 \\ p_3 & p_4 \end{pmatrix} \Rightarrow (p_0)$$

Figure 5.2: Generation of a pyramidal image point p_0 at level $k - 1$ from the quadrupel (p_1, p_2, p_3, p_4) at level k

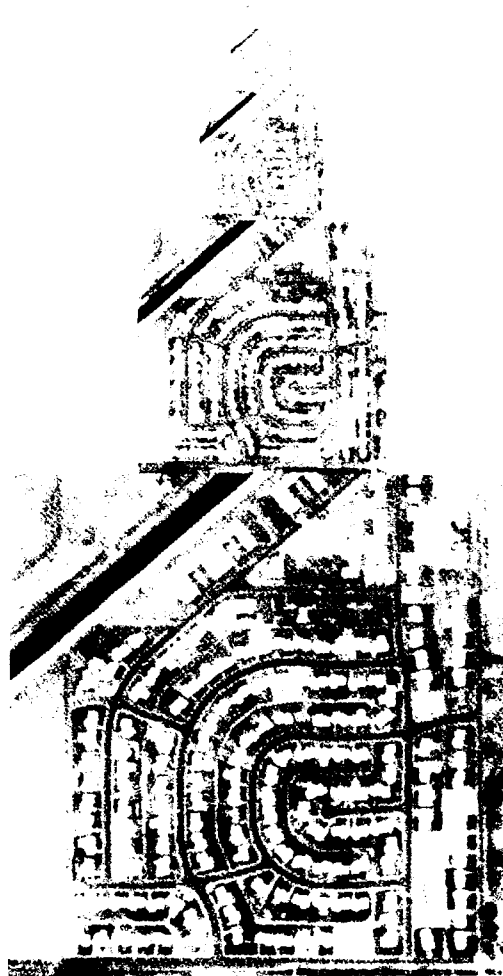


Figure 5.3: Gray level pyramid of PHOENIX_2A (levels 8 to 0)

5.1.2 Projection of an Image Point

For further steps of the image preprocessing it is of interest to project an image point $s_{i,j}^{(k)}$ of level k into the level $k + \iota$, with $\iota \in \{0, 1, \dots, (N - k)\}$. When projecting such a point from level k into level $k + \iota$, another pyramid is formed. The quadratic base of it is called *macro pixel* M . The size $2^i 2^i$ of the macro pixel M depends on the difference between the spot level k , where the point to be projected is located, and the projection level $k + \iota$.

5.2 Contrast Pyramids

Features are locally extracted from each image point of the gray level pyramid by gradient and curvature filtering. This is done by means of difference and Laplacian like operators. Further image analysis is based on the resulting filtered images.

The gradient of an image function s is given by

$$\text{grad } s(x, y) = \begin{pmatrix} \frac{\partial s(x, y)}{\partial x} \\ \frac{\partial s(x, y)}{\partial y} \end{pmatrix} = \begin{pmatrix} \lim_{\Delta x \rightarrow 0} \frac{s(x + \Delta x, y) - s(x, y)}{\Delta x} \\ \lim_{\Delta y \rightarrow 0} \frac{s(x, y + \Delta y) - s(x, y)}{\Delta y} \end{pmatrix}$$

the direction and amount of which are:

$$\varphi(\text{grad } s(x, y)) = \arctan \frac{\frac{\partial s(x, y)}{\partial y}}{\frac{\partial s(x, y)}{\partial x}}$$

$$|\text{grad } s(x, y)| = \sqrt{\left(\frac{\partial s(x, y)}{\partial x}\right)^2 + \left(\frac{\partial s(x, y)}{\partial y}\right)^2}$$

The Laplacian of an image function s is given by

$$\nabla^2 s(x, y) = \left(\frac{\partial^2 s(x, y)}{\partial x^2} + \frac{\partial^2 s(x, y)}{\partial y^2} \right)$$

In the following two sections two simple operators are introduced approximating gradient and curvature filtering from the gray value pyramid.

5.2.1 Detection of Edge Elements

In order to detect edge elements from local differences of intensity in the image function s , the Sobel operator $G = (G_x, G_y)$ is used:

$$G_x = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, \quad G_y = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

The image function $s(x, y)$ is transformed into the functions $g_x(x, y)$ and $g_y(x, y)$ by convolving s with the masks G_x , resp. G_y of the Sobel operator:

$$g_x(x, y) = (s(x+1, y-1) + 2s(x+1, y) + s(x+1, y+1)) - \\ (s(x-1, y-1) + 2s(x-1, y) + s(x-1, y+1))$$

$$g_y(x, y) = (s(x-1, y+1) + 2s(x, y+1) + s(x+1, y+1)) - \\ (s(x-1, y-1) + 2s(x, y-1) + s(x+1, y-1))$$

Then the absolute values $g(x, y)$ and the directions $\varphi(x, y)$ of the gradients are calculated from g_x and g_y :

$$g(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2}$$

$$\varphi(x, y) = \arctan \frac{g_y(x, y)}{g_x(x, y)}$$

The directions of the gradient $\{0, \dots, 2\pi\}$ represent the vectorized orientations of an edge element after a positive rotation of $\pi/2$ of gradients. The orientations of an edge element can also be transformed into $\{0, \dots, \pi\}$ with included notation of its positive or negative contrast step. The gradients are transformed into vectorized orientations of an edge element and quantized into the discrete values $\{0, \dots, 7\}$ representing the directions of the N_8 neighborhood (see Figure 5.8a). After calculation of g and φ their values are entered into the absolute value pyramid or direction pyramid, respectively. Both pyramids are used for the extraction procedure of edge segments (see section 5.3).

The absolute value and the direction of the gradients calculated from the Sobel filtered contrast image PHOENIX_2A1 in best resolution (structural level 9) are shown in Figure 5.4 and Figure 5.5. In order to display the results, the gray values are enhanced to $\{0, \dots, 255\}$. The other levels 8 to 0 of the absolute value and direction pyramid show similar results and are not given here.

Finally it is to be mentioned that instead of the Sobel operator also other edge detecting operators could be used.

5.2.2 Detection of Spots

In order to detect spots from local differences of intensity in the image function s , the spot operator U for curvature is used:

$$U = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

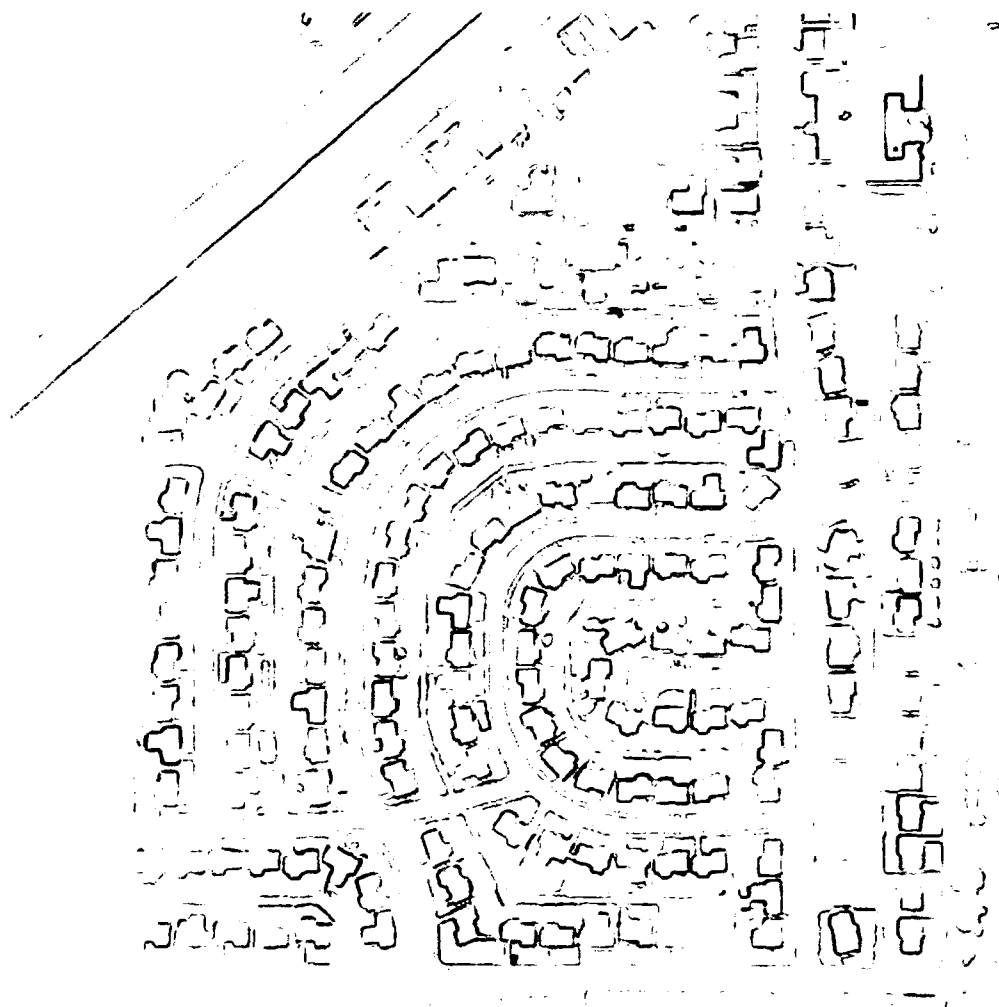


Figure 5.4: Absolute value of gradients of PHOENIX.2A1
in best resolution (level 9)
(absolute values $\{0, \dots, g_{\max}\}$ are enhanced to the gray values $\{0, \dots, 255\}$)

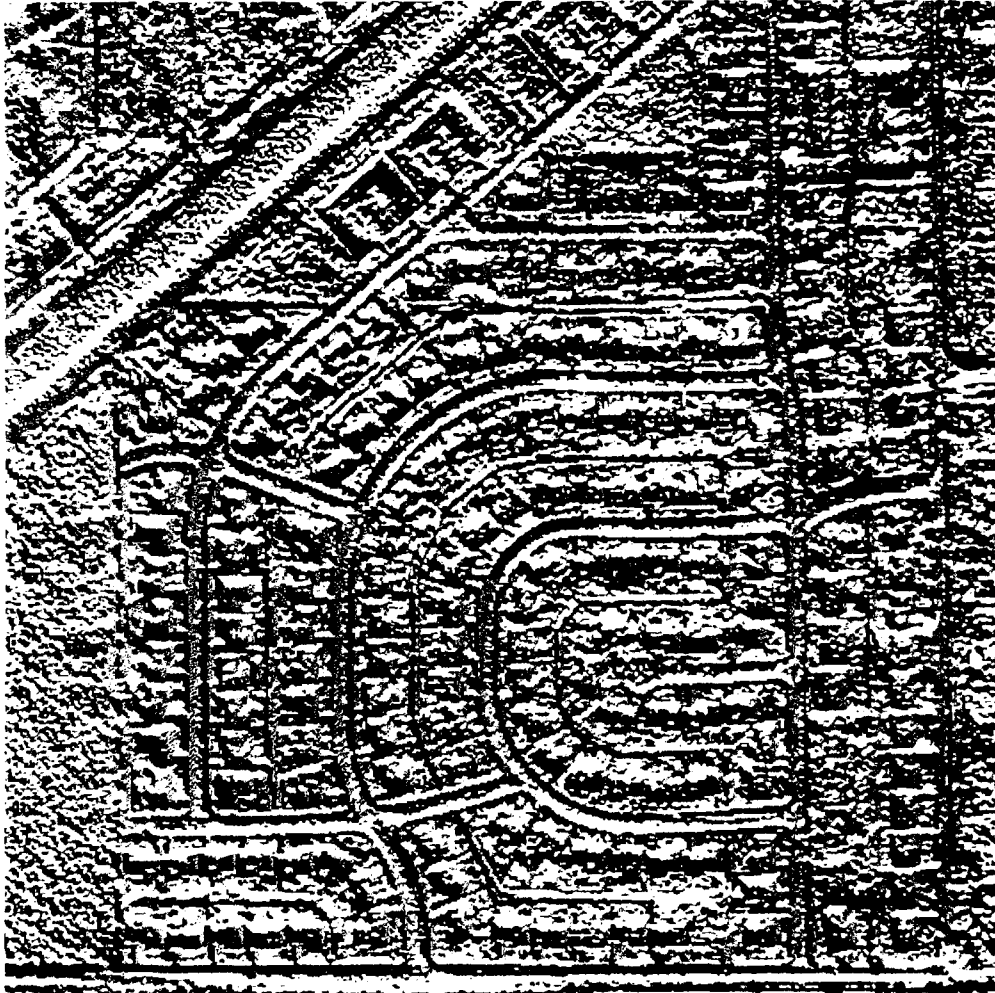


Figure 5.5: Direction of gradients of PHOENIX 2A1 in best resolution (level 9)
(directions $\{0, \dots, 2\pi\}$ are transformed to the gray values $\{0, \dots, 255\}$)

The image function $s(x, y)$ is transformed into the function $u(x, y)$ by convolving s with the mask U of the spot operator:

$$u(x, y) = \begin{array}{rrr} -s(x-1, y-1) & -s(x-1, y) & -s(x-1, y+1) \\ -s(x, y-1) & +8s(x, y) & -s(x, y+1) \\ -s(x+1, y-1) & -s(x+1, y) & -s(x+1, y+1) \end{array}$$

In order to receive shape features from local differences of intensity in the image function s , the operator $V = (V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7)$ is used beside U :

$$V_j = \begin{pmatrix} -\delta_{5,j} & -\delta_{6,j} & -\delta_{7,j} \\ -\delta_{4,j} & 1 & -\delta_{0,j} \\ -\delta_{3,j} & -\delta_{2,j} & -\delta_{1,j} \end{pmatrix} \quad \text{with} \quad \delta_{ij} = \begin{cases} 1 & \text{for } j = i \\ 0 & \text{for } j \neq i \end{cases}$$

The image function $s(x, y)$ is transformed into the functions $v_i(x, y)$ by convolving s with the masks V_j of the shape feature operator V :

$$v_i(x, y) = \begin{array}{rrr} -\delta_{5,i}s(x-1, y-1) & -\delta_{6,i}s(x-1, y) & -\delta_{7,i}s(x-1, y+1) \\ -\delta_{4,i}s(x, y-1) & +s(x, y) & -\delta_{0,i}s(x, y+1) \\ -\delta_{3,i}s(x+1, y-1) & -\delta_{2,i}s(x+1, y) & -\delta_{1,i}s(x+1, y+1) \end{array}$$

A useful shape measurement is the so-called surroundedness $v(x, y)$ featuring spots. This feature denotes the number of image elements in the N_8 neighborhood of a spot which are brighter or darker than the dark, resp. bright contrasting spot:

$$v(x, y) = \frac{1}{2} \sum_{j=1}^8 \text{sgn } v_j(x, y) (\text{sgn } u(x, y) + \text{sgn } v_j(x, y))$$

The next two figures show the results from filtering the images of the gray value pyramid (structural levels 8 to 0) with both the spot detection and spot feature operator. Figure 5.6 presents the results from spot detection. The brighter the displayed values from bright spots and the darker from dark spots are, the greater are the features *curvature* imaged to the values 128 to 255 or 127 to 0, respectively, whereby the value between 127 and 128 represents zero curvature. Figure 5.7 shows the surroundedness of spots from the N_8 neighborhood. The brighter the displayed values from the bright and dark spots are, the greater are the features *surroundedness* imaged to the values 0 to 255. Both pyramids are used for the extraction procedure of area like segments (see section 5.4).

5.3 Edge Oriented Segmentation

The edge oriented segmentation procedure is based on edge tracking. The tracking is performed by gradient controlled following of edge elements. Starting points for edge tracking are such image points in the pyramid where a high

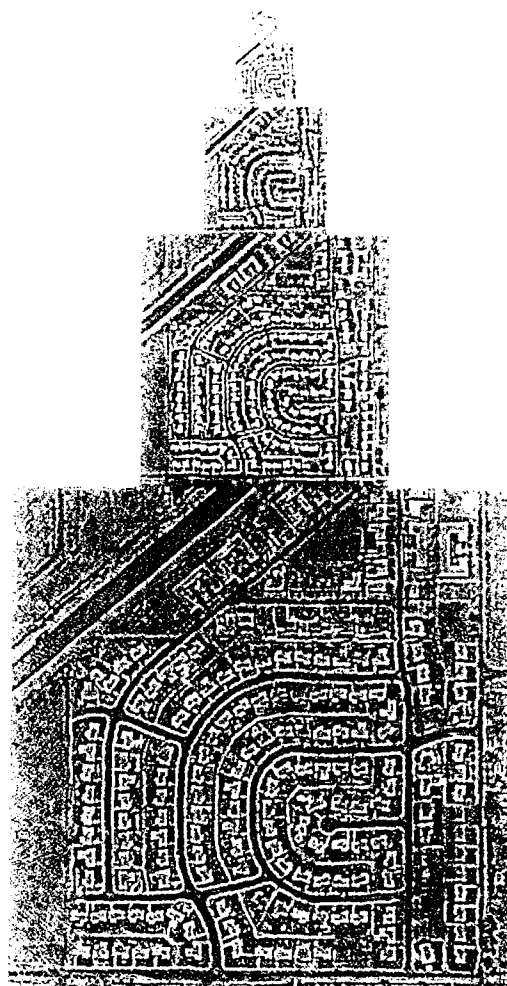


Figure 5.6: Spot curvature pyramid of PHOENIX_2A1 (levels 8 to 0)
(curvatures $\{-|u|_{\max} \dots, 0, \dots, |u|_{\max}\}$ are enhanced to
the gray values $\{0, \dots, 255\}$)

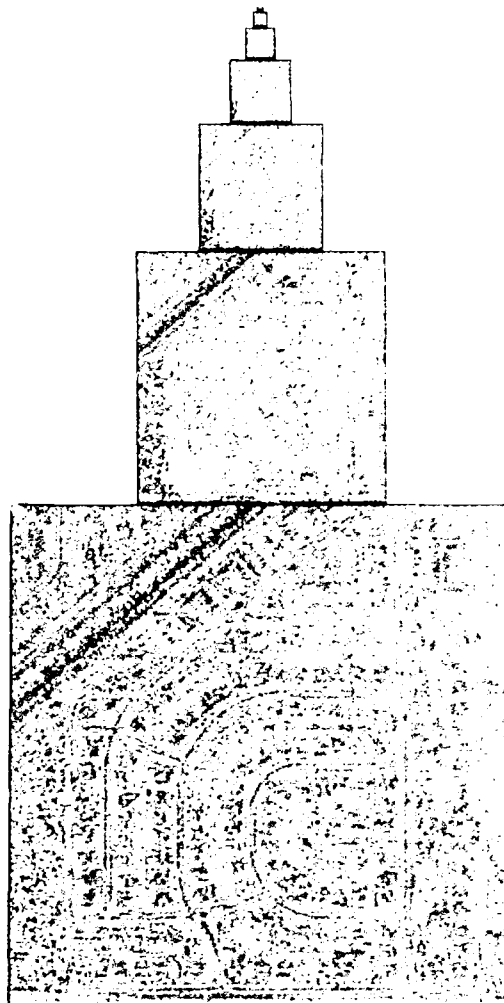


Figure 5.7: Spot surroundedness pyramid of PHOENIX.2A1 (levels 8 to 0) (features $\{0, \dots, 8\}$ for surroundedness are enhanced to the gray values $\{0, \dots, 255\}$)

absolute value of gradients are found after having used the edge operator. Contrary to other edge tracking procedures described in the literature (see, for example, [1,7,13]), it is not aimed at processing the gradient image area totally or in parallel. Instead of that it is tried to get connected edges sequentially by following the orientation of edge elements starting with one of the *most reliable* edge elements. The edge segmentation algorithm is based on the following steps explained in detail in the next subsections:

1. preparation of the starting point list for edge segmentation
2. extraction of edge segments and features by tracking of edge elements
3. assessment of edge elements and segments
4. approximation of edge segments by polygons
5. generation of symbolic description of edge segments

5.3.1 Starting Point List for Edge Segmentation

The coordinates of pixels representing edge elements, which in each resolution level are sorted in descending order corresponding to the magnitude of absolute value of their gradients, are compiled in a starting point list. This list then is read from that end with the highest absolute value. Thereby it is guaranteed that the edge extraction always begins with the most reliable edge element. The procedure stops when a given threshold of the magnitude of absolute value is reached.

5.3.2 Tracking of Edge Elements and Feature Extraction

The edge tracking starts with the track parameter $t = 0$ at the actual starting point $p_0(x_0, y_0)$ contained in the starting point list S . After tracking of edge elements $p_0(x_t, y_t)$ the segment is described by a string x_t, y_t of coordinates of the actual pyramidal level. Possible candidates p_i to be added to each actual pixel p_0 of an edge element at the position x_t, y_t are contained in a subset of pixels in the N_8 neighborhood of p_0 . The pixel p_i can be addressed by the direction Ψ . Figure 5.8 shows the relation between the coding of the directions Ψ and the N_8 neighborhood. That scheme is also used for the coding of the gradient of an actual point p_0 as edge direction Φ .

The pixels p_i connected to the actual pixel p_0 for edge segmentation are distinguished by successors p_i^S and predecessors p_i^P . This means that it is tried to track the edge to both ends with $t > 0$ for (S) and $t < 0$ for (P) beginning with the starting point $t = 0$. Connecting N^- preceding and N^+ succeeding pixels the string $(x_{N^-}, y_{N^-}; \dots; x_{-1}, y_{-1}; x_0, y_0; x_1, y_1; \dots; x_{N^+}, y_{N^+})$ of coordinates of edge elements is generated. For example, Figure 5.9 shows the selected candidates for edge tracking related to an actual point p_0 with an arbitrary gradient coded as edge direction $\Phi = 6$ (symbolized by \uparrow).

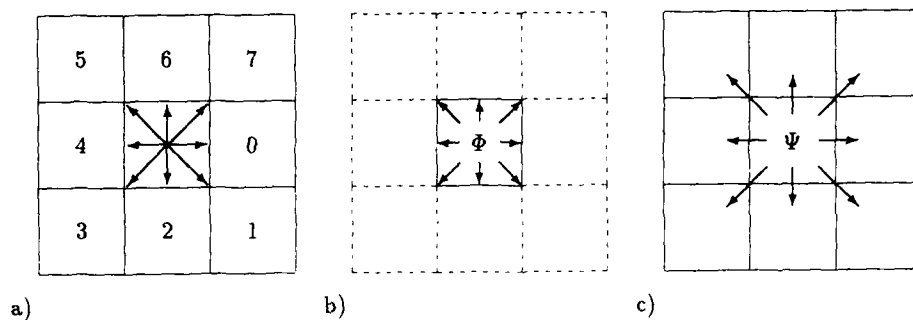


Figure 5.8: Coding scheme a) of directions Φ or Ψ for b) edges, resp. c) neighbors

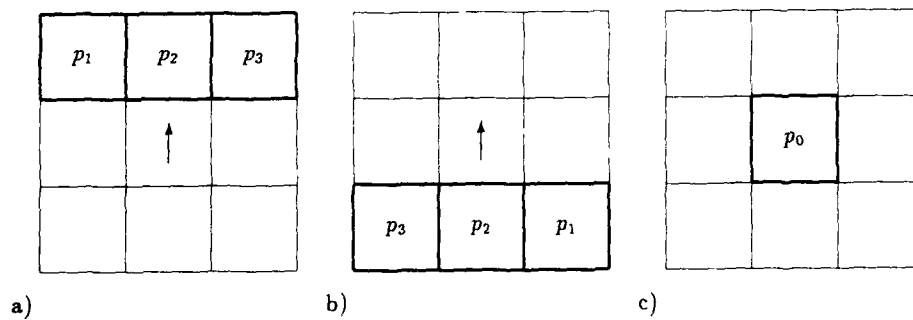


Figure 5.9: Edge tracking candidates for a) successors, resp. b) predecessors with respect to an actual point p_0 with a gradient coded as edge direction $\Phi = 6$

The successors, resp. predecessors are addressed by the direction Ψ derived from the edge direction Φ of the actual point p_0 . In order to select the succeeding p^S , resp. the preceding p^P candidates for a point p_0 , the following rules to compute the direction Ψ in p_0 from Φ are used, where $i = 1$ is the left, $i = 2$ the central, and $i = 3$ the right successor, resp. vice versa for the predecessor:

$$\Psi(p_i^S) = \begin{cases} (\Phi(p_0) - 1) \bmod 8 & \text{for } i = 1 \\ \Phi(p_0) & \text{for } i = 2 \\ (\Phi(p_0) + 1) \bmod 8 & \text{for } i = 3 \end{cases}$$

$$\Psi(p_i^P) = \begin{cases} (\Phi(p_0) - 5) \bmod 8 & \text{for } i = 1 \\ (\Phi(p_0) - 4) \bmod 8 & \text{for } i = 2 \\ (\Phi(p_0) - 3) \bmod 8 & \text{for } i = 3 \end{cases}$$

The surrounding pixels p_i of the actual edge element p_0 are examined in order to choose the *best successor*, resp. the *best predecessor* among the candidates by means of criteria to be defined in the next subsection. Taking the starting point p_0 , it is tried to track the edge elements in both directions until both end points x_{N^+}, y_{N^+} and x_{N^-}, y_{N^-} are reached. Considering a tracking in the direction of successors those end points represent the starting point (for N^-) and the end point (for N^+) of edge segments. After tracking of edge elements radiometric and geometric features of edge segments are generated.

5.3.3 Assessment of Edge Elements and Segments

Assessment of edge segmentation is done by two steps. First, edge elements which are candidates for edge tracking are selected from the N_8 neighborhood at both ends of a segment generated so far. These elements have to meet some minimum requirements for edge tracking, and, if these qualifications are not satisfied, the edge segment with the element concerned is not continued at this end. Each edge element which can be considered as a candidate for edge tracking is assessed with respect to direction and absolute value of the gradient for compatibility with those values of the already existing edge element belonging to the segment. After that a total assessment of all candidates is made, and the candidate with the highest value is taken to be added to the edge segment as next edge element.

Second, if the edge segment cannot be continued at both ends furthermore, a final assessment of the edge segment is done. If this final assessment does not meet the minimum requirements for a segment, the complete edge segment is omitted.

Compatibility of Directions

Testing the compatibility of directions of edge elements means to compare the gradient direction, resp. edge direction of a candidate pixel p_i with that of the

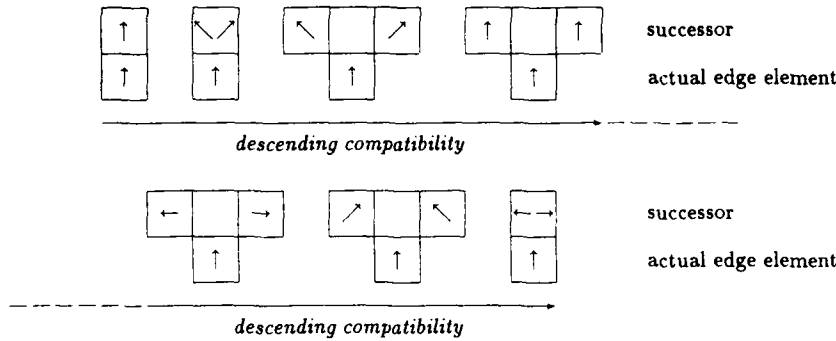


Figure 5.10: Ranking of compatible successors with respect to edge directions

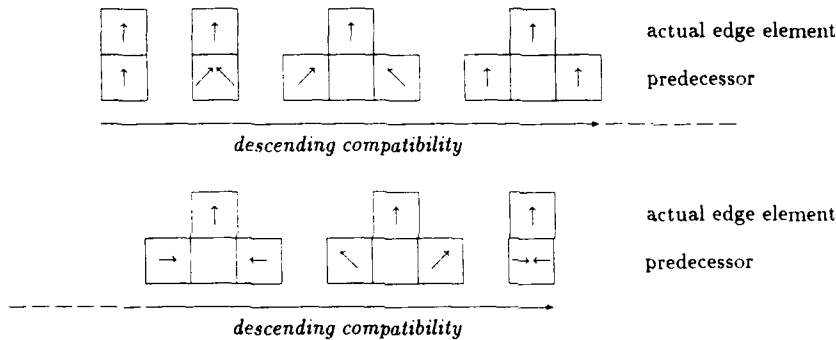


Figure 5.11: Ranking of compatible predecessors with respect to edge directions

actual point p_0 of an edge segment already existing. Which directions between neighboring candidates and actual edge element are compatible and how they are assessed, is shown exemplarily in Figures 5.10 and 5.11 for an edge element with an arbitrary edge direction 6 (symbolized by \uparrow). If, for example, the candidates p_1 and p_3 have the edge direction 5 (symbolized by \searrow), then p_1 gets a higher assessment than p_3 . If the difference of directions between a neighboring candidate and the actual edge element is greater than 2, then the candidate gets the assessment 0 and is not added to the edge segment.

Compatibility of Absolute Values

Concerning the absolute values of gradients the compatibility of a candidate pixel p_i depends on the difference between its own value and the values of that

points of the edge segment already existing. In order to examine this compatibility, a linear regression is made with the absolute values of the gradients belonging to the edge segment existing so far. An extrapolation by means of the computed regression line gives a prediction g' for the absolute gradient value of the candidate to be tested. The smaller the difference between the values g of the candidate and the predicted value g' , the higher is the assessment. If its amount is higher than a given threshold the candidate is assessed to 0 and not added to the edge segment.

Total Compatibility

The total assessment for each of the candidate is a combination of both assessments for direction and absolute value of gradients. The candidate with the highest assessment is added to the edge segment. If the assessment of all candidates is zero no pixel from the N_8 neighborhood is connected to the actual point and the edge segment is not continued at this end.

Final Compatibility

In case that the continuation of an edge is no more possible, the total length of the complete edge segment and its mean absolute value from linear regression is assessed. The entire edge segment is rejected, if a given minimum of mean absolute value and a minimum length are not reached. Figure 5.12 shows the result of edge oriented segmentation by tracking of edge elements. Different edges are displayed by differing colors. Only the best resolution level 9 of the edge pyramid is shown. The results from the other resolution levels 8 to 0 are comparable to that of level 9.

5.3.4 Approximation of Edge Segments by Polygons

After the edge segments have been found, they are analyzed with respect to global segment features and are approximated by polygons the vertices of which are only stored. Then the original strings of pixel coordinates from the edge elements of the segments are cancelled. Each detected edge segment is described by the edge parts between its vertices. The edge parts are received by connecting selected pixels (sample points) of the original edge segment with straight lines. The sample points of the original edge segment represent the vertices of the approximated edge segment (polygon). The first two vertices are the first and the last point of the original edge segment (end points). The next vertex is that point lying on the original edge segment which has the largest distance d to the line drawn through the starting point and the end point. This vertex divides the original edge segment into two edge parts. Then a new vertex is determined for each edge part until the maximum distance d of each edge part to the original edge segment is smaller than a given threshold.

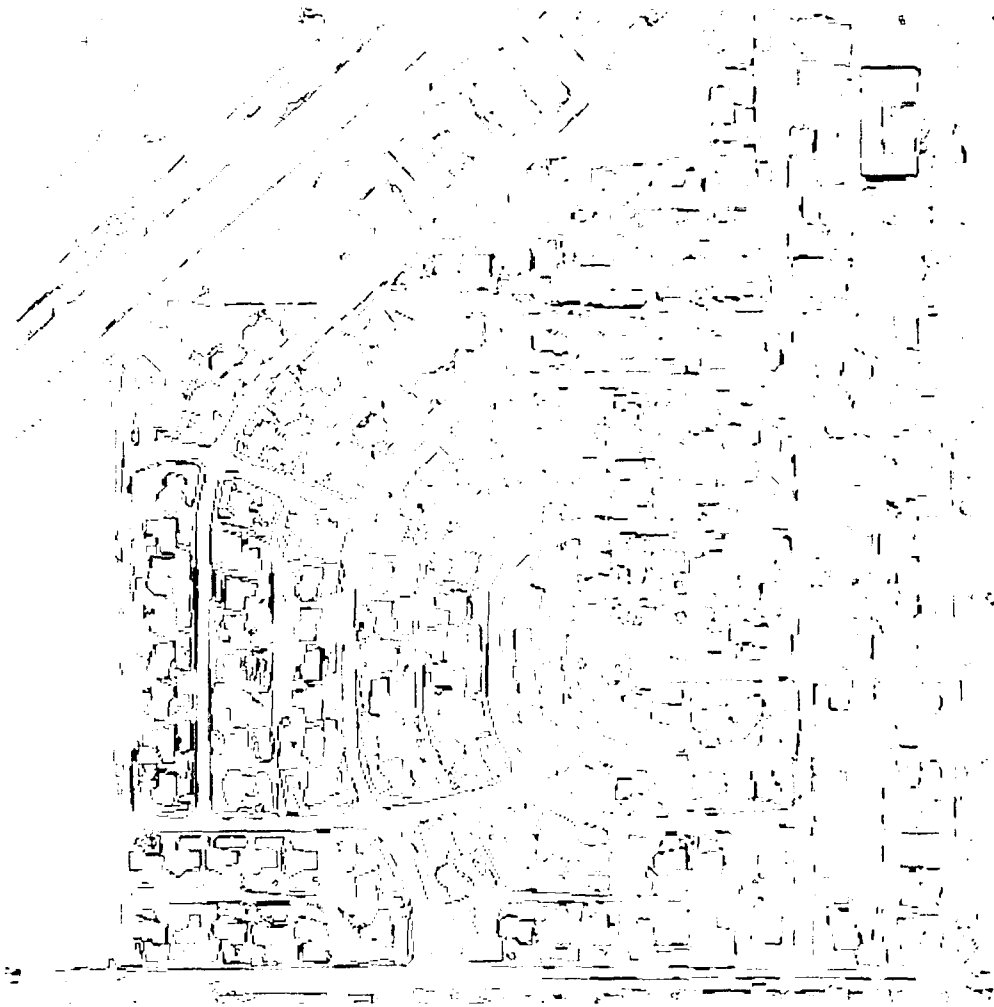


Figure 5.12: Edge oriented segmentation of PHOENIX.2A1 in best resolution (level 9) (different edges are displayed by differing colors)

5.3.5 Symbolic Description of Edge Segments

The iconic results of edge segmentation are transferred to symbolic descriptions for further analyzation. In order to present edge segments symbolically, features (symbols) are generated from each original and approximated edge segment their attributes of which are inserted into two lists. Three different types of attributes are allocated to both lists:

1. key data
2. radiometric features
3. geometric features

In order to link global information of segments and details of edge parts additional attributes are required for the first list:

4. directory data

Figure 5.13 shows the data structure of both lists. The first list (symbol0_list) contains the global features of a nonapproximated edge segment and a pointer to the second list (symbol1_list), in which the corresponding data for the edge parts after edge approximation are contained. The term *contrast* used in the lists means the absolute value of the gradient and the term *slope* refers to the linear regression applied to the contrast.

5.4 Area Oriented Segmentation

The procedure of the area oriented segmentation can be characterized as a *local* binarization method. Homogeneous, contrasting areas of different size in a gray level image are presented in appropriate pyramidal resolution levels as bright, resp. dark contrasting spots. Similar to edge tracking the sequential procedure for area oriented segmentation is driven by the best assessed spots. The segmentation method produces the contours of bright, resp. dark contrasting areas. The segmentation algorithm for areas is based on the following steps explained in detail in the next subsections:

1. preparation of the starting point list for area segmentation
2. projection of spots into levels of higher resolution
3. generation of area segments by binarization
4. extraction of area segments and features by tracking of contour elements
5. assessment of area segments
6. approximation of area segments by polygons
7. generation of symbolic descriptions of area segments

1	key data	edge segment identifier
2		confidence (final assessment)
3		pyramidal level (resolution)
4	radiometric features	slope of contrast
5		min. contrast
6		max. contrast
7		mean contrast
8	geometric features	σ - contrast
9		x.start (end point)
10		y.start (end point)
11		x.end (end point)
12		y.end (end point)
13		encasing rectangle x.min
14		encasing rectangle y.min
15		encasing rectangle x.max
16		encasing rectangle y.max
17	directory data	length of edge (pixels)
18		number of edge parts
19		pointer to symbol1_list

symbol0_list

1	key data	edge part identifier
2		edge segment identifier
3		confidence
4	radiometric features	slope of contrast
5		min. contrast
6		max. contrast
7		mean contrast
8	geometric features	σ - contrast
9		length of edge part
10		direction of edge part
11		x.start (vertex)
12		y.start (vertex)
13		x.end (vertex)
14		y.end (vertex)

symbol1_list

Figure 5.13: Symbolic description of edge segments

5.4.1 Starting Point List for Area Segments

For each resolution level of the spot pyramid the spots are sorted by two steps in descending order with respect to their values of curvature and surroundedness. Their coordinates are compiled in a starting point list. The most reliable spot is that with the greatest negative value (for dark spots), resp. the greatest positive value (for bright spots) of curvature, and with the surroundedness of 8. The spot operator for curvature introduced in subsection 5.2.2 provides values lying between -2040 (dark spots) and $+2040$ (bright spots). These values u are normalized to u' so that $0 \leq u' \leq 255$, i.e. the best assessment for a dark spot is 0, and the best assessment for a bright spot is 255.

For each resolution level first the spots are sorted in descending order according to their surroundedness. Second, the spots are sorted in descending order for each class of surroundedness according to their curvature. Then the starting point list is read *from the bottom* (dark spots to segment dark areas), resp. *from the top* (bright spots to segment bright areas) until a certain threshold for curvature and surroundedness is reached. Thus it is guaranteed that only the most reliable spots are selected for area segmentation. Regarding thresholds experience has shown that values for the surroundedness should lie between 8 and 4, and the values for the curvature below 126 for dark spots and above 129 for bright spots.

5.4.2 Spot Projection into Levels of Higher Resolution

The spots in the resolution level k of the spot pyramid are cues for the detection of homogeneous areas in the levels $k + \iota$ of higher resolution. Furthermore the pyramidal structure gives information about the possible size of the expected areas depending on the difference ι between the level, in which the spot is detected (spot level SL), and that level into which the spot is projected (projection level PL). Figure 5.14 shows schematically the projection of spots from the spot level into the projection level.

A spot at the resolution level k determines an area with the size of $2^\iota \times 2^\iota$ pixels at the resolution level $k + \iota$. This area is the so-called *macro pixel* M the length of which amounts to 2^ι . The border coordinates $m(M), n(M)$ of the macro pixel M depend on the coordinates m, n of the spot at level k and on the difference ι between projection and spot level:

$$m_{start}(M) = (m - 1)2^\iota + 1$$

$$n_{start}(M) = (n - 1)2^\iota + 1$$

$$\text{and} \quad m_{end}(M) = m2^\iota$$

$$n_{end}(M) = n2^\iota$$

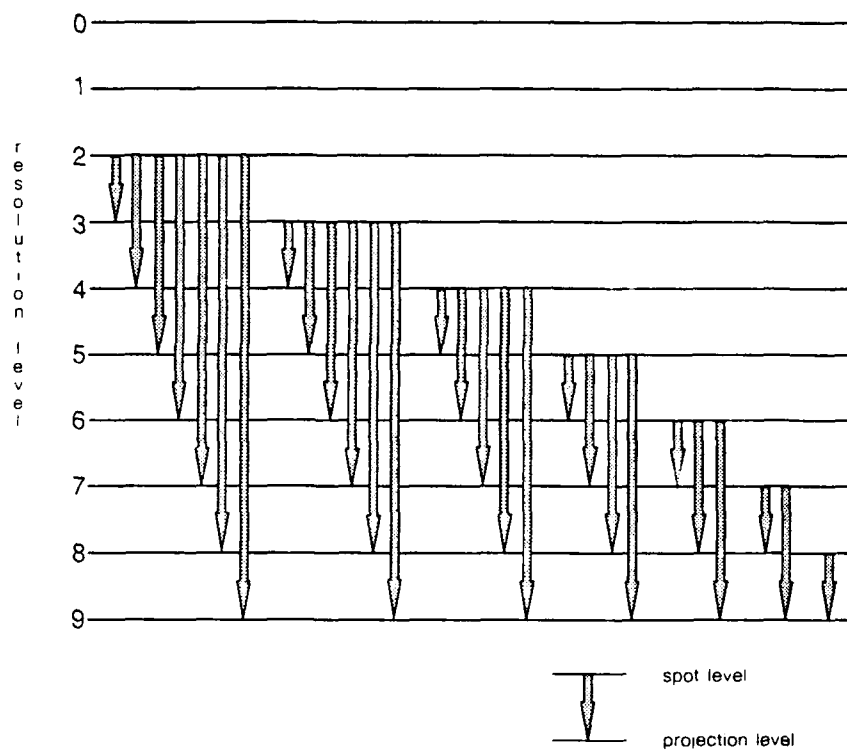


Figure 5.14: Projection of spots into levels of higher resolution

The segmentation area is defined by the so-called *patch* P , the size of which is 3×3 macro pixels. The edge length of a patch amounts to $3 \cdot 2^i$. The border coordinates $m(P), n(P)$ of the patch P depend on the coordinates m, n of the spot at level k and on the difference i between projection and spot level, thus on the coordinates $m(M), n(M)$ of the macro pixel M :

$$m_{start}(P) = 2m_{start}(M) - m_{end}(M) - 1$$

$$n_{start}(P) = 2n_{start}(M) - n_{end}(M) - 1$$

$$\text{and } m_{end}(P) = 2m_{end}(M) - m_{start}(M) + 1$$

$$n_{end}(P) = 2n_{end}(M) - n_{start}(M) + 1$$

The minimum and the maximum size of the area segment to be detected is determined by the difference i of projection level and spot level. The approximate minimum size is as large as the macro pixel, whereas the maximum size can be as large as the patch in which the segmentation is performed.

5.4.3 Area Segmentation by Binarization

The choice of the binarization threshold within the patch is of decisive importance for the segmentation result. This local threshold is separately calculated for each selected spot candidate, and is applied to each structural level of the gray value pyramid with a higher resolution than the spot level (see Figure 5.14). The binarization threshold depends on the gray values of the macro pixel in the projection level of the gray level pyramid concerned (see section 4.1.1). The binarization thresholds Θ_b for bright and Θ_d for dark spots are:

$$\Theta_b = \bar{s} - f\sigma_s$$

$$\Theta_d = \bar{s} + f\sigma_s$$

Thereby \bar{s} is the unweighted medium gray value of the macro pixel and σ_s is the standard deviation of the gray values. The factor f allows to raise or to lower the effect of area binarization. An increase of the factor leads to a decrease, resp. increase of the binarization threshold for bright, resp. dark spots, which leads to the detection of more and larger area segments. Concerning bright, resp. dark spots having gray values being greater, resp. less than the threshold, then connecting pixels are merged to a bright, resp. dark area. The connecting pixels are labelled by 1 or any other positive numeral, the background by 0.

5.4.4 Tracking of Contour Elements and Feature Extraction

In order to determine the contour of an area segment, a contour tracking algorithm given in [14] is used. The aim is to encircle the connected labelled points of the segment first and then to fill possible gaps within the area. The contour tracking algorithm can be described as follows: The upper left 1-labelled border point within the patch is chosen as a starting point S . Starting with this point the contour points are surrounded by a 2×2 mask and new labelled points depending on the change of the direction of the inner, resp. outer contour points. The elements of the mask are denoted by P, Q, V, U with an arrow between P and Q pointing to the centre of the mask.

At the beginning of the contour tracking the element P of the mask is fixed to the starting point S of the segment and Q, V, U lie outside the area segment. Then the mask is rotated and shifted such that always at least P of the four elements of the mask lies on the inner contour of the area segment. The mask is rotated around P or Q , and shifted in the direction of the arrow substituting the old V, U by the new Q, P . The new orientation or position of the mask is always determined by substituting only the two elements P and Q . Figure 5.15 shows the transfer table

$$(P, Q, V, U)_i \xrightarrow{L(U)_i, L(V)_i} (P, Q)_{i+1}$$

depending on the labels $\{1, 0\}$ of area points of the corresponding mask elements U, V denoted by $L(U), L(V)$. The contour tracking has finished if the starting point S is reached again. Between the marked points the area segment is filled up with the given 1-label. Thereby 0-distortions resulting from the binarization procedure are eliminated. After tracking of the contour elements radiometric and geometric features of the area segment are generated.

5.4.5 Assessment of Area Segments

The area segments are assessed with respect to geometric and radiometric features. These features serve as thresholds in order to decide if a segment is to be accepted or has to be rejected. The following features are used for thresholding:

- area size
- compactness
- absolute values of contour gradients
- homogeneity of gray values within the area

As segmentation occurs only within the patch, the magnitude of area sizes to be obtained depends on the difference between projection and spot level. A

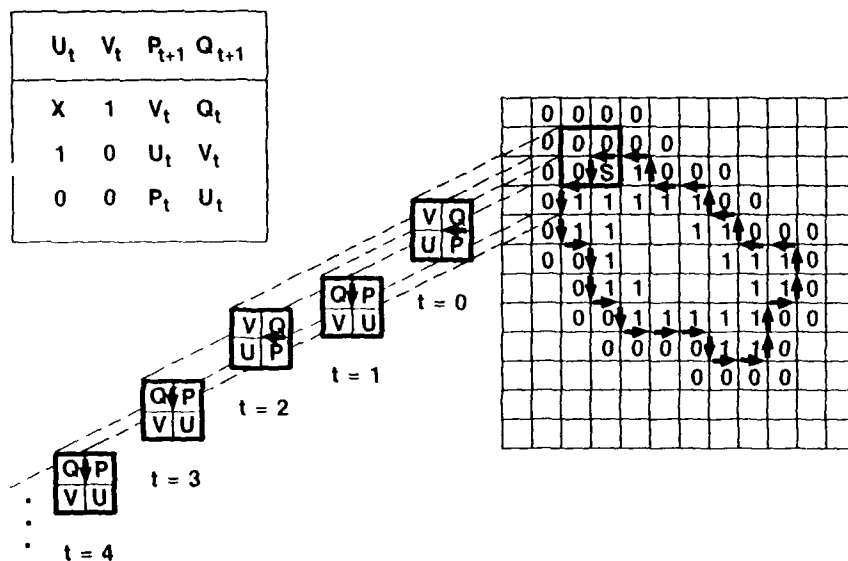


Figure 5.15: Contour tracking starting from point S ([14])

segment is then accepted, if its area A lies between $f_1 M \leq A \leq f_2 M$. Thereby M is the size of the macro pixel, f_1 and f_2 are factors with $0.0 \leq f_1 \leq 1.0$ and $1.0 \leq f_2 \leq 9.0$. In order to avoid finding too *fuzzy* segments, only segments above a minimum compactness are allowed. Segments with a mean absolute value of contour gradients below a given threshold are rejected as well as segments with a standard deviation of gray values being greater than a given threshold for low homogeneity.

A confidence measure for each accepted segment is calculated by the weighted sum of single confidence values. Those single values are the area size A , the compactness C , the mean absolute value of contour gradients K , and the standard deviation of the gray values Σ of the area segment, which are provided with eligible weighting factors g_i with the sum of them is equal to 1:

$$V = g_1 A + g_2 C + g_3 K + g_4 \Sigma \quad \text{with} \quad g_1 + g_2 + g_3 + g_4 = 1$$

5.4.6 Approximation of Area Segments by Polygons

The tracked contour lines of area segments are temporarily recorded as lists of coordinates for feature extraction. In order to reduce storage and computing time the area segments are approximated by closed polygons the vertices of which are stored permanently. The procedure for approximating area segments works similar to that for edge segments described in section 5.3.4. The detected contour is described by its contour parts. These parts are found by connecting sample points of the original contour with straight lines, the sample points becoming the vertices of the approximating polygon. The first both sample points are an arbitrarily selected contour point P_1 , e.g. the starting point S , and that point P_2 on the contour with the maximum distance to P_1 , resp. S . Both points are connected by a line which divides the original contour into two parts. The next two vertices are those two points of the contour which are most far away from this line. By this means the original contour line is replaced by four contour parts. This procedure is continued until the maximum distance of a line to the contour is smaller than a given threshold.

Segmentation results for two different local binarization thresholds with factors f chosen as $f = -0.2$, resp. $f = +0.2$ are shown in Figures 5.16 and 5.17 (bright areas), resp. in Figures 5.18 and 5.19 (dark areas). The bright and dark areas are segmented from the best resolution level 9 of the pyramid with the spots as cues projected from level 6 to level 9. The different areas are displayed by different colors in order to distinguish the detected segments. The horizontal and vertical contour lines are due to the borders of the patches.

When regarding other spot and projection levels, smaller or larger area segments would be extracted by this method. For example, segmentation results for one local binarization threshold with factor f chosen as $f = -0.2$ are shown in Figure 5.20 (bright areas) and in Figure 5.21 (dark areas). The bright and dark areas are segmented from the best resolution level 9 of the pyramid with the spots as cues projected from level 5 to level 9.



Figure 5.16: Area oriented segmentation of PHOENIX.2A1
in best resolution (level 9)
(bright areas, binarization factor $f = -0.2$, projection from level 6 to level 9)

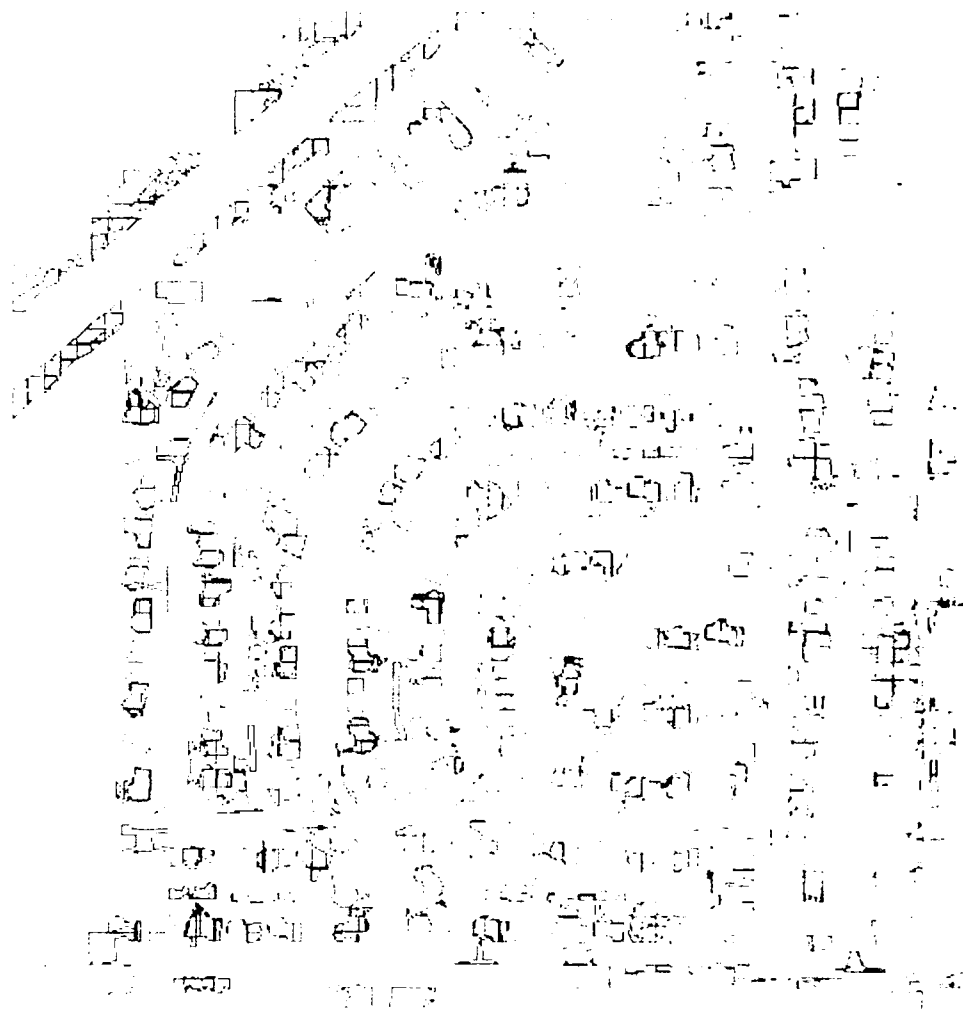


Figure 5.17: Area oriented segmentation of PHOENIX_2A1
in best resolution (level 9)
(bright areas, binarization factor $f = +0.2$, projection from level 6 to level 9)

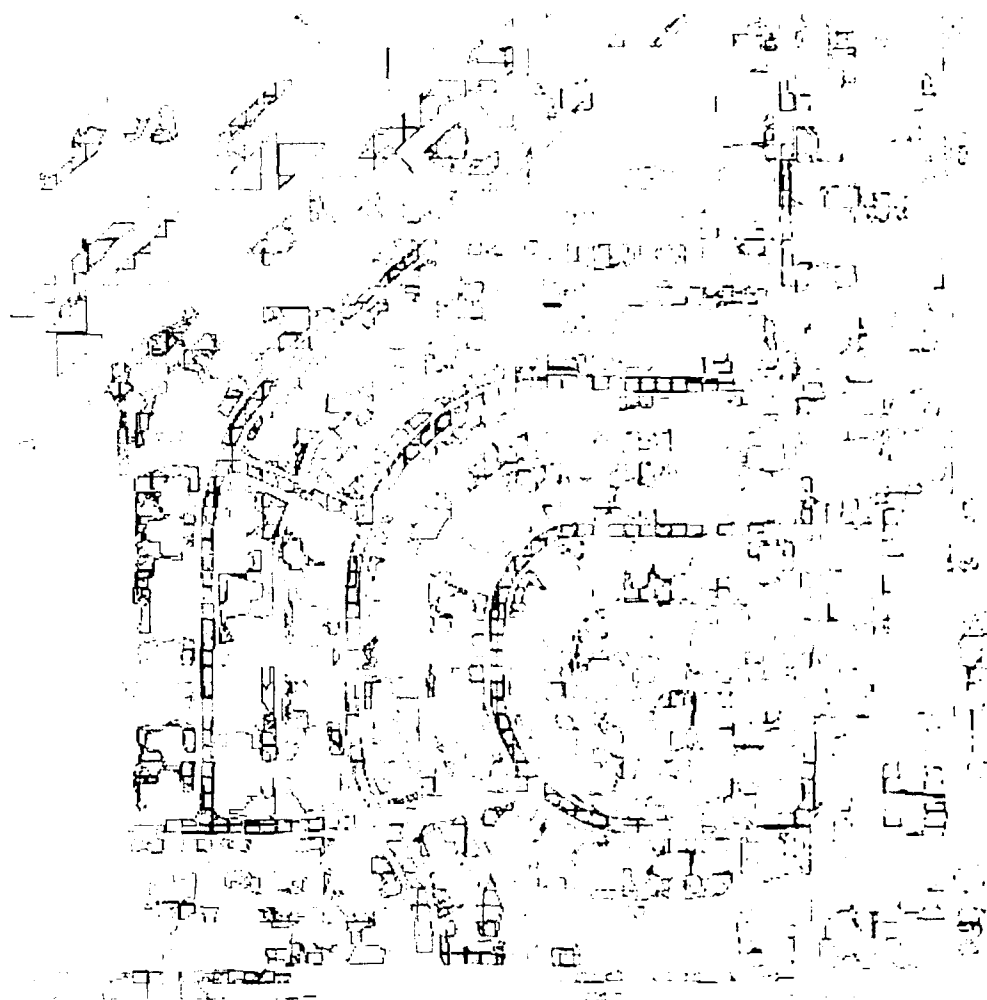


Figure 5.18: Area oriented segmentation of PHOENIX.2A1
in best resolution (level 9)
(dark areas, binarization factor $f = -0.2$, projection from level 6 to level 9)



Figure 5.19: Area oriented segmentation of PHOENIX.2A1
in best resolution (level 9)
(dark areas, binarization factor $f = +0.2$, projection from level 6 to level 9)

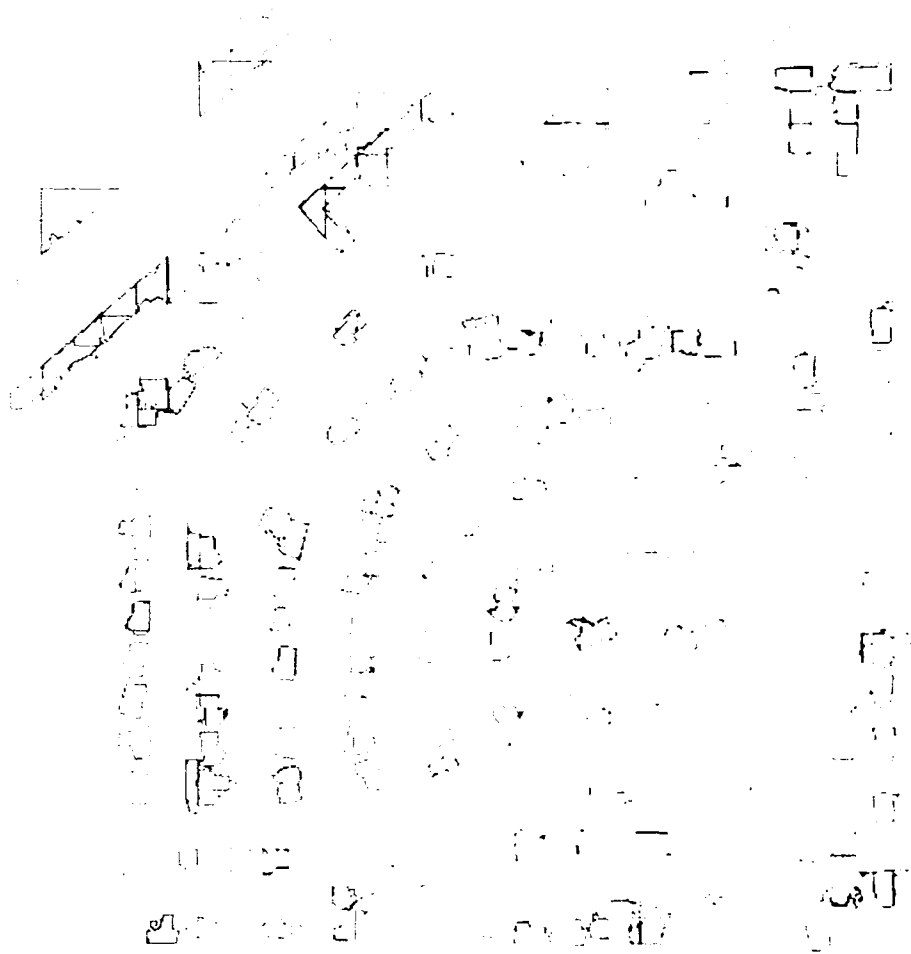


Figure 5.20: Area oriented segmentation of PHOENIX-2A1
 in best resolution (level 9)
 (bright areas, binarization factor $f = -0.2$, projection from level 5 to level 9)



Figure 5.21: Area oriented segmentation of PHOENIX_2A1
in best resolution (level 9)
(dark areas, binarization factor $f = -0.2$, projection from level 5 to level 9)

5.4.7 Symbolic Description of Area Segments

The iconic results of area segmentation are transferred to symbolic descriptions for further analyzation. In order to present area segments symbolically, features (symbols) are generated from each original and approximated area segment their attributes of which are inserted into two lists. Analog to the treatment of edge segments (see section 5.3.5) three different types of attributes are allocated to both lists:

1. key data
2. radiometric features
3. geometric features

In order to link global information of segments and details of contour parts additional attributes are required for the first list:

4. directory data

Figure 5.22 shows the data structure of both lists. The first list (symbol0.list) contains the global features of a nonapproximated area segment and a pointer to the second list (symbol1.list), in which the corresponding data for the contour parts after contour approximation are contained. The term *contrast* used in the lists means the absolute value of the gradient on the contour and the terms *brightness*, resp. *darkness* refer to the gray values of the area segment.

5.5 Results from Edge and Area Segmentation

The segmentation procedures introduced in this chapter are provided for the iconic preprocessing of symbolic blackboard oriented structural image analysis. They produce an enormous amount of edge and area segments in each resolution level of the gray value pyramid. Because of capacity it is necessary for further symbolic processing to reduce the large number of segments by selecting certain groups of segments as base objects according to the encountered task.

For the task of street extraction in urban imagery the edge elements in the best resolution level 9 should be selected as base elements for the prolongation of short lines to long lines. Concerning areas the dark, resp. bright area segments in the best resolution level 9 projected from spot level 6 should be selected as base elements for the enlargement of small areas to large areas.

1	key data	area identifier
2		confidence
3		spot level
4		projection level
5	radiometric features	type of contrast (dark, bright)
6		min. brightness/darkness
7		max. brightness/darkness
8		mean brightness/darkness
9		σ - brightness/darkness
10		min. contour contrast
11		max. contour contrast
12		mean contour contrast
13		σ - contour contrast
14	geometric features	area size
15		compactness
16		centre of gravity x-grav
17		centre of gravity y-grav
18		encasing rectangle x-min
19		encasing rectangle y-min
20		encasing rectangle x-max
21		encasing rectangle y-max
22		contour length (pixels)
23	directory data	number of edge parts
24		pointer to symbol1_list

symbol0_list

1	key data	contour part identifier
2		area identifier
3		confidence
4	radiometric features	behavior of contrast
5		min. contrast
6		max. contrast
7		mean contrast
8		σ - contrast
9	geometric features	length of contour part
10		direction of contour part
11		x_start
12		y_start
13		x_end
14		y_end

symbol1_list

Figure 5.22: Symbolic description of area segments

Chapter 6

Extended Production Model for Street Network Extraction

As shown in chapter 5, a huge amount of segmentation results is obtained from improved image preprocessing. Because of the limited running time of the project the results achieved, however, could not yet be used for structure oriented image analysis in general. In order to test the direct applicability of the original BPI model for line prolongation used in chapter 4. (see Figure 4.9), a section of 256×256 pixels from PHOENIX.2A1 was segmented into edges in the best resolution level 9. All edges received from improved preprocessing were equally assessed and stored in a blackboard as base objects *line*. In Figure 6.1 those base objects are plotted as thin lines. The base objects *line* were used to generate more complex partial objects *long line*, plotted as thick lines in Figure 6.1. The production of long lines showed that because of many gaps between single long lines, the target object *street network* would not be extracted unless a modification of the production *P01* is carried out. Furthermore the original model does not allow to construct bending lines, so that line prolongation and stripe composition would not be done on curves. Therefore most of the productions following *P01* also have to be modified.

The direct application of the original model to edge elements received from improved preprocessing has shown that the extraction of streets would not be successful. This is even more true when using edge and area segments together unless the model is substantially extended. Because of time a general modification of the model could not be realized. On the other hand edge oriented structural image analysis using the original model has been demonstrated in chapter 4. in principle. Therefore an extended model for street network extraction was developed using only the area segments as base objects. Furthermore the dark, resp. bright contrasting small areas segmented by projecting the spots from level 6 to gray value level 9 are used only.

In the first section of this chapter an extended model for the detection of intersections in urban areas is described that combines appropriate small areas to streets connected by their intersections. In the next section the set of productions required for connecting the partial objects is given. In the last sec-

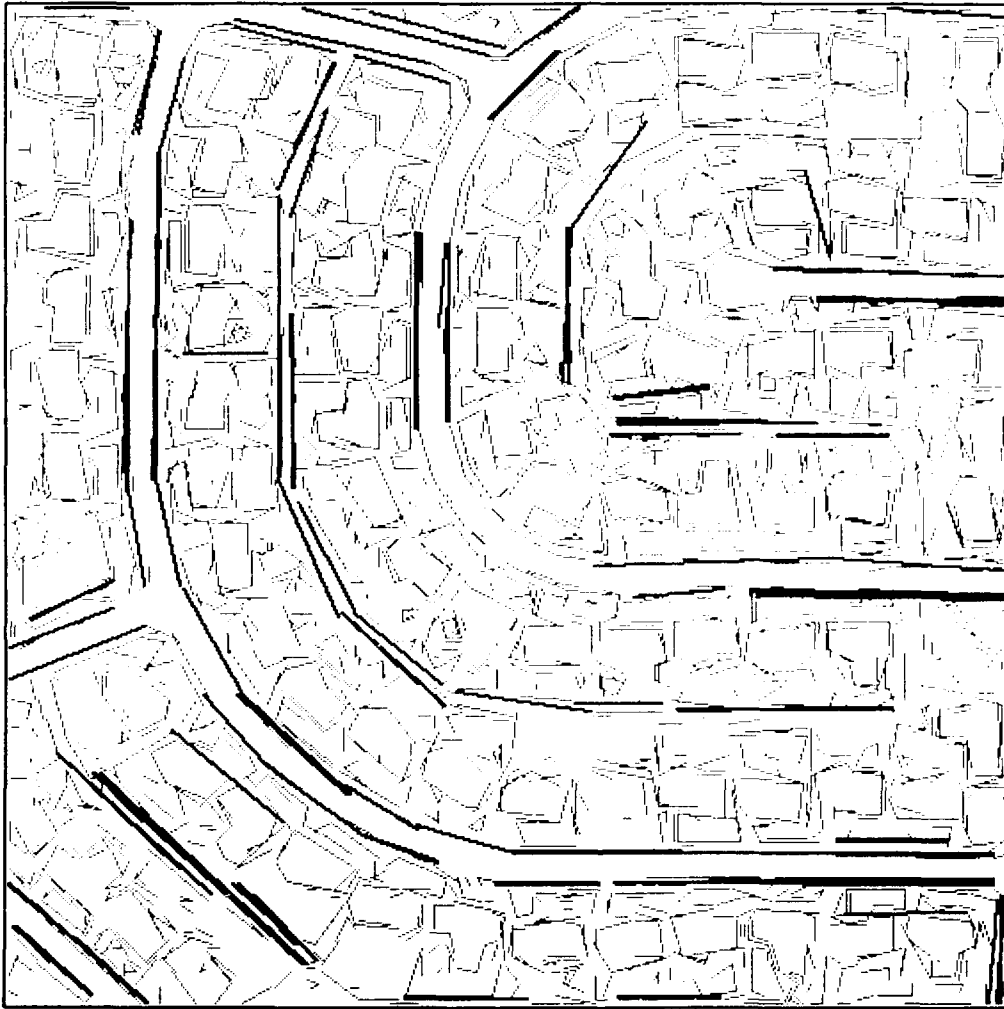


Figure 6.1: Analysis of a 256×256 section from PHOENIX.2A1/
base objects: *line* (thin), partial objects: *long line* (thick)

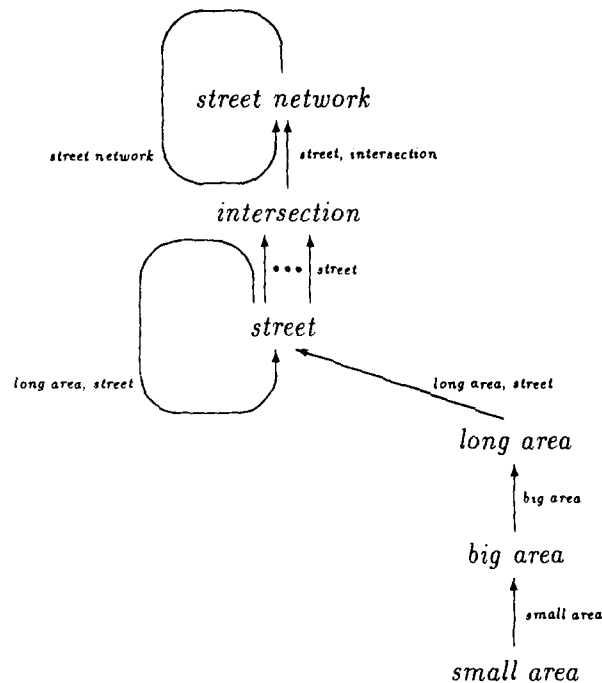


Figure 6.2: Production net for the area oriented detection of intersections

tion the results of the street network extraction are shown for the four images PHOENIX_2A1, PHOENIX_2B1, BIETIGHEIM1_31, and BIETIGHEIM2_31 used in chapter 4. for line oriented image analysis with a BPI system.

6.1 Area Oriented Detection of Intersections

The street network as target object consists of *streets* connected by their intersections. In the selected images the streets appear as more or less bending homogeneous stripes. Those streets can be constructed, for example, from big and long areas outgoing from small areas as base objects received by the segmentation procedure given in chapter 5.

As shown in Figure 6.2 two or more base objects (terminals) *small area* described by their contour parts and radiometric features are connected to the partial object *big area* triggered by the base object *small area*. Then the partial objects *big area* are combined to the new partial object *long area*, and long ar-

as build up the partial objects *street*. If two or more streets are meeting with an angle greater than a given threshold it is assumed that there must be an object *intersection* to be detected. The partial objects *intersection* are representing the nodes in the target object *street network*.

In the next section the productions required for generating the partial and target objects are given in detail.

6.2 Extended Set of Productions

In the selected images the streets appear as a network of more or less connected dark or bright long areas. This statement remains true even for more complex scenes. In the chosen resolution the width of the long areas lies between 5 to 12 pixels. The long areas and by this the street network can be obtained by merging selected area segments received from appropriately chosen spot and projection levels. Before a merging of segments can take place, three main criteria for selecting candidates must be fulfilled:

1. a neighborhood relation of the kind *overlapping* or *adjacent* must hold between reference segment and candidate segments
2. there must be a certain similarity between *geometric* features of reference segment and candidate segments
3. there must also be a certain similarity between *radiometric* features of reference segment and candidate segments

According to the definitions

$$R = (V_t, V_n, S, P) \quad \text{with}$$

V_t = set of base objects
 V_n = set of partial objects
 S = set of target objects
 P = set of productions

given in subsection 4.3.1 the following productions are applied for the detection of intersections and extraction of the street network:

$V_t = \{small\ area\}$
 $V_n = \{big\ area, long\ area, street, intersection\}$
 $S = \{street\ network\}$
 $P = \{$ P1: *small area (small area) \longrightarrow big area;*
 P2: *big area (big area) \longrightarrow long area;*
 P3: *long area (long area, street) \longrightarrow street;*
 P4: *street (long area, street) \longrightarrow street;*

P5: street (street) \rightarrow intersection, ..., intersection;
P6: intersection (street, intersection) \rightarrow street network;
P7: street network (street network) \rightarrow street network}

The productions *P1* to *P7* are described in detail in the following subsections. Thereby the so-called *triggering* partial object denotes that object which is chosen from the supervisor as the (at present) *best* object for further processing (see Figure 4.6).

P1: Small Area \rightarrow Big Area

If a base object *small area* is selected as best object, then the hypothesis for the partial object *big area* has to be tested. The associated verification program for testing that hypothesis is triggered by the base object *small area* selected as best object. For testing purpose two examination steps are performed. First a narrow search area around that contour part of the triggering small area which has the weakest contrast of all contour parts is examined. The aim is to detect contour parts of other small areas which have similar features, namely a similar orientation and contrast as the reference contour part. Second, in case of success it is tested whether the small area belonging to the found contour part is overlapping or (at least) adjacent with the triggering small area, and having similar mean brightness and mean contrast. In case of a successful examination the partial object *big area* is built up by the selected small area together with those small areas found. The big area is described by the following attributes: endpoints on the contour of the big area determined by the regression line laid through the gravity points of all participating small areas, mean brightness, mean contrast to its surrounding, length, orientation, mean width and variance of width. After composition of the big area an assessment is accomplished according to its variance of width, that means the degree of *fitting* of the small areas with each other.

P2: Big Area \rightarrow Long Area

If a partial object *big area* is selected as best object, then the hypothesis for the partial object *long area* has to be tested. The associated verification program for testing that hypothesis is triggered by the partial object *big area* selected as best object. For testing purpose a search area around the big area is examined to find other big areas with similar orientation, mean width, mean contrast and mean brightness. In case of a successful examination the partial object *long area* is built up by the selected big area together with those big areas found. The long area is described by its attributes mean brightness, mean contrast to its surrounding, mean width, variance of width, orientation of the corresponding big areas at both ends, and the length. After composition of the long area an assessment is accomplished according to the variance of the width.

P3: Long Area \rightarrow Street

If a partial object *long area* is selected as best object, then the hypothesis for the partial object *street* has to be tested. The associated verification program for testing that hypothesis is triggered by the partial object *long area* selected as best object. For testing purpose a narrow search area around both ends of the long area are examined to detect other long areas or streets with similar brightness and contrast values, similar width, and similar orientation at their ends. In case of a successful examination the partial object *street* is built up by the selected long area together with those long areas or streets found. The street is described by the union of all small areas belonging to that street. After composition of the street an assessment is accomplished according to the assessment of the triggering *long area*.

P4: Street \rightarrow Street

If a partial object *street* is selected as best object, then the hypothesis for the partial object *street* has to be tested in addition to the hypothesis *P5*. The associated verification program for testing that hypothesis is triggered by the partial object *street* selected as best object. For testing purpose a narrow search area around the street is examined to detect other long areas or streets with similar brightness and contrast values, similar width, and similar orientation. In case of a successful examination the partial object *street* is built up by the selected street together with those long areas and streets found. After composition of the street an assessment is accomplished according to the assessment of the triggering *street*.

P5: Street \rightarrow Intersection, ..., Intersection

If a partial object *street* is selected as best object, then the hypothesis for the partial object *intersection* has to be tested in addition to the hypothesis *P4*. The associated verification program for testing that hypothesis is triggered by the partial object *street* selected as best object. For testing purpose many search areas originating from the positions of the long areas along the selected street are oriented at different angles with respect to the direction of the street. The selected street is used to find other streets which have a crossing with this street within the image. The streets found have to meet the criterion *closeness to the selected street*. In case of a successful examination the partial objects *intersection* are built up by the selected street together with those streets found. After composition of the intersection an assessment is accomplished according to the above mentioned criterion.

P6: Intersection \rightarrow Street Network

If a partial object *intersection* is selected as best object, then the hypothesis for the target object *street network* has to be tested. The associated verification

program for testing that hypothesis is triggered by the partial object *intersection* selected as best object. For testing purpose all streets belonging to the selected intersection are accessed. The target object *street network* is built up by the selected intersection together with those streets accessed. After composition of the street network no particular assessment is accomplished.

P7: Street Network \rightarrow Street Network

All partial objects *intersection* composed by testing the hypothesis *P6* are assembled. The intersections with their associated streets can be accessed by searching for assessment values greater than the threshold given for successful examination of intersections.

6.3 Results from Area Oriented Extraction of Street Networks

In order to test the extended model for the street network analysis described in the previous section the same images as in chapter 4. were used:

- PHOENIX.2A1 (1/4 section of PHOENIX.2, see Figure 4.10)
 - 512 \times 512 pixels
 - 1 : 20,000 real scale
 - 50 μm pixel size (on image)
 - 1 m pixel size (on earth)
- PHOENIX.2B1 (1/4 section of PHOENIX.2, see Figure 4.14)
 - 512 \times 512 pixels
 - 1 : 20,000 real scale
 - 50 μm pixel size (on image)
 - 1 m pixel size (on earth)
- BIETIGHEIM1.31 (1/4 section of BIETIGHEIM1.3, see Figure 4.18)
 - 512 \times 512 pixels
 - 1 : 14,000 real scale
 - 50 μm pixel size (on image)
 - 0.7 m pixel size (on earth)
- BIETIGHEIM2.31 (1/4 section of BIETIGHEIM2.3, see Figure 4.22)
 - 512 \times 512 pixels

1 : 4,000 real scale
100 μm pixel size (on image)
0.4 m pixel size (on earth)

The results from all test images are presented in the same way (see Figure 6.3 to Figure 6.30):

1. preprocessing of test image/base objects: *small area* (blue)
2. analysis of test image/generated partial objects: *big area* (yellow) triggered by *small area*
3. analysis of test image/generated partial objects: *long area* (green) triggered by *big area*
4. analysis of test image/generated partial objects: *street* (green) triggered by *long area* or *street*
5. analysis of test image/generated partial objects: *intersection* (red) triggered by *street*
6. analysis of test image/generated target objects: *street network* (yellow) triggered by *intersection* or *street network*
7. analysis of test image/target objects: *street network* (yellow), partial objects: *intersection* (red), noncomposing partial objects: *street* (green), *long area* (green), *big area* (yellow)

Every last figure shows the complete extraction result of the street network analysis where the generated partial objects *big area*, *long area*, *street*, *intersection*, and the target object *street network* are superimposed such that only the noncomposing partial objects are seen and the underlying composing objects are not seen.

6.3.1 PHOENIX_2A1

Because of its quite clear structures this scene was used to develop and test the production modules and to adjust the parameters for the structure oriented image analysis with a blackboard system. In this scene the street network consists of more or less dark contrasting long areas. Therefore the dark areas segmented by projecting the spots from level 6 to gray value level 9 are used from preprocessing and applied as base objects *small area* for structural analysis.

As shown in Figure 6.3, bigger areas in the street network (e.g. intersections of streets) sometimes are not segmented from projection level 9 with a level difference of 3. The generated partial objects *big area* are shown in Figure 6.4. The isolated small areas which could not be connected to big areas are thrown away. Figure 6.5 shows all partial objects *long area*. Those big areas which do not meet the connection rules are suppressed.

In Figure 6.6 the partial objects *street* generated by connecting long areas or streets are shown. Only few areas in the fields and some segments of roofs are detected as streets. On the other hand some parts of the streets are not found because of their inhomogeneities with respect to gray values or shape and because of misadaption to the width of the street (level difference of 3).

Figure 6.7 shows the position of the detected objects *intersection*. Ten intersections are found. All streets which are connected by these intersections are plotted in the next figure. Figure 6.8 represents the network of streets originating from the intersections. The street network is detected nearly completely with respect to the adapted width of streets. In Figure 6.9 all object types from *big area* up to the *street network* are superimposed. Only the base objects (*small area*) are omitted for clearness.

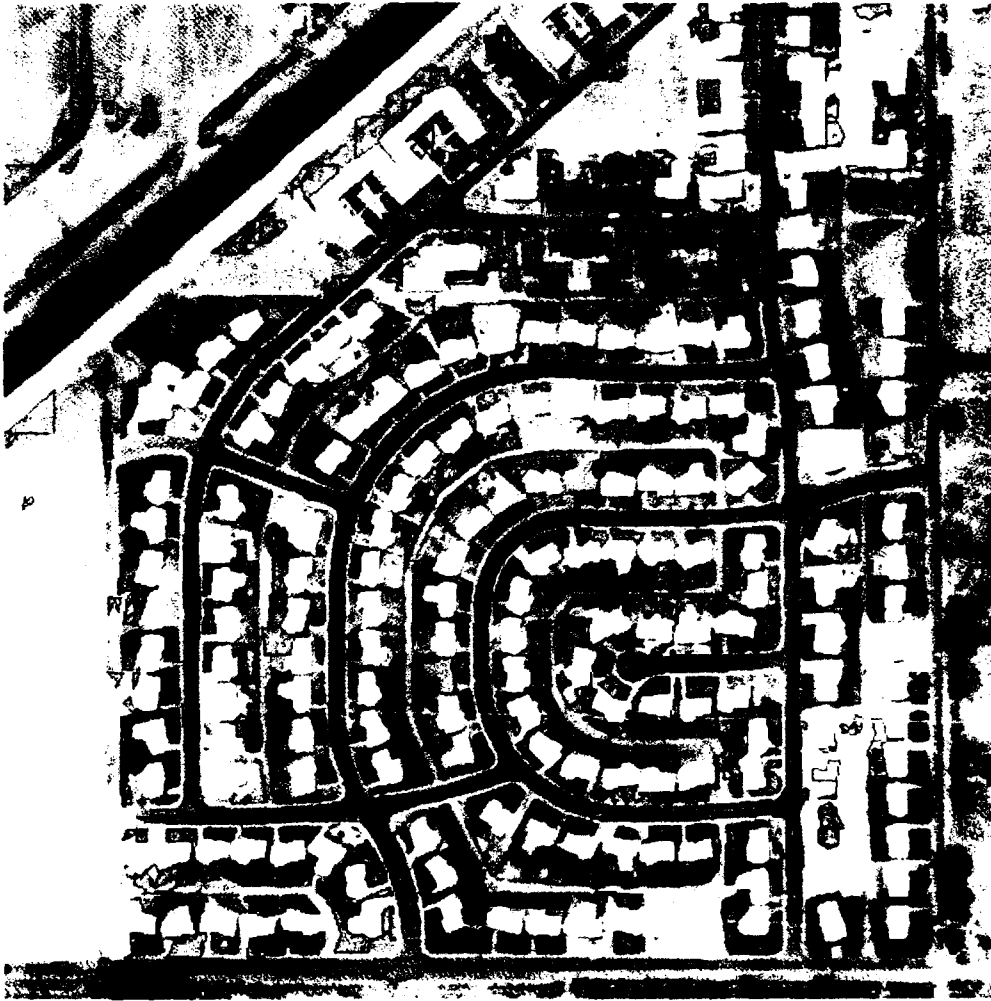


Figure 6.3: Preprocessing of PHOENIX.2A1/base objects: *small area* (blue)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 6.4: Analysis of PHOENIX.2A1/triggering base objects: *small area*, generated partial objects: *big area* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)

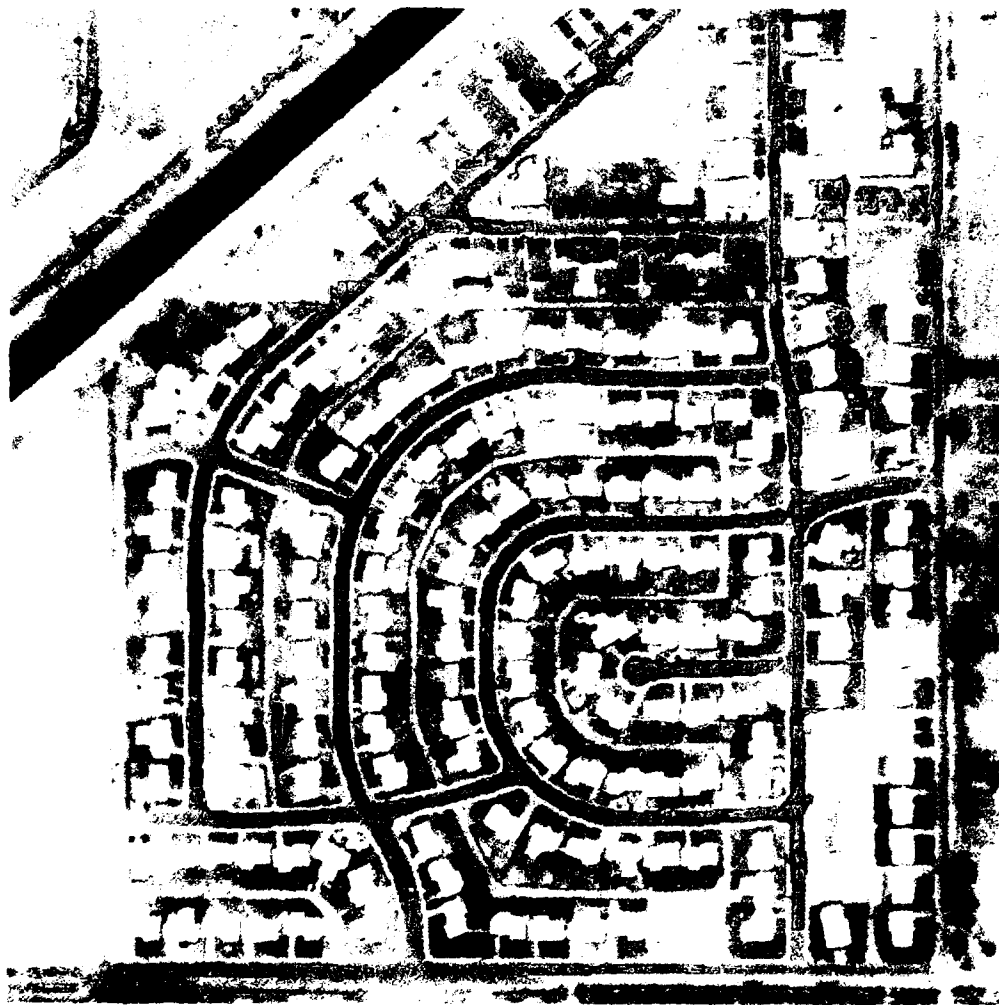


Figure 6.5: Analysis of PHOENIX.2A1/triggering partial objects: *big area*, generated partial objects: *long area* (green)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)

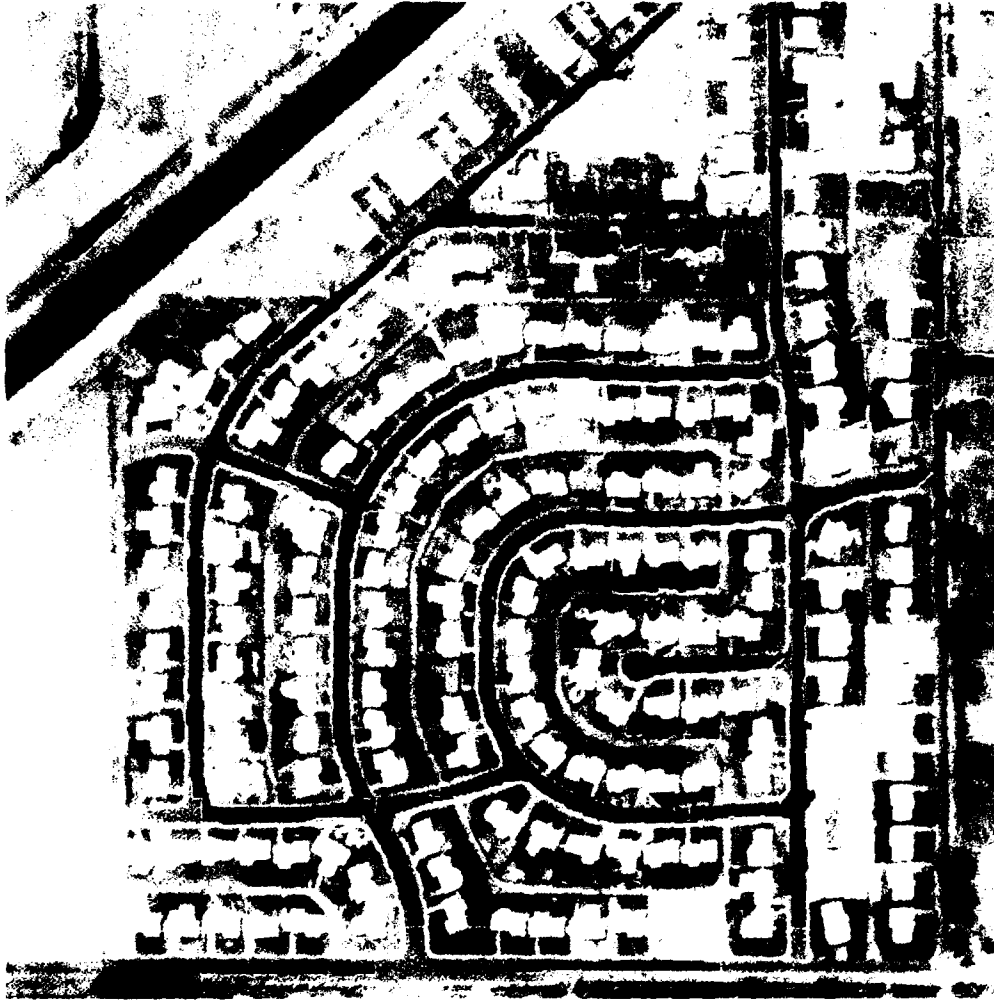


Figure 6.6: Analysis of PHOENIX_2A1/triggering partial objects: *long area*, *street*, generated partial objects: *street* (green)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 1\text{m}$)

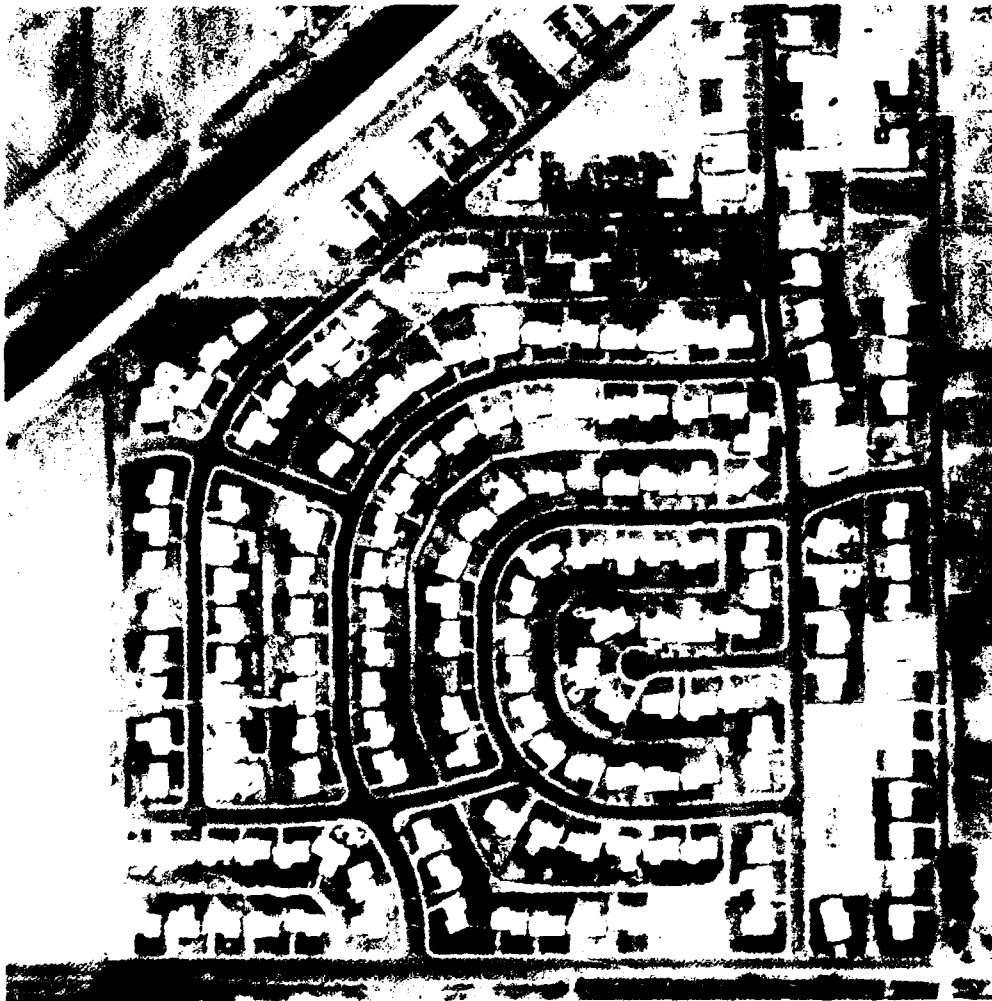


Figure 6.7: Analysis of PHOENIX_2A1/triggering partial objects: *street*,
generated partial objects: *intersection* (red)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 6.8: Analysis of PHOENIX_2A1/triggering partial objects: *intersection*, *street network*, generated partial objects: *street network* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 1\text{m}$)

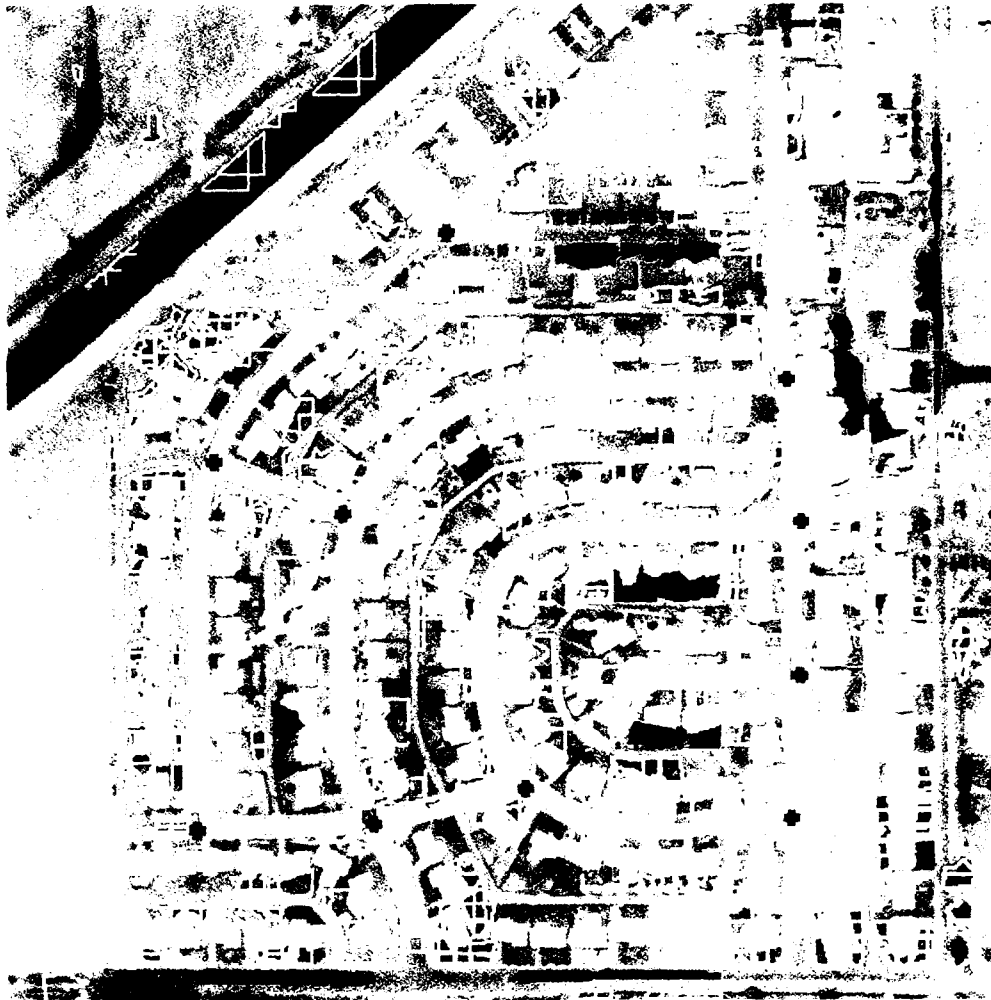


Figure 6.9: Analysis of PHOENIX_2A1/target objects: *street network* (yellow), partial objects: *intersection* (red), *street* (green), noncomposing partial objects: *long area* (green), *big area* (yellow) (matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 1\text{m}$)

6.3.2 PHOENIX_2B1

As in PHOENIX_2A1 the dark contrasting small areas segmented by projecting the spots from level 6 to gray value level 9 are taken as base objects. The adjustment of the parameters, however, is modified with respect to a smaller variance of the width of long areas for the analysis of PHOENIX_B1. The street network is detected nearly completely with respect to the adapted width of streets as in PHOENIX_2A1 (see Figure 6.10 to Figure 6.16). Some connecting dark areas are interpreted as streets, thus misdetection of intersections could not be avoided in a few cases.

6.3.3 BIETIGHEIM1_31

In this scene the bright contrasting small areas segmented by projecting the spots from level 6 to gray value level 9 are taken as base objects. In order to test the applicability of the BPI model and the adjustment of the parameters to this more complex scene, the same parameter setting as for the analysis of the scene PHOENIX_2B1 is taken. Because of misadjustment with respect to the width of streets one can expect that the results are worse than from PHOENIX_2A1 and PHOENIX_2B1. Though the result of the street network extraction seems quite poor, no misdetection of intersections occurs (see Figure 6.17 to Figure 6.23).

Figure 6.23 shows that especially the broader streets, where the base objects *small area* (Figure 6.17) are lying side by side and not only one behind the other, are not extracted very well. This is due to the fact that the connection from big areas (Figure 6.18) to long areas (Figure 6.19) and then up to streets (Figure 6.20) fails.

6.3.4 BIETIGHEIM2_31

As in BIETIGHEIM1_31 the bright contrasting small areas segmented by projecting the spots from level 6 to gray value level 9 are taken as base objects. Because of misadjustment with respect to the width of streets one can expect that the results as in BIETIGHEIM1_31 are worse than from PHOENIX_2A1 and PHOENIX_2B1. Though the BPI model and the parameter setting are not changed, the result of the analysis from BIETIGHEIM2_31 is obvious better than that of the scene BIETIGHEIM1_31. One misdetection of an intersection occurs in the horizontal street in the lower part of the figure. Some streets are not taken in their total width. This is due to the strong variation of their gray values.



Figure 6.10: Preprocessing of PHOENIX_2B1/base objects: *small area* (blue)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 1\text{m}$)

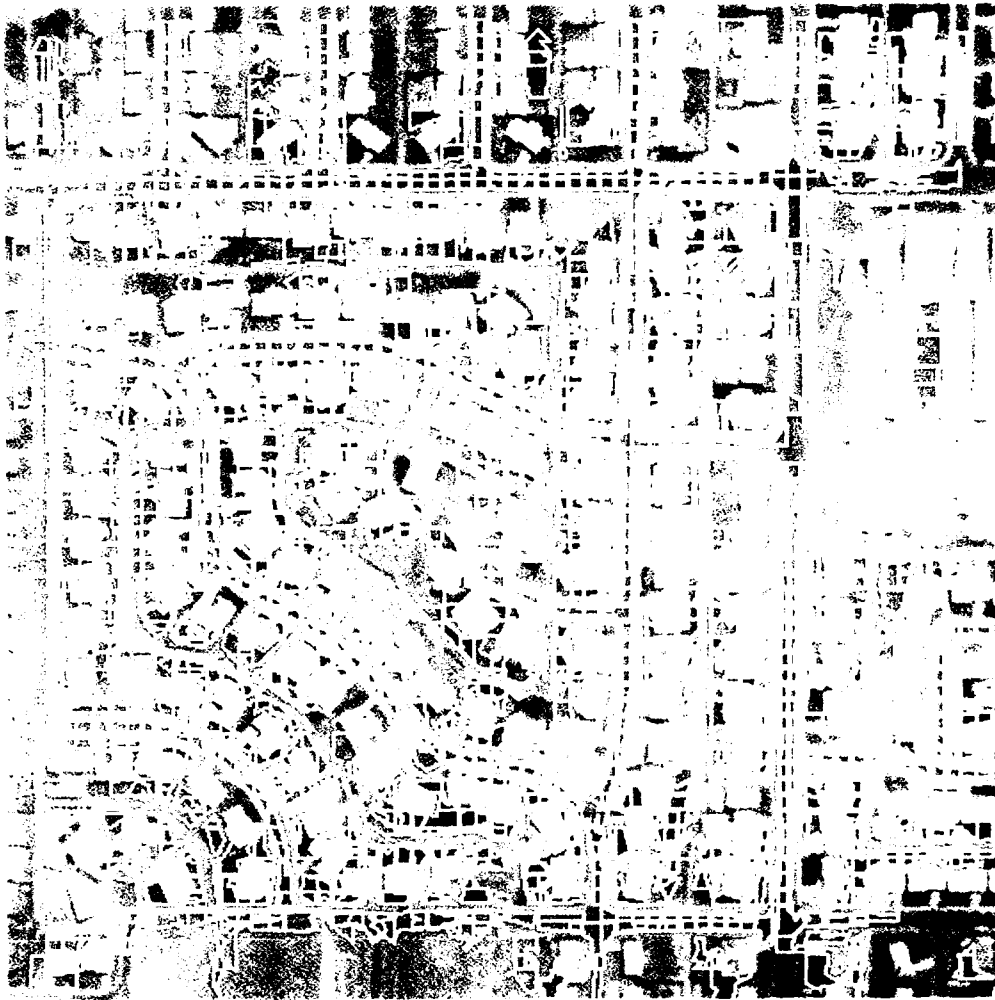


Figure 6.11: Analysis of PHOENIX.2B1/triggering base objects: *small area*, generated partial objects: *big area* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)

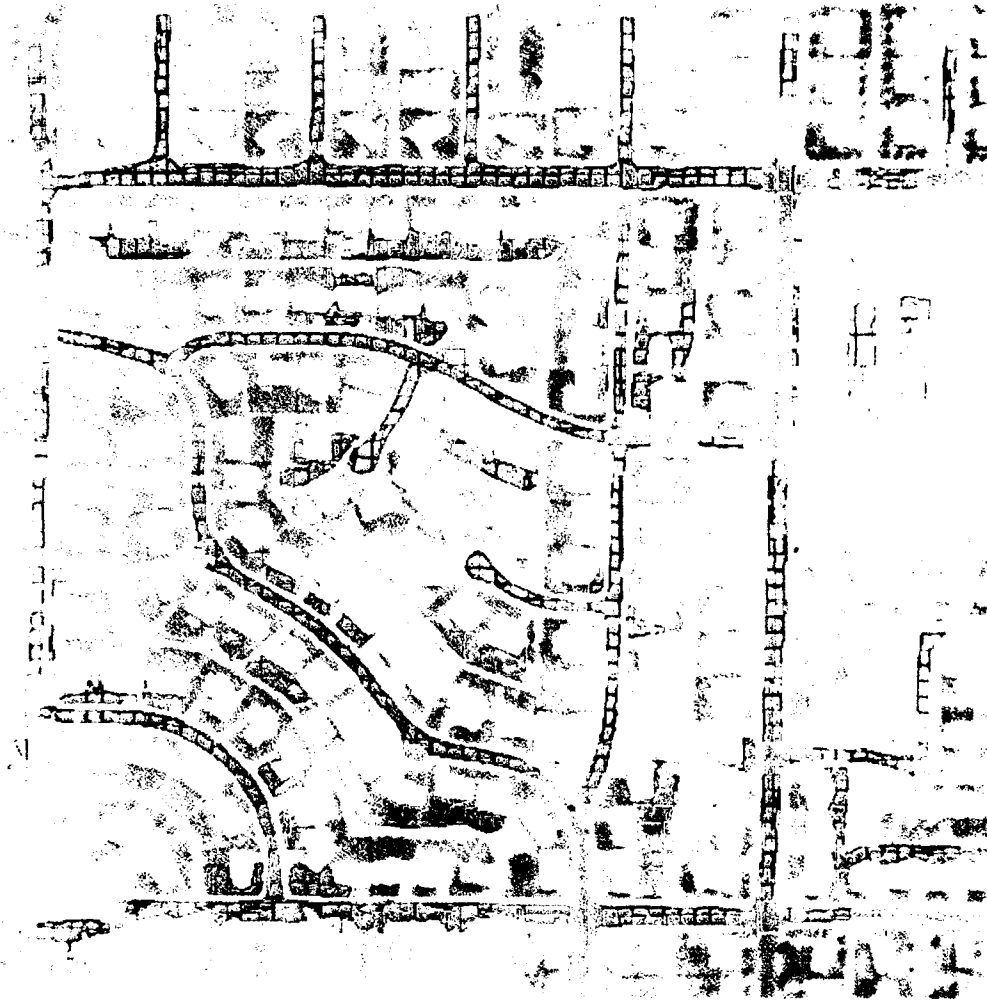


Figure 6.12: Analysis of PHOENIX_2B1/triggering partial objects: *big area*, generated partial objects: *long area* (green)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)



Figure 6.13: Analysis of PHOENIX.2B1/triggering partial objects: *long area*, *street*, generated partial objects: *street* (green)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)

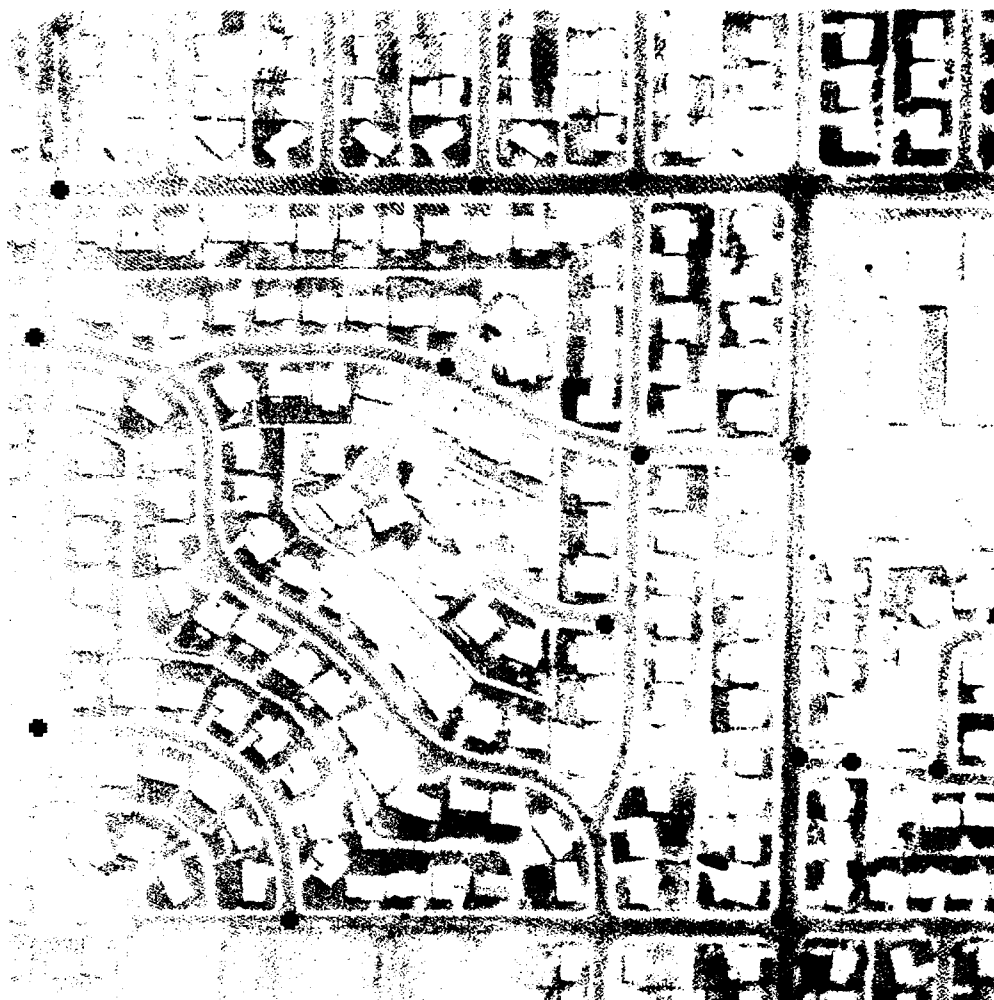


Figure 6.14: Analysis of PHOENIX.2B1/triggering partial objects: *street*,
generated partial objects: *intersection* (red)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 1\text{m}$)



Figure 6.15: Analysis of PHOENIX.2B1/triggering partial objects:
intersection, street network, generated partial objects: *street network* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 1m$)

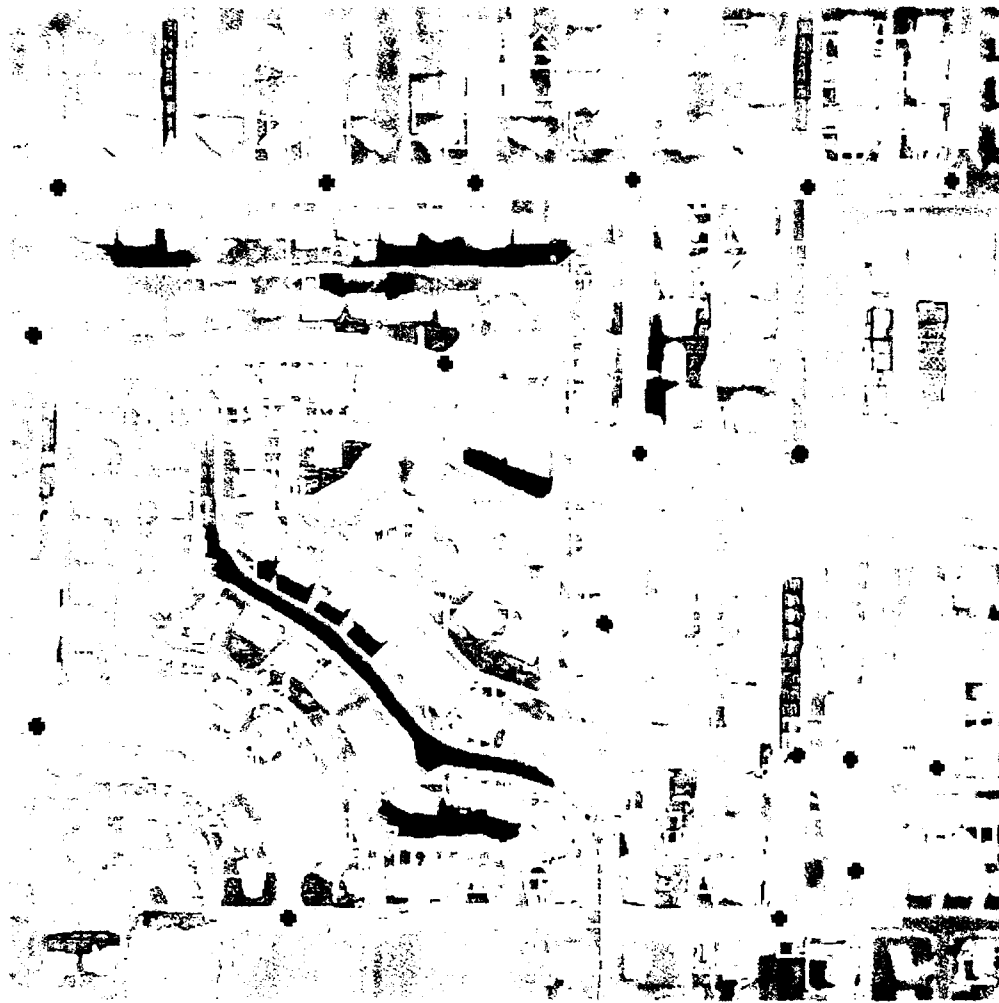


Figure 6.16: Analysis of PHOENIX_2B1/target objects: *street network* (yellow),
partial objects: *intersection* (red), *street* (green),
noncomposing partial objects: *long area* (green), *big area* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 1\text{m}$)



Figure 6.17: Preprocessing of BIETIGHEIM1.31/base objects: *small area* (blue) (matrix size 512×512 pixels, pixel size and scale $50\mu m : 0.7m$)



Figure 6.18: Analysis of BIETIGHEIM1.31/triggering base objects: *small area*,
generated partial objects: *big area* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 0.7\text{m}$)



Figure 6.19: Analysis of BIETIGHEIM1.31/triggering partial objects: *big area*, generated partial objects: *long area* (green)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 0.7m$)

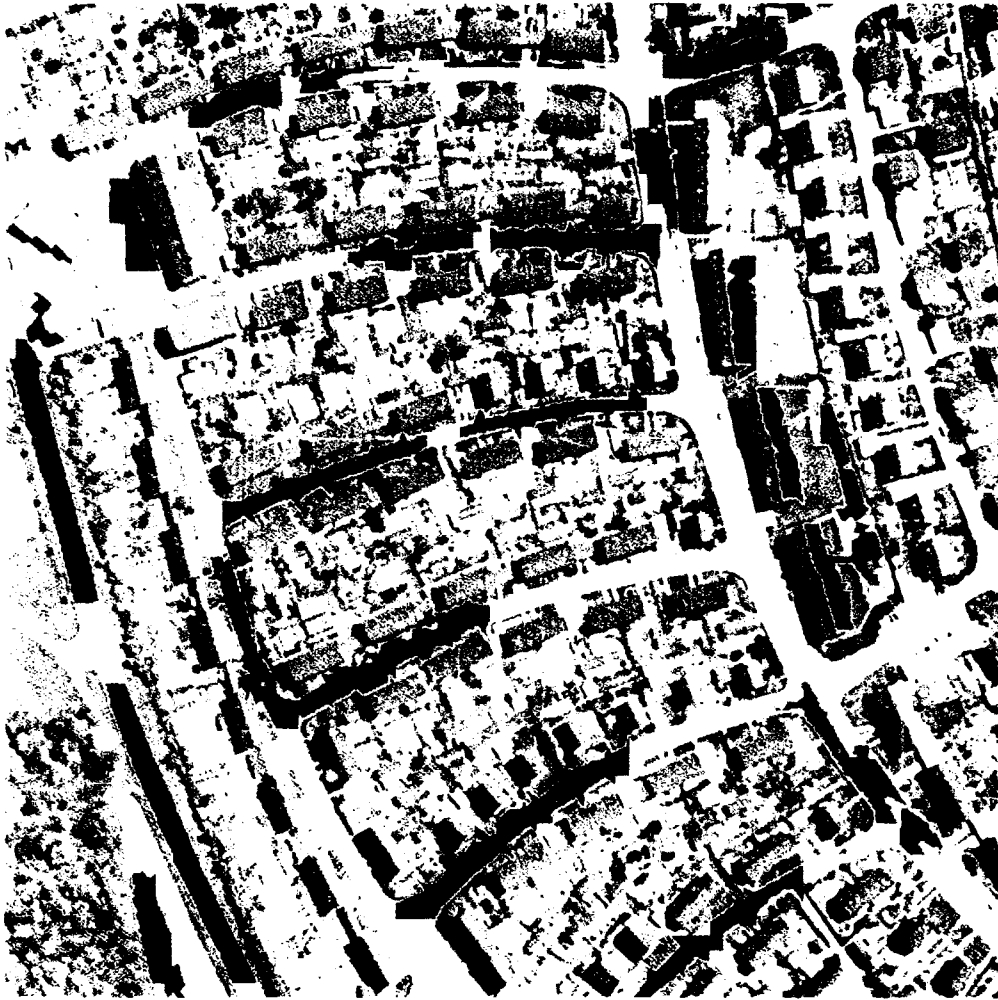


Figure 6.20: Analysis of BIETIGHEIM1.31/triggering partial objects:
long area, street, generated partial objects: *street* (green)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 0.7m$)



Figure 6.21: Analysis of BIETIGHEIM1_31/triggering partial objects: *street*,
generated partial objects: *intersection* (red)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 0.7m$)



Figure 6.22: Analysis of BIETIGHEIM1.31/triggering partial objects:
intersection, street network, generated partial objects: *street network* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu\text{m} : 0.7\text{m}$)



Figure 6.23: Analysis of BIETIGHEIM131/target objects:
street network (yellow), partial objects: *intersection* (red), *street* (green),
noncomposing partial objects: *long area* (green), *big area* (yellow)
(matrix size 512×512 pixels, pixel size and scale $50\mu m : 0.7m$)

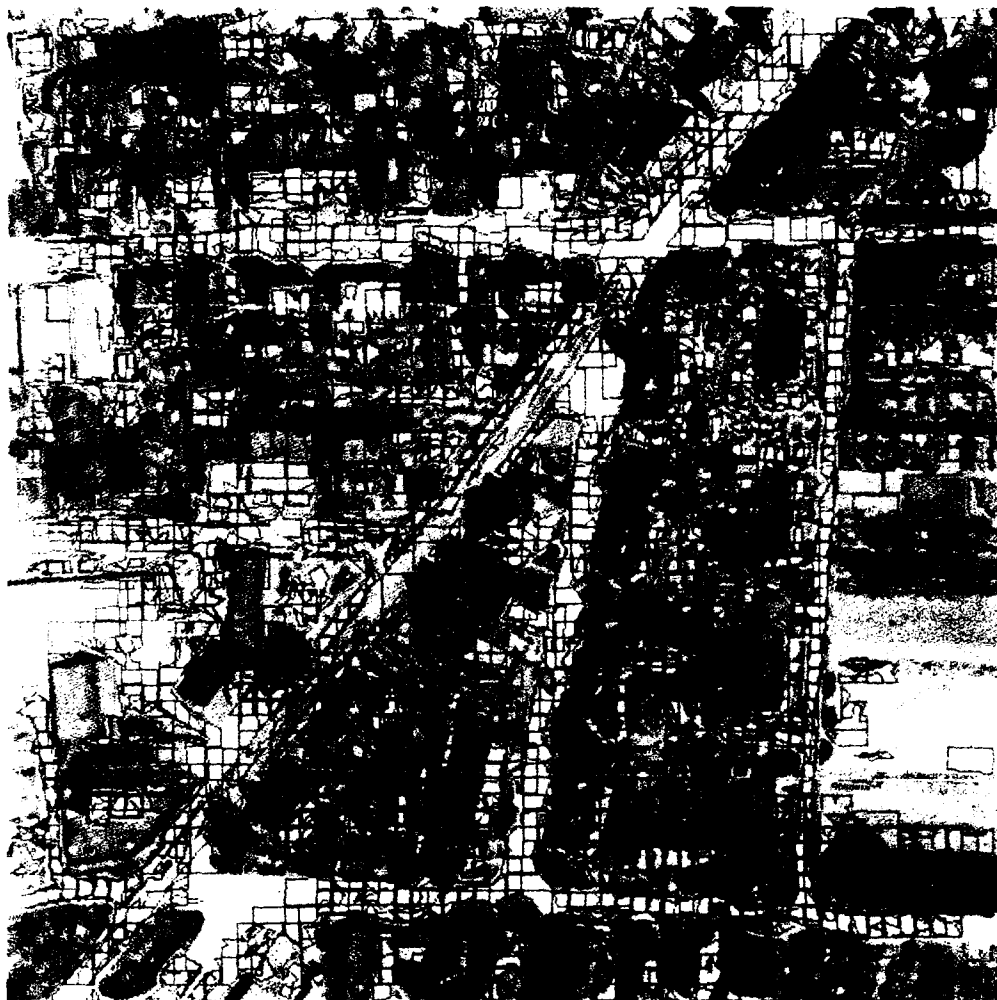


Figure 6.24: Preprocessing of BIETIGHEIM2.31/base objects: *small area* (blue) (matrix size 512×512 pixels, pixel size and scale $100\mu\text{m} : 0.4\text{m}$)



Figure 6.25: Analysis of BIETIGHEIM2.31/triggering base objects: *small area*,
generated partial objects: *big area* (yellow)
(matrix size 512×512 pixels, pixel size and scale $100\mu m : 0.4m$)



Figure 6.26: Analysis of BIETIGHEIM2.31/triggering partial objects: *big area*, generated partial objects: *long area* (green)
(matrix size 512×512 pixels, pixel size and scale $100\mu m : 0.4m$)



Figure 6.27: Analysis of BIETIGHEIM2.31/triggering partial objects:
long area, street, generated partial objects: *street* (green)
(matrix size 512×512 pixels, pixel size and scale $100\mu m : 0.4m$)



Figure 6.28: Analysis of BIETIGHEIM2.31/triggering partial objects: *street*,
generated partial objects: *intersection* (red)
(matrix size 512×512 pixels, pixel size and scale $100\mu\text{m} : 0.4\text{m}$)



Figure 6.29: Analysis of BIETIGHEIM2.31/triggering partial objects:
intersection, street network, generated partial objects: *street network* (yellow)
(matrix size 512×512 pixels, pixel size and scale $100\mu\text{m} : 0.4\text{m}$)



Figure 6.30: Analysis of BIETIGHEIM2.31/target objects:
street network (yellow), partial objects: *intersection* (red), *street* (green),
noncomposing partial objects: *long area* (green), *big area* (yellow)
(matrix size 512×512 pixels, pixel size and scale $100\mu m : 0.4m$)

Chapter 7

Conclusions

The results about street network extraction in urban imagery achieved so far from solutions of low, medium, and high level image processing are discussed and some concluding remarks are presented in the following:

- The line extraction method using stream following works quite well when applied to image scales where the line objects really appear as line shaped objects (only a few pixels wide). In urban areas the detection of those objects is more difficult than in rural scenes because of greater noise from variations of shape and gray value (contrast). Therefore an image has to be processed with different sets of parameters and with changing image resolution in order to find most of the line objects. With respect to the extraction of street networks misrecognition of streets and their intersections may occur because no image understanding knowledge is applied.
- The line extraction method using structured parallel image operations (SIO) produces similar results as compared with that using stream following. However, the SIO method is more robust with respect to parameter adjustment, clearer to parameter adaption, and more flexible to the extension of models and algorithms. Although no essential difference exists taking the sequential or parallel image processing method, the parallel approach should be favored because of its easier application.
- The edge oriented segmentation method produces more complete results than the preprocessing procedure for line primitives of given length. The completeness is meant in the sense of a higher degree of the multiresolutional radiometric and geometric description of an edge segment.
- The area oriented segmentation method produces also more complete results than the preprocessing procedure for line primitives of given length. The completeness is meant in the sense of a higher degree of the multiresolutional radiometric and geometric description of an area segment. Furthermore the area oriented segmentation method supplements the edge

oriented segmentation method. If only one method is to be applied then the edge oriented segmentation should be preferred because of more clearness of its elements compared to those from area oriented segmentation.

- In order to extract street networks in urban areas with higher performance more intelligent, knowledge based image understanding methods are used. The complexity of the first models used for blackboard oriented structural image analysis (BPI) is still low and depends on the applied preprocessing method. Depending on the kind of line primitives from low level and of base objects (edge or area segments) from medium level processing, different BPI models and sets of productions are (partly) developed and tested. Because of running out of time it was decided on the one hand to use line primitives and on the other hand only one type of base objects (area segments) in only one projection level for structural analysis.
- In case of preprocessed images providing the BPI system with line primitives no BPI model was developed but an existing model adapted only to the task of street network recognition. Therefore no curved streets are extracted because they are not provided by the model. In spite of that the results obtained from the scenes with straight streets and their intersections show already the quality of advanced recognition inherent in the BPI method.
- In case of preprocessed images providing the BPI system with area segments an existing BPI model was modified adapting and interfacing the lower levels of the model to the preprocessing. This model provides curved street as well. However, the complexity inherent in the BPI model and in the provided descriptions of the area segments are not yet used appropriately for street network recognition. Even the preliminary results show that in principle the quality of the structural image analysis is much higher than that which can be received from the line extraction by low level processing.

The test imagery has two levels of complexity. The urban structures of Bietigheim are more complicated than those of Phoenix. Comparisons between low level line extraction and high level street network analysis show, that an already simple image processing method produces rather good results from both types of scenes. It could be shown that application of the developed and implemented improved methods and models produce better results in case of radiometric and geometric ambiguities of image structures. However, the more expensive the preprocessing methods are, the more complex the high level models have to be in order to make use of all relevant base objects produced by low level processing. Furthermore the dependence of the results on the type of a scene is greater in this case than in the case of low level line extraction.

At the moment no attention was paid to the problems of optimization of computer run time and capacity of memory. In this respect, optimization has to be done with regard to the refinement of the software as well as to the application of special blackboard techniques realized in hardware. For further investigations in street network extraction from aerial imagery a combination is favored of the *preprocessing* by SIO operations using the procedures of improved segmentation with the BPI analysis using a more advanced structural BPI model.

Acknowledgement

The principal investigator and the authors gratefully appreciate the valuable help of Dr. - Ing. Manfred Sties from the University of Karlsruhe preparing the research proposal and Dr. rer. nat. Peter Gemmar and Dipl. - Ing. Karl Lütjen from the FGAN laboratory FIM who contributed significantly to the results produced during the project. P. Gemmar and K. Lütjen supported the work with methods and software, P. Gemmar by providing the structured parallel image operations (SIO), and K. Lütjen by providing the blackboard oriented structural image analysis (BPI).

Bibliography

- [1] D. H. Ballard, C. M. Brown
Computer Vision
Prentice Hall, Englewood Cliffs 1982
- [2] U. Bausch, M. Bohner, W.-D. Groch, H. Kazmierczak, M. Sties
Automatic Extraction of Linear Features from Aerial Photographs
Final Technical Report DAJA 37-80-C-0006, ERO, London, April 1981
- [3] P. Gemmar
Ein grundlegendes Operationsmodell für die strukturierte parallele Bildverarbeitung - Entwurf und Anwendung (Dissertation, Universität Karlsruhe)
FIM-Bericht Nr. 208, Ettlingen, Mai 1989
- [4] P. Gemmar, H. Ischen, K. Lütjen
FLIP: A Multiprocessor System for Image Processing
in *Languages and Architecture for Image Processing*
(Eds.: M. J. B. Duff and S. Levialdi), pp. 245-256
Academic Press, London New York 1981
- [5] H. Föger, K. Jurkiewicz
Objektklassifikation mit einem blackboardorientierten Inferenzmechanismus
DFG-Sachbeihilfe Ka 414/8-1, Ettlingen, Januar 1988
- [6] K. Jurkiewicz
Objektklassifikation mit einem blackboardorientierten Inferenzmechanismus
DFG-Sachbeihilfe Ka 414/8-2, Ettlingen, Juli 1988
- [7] *Erfassung und maschinelle Verarbeitung von Bilddaten - Grundlagen und Anwendungen* (Hrsg.: H. Kazmierczak)
Springer, Wien New York 1980
- [8] H. Kazmierczak, M. Sties, R. Lubkowitz, M. Bohner
Coincident Extraction of Line Objects from Stereo Image Pairs
Final Technical Report DAJA 37-82-C-0243, ERO, London, September 1983

- [9] H. Kazmierczak
The Gap between Software Implementation and Hardware Realization of Image Processing
in *Proceedings of the COMP EURO 87 Congress*
(Eds.: E. Proebster, H. Reiner), pp. 126-131
IEEE-Press, Washington, May 1987
- [10] K. Lütjen
Ein wissensbasierter Ansatz zur Wiedererkennung von symbolisch beschriebenen Objekten für die bildgestützte Navigation
FIM-Bericht Nr. 147, Ettlingen, Dezember 1985
- [11] K. Lütjen, H. Füger, H.-J. Greif, K. Jurkiewicz
Auswahlverfahren für die wissensbasierte Bildauswertung mit dem blackboard-orientierten Produktionssystem BPI
in *Proceedings Mustererkennung 87* (Ed.: E. Paulus), pp. 290-294
Springer, Braunschweig 1987
- [12] R. Neu, W. Heißler, H. Kazmierczak, M. Sties
Integration of Artificial Intelligence Concepts into the Methods for Extracting Line Objects from Monochromatic Aerial Imagery
Final Technical Report DAJA 45-84-C-0014, ERO, London, March 1986
- [13] *Multiresolving Image Processing and Analysis* (Ed.: A. Rosenfeld)
Springer, Berlin 1982
- [14] A. Rosenfeld, A. C. Kak
Digital Picture Processing
Academic Press, Orlando 1982
- [15] U. Thoennessen, C. Anderer, H. Groß, R. Schulz
Sensordatenauswertung zur Zielbekämpfung, Teil II
FIM-Bericht Nr. 198, Ettlingen, April 1989