The Use of Image Processing Techniques for the Automated Detection of Blue-Green Algae

Stefan U. Thiel

A thesis submitted in partial fulfilment of the requirements of the University of Glamorgan for the degree of Doctor of Philosophy

November 1994

Abstract

The determination of water quality in freshwater lakes and reservoirs is an important task, which must be carried out on a regular basis. Information about long term water quality must be provided by the existence of particular organisms, for example blue-green algae. Currently the detection of these algae is done in a very time consuming manual way, involving highly trained biologists, for example those employed by the National Rivers Authority. This thesis is a first investigation in the automatic detection of blue-green algae using image processing techniques.

Samples of seven species of blue-green algae and two species of green algae were examined under a microscope and transferred to a computer. The microscope pictures were then stored as digital images.

In order to locate the organisms Image Segmentation routines were applied. In particular, a newly developed LoG Thresholding Operator proved to be effective for the segmentation of biological organisms. Image Enhancement improved the quality and appearance of the segmented species in the images. In the identification process the biological key, which describes some important features of each species, needed to be implemented. With the aid of shape algorithms and textural algorithms both occluding and non occluding organisms were analyzed and meaningful features were extracted.

The obtained features were then used to classify the organisms into different species. Both, discriminant analysis and neural networks were used for classification purposes. A detection rate of approximately 90% was achieved.

The approach has produced promising results and it is hoped that further investigations will be encouraged.

Certification of Research

This is to certify that except where specific reference is made, the work described in this thesis is the result of the candidate. Neither this thesis nor any part of it has been presented or is currently submitted in candidature for any other University.

Stefan Rink
Candidate

Ron Wiltshire
Director of Studies

03.02.1995
date

Acknowledgement

I wish to take this opportunity to thank the University of Glamorgan for providing financial support and the computer equipment. Thanks must go to Professor Alan Ryley for departmental support, to Dr. Ron J. Wiltshire for supervision and guidance and to Lance J. Davies for help on blue-green algae.

I would also like to extend my thanks to the National Rivers Authority, for providing the microscope facilities and invaluable advice in respect to the biological application.

Finally, I also wish to acknowledge all those many unnamed people, from whom I received hints, tips and constructive criticism.

Stefan Thiel

Contents

1 Introduction **1**
- 1.1 Aims and Objectives . . . 2
- 1.2 Problem Approach . . . 3
- 1.3 Test Data . . . 6
- 1.4 The Image Processing Toolkit . . . 6
- 1.5 The Organization of the Thesis . . . 7

2 Literature Review **10**
- 2.1 Blue-Green Algae . . . 12
- 2.2 Image Segmentation and Enhancement . . . 14
  - 2.2.1 Thresholding . . . 14
  - 2.2.2 Edge Detection . . . 15
  - 2.2.3 Morphological Operations . . . 16
- 2.3 Object Identification using Boundary Pixels . . . 17
  - 2.3.1 Fourier Descriptors . . . 18
  - 2.3.2 Moment Invariants . . . 19
  - 2.3.3 Hough Transform . . . 20
  - 2.3.4 Points of High Curvature . . . 21
- 2.4 Object Identification using Texture . . . 21
  - 2.4.1 Structural Approach . . . 22
  - 2.4.2 Moments . . . 22
  - 2.4.3 Spatial Grey Level Dependency Matrix . . . 23
  - 2.4.4 Other Texture Classification Methods . . . 24
- 2.5 Classification . . . 24
  - 2.5.1 Statistical Classification, Discriminant Analysis . . . 24
  - 2.5.2 Neural Networks . . . 26
- 2.6 Automated Detection of Biological Organisms . . . 27

3 Water Pollution and Blue-Green Algae **30**
- 3.1 Water Pollution . . . 31
- 3.2 Blue-Green Algae . . . 31
- 3.3 A Comparison between the Current and Future Methods of Detecting Blue-Green Algae . . . 34
- 3.4 Flow Cytometry . . . 35

4 The Image Segmentation of Algae **37**
4.1 Thresholding 40
4.2 Edge Detection 41
4.2.1 Sobel Edge Detector 42
4.2.2 Laplacian of the Gaussian Operator as an Edge Detector . 43
4.2.3 Setting a Threshold 44
4.2.4 The use of Edge Detection to Enhance the Appearance of Edges 45
4.3 LoG Thresholding 45
4.4 Morphological Operations on Binary Images 50
4.4.1 The Erosion Operation 51
4.4.2 The Dilation Operation 52
4.4.3 The Morphological Open and Close Operations 52
4.4.4 The Application of Morphological Operations 54
4.5 Boundary Tracing 55
4.5.1 Removing an Object from the Input Image 56
4.5.2 The Length of the Boundary 57

5 The Enhancement of Algal Organisms **59**
5.1 Fourier Transform 61
5.2 Low Pass Filtering 63
5.2.1 LPF using the Frequency Domain 63
5.2.2 LPF using Convolution 63
5.3 High Pass Filtering 64
5.3.1 HPF using the Frequency Domain 64
5.3.2 HPF using Convolution 64
5.3.3 Neighbourhood Averaging 66
5.4 Morphological Operations on Grey Scale Images 67
5.5 Contrast Stretching 69
5.6 Histogram Equalisation 70

6 Object Identification using Boundary Pixels **72**
6.1 Fourier Descriptors 75
6.2 Moment Invariants 77
6.2.1 Circularity 81
6.3 Hough Transform 83
6.4 Straightness Algorithm 87
6.5 Important Boundary Points 90

7 Object Identification using Texture **96**
7.1 Structural Approach 98
7.1.1 Cell Finding Algorithm 98
7.2 Statistical Approach 101
7.2.1 Grey Level Distribution Moments 102
7.2.2 Moment Invariants 104
7.2.3 Spatial Grey Level Dependency Matrix 105

8 The Classification of Algae **113**
8.1 Minimum Distance Classifier 116
8.2 Discriminant Analysis 116
8.2.1 Bayes Rule 117
8.2.2 Mahalanobis Distance 118
8.2.3 Simple Discriminant Function 119
8.2.4 Canonical Discriminant Function 121
8.3 Neural Networks 122
8.3.1 Back-Error Propagation 122
8.3.2 Kohonen Maps 124
8.4 The Major Differences between Discriminant Analysis and Neural Networks 125

9 Results **126**
9.1 The individual Classification Results 130
9.1.1 Fourier Descriptors 130
9.1.2 Moment Invariants (Boundary) and Circularity 132
9.1.3 Straightness Algorithm 133
9.1.4 Grey Level Distribution Moments and Mean 136
9.1.5 Moment Invariants (Texture) 137
9.1.6 Spatial Grey Level Dependency Matrix 138
9.1.7 Cell Finding Algorithm 142
9.2 The combined Classification Results 143

10 Algal Identification of Larger Populations **147**
10.1 Methodology 150
10.2 Results 150

11 Conclusions **153**
11.1 Discussion 154
11.1.1 Segmentation 154
11.1.2 Enhancement 155
11.1.3 Object Identification using Boundary Pixel 155
11.1.4 Object Identification using Texture 156
11.1.5 Classification 157
11.2 The Detection of Blue-Green Algae 157
11.2.1 The Biological Key 158
11.3 Future Work 159

A Images **162**

B The Computer System **183**
B.1 Manual of the System 184
B.1.1 How to use *hunt.bin* and what it does 184
B.1.2 How to use *final.bin* and what it does 186
B.1.3 How to use *make_da.bin* and what it does 188

B.2 The Program Structure 190

Bibliography **192**

List of Figures

1.1 The Structure of the Detection Algorithm 4
1.2 Segmentation and Enhancement 5
1.3 The Two Types of Descriptors 5

4.1 The Two Masks of the Sobel Edge Detector 42
4.2 The Laplacian of the Gaussian, $\sigma = 0.5$ 44
4.3 Cross-Section of the LoG Operator with Different Values of σ . . 47
4.4 The LoG with $\sigma = 4.75$ and a Mask Size of 9×9 48
4.5 Masks for Morphological Operations 51
4.6 Example of the Four Morphological Operations 53
4.7 The Structuring Element for the Morphological Operation 54
4.8 The Error Involved with a Digital Boundary 58

5.1 Filtering in the Frequency Domain 62
5.2 Masks for Low Pass Filtering 64
5.3 Masks for High Pass Filtering 65
5.4 The Laplacian of the Gaussian, $\sigma = 1$ 65
5.5 One-Dimensional Neighbourhood Averaging 66
5.6 3×3 Masks for Neighbourhood Averaging 67
5.7 Example of the Grey Level MO 68
5.8 Linear Contrast Stretching 69
5.9 Histograms before and after Histogram Equalisation 71

6.1 Artificial Test Data 74
6.2 The Results of the FD with $N = 16$ 76
6.3 The Results of the MI 82
6.4 Example of the HT with Varying Parameters 86
6.5 The k-cosine 91
6.6 The Important Points 92
6.7 The Squared Distance of the Boundary Points to the Centre Point 93
6.8 The Important Points Found by the Algorithm 93
6.9 The Important Points of the Artificial Data 95

7.1 The Average Grey Levels before and after Smoothing with the Hamming Window 100
7.2 The Histograms of the Textures of Plate A.10 103
7.3 The MI applied to the Artificial Texture 105

7.4 Example for Calculating the SGLDM's 107
7.5 The Five Measures Applied to the Artificial Texture 111

8.1 Back-Error Propagation Network 123
8.2 Kohonen Network 124

B.1 The Program Structure 191

List of Tables

5.1 Example of Histogram Equalisation 71

6.1 The MI of a Rectangle 81
6.2 The Circularity Tested on Artificial Data 84
6.3 The Results of the Straight Line Algorithm 89

7.1 Grey Level Distribution Moments Applied to Artificial Data . . . 103
7.2 Example Values of the Descriptors Based on Figure 7.4 111

9.1 The Species used for the Results 128
9.2 Summary of the Features used for Classification 129
9.3 The Results of the Fourier Descriptors with $N = 16$ 130
9.4 The Results when Separating Filamentous from Curved Algae . . 131
9.5 The Results of the Fourier Descriptors with $N = 256$ 131
9.6 The Results of the Moment Invariants (Boundary) 132
9.7 The Results when Separating Filamentous from Curved Algae . . 133
9.8 The Results of the Measure of Circularity 134
9.9 The Results of the Straightness Algorithm 135
9.10 The Results when Separating Filamentous from Curved Algae . . 135
9.11 The Results of the Grey Level Distribution Moments and Mean . 136
9.12 The Results of the Moment Invariants (Texture) 137
9.13 The Results of the different SGLDM 138
9.14 The Results of the H-SGLDM with $P(2, 0, 45, 90, 135)$ 139
9.15 The Kohonen Map after Training with the H-SGLDM 140
9.16 The Results of the SGLDM Combined 141
9.17 The Results of the Cell Finding Algorithm 142
9.18 The Selected Descriptors 143
9.19 The Results of the Combined Features 144
9.20 The Results of the Combined Features with *Reject* 145

10.1 The Individual Results of the Classification of Scum 151

List of Plates

A.1 Anabaena 163
A.2 Aphanizomenon 164
A.3 Coelosphaerium 165
A.4 Merismopedia 166
A.5 Microcystis 167
A.6 Nostoc 168
A.7 Oscillatoria 169
A.8 Eudorina 170
A.9 Pandorina 171
A.10 Artificial Texture 172
A.11 Test Picture (Crane on Campus) 173
A.12 Plate A.1 after Edge Detection 174
A.13 Plate A.1 after LoG Thresholding and Morphological Close 175
A.14 The Histogram Equalised Objects of Plate A.1 176
A.15 The Boundary with Important Points and Detected Cells 177
A.16 Scum of Anabaena 178
A.17 Scum of Aphanizomenon 179
A.18 Scum of Microcystis 180
A.19 Scum of Nostoc 181
A.20 Empty Microscope Image 182

Chapter 1

Introduction

1.1 Aims and Objectives

It is the aim of this study to create a novel system for the automated detection of blue-green algae using image processing techniques. The purpose of the system is firstly a speeding up of the process of monitoring water quality and secondly to assist in the classification procedure.

The automated system should be able to segment objects in an image obtained using a microscope. The segmented objects are enhanced and certain features extracted. These features provide the necessary information for object classification. The whole system must be able to complete the task without any user interaction. Such an automated system appears not to exist within a biological background.

For the development of a complete system it is important to incorporate already established image processing techniques. However, to tackle certain specific problems it is also necessary to develop and investigate new techniques. It is especially important that all these techniques work closely together to form a working detection algorithm. The possibility to implement the biological key, used by the biologist to identify different algae, needs to be investigated. Information on the biological key can be found in books, which describe algae. Another invaluable source on the biological key are biologists, who work in this field and can input their experience in identifying algae.

Microscope pictures of blue-green algae were obtained and saved as images on the computer for analysis. The computer should then be able to detect the organisms from the images. The difficulties which arise in the initial step of detection of algae are as follows:

- The locations of the organisms are unknown.
- There can be more than one organism in one single image.

- The organisms may be occluded.

Once the image has been scanned for probable objects a final detection algorithm can be applied. The difficulties arising here are:

- The located organism could be any species of blue-green algae as well as any other type of algae or simply just background clutter, i.e. no organism.
- The boundary of the object does not contain enough information to identify an organism.
- Textural information is limited. Texture is important in the identification process.
- The biological key needs to be implemented.

The final identification algorithm needs to take all these points into account and needs to be able to either identify an object or reject it. As shown in the literature search in Chapter 2 such algorithms have not been developed previously.

1.2 Problem Approach

The first step in an automated recognition process is to transfer images of the specified algae to the computer. The microscope slides containing algae are prepared in the NRA Laboratory in Llanelli, South Wales. A video camera attached to a light microscope is used to produce the pictures on video tape.

This tape is then read by a computer program which is able to transfer the pictures as images into computer memory. This process is called *grabbing*. These images are then stored on the hard disk.

In a fully automated recognition system the computer would be required to be directly connected to the light microscope via a video camera. The algorithm

is able to read video images, whether they are played from a tape or directly recorded with a video camera.

Initially two species of blue-green algae were selected, *Anabaena* and *Oscillatoria*. Based on the results obtained with sample images of these two species a first automated detection algorithm was developed. After that, other species of blue-green algae were implemented in the algorithm whilst always updating the algorithm with new routines. Within the automated detection algorithm several Image Processing routines were tested and their results compared.

The overall structure of the detection algorithm is shown in Figure 1.1.

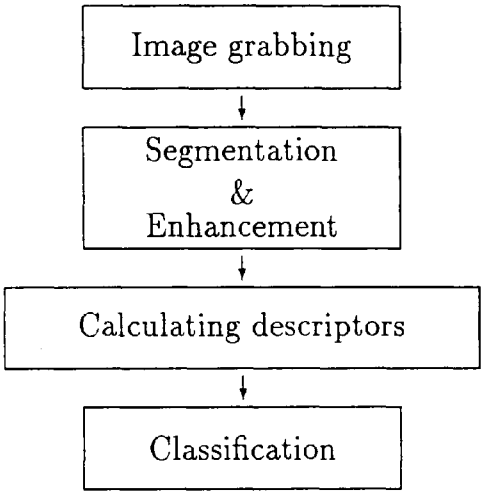


Figure 1.1: The Structure of the Detection Algorithm

The detection process involves locating the organism. This is achieved using image segmentation. In addition, it is necessary to improve the quality of the images and image enhancement algorithms are used for this purpose. Firstly the whole input image was enhanced and then a further enhancement was carried out for each segmented part of the image. This process is outlined in Figure 1.2.

Several image identification algorithms were tested on these enhanced images. The tried identification routines cover a wide range of automated detection techniques, which are based both on boundary information and texture contents

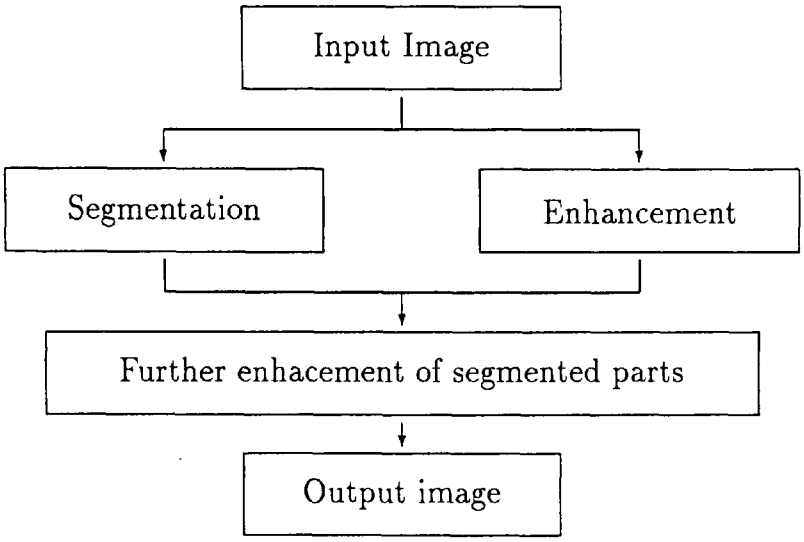


Figure 1.2: Segmentation and Enhancement

of the algal organism as can be seen in Figure 1.3.

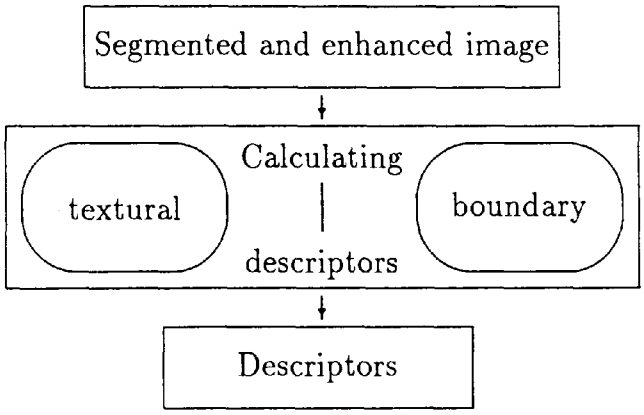


Figure 1.3: The Two Types of Descriptors

A more detailed display of the overall algorithm, showing the linear arrangement of the individual functions and algorithms can be found the Appendix, Figure B.1.

1.3 Test Data

In order to provide sufficient test data for the algorithms, in addition to obtaining data from the NRA, it was also necessary to order artificially grown samples from THE CULTURE COLLECTION OF ALGAE AND PROTOZOA in Ambleside, Cumbria, and from Göttingen University, Germany. This was necessary because natural grown blue-green algae can only be obtained in the summer months and even then their availability is variable. In the summer of 1992 the data provided by the NRA was of variable quality. In fact, the problem of obtaining feasible test data proved to be a serious threat to the project and throughout almost the whole research period attempts were made to obtain more algal samples. However, the mixture of both artificially grown and natural data is important for the testing of the computer methodology.

Several images of test data were acquired and used in the automated system. 178 objects were segmented. 20 segmented objects were background clutter and the remaining 158 objects were one of nine different species of algae. As said above the test samples were recorded over a period of more than one year and it is hoped that the samples represent a working cross section of blue-green algae. The algorithm can therefore be used to identify new images of species which were already included and should reject unknown organisms.

1.4 The Image Processing Toolkit

The work was carried out on a SUN-SPARCstation. The disk space required for the development of the automated recognition algorithm is approximately 20 Mb. Further disk space is required for each image. Each image has a size of 768×575 pixels and each pixel requires one byte because of its 256 grey levels. Therefore the space required for each image is about 450 Kb. In order to save disk space each image can be compressed to a quarter of its size by an internal compression

algorithm.

Although the images are only black and white, the 255 grey levels are giving them a sufficient enough appearance for this investigation. As usual the grey level value 0 represents *black* and the grey level value 255 *white*.

The format of the images is set by SUN VISION and SUN IP. SUN VISION is an interactive software package where images can be displayed and some common image manipulations can be carried out. SUN IP provides basic image processing subroutines. The interface used to convert the input data from video format to image format was VFCTOOL. This interface could also be used to copy images into PC-DOS readable Postscript or TIFF format.

1.5 The Organization of the Thesis

In Chapter 2 a review over the literature is given. Chapter 3 is an overview of the biological application and the connection of Water Pollution and blue-green algae is described. The advantages of an automated detection system are highlighted.

In the Chapters 4 and 5 several image processing routines are described, which were considered for the necessary pre-processing step. Chapter 4 contains information about image segmentation. In particular, the following segmentation routines are considered:

- Thresholding
- Edge Detection
- LoG Thresholding
- Morphological Operations
- Boundary Tracing

Enhancement routines are described in Chapter 5 and the Fourier Transform is also introduced. The enhancement routines are as follows:

- Low Pass Filtering
- High Pass Filtering
- Morphological Operations
- Contrast Stretching
- Histogram Equalisation

Chapters 6 and 7 contain all identification routines which have been considered and implemented on blue-green algae. Chapter 6 contains identification routines which use the shape or boundary of an object:

- Fourier Descriptors
- Moment Invariants
- The Hough Transform
- Straightness Algorithm
- Important Boundary Points Algorithm

Chapter 7 is concerned about object identification by means of texture. The algorithms are as follows:

- Cell Finding Algorithm
- Grey Level Distribution Moments
- Moment Invariants
- Spatial Grey Level Dependency Matrix

Finally, once all the descriptors were calculated, Chapter 8 explains the classification. Both, statistical classification and neural networks were examined. The classification results are then documented in Chapter 9.

Chapter 10 investigates the identification of populations of algae or algal scum. The core body of the thesis is then ended with the conclusions, Chapter 11.

Appendix A contains sample input images of all investigated algal species and also modified images, which were obtained after some image processing algorithms were applied to the original image. A manual of the computer system together with a short display of the program structure is provided in Appendix B.

Chapter 2

Literature Review

The investigated literature is presented in this chapter. The literature review is divided into different sections. Firstly, literature that details the biological background and rationale for the work is documented.

Secondly, literature concerned with several image processing topics, structured into segmentation and enhancement and also into object identification by boundary pixels and texture, is discussed. It is then concluded by a review of the literature concerning object classification.

Thirdly, the literature review covers the important aspect of other work done in the field of automatic detection. A critical review is presented, especially referring to image processing related literature with similar applications.

2.1 Blue-Green Algae

The researcher has a mathematical background and all information about the biological application, blue-green algae, had to be investigated without any prior knowledge. There were two main sources of information: 1) The people working at the NRA Laboratory in Llanelli, their help, which was very useful is greatly appreciated; 2) books, which will now be reviewed.

The literature described here refers to the following topics:

- Algae
- Algal pollution of freshwater, or the monitoring of pollution through the existence of particular algae.
- A key to classify algae

There are many excellent books concerning blue-green algae, for example: Boney [A3], Carr and Whitton [A4], Fay [A5], Fogg et al. [A6], Hynes [A8] and Morris [A10]. Within their first chapters they present a good introduction to the biological background necessary for this research work.

In addition these books were scanned for two main types of information. Firstly, the description of distinct features of the algae, in order to gain information on how the final identification algorithm should work. Secondly, about the ecology of the algae. In particular the influences water quality has on the occurrence of algae and more important the influence of the algae on water quality. This is especially important for the case of blue-green algae.

The book which contains a most complete introduction with a very readable description of other aspects of blue-green algal life was by Fogg et al. [A6]. Another excellent in depth summary on the ecology of blue-green algae is given in the paper by Whitton [A16].

For the identification of algae it is important to have a key for the features which separate one algae from another. Five such books were used and taken into consideration.

Firstly Belcher and Swale [A1] should be mentioned. This book is a very short guide for the reader and contains only limited information. However it is very easy to read and probably the best first introduction for a novice. In addition the classification into species of blue-green algae is the same as the NRA uses for their purpose.

A very complete guide into freshwater algae and how to identify them is presented by Smith [A13]. Even though this book is very old, it is still used in the NRA laboratory to check for certain keys. This book is not only a key, but also a complete book on algae.

The next two books, Bellinger [A2] and Prescott [A12] are only keys, but cover a whole variety of algae. Prescott [A12] is structured by the biological genus, whereas Bellinger [A2] is structured purely by morphology. Therefore, in Bellinger [A2] the different species of blue-green algae are scattered all over the book, and similar looking species, even of a totally different group, are described next to each other. It is quite sensible to structure a key by morphology. However, if only information about one algal group is required, the whole book has to be browsed.

Another key for blue-green algae, which also provides detailed information about the environmental living requirements of the algae is by VanLandingham [A14]. This key lacks however a detailed description of the morphology of the individual blue-green algae.

Reasons for monitoring blue-green algae are given in the two papers by Harding and Hawley [A7] and by Whitton [A15] and they are discussed in the next chapter. Mason [A9] presents in his book the overall picture of freshwater pollution. All different kinds of freshwater pollution are discussed and two case

studies about the appearance of blue-green algae are explained.

2.2 Image Segmentation and Enhancement

2.2.1 Thresholding

Thresholding is the most common segmentation method and has therefore received great interest in the literature.

There is a whole range of different approaches to Thresholding and the papers by Weszka [105] and Sahoo et al. [81] are two excellent Thresholding surveys. These two surveys cover all aspects of Thresholding, however they lack detail. If there is information about a special Thresholding method required, other literature provides much more information. Good introductions into Thresholding are presented in overall Image Processing Books, such as Gonzalez and Woods [27], Niblack [64] and also Rosenfeld and Kak [80]. But there are more specific books, some Thresholding is used for robotics (Fu et al. [24], Horn [37]) and pattern recognition (James [41]). All these books just explain the technique, examples of applications are very rare. It is necessary to refer to papers, to obtain examples on Thresholding applications. It is also noted that Thresholding can in fact be used for various applications, however the Thresholding method varies, depending on the specific requirements of the particular problem.

For example Hills [35] uses Thresholding to find the boundary of microfossils in reflected light images. The image histogram and its second derivative is calculated and the threshold is set at the maximum of the second derivative between the object and background pixels.

The automated Thresholding by Jiang et al. [42] is based on modifying the histogram by using a pyramid data structure and obtaining a valley between the object and boundary pixels which can be used as threshold value.

Spatial Thresholding, which takes, in addition to the pixel value, the location of the pixels into account is described by Mardia and Hainsworth [58].

Local Thresholding is described by Nakagawa and Rosenfeld [63]. The image is divided into smaller windows and for each window a threshold is calculated based on a smoothed histogram. However a threshold is only needed if there are both object and background pixels occurring within the window and this is decided by calculating the mean and variance of the histogram.

Ohta et al. [66] presents Thresholding into three regions, requiring two thresholds.

An iterative Thresholding is presented by Ridler and Calvard [76]. A switching mechanism decides always between background and object, where the corners of the image are taken as initial values for the background.

Documents are thresholded by Taxt and Flynn [88] and excellent results are achieved using a Thresholding method which incorporated several different approaches.

2.2.2 Edge Detection

Edge detection is a method as significant as Thresholding. However, within image processing literature it has received much less attention.

A survey into the differences of particular edge detectors is presented by Arthurs [2]. Four different edge detector operators are examined and it shows that the Sobel edge detector provides very *thick* and sometimes very inaccurate edges, especially when applied to noisy images. The LoG operator provides slightly better results. The best results are achieved with the Maxgrad edge detector and especially the Canny operator. Even in very noisy conditions The results are very good with these two detectors. The disadvantage with these techniques is

however, that the better an edge detector the more processing time is required. Arthurs [2] applies the Canny operator and the Maxgrad edge detector only to a very small part of the test image. The Sobel edge detector however can be easily applied on the whole image.

An important paper concerning edge detection is by Marr and Hildreth [59]. Not only is it the first paper presenting the LoG operator, sometimes referred to as Marr-Hildreth operator, but it also investigates all the features, which an edge detector should provide, for example the detection of edges in all directions and a simultaneous smoothing of the result. Marr-Hildreth's LoG operator is for example applied by Höhne et al. [36] to segment X-ray images of the human brain. An investigation on the performance of the LoG operator under non ideal conditions (corners in edges, non-linearity, noise) can be found in the paper by Berzins [6].

For information on other edge detectors it is best to seek advice in general image processing books such as Gonzalez and Woods [27] and also Rosenfeld and Kak [80]. Both present a thorough overview of several edge detection techniques.

2.2.3 Morphological Operations

Morphological Operations are, in comparison with the two earlier mentioned techniques, much younger and hence are contained only in recent literature. However the demand for them is increasing.

The most important and most frequently cited book on Morphological Operations is the book by Serra [83], which can be highly recommended. It forms together with the book by Matheron [60], which was the first book to define Morphological Operations, the bases of morphology. In these books the operations are explained and theoretically evaluated. In terms of image processing and morphology the paper by Haralick et al. [32] provides a very understandable

introduction and it shows some simple examples. New overall image processing books, for example Low [53], now also include texts concerning morphology. This is especially noticeable with books, where a new edition was recently published and morphology included, for example Gonzalez and Woods [27].

The Morphological Operators are very easy to apply on binary images and have been applied in several different fields. Katsinis et al. [43] uses them on marine biological images. With their help he is able to remove *legs* or *antennae* from some organisms and also to extract the boundary. In the paper by Bovik et al. [9] the smoothing characteristic of the Morphological Operations is used to measure automatically the cellular area after an initial Thresholding procedure.

A grey level application of the Morphological Operations is given by Werman and Peleg [103]. A signature of texture is defined using grey level morphology and hence to assist in texture analysis.

Goetcherian [26] uses the grey level morphological operations to obtain an efficient edge detector. An edge is in this case the difference of the original to its expanded version.

Another example is given in Gonzalez and Woods [27], where grey level morphology is used to enhance images. A blurring effect is noticeable after both, the *min* and the *max* operator, are applied.

2.3 Object Identification using Boundary Pixels

A survey of algorithms for the identification by means of boundary pixels is presented by Pavlidis [71]. This survey contains transform techniques (Fourier Descriptors, Moment Invariants) and the processing of contours (Points of high curvature) as well as geometrical transformations (Thinning Algorithm, Projec-

tion, Decomposition). In the following section the literature for the techniques which were important for this work is discussed.

2.3.1 Fourier Descriptors

The method of Fourier Descriptors is well known and is a frequently used method for the identification of objects with the aid of the boundary.

Of all reviewed literature Wallace and Wintz [102] provide the most complete paper on Fourier Descriptors. The Fourier Descriptors are introduced and normalisation is described in detail. An example, the identification of different types of aircraft is then described. Both a 2-D and a 3-D identification are presented. In addition a possible chain code error is carefully monitored and taken into account.

Another example of aircraft recognition is presented by Reeves et al. [75]. This very short paper consists mainly of a display of results. The Fourier Descriptors and Moment Invariants are compared, with the Fourier Descriptors producing a much better identification accuracy. These findings are confirmed also by Illing [39], who in addition, investigates their accuracy in noisy and occluded conditions.

In older papers by Zahn and Roskies [108] and also by Granlund [28] the Fourier Descriptors are defined differently concerning the set of required boundary points. The boundary points are represented as polar coordinates (r, θ) and successive points are considered by a uniform increase in θ. With this definition, the results in identifying plane closed curves, in general, and hand print characters or numbers, in particular, are also very good. Zahn and Roskies [108] not only use the Fourier Descriptors to classify curves but also reverse the effect to generate artificial figures.

Peerson and Fu [74] show, in addition to the shape discrimination, a further

interesting feature of the Fourier Descriptors. With their aid they are able to obtain skeletons of objects.

The number of required Fourier Descriptors is partially examined by Rösler et al. [77]. For the classification of artificial shapes only one Fourier Descriptor is sometimes sufficient. The importance of the number of Fourier Descriptors can also be seen in Gonzalez and Woods [27], where a square is reconstructed from Fourier Descriptors. When only few descriptors are taken the original square appears more like a circle. When a higher number of descriptors are considered, the shape becomes more square.

2.3.2 Moment Invariants

When considering Moment Invariants the paper by Hu [38] is the first significant publication. In this paper the seven moment invariants are derived from first, second and third order moments. An example of an application is given by the identification of certain characters. Hu [38] presents a complete and readable introduction to Moment Invariants. Even though it is old, it has yet not been matched by other literature on Moment Invariants.

Dudani et al. [20] follows Hu's approach to classify aircraft. In addition Dudani et al. [20] introduces a further size normalisation of the Moment Invariants. The results shown are promising. However Reeves et al. [75], who also classify aircraft, obtain much better classification results with Fourier Descriptors than by Moment Invariants.

Teague [90] introduces in his paper the so called Zernike Moments. In general these Moments are similar to the Moment Invariants by Hu [38], however moments of higher order are also used.

2.3.3 Hough Transform

The Hough Transform is very common in applications where the detection of straight lines is required. An early paper is Duda and Hart [18], where the use of polar coordinates is proposed in contrast to the traditional cartesian coordinates. Krahe and Pousset [48] use a straight line algorithm to detect parallel lines.

Sometimes the straight line Hough Transform is efficient enough to detect features such as artificial curves as shown by Murakami et al. [62] or Pao et al. [68], who also investigate its robustness in the case of occlusions.

An application for the detection of ellipses is presented by Wang et al. [101]. A two step approach is proposed. Firstly possible locations of ellipses are found and secondly the Hough Transform is applied only to the sub-images to locate the ellipses.

The most important paper on the Hough Transform is by Ballard [4], where not only the detection of lines and ellipses is explained, but the Hough Transform is also extended to detect all arbitrary shapes.

Illingworth and Kittler [40] suggest a method to speed up the process of detection by changing the image resolution from coarse to fine. Atiquzzaman [3] uses this approach and extends it to solving the problems which occur with the discretization error. Another possible way of speeding up the Hough Transform is presented by Ma and Chen [56] in using a reduced accumulator array.

Kiryati and Bruckstein [45] investigate the differences of the straight line Hough Transform, to least square fitting and *by eye* fitting. In another paper (Sklansky [85]), not only the detectional characteristics of the Hough Transform are shown, but in addition it is proved that the Hough Transform can also have filtering properties.

Most detection routines which rely on the boundary require a complete set

of boundary points. Turney et al. [99] describe how the Hough Transform is able to detect objects using segments of the boundary rather than a complete curve, for the description of objects.

2.3.4 Points of High Curvature

The first paper to present an approach to detecting angles in curves was by Rosenfeld and Johnston [78]. It shows, that the k-cosine detects changes in direction in curves, hence points of high curvature or dominant points. This paper was soon improved (Rosenfeld and Weszka [79]) and since then several other methods have been developed. A recent summary of most angle or dominant point detection algorithms can be found in the paper by Teh and Chin [91]. It briefly introduces various methods and shows their differences in several examples. Illing [39] describes the results of the algorithms described by Rosenfeld and Johnston [78] and also by Rosenfeld and Weszka [79] on aircraft in noisy and occluded conditions. It is pointed out that the algorithms are not very robust in the presence of noise.

2.4 Object Identification using Texture

It is very hard to explain texture, and Ballard and Brown [5] started their chapter about texture with the words:

> *The notion of texture admits to no rigid description, but a dictionary definition of texture as "something composed of closely interwoven elements" is fairly apt.*

These words form the start of a very excellent and fairly complete introduction about texture. It covers various aspects about texture, from texture primitives to statistical models and texture gradients.

A very good summary is also provided by Haralick [30] or its revised version (Haralick [31]). In order to be able to cover most aspects of texture, explaining words are however rarely found in these surveys.

Basic image processing books such as Gonzalez and Woods [27] or Rosenfeld and Kak [80] cover as well the object identification using texture. Gonzalez and Woods [27] explain texture very detailed and examples are often given.

2.4.1 Structural Approach

The Structural Approach is often divided into two parts (Ballard and Brown [5]). Texture Primitives and Structural Models, where the first are the basic textural elements and the second is concerned with their relational occurrence.

Applications are presented for example by Tomita et al. [98], where it can be seen, that actually great care is needed to segment the images in order to achieve the textural elements, which can then be used for structural texture analysis. An application about the definition of tree grammars and their representation for classification can be found in the paper by Lu and Fu [54].

2.4.2 Moments

The very simple, but sometimes not less successful, method of Grey Level Distribution Moments or First Order Moments is best introduced in Gonzalez and Woods [27], where as well the Moment Invariants are covered. The introductory literature to Moment Invariants for textural classification is the same than for Moment Invariants to classify the shape of an object as introduced in Section 2.3.2. The theory is the same, just the applications are different. An application, where as well further normalisation of the Moment Invariants is suggested can be found for example in the paper by Maitra [57].

2.4.3 Spatial Grey Level Dependency Matrix

The first two papers to introduce the Spatial Grey Level Dependency Matrix are Dwyer et al. [21] and Haralick et al. [29]. Dwyer et al. [21] define the five statistical measures of *Energy*, *Entropy*, *Correlation*, *local Homogeneity* and *Inertia* and then apply these measures to some textural patterns. A more theoretical introduction is given by Haralick et al. [29] and several other textural measures are also defined. In almost all applications however only the five already named descriptors are taken. This is the case for example in the paper by Siew et al. [84], where carpet is examined for wear with the aid of textural measures. The measures are also very successfully implemented by Thomas [93], to distinguish between sand quartz grain.

Two comparisons of several different textural algorithms are presented by Weszka [104] and later Conners and Harlow [13]. Weszka [104] compares the earlier mentioned measures based upon the Spatial Grey Level Dependency Matrix with the Fourier Power Spectrum, the Grey Level Difference Statistics and the Grey Level Run Length Statistics. It concludes that the Fourier Features are more difficult to obtain and contain less information. The results obtained with the other measures are similar. Conners and Harlow [13] continue the comparison with a theoretical evaluation and conclude, that the Spatial Grey Level Dependency Matrix contains more textural information than the other methods. However the five commonly used statistical measures do not extract all the information, which is provided by the matrix.

As seen, there is a whole range of different texture measures and He and Wang [34] suggest three further descriptors. On the other hand Vickers and Modestino [100] feed directly the Spatial Grey Level Dependency Matrix into a Log-likelihood classifier. However in real world applications the five Spatial Grey Level Dependency Measures have been as popular as successful.

2.4.4 Other Texture Classification Methods

The following papers were also browsed, however their methods were not considered for inclusion into this work, because they did not seem to be more promising than the earlier mentioned techniques.

Schachter et al. [82] and Ahuya and Rosenfeld [1] use Mosaic Models for textural classification. Region based statistical features are defined and applied to classify texture.

Dapeng and Zhongrong [15] define upon the Spatial Grey Level Dependency Matrix a *Difference Matrix* and use then similar statistical features to classify terrain types.

A spectral approach for obtaining textural features is presented by He and Wang [34]. For the classification of natural texture images, three texture spectrum features are introduced and produce in their application better results than the five statistical features based on the Spatial Grey Level Dependency Matrix.

Another, totally new, approach is the use of Wavelets for textural analysis (Laine and Fan [52]). Future applications will show, if the Wavelet transform is able to extract useful features and to discriminate better than the classic textural features.

The possibility of segmenting an image with the aid of a textural algorithm is investigated by Bouman and Liu [8].

2.5 Classification

2.5.1 Statistical Classification, Discriminant Analysis

In 1936 Fisher [23] published his famous paper on 'The Use of Multiple Mea-

surements in Taxonomic Problems'. He describes a linear function, which makes it possible to discriminate two species of flowers, *Iris setosa* and *I. versicolor*, with the aid of four 'flower measurements'. Since then, much work has been done, to continue his research.

Lachenbruch published several papers or books about discriminant analysis. Lachenbruch and Mickey [49] investigate the estimation of error rates, in particular the U method is recommended. This method was later known as the *leave-one-out* method or estimation by *cross-validation* and is also used in this work. In his book on discriminant analysis Lachenbruch [50] summarises all knowledge to date. Following an introduction on discriminant analysis, with detail of how to calculate the linear discriminant functions, tests on the feasibility of the method are given. The robustness of discriminant analysis is described and also classification into more than two groups, until then hardly used, is investigated. Later Lachenbruch and Goldstein [51] published another summary. This paper mainly emphasizes further work in need of investigation.

Klecka [46] also wrote a book on discriminant analysis, which, in addition to the work covered in the above mentioned literature, introduces Canonical Discriminant Functions. This book can not deny its origin, as it seems to be an extension to the SPSS manual on this particular topic. This is not surprising, as the author was also involved in the writing of the manual. At this stage it should be mentioned that all statistical packages examined provide manuals, and always present short introductions on discriminant analysis.

Duda [17] presents a short, but excellent introduction into classification. A simple problem is presented, features are extracted and then the statistical classifier is derived. Firstly a simple classifier is taken and then improved using Bayes Rule. The first part of the book by Duda and Hart [19] presents a much deeper insight into Bayes Rule and Linear Discriminant Functions. The second part of this book is then concerned with Scene Analysis, for example shape descriptors.

Another book, which introduces and theoretically examines classification procedures is Therrien [92]. Even though practical examples are simplified, the theoretical derivation of different classification procedures is complete and excellent. Pavlidis [70] also designates one chapter for classification. The interesting point about it is, that he explains a simple iterative procedure to calculate the linear discriminant functions. It is surprising that his procedure works as the algorithm is free of any statistical procedures.

2.5.2 Neural Networks

A neural network has been described by Tazelaar [89] as:

> *humanity's attempt to mimic the way the brain does things in order to harness its versatility and its ability to infer and intuit from incomplete or confusing information.*

Compared to discriminant analysis the use of neural networks is a relatively new technique related to Artificial Intelligence. Neural networks can be used for several purposes but, in the following, only classification with the aid of neural networks will be considered and described.

Classification with the aid of neural networks can be achieved directly without feature pre-processing or with the aid of features already established by other methods. Applications of direct classification are quite rare, as direct classification is to date only possible on very simplified problems. An example of such an application is Dayhoff and Dayhoff [16], where a back-error propagation neural network is applied to an image of size 9×7 to distinguish between four types of amoebae nuclei. All the 63 image pixel elements are used as direct input to the single hidden layer of 16 nodes.

More common is the usage of previously extracted features as input into a neural network. This is the way in which a neural network would be used in the

automated detection of algae. There are several examples where neural networks are used to analyze flow cytometry[1] data. An example of the classification of Phytoplankton is presented by Frankel et al. [25]. Similar to the paper by Morris et al. [61], where fungal species are classified, a back-error propagation neural network approach was used. A summary of some application results for the classification of flow cytometry data using neural networks can be found in the paper by Boddy and Morris [7].

Another type of networks, known as Kohonen networks, are used by Wilkins et. at [107] to classify phytoplankton.

2.6 Automated Detection of Biological Organisms

When searching for applications of image processing to problems of recognition it is noticed that much work has been done on objects with fixed shapes, such as aeroplanes (Illing [39], Wallace and Wintz [102]). Examples of object recognition with Fourier Descriptors and the Hough Transform are very common in these areas, because the information given by the shape is very reliable. However such reliable shape information is not so common in the field of biological and medical image processing. This is the case in image topography, for example Höhne et al. [36], where segments of brain are displayed. In addition neural networks (Dayhoff and Dayhoff [16]) are used in medical image processing. Of more interest in the present analysis is the paper by Chow and Kanenko [12], where the detection of boundaries within the human heart is discussed. In addition the use of image processing techniques on blood cells are well researched, for example Parthenis et al. [69] and Harris and Deskoy [33]. In the case of Harris and Deskoy [33] a whole classification algorithm is presented.

[1] see also Section 3.4

However, the above medical applications do not really conform with the present biological application. In the cases above, the images are enhanced, but the final identification is left to the human eye. Where medical applications do identify some objects, it is mainly a simple yes-no decision, as it is in the case described by Ohta et al. [66] and Tomaru et al. [97], where cancer or rosette forming cells are identified.

In the problem of the automated detection of blue-green algae simple yes-no decisions are no longer appropriate and more sophisticated techniques are required. In the case of biological applications it is necessary to consider a combination of variable and fixed shape applications. As in medical applications automated recognition classification cannot be achieved purely on the bases of shape. Similar to fixed shape applications there are a whole range of possible objects to detect.

In the case of biological applications automated recognition has to find an object and then to decide which one out a whole range of possibilities matches the features stored in a database. The only application of this is by Katsinis et al. [43], where a marine organism classification is described. In that case nine features out of fifty are selected to do the final classification. The classification is achieved mainly with the aid of Moment Invariants (Section 6.2) and Fourier Descriptors (Section 6.1). In addition Katsinis et al. [43] as well as Bovik et al. [9] use Morphological Operations (Section 4.4) in pre-processing which prove to be very useful in biological applications.

An example of how a computer can be connected to a microscope and then be able to supervise the scanning of a slide is shown by Courtney et al. [14], where a robot is used to distinguish between two plaque types and then to count them. Such a scanning mechanism could be as well of interest for this work, as the images are grabbed of microscope slides.

The literature search provided several different applications and methods

which can be used for pre-processing or the automated identification and classification of blue-green algae. Literature concerning the whole process of automatic detection of biological organisms with the aid of image processing techniques can hardly be found. However, instead of using image processing, some literature describe the detection of phytoplankton with the aid of a flow cytometer (Edwards et al. [22], Spinrad and Yentsch [87]). Flow Cytometry is further discussed in Section 3.4.

Chapter 3

Water Pollution and Blue-Green Algae

3.1 Water Pollution

Water is very vulnerable to several kinds of pollution. Acids, detergents, gases, heat, metals, oil, pesticides and others, all affect water (Mason [A9]). Therefore the determination of water quality in freshwater lakes and reservoirs is an important task which must be carried out on a regular basis. However, it is not enough just to measure the levels of chemicals in the water at certain times, it is also necessary to obtain information on the biota of water bodies, for these can provide information about the long term water quality. In addition, organisms are also important, because they all belong to a food chain, and a pollutant that has a direct effect on one link in a food chain can have indirect effects on other components in the chain.

Algae can play an important role in the determination of water quality in freshwater lakes, reservoirs and rivers. Whitton [A15] is convinced, that when the algal flora is used to assist a chemical approach it *'may actually be cheaper than an entirely chemical approach'*. Harding and Hawley [A7], who give examples of the monitoring of water quality using algae within the UK and also outline the current and future aims of the National Rivers Authority (NRA), state however that *'where algae have been investigated by water management bodies they have usually been the problem rather than part of the solution'*. Therefore, algal monitoring is to date only commonly used with blue-green algae, who because of their toxicity directly affect water quality.

3.2 Blue-Green Algae

Blue-green algae are also known as *Myxophyceae*, *Cyanophyceae* or *Cyanobacteria*. Recent literature (Carr and Whitton [A4], Mason [A9] and Morris [A10]) prefer to address blue-green algae as *Cyanobacteria*, stressing the biological features of blue-green algae, for instance their cellular organisation. However, in

keeping with the terminology used by the co-operating body, the National Rivers Authority (NRA), the term blue-green algae will be used throughout this thesis.

Note that although blue-green algae occur in soil, on rocks and in the air (Fogg et al. [A6]), it is only those which occur in lakes and rivers that will be considered.

The classification into different algal classes is based on the pigment arrays in the chloroplasts. For example, blue-green algae contain only chlorophyll *a* whilst green algae contain chlorophyll *a* and *b*. The terms blue-green algae or *Cyanophyceae* clearly indicate the colour of the organism, and this is easily recognisable by examining them under a light microscope. When they bloom, they give the water a blue-green or greenish-brown appearance.

Blue-green algae are classed into different species and genera. The NRA identified nine different species/genera of blue-green algae which are important in water quality and they were used in this investigation. They were as follows:

- Anabaena
- Aphanizomenon
- Aphanocapsa
- Gloeotrichia
- Coelosphaerium
- Merismopedia
- Microcystis
- Nostoc
- Oscillatoria

The grouping into these different genera is based on Belcher and Swale [A1]. In this work a typical species is thought to represent one genus and seven species are considered for this investigation. The two species/genera not considered are Gloeotrichia and Aphanocapsa. This was due to problems in obtaining representative samples. However two green algae are included, namely

- Eudorina and
- Pandorina.

It should be noted that in many books, for example Fay [A5] or Morris [A10], different classifications are used. However this does not cause a problem in this work. For example if in future a reduced classification is required, then this

new class can be obtained by merging the above classes, while if an expansion is required, then the principles established here can still be used to take this into account.

The reason why blue-green algae were chosen for this study are as follows. Firstly, the problem of the detection of blue-green algae is a closed problem in a sense that there are only nine different species to consider. Even though it may also be necessary to consider green algae, the problem will remain closed. By working with a relatively small group of algae many principles of automated recognition can be established, and these principles may be extended to other classes in future work.

Secondly, blue-green algae are not only indicators of water quality, but they also present a problem to water quality due to their toxic nature. It is because of this toxicity that the occurrence of these algae is of special interest (NRA [A11]). Some lakes are only monitored by the NRA for this reason.

The toxins which are produced by blue-green algae affect human health as well as that of domestic livestock and wildlife. There have been several reported incidences of the death of fish and of animals which drank water contaminated by blue-green algae (Fogg et al. [A6], Hynes [A8] and Mason [A9]).

When blue-green algae are detected, their density in the water has to be estimated. The following classification of density is used by the NRA:

- If there is *at least one* species of blue-green algae in one millilitre, then they are considered to be *present.*
- If there are *more than twelve* species of blue-green algae in one millilitre, then it is considered as an *algal bloom.*

At both densities the water is classed as toxic and therefore swimming is prohibited and the water cannot be used as tap water or for pets and livestock.

3.3 A Comparison between the Current and Future Methods of Detecting Blue-Green Algae

The NRA carries out a monitoring program to establish and ensure the water quality throughout the country. Currently water is checked for blue-green algae in the following way. A water bottle is filled with approximately one pint of water from the lake or reservoir to be examined. Often, of course, the algal scum is easily recognisable with the human eye without any knowledge of algal type. This bottle is then sent to a NRA Laboratory, and an experienced biologist puts drops of the water onto a microscope slide, using a pipette. This slide is then scanned under a light microscope and examined for the existence of blue-green algae. Every part of the slide has to be focused and carefully examined. Sometimes blue-green algae can be very close to other algae and it is very easy to miss them. Clearly this is an extremely time consuming process, especially when it is carried out frequently and at a wide range of locations. Furthermore the work must be carried out by a qualified biologist.

Identification of algae presents a problem and according to Whitton [A15] *'computer-based identification systems have scarcely been introduced for algae'*. This lack of research was also noticed in the literature search. Hardly any literature was found on this subject. A somewhat similar application to that used in this investigation was performed by Katsinis et al. [43], who used image processing techniques to classify marine organisms. Another, competing method is flow cytometry and it is further discussed in the next section.

The aim of this research is to find a way of automating some of the steps in detecting blue-green algae. Clearly, the filling of a bottle and sending it to the NRA must remain the same. But the person who opens the bottle does not necessarily need to be very experienced with algae. Essentially, some drops

of water have to be placed on a slide and these must be scanned under a light microscope. From this point, the automated recognition process may be carried out as described below.

A computer grabs the microscope picture and stores it as a digital image. Now an algorithm can be applied to the image. Since the images have a different ambient appearance and are often unclear, pre-processing has to be done. For automated recognition the image needs to have a predefined brightness and also has to be enhanced. In this step the possible objects are found and their location is stored. Furthermore, some of the algal features will be extracted and then the organism can be classed into the species of blue-green algae or rejected. The result is printed on the screen and the biologist can take necessary action and continue with their work.

On the SUN-SPARCstation 10 the processing time required to analyze one image is approximately 30s+10s per object found.

3.4 Flow Cytometry

Flow cytometry is a relatively new technique (Edwards et al. [22]), and within the last five years its popularity has increased continuously. A flow cytometer is able to examine fluids with respect to the presence of microorganisms, and works in the following way. The fluid to be investigated is filled into a flow chamber, where the individual cells, in a single stream, pass through a laser or light beam. The effect of the individual cells on the beam is then measured. The measured parameters depend upon the manufacturer of the flow cytometer, but commonly measured parameters include the time of flight (size), light scatter, and the red, green and blue fluorescence. These measures represent features which can then be used to identify the organisms in the water with the aid of classification routines (Chapter 8). Some flow cytometers can also sort the examined cells, depending

upon the classification outcome or some direct features (Edwards et al. [22]).

The price of a flow cytometer is around £100,000 depending on the specification. Flow cytometers are able to examine large numbers of samples very quickly and efficiently, as they provide real time analysis. The speed to process the samples is approximately 10^3 cells per second.

More background reading can be found in the already mentioned paper by Edwards et al. [22] or in Boddy and Morris [7], who present a neural network approach to classify flow cytometry data.

With respect to this work a flow cytometer would take the part of the image processor. Instead of measuring features based upon the shape and texture of an object, as shown in the following chapters, flow cytometry introduces totally new, non morphometric features. A flow cytometer would be able to detect algae in water and also could certainly examine a large number of samples more quickly. However in this work it was the aim to identify the examined species by using the biological key, hence similar to a biologist. In addition, when only few samples are investigated, as is the case in the NRA Laboratory in Llanelli, the use of an already present microscope camera coupled with a computer is a more sensible choice.

Chapter 4

The Image Segmentation of Algae

When a computer is used to identify important features, image segmentation is always the first process to be employed. Segmentation means the identification of the foreground and background of an image and also the extraction of the edges in an image. Segmentation is the necessary pre-processing in order to provide data for the use of further Pattern Recognition/Image Processing routines. Image segmentation is used here to identify the location of probable objects and their boundaries. It should be noted however, that image segmentation works closely with image enhancement. Together they provide the information on which further identification routines will be applied.

The following segmentation routines are discussed here. The basic technique of Thresholding, which converts a grey level image into a binary image, is introduced. In contrast to Thresholding, where only the grey level value of the individual pixels is of interest, Edge Detection compares the grey level values of each pixel with the grey levels which occur in the surrounding area. There exist several different Edge Detectors which treat this surrounding area differently. These detectors and also a newly developed technique, LoG Thresholding is then described. It was necessary to develop a new technique, because the traditional Thresholding and Edge Detection techniques could not segment all the different objects sufficiently.

The Morphological Operations which have the ability to change the object outline according to several rules are then introduced. Defined on binary images they help to locate the boundary more accurately. They are extended in Chapter 5 for grey scale images. The final step in the segmentation process is Boundary Tracing.

Finally, it should be mentioned that the segmentation of biological images presents a very difficult task. The difficulties are as follows:

- The object could appear anywhere in the image. No point was *a priori* known to belong to the object rather than the background.

- There could be more than one object in the image.
- The objects could have any shape.
- The objects could have any size.
- The occurrence of the grey levels inside an object could vary from species to species or even from observation to observation.
- The frequency of occurrence of the grey levels over the whole image (histogram) varied from image to image and did not provide much information.

Image Segmentation has to take all these points into account and provide an accurate location of all occurring objects within the images.

Segmentation is a very difficult task and is always the first problem which needs to be overcome. Without a reliable segmentation any further processing is not possible. It was felt here, that the traditional Thresholding and Edge Detection routines did not process the images adequately enough and hence great effort has been spent to develop a new routine, LoG Thresholding, which could separate foreground and background efficiently.

4.1 Thresholding

A digital image consists of several grey levels, for example on the SUN-Workstation 256 grey levels are employed. Thresholding is the reduction of an image to a binary image, normally the minimum and maximum grey level. In order to isolate an object from the background, Thresholding is a well known technique.

In the following $0 \leq f(x,y) \leq 255$ will denote the grey level of an image pixel at the position (x,y). In addition $h(x,y)$ will denote the application of some algorithm on the original image $f(x,y)$. In the use of Thresholding, $h(x,y)$ will be formed according to the rule:

$$h(x,y) = \begin{cases} 0 & \text{if } f(x,y) < T \\ 255 & \text{otherwise} \end{cases} \tag{4.1}$$

where $0 \leq T \leq 255$ is known as the threshold value. The pixels with the value of 0 represent the object, all the others the background or vice versa.

It can be distinguished between *local* and *global* Thresholding. In local Thresholding the image will be divided into windows and Equation (4.1) will be applied for each window with a varying T. In global Thresholding T remains the same over the whole image. Local Thresholding is necessary, if the appearance of the background changes and therefore the threshold value needs adjusting to these local needs. For example Nakagawa and Rosenfeld [63] used local Thresholding with good success, however it was shown by Taxt and Flynn [88], that if global Thresholding is sufficient it normally provides a better result.

The grabbed microscope images of blue-green algae had a similar background appearance over the whole image, and an object could stretch through the whole image also. For this reason global Thresholding was thought to be sufficient.

The main problem concerning Thresholding is to find a suitable threshold,

and Sahoo [81] considers several approaches. The two most common methods are by trial and error or to select a threshold based on the image grey level histogram, which gives the frequency of occurrence of the pixel intensities. The histogram should clearly separate object and background and the threshold can then be set to a value corresponding with the valley in between. Mostly, the histogram does not provide enough information for Thresholding and it has to be smoothed or modified as for example by Jiang et al. [42] and also Nakagawa and Rosenfeld [63].

For many biological images, including those in this application, these basic Thresholding methods do not adequately separate the object from the background. This was proven by interactively selecting several thresholds and then considering their result. In addition, the algorithm should work without any user interaction and it was found, that setting automatically a threshold would introduce even more problems and was not possible in the general case.

4.2 Edge Detection

The identification of edges is a difficult process since boundaries are often blurred. However in this application a well known definition will be used.

An edge is a set of pixels. An edge pixel is found by evaluating the accumulated weighted differences in grey levels with the neighbourhood and determining which differences are greater than a set threshold.

The main problem in this definition is to determine which neighbours to take into account and then what weight to give them. For example the more distant the neighbours, the bigger are the arrays which need to be stored and the longer the processing time. However the larger the mask, the less sensitive is the algorithm against the pixel noise.

In the following, the Sobel Edge Detector and the Laplacian of the Gaussian

will be introduced. The Sobel Edge Detector was chosen because of computer efficiency and its noise reducing properties. The Laplacian of the Gaussian, although more costly in computer time, has even more noise suppression characteristics and has a wider range of applications.

4.2.1 Sobel Edge Detector

The Sobel Edge Detector is defined by:

$$h(x,y) = \mid h_x(x,y) \mid + \mid h_y(x,y) \mid \tag{4.2}$$

where:

$$\begin{aligned} h_x(x,y) = \; & -1 \cdot f(x-1,y-1) - 2 \cdot f(x,y-1) - 1 \cdot f(x+1,y-1) \\ & +1 \cdot f(x-1,y+1) + 2 \cdot f(x,y+1) + 1 \cdot f(x+1,y+1) \end{aligned} \tag{4.3}$$

and

$$\begin{aligned} h_y(x,y) = \; & -1 \cdot f(x-1,y-1) - 2 \cdot f(x-1,y) - 1 \cdot f(x-1,y+1) \\ & +1 \cdot f(x+1,y-1) + 2 \cdot f(x+1,y) + 1 \cdot f(x+1,y+1) \end{aligned} \tag{4.4}$$

h_x detects the location of horizontal edges, whereas h_y corresponds to vertical edges. The corresponding masks are shown in Figure 4.1.

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

Figure 4.1: The Two Masks of the Sobel Edge Detector

The reference point is always the centre of each mask. Mask 4.1(a) detects the horizontal edges $h_x(x, y)$ and mask 4.1(b) the vertical edges $h_y(x, y)$.

While the Sobel Edge Detector is very efficient, the accuracy of the edges decrease in the presence of noise. Plate A.12 shows an image of *Anabaena* after the Sobel Edge Detector was applied. In order to see the edges more clearly, the image was Contrast Stretched (Section 5.5) to ensure that the full dynamic range of the computer was used.

4.2.2 Laplacian of the Gaussian Operator as an Edge Detector

Marr and Hildreth [59] were the first to introduce the Laplacian of the Gaussian operator (LoG) as an edge detector. They used the two-dimensional operator defined by:

$$LoG(i,j) = \frac{1}{\pi\sigma^4}\left(1 - \frac{i^2+j^2}{2\sigma^2}\right) e^{-(i^2+j^2)/(2\sigma^2)} \tag{4.5}$$

where i and j are the signed distances in x and y direction from some reference point (x_0, y_0). Thus edges are found according to the convolution:

$$h(x,y) = \sum_i \sum_j LoG(i,j) \cdot f(x+i, y+i) \tag{4.6}$$

Examination of Equation (4.5) shows that σ affects the variance of the operator and clearly the choice of its value determines the effectiveness of this edge detection procedure. In this work it was found that $\sigma = 0.5$ results in a good edge detector. Its mask is shown in Figure 4.2. The 5×5 mask size is derived from Equation (4.5) and clearly all pixels further away from the reference point are of no great importance, because the values of the LoG operator become insignificant.

The LoG operator is more effective against noise than the Sobel Edge Detector, however the LoG operator requires more processing time. It is important

-0.0000	-0.0021	-0.0120	-0.0021	-0.0000
-0.0021	-0.2798	-0.6893	-0.2798	-0.0021
-0.0120	-0.6893	5.0930	-0.6893	-0.0120
-0.0021	-0.2798	-0.6893	-0.2798	-0.0021
-0.0000	-0.0021	-0.0120	-0.0021	-0.0000

Figure 4.2: The Laplacian of the Gaussian, $\sigma = 0.5$

to justify the use of increased processing time for each application.

4.2.3 Setting a Threshold

Once an edge detector is applied to an image, the resulting image can have grey level values over the whole possible dynamic range. The higher the pixel value, that is the brighter it appears, the higher the probability, that this pixel is an edge. In order to distinguish between *edge* and *no edge*, a threshold has to be applied on the resulting image. As stated before (Section 4.1), this threshold has to be selected manually or determined with the aid of the grey level histogram.

In general, the use of the Sobel Edge Detector or the Laplacian of the Gaussian as an edge detector is very effective. For example Plate A.12 shows very clearly the object boundaries and other occurring edges within the input image, Plate A.1. However, for the location of the object boundaries, the result needs to be thresholded. It was then found, that in this application edges were not adequately located. This was primarily due to the low resolution of the input image

and the presence of image noise. The edges in the resulting image were found to be discontinuous and the noise was still present. Some discontinuities could be already found in the edge detected image (Plate A.12), before the thresholding was applied. In fact, the results were not very promising and therefore alternative techniques were considered. It was found, that a better approach was based upon a combination of LoG Thresholding and Boundary Tracing.

4.2.4 The use of Edge Detection to Enhance the Appearance of Edges

Rather than detecting the edges for segmentation purposes it is also possible to use edge detection to enhance the appearance of an image. Edge detectors can also be used to give edges a more darker appearance.

This is simply done by applying an edge detector to an image and then subtracting the result form the original. This results in darker pixel values at places were the edge detector produced a high output.

This method was used to separate the object more clearly from the background and to make further segmentation more safe, irrespective of the appearance of the input image. The chosen edge detector was the Sobel Edge Detector. The Sobel Edge Detector provided very fast results which were good enough for this particular purpose.

4.3 LoG Thresholding

The traditional Thresholding techniques and the various Edge Detection Operators achieved inadequate results and therefore a new segmentation technique had to be developed and tested. This technique which is now explained in detail was especially developed during the course of this research for this particular

problem of the segmentation of blue-green algae and to cater for the special needs involved.

LoG Thresholding is a form of Thresholding based on the LoG operator, Equation (4.5). The actual value of σ effects the role of the LoG operator in image processing. For example $\sigma = 0.5$ enables the LoG operator to be used as an edge detector, however values of σ around one provide a High Pass Filter (Section 5.3.2), whereas larger values enable image smoothing.

This is very surprising as the value of σ does not affect the shape of the LoG operator. σ represents the variance of the operator. However the difference can clearly be seen by analysing the plots of the LoG operator with different values of σ (Figure 4.3). These plots represent a cross-section of the two-dimensional operator. Because image convolution is a digital operation, only distinct values of the curve are taken. These distinct values are highlighted in Figure 4.3. By only considering these distinct values and omitting the global curve shape, the difference which appears after applying the LoG operator with different values of σ is obvious.

In particular in this investigation the LoG operator with a high σ was found to be very useful to threshold an image. As can be seen from Equation (4.5), a value of $\sigma = 4.75$ would normally require a mask of 25×25 elements and, in this particular case, the sum of the mask elements would be approximately zero, resulting in the convoluted image disappearing from view. However, if only a truncated mask size of 9×9 (Figure 4.4) is considered, so only positive elements are taken, the sum of the mask elements is 0.0282 and the resulting operation may be used for Thresholding.

In order for the algorithm to be capable of Thresholding, it is necessary to standardise the mean of all the sample images. A mean of 75 was chosen simply by adding the difference between 75 and the old mean to each pixel. This operation is possible on the images without any loss of information, because all

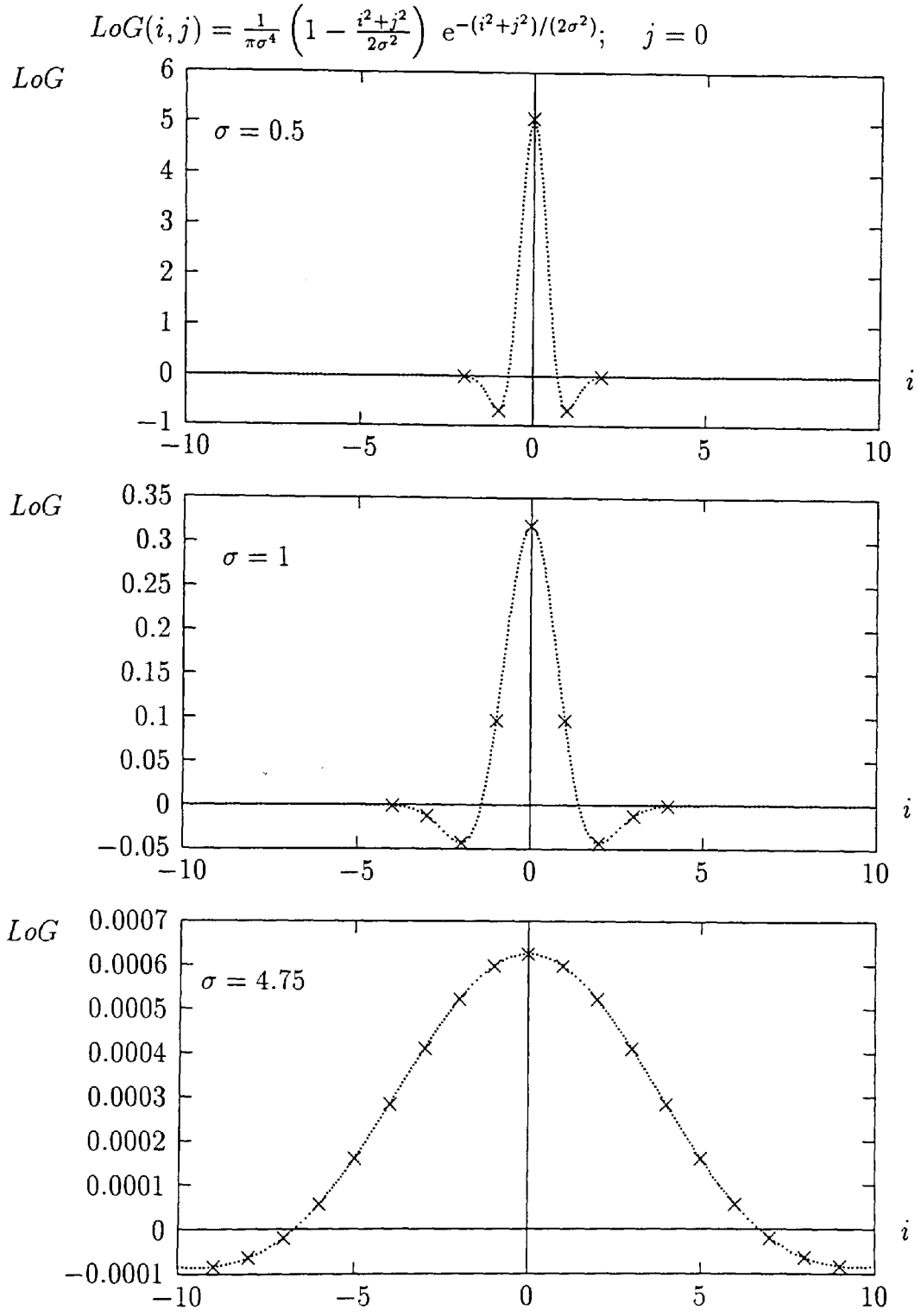


Figure 4.3: Cross-Section of the LoG Operator with Different Values of σ

0.0001	0.0002	0.0002	0.0003	0.0003	0.0003	0.0002	0.0002	0.0001
0.0002	0.0003	0.0003	0.0004	0.0004	0.0004	0.0003	0.0003	0.0002
0.0002	0.0003	0.0004	0.0005	0.0005	0.0005	0.0004	0.0003	0.0002
0.0003	0.0004	0.0005	0.0006	0.0006	0.0006	0.0005	0.0004	0.0003
0.0003	0.0004	0.0005	0.0006	0.0006	0.0006	0.0005	0.0004	0.0003
0.0003	0.0004	0.0005	0.0006	0.0006	0.0006	0.0005	0.0004	0.0003
0.0002	0.0003	0.0004	0.0005	0.0005	0.0005	0.0004	0.0003	0.0002
0.0002	0.0003	0.0003	0.0004	0.0004	0.0004	0.0003	0.0003	0.0002
0.0001	0.0002	0.0002	0.0003	0.0003	0.0003	0.0002	0.0002	0.0001

Figure 4.4: The LoG with $\sigma = 4.75$ and a Mask Size of 9×9

the images were dark and of low contrast. In addition, note that the sum of the mask elements multiplied by the mean pixel intensity results in a value of approximately two, that is:

$$0.0282 \times 75 \approx 2.1$$

This is important, because a typical pixel intensity value after convolution will be two. Such a typical pixel represents the image background. After the convolution, the darker organism pixels will have a pixel intensity of one. In addition, pixels close to the border, which were black due to the camera, remained black and so had a pixel value of zero. The resulting image could then easily be thresholded, setting all the pixels with the pixel value of one to zero and all the others to 255.

As a summary it can be said that this algorithm is a global Thresholding technique, where not only a single pixel value but a whole region is taken into account. The real advantage of this algorithm is based on the fact that it is fully automated and yet still very reliable.

While the use of a truncated LoG operator has not appeared in the literature before, the use of the LoG operator with large values of σ is not uncommon (Nishihara and Crossley [65]). There have been some algorithms developed, for example the use of a sequence of one-dimensional convolutions instead of a the cost expensive multi-dimensional convolution (Wiejak et al. [106]). In addition Chen et al. [11] and also Sotak and Boyer [86] propose the use of the spectral domain to achieve a reduction of the resolution. Both methods speed up the computing process with big LoG masks, which are required because of large values of σ.

However, the truncation of the LoG, which also speeds up the processing time, is not mentioned in any papers found. After the truncation of the mask to a size of 9×9 a further speeding up of the process was not undertaken, as a fast convolution algorithm was provided by the image processing software.

It should be noted here that the truncation of the LoG with $\sigma = 4.75$

appears to result in a similar mask as a raised cosine would produce. However, the raised cosine was also considered in this application but the LoG operator provided better results.

As said earlier the LoG Thresholding proves to be very effective, but there still remained some problems after the LoG Thresholding was applied to an image. The organism found, had sometimes discontinuous boundaries and the boundary at some parts was not very smooth. In order to overcome these problems Morphological Operations were applied.

4.4 Morphological Operations on Binary Images

The Morphological Operations (MO) are well known techniques for the segmentation and enhancement of images and also for recognition purposes. For image segmentation it is only necessary to consider binary images and the MO are significantly simpler to apply. Therefore the MO are introduced in this section on binary images only.

MO for the enhancement of images, i.e. MO on grey scale images will be introduced in the next chapter. The possibility of automated recognition with MO as shown by Werman and Peleg [103] is not covered within this thesis. This was mainly because the automated detection with MO did not seem to give better results than obtained through other routines.

Traditionally MO are set operations and the MO operations are commonly introduced on sets in an N-space (Serra [83]). However when they are applied on images only, such a global and theoretical definition would lead to more confusion than necessary. Therefore for this thesis they are only defined and applied on images.

When MO are applied on binary images they change the boundary of an object according to their rules. Therefore the MO on binary images are clearly image segmentation routines. The MO on grey scale images are introduced in Chapter 5, because in this case they change the appearance of the different grey levels, that means they can be used to enhance images.

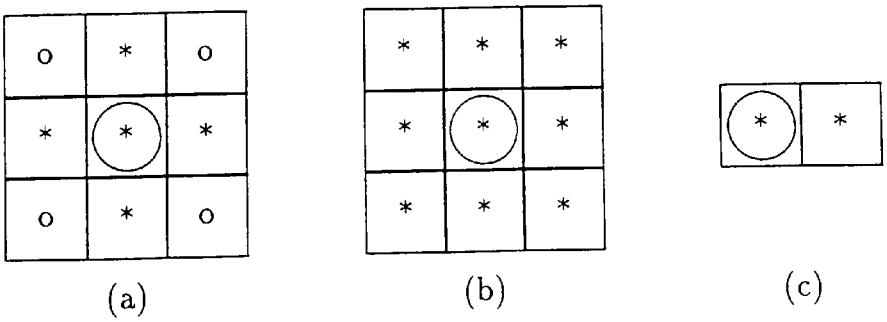
4.4.1 The Erosion Operation

The Erosion of an image is defined by:

$$f(x,y) \ominus m(i,j) = \begin{cases} * & \text{if } \forall_{i,j}\ m(i,j) = * \wedge f(x+i, y+j) = * \\ o & \text{otherwise} \end{cases} \tag{4.7}$$

where * is the grey level value of the object, o the value of the background and $m(i,j)$ the mask of the erosion. In terms of practical implementation, Equation (4.7) means that a mask is placed over an image and if all the mask pixels of intensity * correspond to image pixels with * then the output image pixel will have the intensity * at the reference point of the mask. Otherwise the result is o.

The Erosion results in the thinning or shrinking of an object. Some common Erosion masks are shown in Figure 4.5, although any binary mask can be used for eroding an image. The circled elements represent the reference point $(i,j) = (0,0)$ of the Morphological Operation. The morphological masks are sometimes referred to as the structuring element.



(a) (b) (c)

Figure 4.5: Masks for Morphological Operations

Figure 4.5(a) shows a 4-connect Erosion mask whereas Figure 4.5(b) transforms an 8-connect Erosion. The simple Erosion mask 4.5(c) changes all the object elements which lie on the right border to background.

4.4.2 The Dilation Operation

Dilation is used to fill or expand a region. The operation of Dilation and Erosion are closely related. Dilation is defined by:

$$f(x,y) \oplus m(i,j) = (f(x,y)^c \ominus m(i,j))^c \tag{4.8}$$

where $f(x,y)^c$ is the negative or complement image of $f(x,y)$. Therefore Dilation of the background is equivalent to Erosion of the foreground (object) and vice versa. The equivalent of the definition which was used for Erosion would be:

$$f(x,y) \oplus m(i,j) = \begin{cases} * & \text{if } \forall_{i,j}\ m(i,j) = * \vee f(x+i,y+j) = * \\ o & \text{otherwise} \end{cases} \tag{4.9}$$

Most image processing packages which have a pre-programmed MO routine treat the grey level value 0 as the background value and the other value as the object value. If this is the case, however the object has actually the grey level value 0, then the pre-programmed Erosion transforms a Dilation and vice versa. This happened in this application of blue-green algae on the SUN-IP also.

4.4.3 The Morphological Open and Close Operations

It should be noted that Dilation and Erosion are dual operators, however they are not inverse operations of each other. Dilation after Erosion or Erosion after Dilation does not recover the original image.

The Closing operation is defined to be Dilation followed by Erosion.

$$f(x,y) \bullet m(i,j) = (f(x,y) \oplus m(i,j)) \ominus m(i,j) \tag{4.10}$$

Similarly, the Opening operation is taken to mean Erosion followed by Dilation.

$$f(x,y) \circ m(i,j) = (f(x,y) \ominus m(i,j)) \oplus m(i,j) \tag{4.11}$$

Closing is an important operation as it is the operation which fills some holes and connects certain discontinuities (Bovik et al. [9]). Opening is used to eliminate small details (Katsinis et al. [43]). Both operations smooth the image boundaries.

The Morphological Operations are illustrated in Figure 4.6.

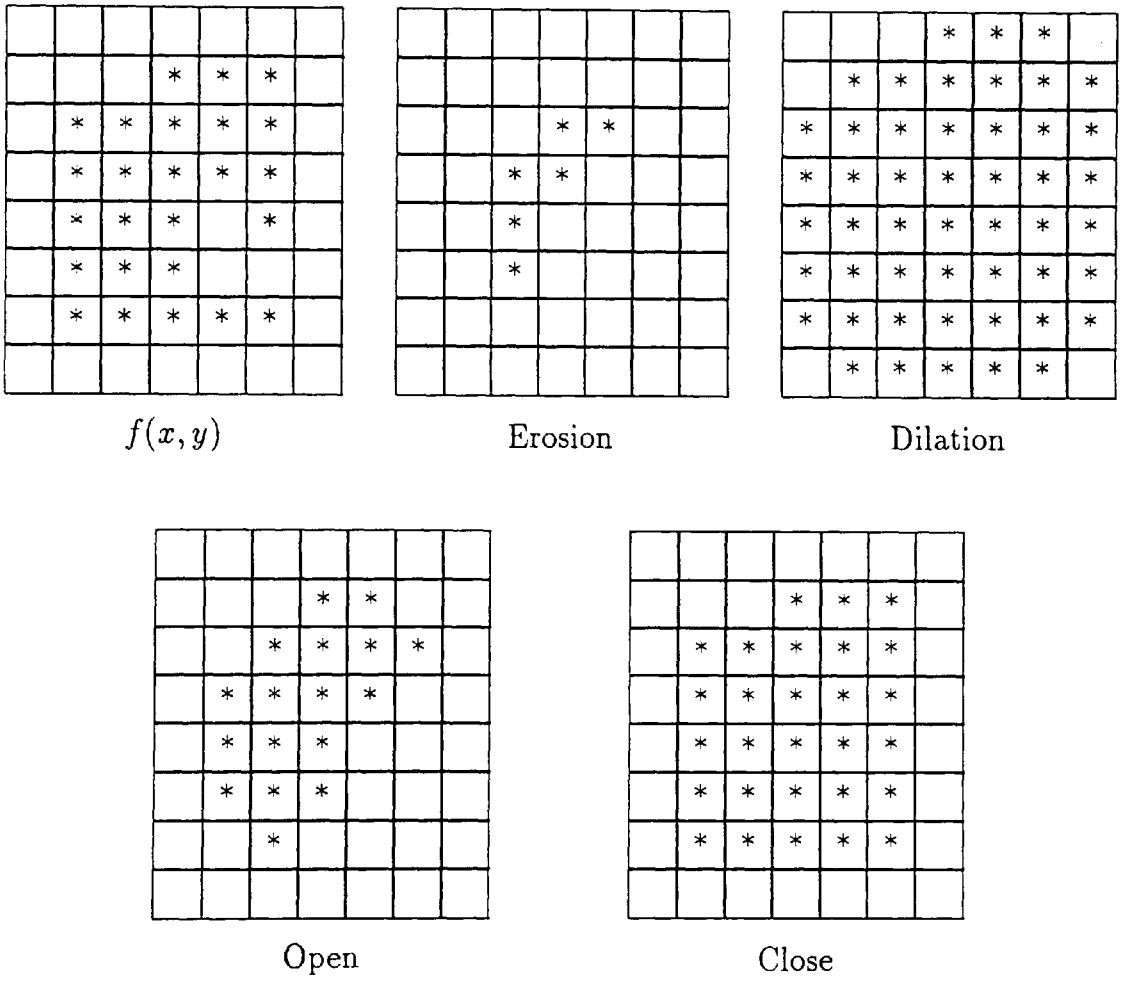


Figure 4.6: Example of the Four Morphological Operations with 4-Connect Structuring Element

4.4.4 The Application of Morphological Operations

In this work, the Closing operation was applied to connect discontinuities in the LoG Thresholded image and to give a smoother boundary. Experimental investigations were carried out with different masks in respect to shape and size. The mask illustrated in Figure 4.7 produced the most effective result and is used to assist in the segmentation process.

o	o	o	o	*	*	*	o	o	o	o
o	o	o	*	*	*	*	*	o	o	o
o	o	*	*	*	*	*	*	*	o	o
o	*	*	*	*	*	*	*	*	*	o
*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	(*)	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*
o	*	*	*	*	*	*	*	*	*	o
o	o	*	*	*	*	*	*	*	o	o
o	o	o	*	*	*	*	*	o	o	o
o	o	o	o	*	*	*	o	o	o	o

Figure 4.7: The Structuring Element for the Morphological Operation

If, like in this case, the structuring element exceeds the size of 3×3 it is advisable to produce the MO in steps to speed up the convolution. Equation (4.12) is valid for Erosion and Dilation and for all images and structuring elements.

$$f \oplus (m_1 \oplus m_2) = (f \oplus m_1) \oplus m_2 \tag{4.12}$$

For a proof please refer to Katsinis and Poularikas [44].

Therefore, the Erosion or Dilation with the mask shown in Figure 4.7 should be carried out by eroding or dilating the image 3 times with the 8-connect struc-

turing element followed twice by the operation with the 4-connect structuring element.

The combined effect of LoG Thresholding and Morphological Operations is shown in Plate A.13.

4.5 Boundary Tracing

The boundary of an object contains important information which is used in later algorithms to identify the organism.

Since the images have a binary form, it is possible to identify the boundary easily by Boundary Tracing. This is achieved by scanning the image for the first object point and then tracing around the boundary of the organism in order to detect the boundary pixels. The boundary is thought to be 8-connect, that is two neighbouring pixels could be on a diagonal also.

The Boundary Tracing algorithm is described for example by Pavlidis [72]. This algorithm is used in order to find object boundaries after LoG Thresholding and a morphological close was applied to the images.

When it is possible that there is more than one object in one image, the Boundary Tracing has to be carried out in a loop, where after one object was found it has to be deleted from the input image in order to make it possible to continue to trace for more objects. This is actually a difficult process because when deleting the object from the input image it must be made certain, that no parts of other objects will get deleted also. The implementing of the Boundary Tracing algorithms with the ability to find more than one object was done as follows:

- tracing around the boundary. The boundary was traced as an 8-connect boundary.

- if the boundary length was less than 200 pixels it is assumed that the found object was because of the small size just a background clutter. The background clutter was removed by simply taking the smallest rectangle around the clutter and than deleting this rectangle in the input image.
- if the boundary was longer than 200 pixels the inside of the object was scanned for holes with the same algorithm. Then the object was removed from the input image.

Always when one object was found through boundary tracing the automated detection algorithm was applied on that object and the result stored. Only after this step was finished the Boundary Tracing was continued to find another object.

4.5.1 Removing an Object from the Input Image

As described above, when an object was found and the boundary was traced it needed to be removed from the input image in order to search for the next possible object. This removing is anything but trivial, and has to be done very carefully.

A small object, that is an object with a boundary length of less than 200 points, is just removed by taking the smallest rectangle around the object and then deleting the whole rectangle. This is a very fast method, however for larger objects the rectangle would cover too many non object points which are probable points for other objects and could therefore destroy data.

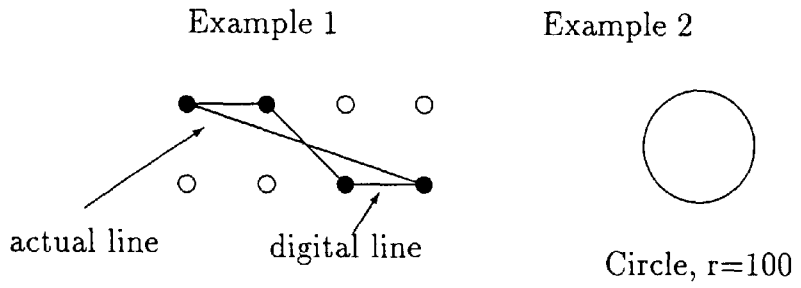
The obvious way of removing a large object is to fill the inside of the object and then removing all the filled and boundary points. Several filling algorithms are described by Pavlidis [72]. However there is one difficulty which was impossible to overcome: in order to fill an object a starting point for the filling operation has to be specified. This starting point must lie inside the boundary of the object and in fact it was not possible to find a method to automatically decide whether

some points were inside or outside the boundary. However, it is possible to find a point outside the boundary. Outside the boundary are all the points which are outside the smallest rectangle which fits around the boundary. If a fill is performed from such a point and then the boundary is removed a negative of the wanted result is achieved. With this negative the object can be removed from the input image.

The type of fill algorithm chosen was the 4-connect non-recursive connectivity filling described by Pavlidis [72]. Always when there is a 8-connect boundary, the fill is required to be 4-connect and vice versa. A 8-connect fill on a 8-connect boundary would leak through the boundary, whereas a 4-connect fill on a 4-connect boundary would not fill the whole object.

4.5.2 The Length of the Boundary

One important point which should never be underestimated is the digital character of the boundary and the actual inaccuracy of the boundary and the boundary length. In the above tracing algorithm the boundary consists of 8-connect points, that is that two neighbouring points have either the distance 1 or $\sqrt{2}$. Summing all the distances between the points provides the length of the boundary. The two examples in Figure 4.8 show the discrepancies between the digital and a real line. The researcher should always be aware of this problem when applying algorithms involving the length of the boundary.



Digital Length:	$1 + \sqrt{2} + 1 = 3.414$	664 (experimental result)
Actual Length:	$\sqrt{3^2 + 1^2} = 3.162$	$2\pi r = 628$

Figure 4.8: The Error Involved with a Digital Boundary

Chapter 5

The Enhancement of Algal Organisms

In Chapter 4, algorithms defining the boundary are presented. However, for the classification of algae it is also necessary to identify features within the boundary. In order to do this, enhancement of the object pixels is necessary.

This enhancement is the second step of the pre-processing. The enhancement should never be treated alone, as it is the further application which requires it to be carried out. The better the further application works with an enhanced image, the better is the enhancement. Apart from that, the enhancement also has to work closely with the segmentation. Sometimes parts of the enhancement are carried out before the segmentation in order to improve the segmentation. In this application, the Neighbourhood Averaging was done parallel to the initial segmentation.

The enhancement can be classed into different groups. *Spatial* enhancement or enhancement with the aid of the *frequency domain.* Masks are mainly used for spatial enhancement purposes, whereas the enhancement in the frequency domain requires the Fourier Transform. For this reason the Fourier Transform is introduced in the next section.

Furthermore the enhancement techniques with Low and High Pass Filters, which can be done in the frequency domain or direct with masks are described in this chapter. The Low Pass Filter smooths the image whereas the High Pass Filter sharpens the image. The Morphological Operations, which were already introduced in Chapter 4 on binary images are extended to grey level images. Finally two further enhancement techniques, Contrast Stretching and Histogram Equalisation are explained at the end of this chapter. Contrast Stretching is simply a mapping of current grey level values to new grey levels. Histogram Equalisation is also a mapping of grey levels, however the frequency of occurrence of each distinct grey level is taken into account.

5.1 Fourier Transform

In the design of Low and High Pass filters for the enhancement of images Fourier Transform techniques are often used. The Fourier Transform (FT) $\mathcal{F}$ of a function $f(x)$ is defined by:

$$F(u) = \mathcal{F}\{f(x)\} = \int_{-\infty}^{\infty} f(x)\, e^{-j2\pi ux} dx \tag{5.1}$$

where $j = \sqrt{-1}$. The function f(x) can be obtained from F(u) with the inverse FT, $\mathcal{F}^{-1}$ where:

$$f(x) = \mathcal{F}^{-1}\{F(u)\} = \int_{-\infty}^{\infty} F(u)\, e^{j2\pi ux} du \tag{5.2}$$

The FT exists, if $f(x)$ is continuous and integrable and $F(u)$ integrable. The discrete forms of Equations (5.1) and (5.2) are defined as:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)\, e^{-j2\pi ux/N} \tag{5.3}$$

and

$$f(x) = \sum_{u=0}^{N-1} F(u)\, e^{j2\pi ux/N} \tag{5.4}$$

It should be noted, that in the definition (5.1) and (5.2) x and u are independent continuous variables, whereas in the discrete form (5.3) and (5.4) x and u are discrete independent values. The conflict in notation will not prove to be confusing as we shall be using the discrete definition in the following work. Furthermore the discrete FT assumes that $f(x)$ is a periodic function.

The FT has many applications in signal processing, whereas in Image Processing the two-dimensional FT is of more interest. The discrete two-dimensional FT and its inverse are shown below:

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(ux+vy)/N} \tag{5.5}$$

$$f(x,y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi(ux+vy)/N} \tag{5.6}$$

In practice, these formulae will never be applied directly. If it is assumed that $N = 2^n$, the complexity of the calculations can be reduced significantly with an algorithm called the Fast Fourier Transform (FFT). A program listing of the FFT can be found for example in Gonzalez and Woods [27] and all common signal processing books. However, when transforming an image, the value of N is at least 512, and so with the FFT this transformation takes a considerable amount of time.

The images used in this research have a size of 768×575 pixels and applying the FFT on only 512×512 pixels loses information. However, by taking the larger value $N = 1024$, a transformation of 1024×1024 takes a very long time and proves to be impractical. Furthermore, such an image needs to have floating point pixel values defined and, therefore the storage space required for such an image is $1024 \times 1024 \times 4 \approx 4.2$Mb, which is approximately 10 times more than the original input image.

After applying the FFT on an image $f(x, y)$ the result $F(u, v)$ gives the frequency description. The frequency domain shows the phase and amplitude of an image structure. The enhancement of an image by means of filtering and use of the frequency domain is well known and illustrated in Figure 5.1. In Figure 5.1 $h(x, y)$ is the final enhanced image and $H(u, v)$ is its Fourier Transform.

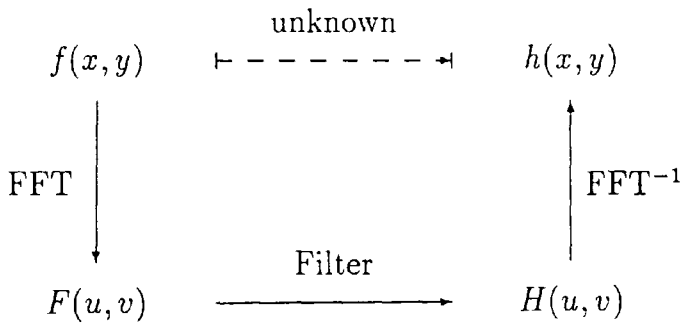


Figure 5.1: Filtering in the Frequency Domain

5.2 Low Pass Filtering

A Low Pass Filter (LPF) blurs or smooths an image. It can also be used to reduce noise in an image.

5.2.1 LPF using the Frequency Domain

Edges and sharp changes in grey level values in an image result in high-frequency components in its frequency domain. Blurring or smoothing can be done by suppressing these high frequency components.

The ideal LPF is defined by:

$$H(u,v) = \begin{cases} F(u,v) & \text{if } d(u,v) \leq d_0 \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

where $d(u,v)$ is the distance in the frequency domain from the origin and d_0 the threshold. With this technique all high frequency components will be filtered out and only low frequency components will be preserved. In practical implementations the ideal LPF is not used. Smooth approximations are used instead.

5.2.2 LPF using Convolution

In most applications LPF is done by direct convolution of the image values rather than using the frequency domain. Typical convolution masks are the Gaussian mask (Figure 5.2(a)) and the uniform mask (Figure 5.2(b)).

Note that the mask used for LoG Thresholding is similar to the mask in Figure 5.2(a).

0.0449	0.1221	0.0449
0.1221	0.3319	0.1221
0.0449	0.1221	0.0449

(a)

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

(b)

Figure 5.2: Masks for Low Pass Filtering

5.3 High Pass Filtering

The effect of a High Pass Filter (HPF) is to sharpen the edges and enhance some details in an image.

5.3.1 HPF using the Frequency Domain

A HPF using the frequency domain is very similar to the LPF. A HPF preserves only the high frequency components in an image. The ideal HPF is defined by:

$$H(x,y) = \begin{cases} F(x,y) & \text{if } d(x,y) > d_0 \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

Note the similarity between the Equations (5.8) and (5.7). Again, only smoothed versions of Equation (5.8) are used in practice.

5.3.2 HPF using Convolution

Matrix based High Pass Filtering can be done with the masks shown in Figure 5.3 and 5.4.

High Pass Filtering with mask 5.3(a) gave good results, in the sense that

0	-1	0
-1	5	-1
0	-1	0

(a)

-1	-1	-1
-1	9	-1
-1	-1	-1

(b)

Figure 5.3: Masks for High Pass Filtering

-0.0000	-0.0000	-0.0001	-0.0005	-0.0007	-0.0005	-0.0001	-0.0000	-0.0000
-0.0000	-0.0003	-0.0026	-0.0086	-0.0124	-0.0086	-0.0026	-0.0003	-0.0000
-0.0001	-0.0026	-0.0175	-0.0392	-0.0431	-0.0392	-0.0175	-0.0026	-0.0001
-0.0005	-0.0086	-0.0392	0.0000	0.0965	0.0000	-0.0392	-0.0086	-0.0005
-0.0007	-0.0124	-0.0431	0.0965	0.3183	0.0965	-0.0431	-0.0124	-0.0007
-0.0005	-0.0086	-0.0392	0.0000	0.0965	0.0000	-0.0392	-0.0086	-0.0005
-0.0001	-0.0026	-0.0175	-0.0392	-0.0431	-0.0392	-0.0175	-0.0026	-0.0001
-0.0000	-0.0003	-0.0026	-0.0086	-0.0124	-0.0086	-0.0026	-0.0003	-0.0000
-0.0000	-0.0000	-0.0001	-0.0005	-0.0007	-0.0005	-0.0001	-0.0000	-0.0000

Figure 5.4: The Laplacian of the Gaussian, $\sigma = 1$

the visual effect of this convolution was very good. However, enhancement by means of Neighbourhood Averaging (Section 5.3.3) will give even better overall results.

5.3.3 Neighbourhood Averaging

Neighbourhood Averaging (NA) is implemented in the following way:

$$h(x,y) = f(x,y) - \sum_m \sum_n g(x-m, y-n) f(x,y) + c \tag{5.9}$$

By applying NA, the output image $h(x,y)$ is obtained from a smoothed version of the original, subtracted from the original image. A constant c is added to ensure positive values. In this case $c = 128$ was chosen.

The smoothing filter may be one-dimensional of length $2N + 1$ elements as given by Figure 5.5 or it may be two-dimensional where averaging is often implemented using 3×3 masks (Figure 5.6).

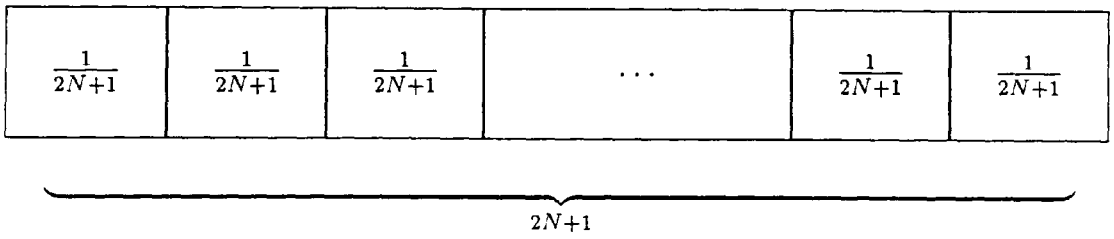


Figure 5.5: One-Dimensional Neighbourhood Averaging

In the case study by the Open University [67], where the enhancement of TV-video images of Mars, which were recorded and send to earth from the Mars orbiter Mariner 9, are described, the one-dimensional mask with a value of $N = 50$ is used. In this work the one-dimensional mask was found to be also effective and the value of $N = 50$ was also used. This was applied by taking two one-dimensional cases with right angles to each other. Thus the final NA

0	0.25	0
0.25	0	0.25
0	0.25	0

(a)

0.128	0.128	0.128
0.128	0	0.128
0.128	0.128	0.128

(b)

Figure 5.6: 3×3 Masks for Neighbourhood Averaging

algorithm has the form:

$$\begin{aligned} h(x,y) = \; & f(x,y) \; -\tfrac{1}{2}\times \\ & \left[\tfrac{1}{101}\left(f(x-50,y)+f(x-49,y)+\cdots+f(x+50,y)\right)\right. \\ & \left.+\tfrac{1}{101}\left(f(x,y-50)+f(x,y-49)+\cdots+f(x,y+50)\right)\right] \\ & +128 \end{aligned} \tag{5.10}$$

Neighbourhood averaging was applied on the unsegmented input image. The resulting image was taken and further enhanced on the segmented objects found by the segmentation process.

5.4 Morphological Operations on Grey Scale Images

The MO, which were already introduced in Chapter 4 are now being extended for grey scale images. A good summary over MO on grey scale images can be found in the paper by Haralick et al. [32].

The definition for Erosion is now:

$$f(x,y) \ominus m(i,j) = \min_{i,j}\left(f(x+i,y+j) - m(i,j)\right) \tag{5.11}$$

Similarly the definition for Dilation:

$$f(x,y) \oplus m(i,j) = \max_{i,j}(f(x+i,y+j) + m(i,j)) \tag{5.12}$$

These definitions are the same than the binary definition (Equations (4.7) and (4.9)) when applied on binary images. The definition for Closing and Opening remains the same (Equations (4.10) and (4.11)). Closing can be described like a *ball* rolling over an image. A one dimensional example of grey level Erosion and Dilation is presented in Figure 5.7.

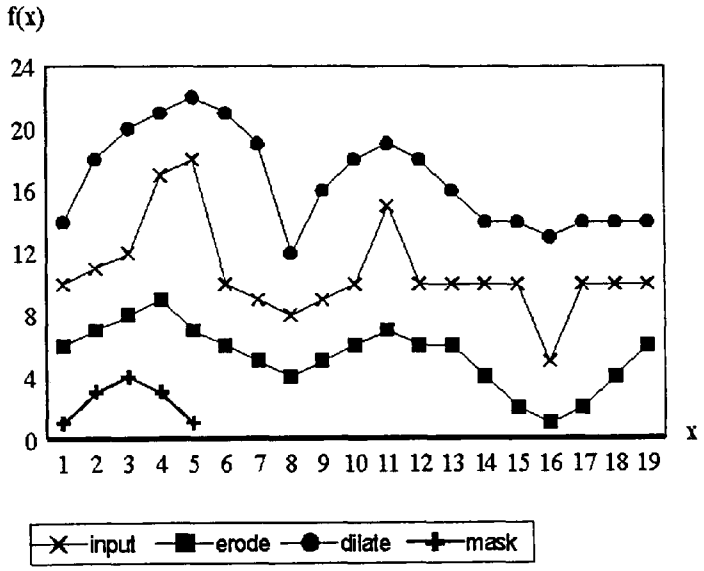


Figure 5.7: Example of the Grey Level MO

The effects of the operators are roughly the same than on binary images. However it is noted that both, the opening and closing operation are blurring the grey level image. Such an effect was not wanted for the purpose of automated detection and therefore MO were no longer investigated within this work.

5.5 Contrast Stretching

Contrast Stretching is one of the simplest methods in enhancing an image. It is a mapping of each of the pixel intensity of $f(x,y)$ to a larger range of output intensities of the image $h(x,y)$. An example for a linear mapping is shown in Figure 5.8.

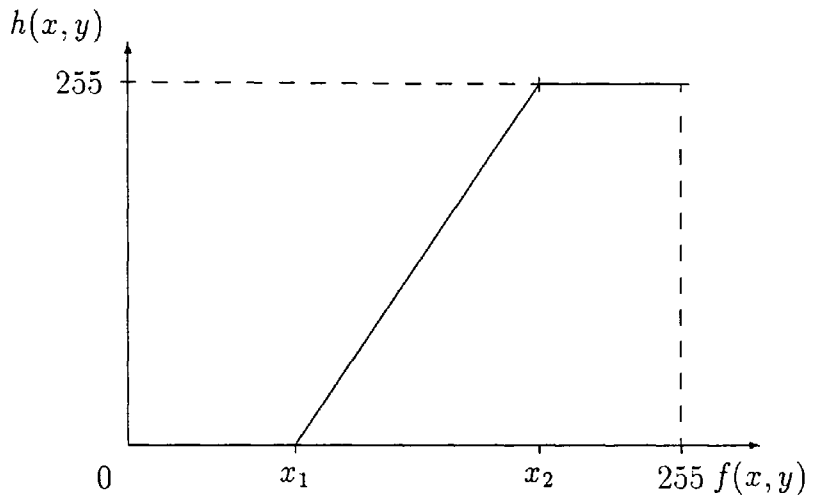


Figure 5.8: Linear Contrast Stretching

The equation for a linear contrast stretch is:

$$h(x,y) = \begin{cases} 0 & \text{if } f(x,y) < x_1 \\ 255 & \text{if } f(x,y) > x_2 \\ 255\left(\frac{f(x,y)-x_1}{x_2-x_1}\right) & \text{otherwise} \end{cases} \tag{5.13}$$

From the above equation it can be seen that all the grey levels $> x_2$ and $< x_1$ will be lost and set to 255 and 0 respectively.

It is often a good idea, to contrast stretch an image with low resolution. If x_1 is set to the minimum and x_2 to the maximum grey level of the image, then no information will be lost. This was also done in this application.

It is, of course, possible to exchange the linear function with any other function, for example, a parabolic, or logarithmic function.

5.6 Histogram Equalisation

Histogram Equalisation (HE) is the only statistical method of image enhancement presented here. It is a method, which maps approximately the frequency occurrence of pixel intensities, given by a histogram, to a uniform distribution. HE is carried out in the following steps:

- By reducing the distinct number of grey levels, i.e. from 256 to 8, where $m = 8$ the number of remaining grey levels.
- Calculation of n_k, the number of times the grey level k appears in an image.
- Calculation of p_k, the frequency of occurrence of each grey level,

$$p_k = \frac{n_k}{n}; \qquad n = \sum_{j=0}^{m-1} n_j$$

and also of s_k, the accumulated probabilities

$$s_k = \sum_{j=0}^{k} p_j$$

- Calculation of the number r_k

$$r_k = \frac{k}{m-1}$$

- Assigning all the probabilities s_k to their closest values r_k.

It is surprising that even with less distinct grey levels than in the original image, HE improves the appearance of an image significantly.

HE was applied in the algorithm to a part of the image, that is a region of interest (ROI). This ROI represents the object in the image. Before the operation of HE, a linear contrast stretch was applied to the ROI in order to give the full grey level range. This proves to be very effective as can be seen here.

n_k	p_k	s_k	mapping	r_k	new n_k
111536	0.2526	0.2526		0	0
56076	0.1270	0.3796		1/7	0
47516	0.1076	0.4872		2/7	111536
98908	0.2240	0.7111		3/7	103592
123578	0.2798	0.9910		4/7	0
2862	0.0065	0.9975		5/7	98908
478	0.0011	0.9985		6/7	0
646	0.0015	1		1	127564
441600	1				

Table 5.1: Example of Histogram Equalisation

Plate A.14 shows histogram equalised objects, and the histograms of the topmost *Anabaena* are shown in Figure 5.9. The HE was applied over 64 distinct grey levels.

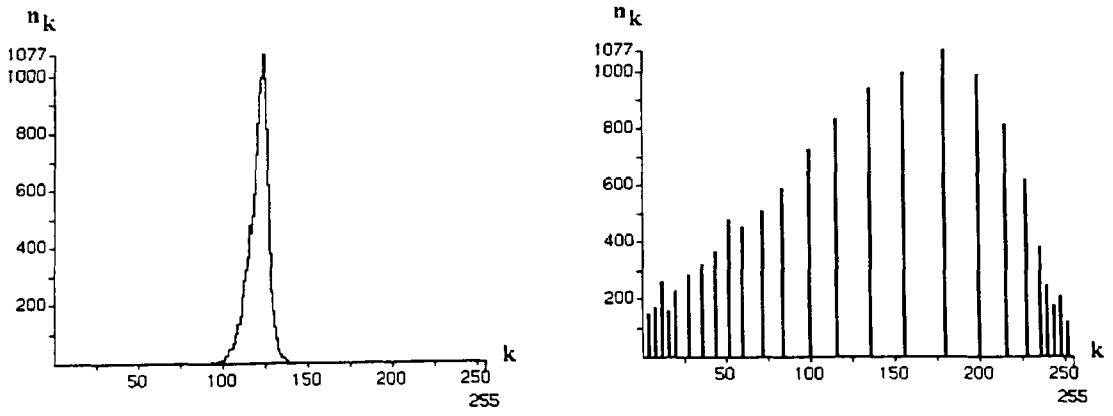


Figure 5.9: Histograms before and after Histogram Equalisation

Chapter 6

Object Identification using Boundary Pixels

When all the pre-processing has been completed, that is, the boundary has been detected, the object pixels of the organism have been enhanced and the background clutter has been removed, it is possible to consider feature extraction algorithms.

The approach on feature extraction is to be based on the information given by the biological key. The key describes the important features, which enable a distinction to be made from one species to another, for example the clear visibility of the cells. In order to extract these features, it is necessary to use Image Processing techniques. In general terms such techniques divide into two groups. Those, which rely on boundary pixels or shape of the organisms only and others which examine the texture of the object. Main features on the boundary may be identified by two classes of methods. In the first all boundary points are required to give a classification, for example Fourier Descriptors or Moment Invariants. While in the second class discrete parts of the boundary may be analyzed separately, for example Hough Transform and a line straightness algorithm. An algorithm for detecting important points on the boundary is also discussed in this chapter.

All the algorithms were not only tested on real data, but also on artificial test data. This had the effect of verifying the results and to highlight some aspects of the different descriptors. The artificial test data is shown in Figure 6.1. The test data consists of 5 objects:

- perpendicular triangle with a height of 200 pixels and width of 200 pixels.
- circle with a radius of 100 pixels
- rectangle with a height of 370 pixels and a width of 20 pixel
- square with a length of 200 pixels
- occlusion of the above square and circle

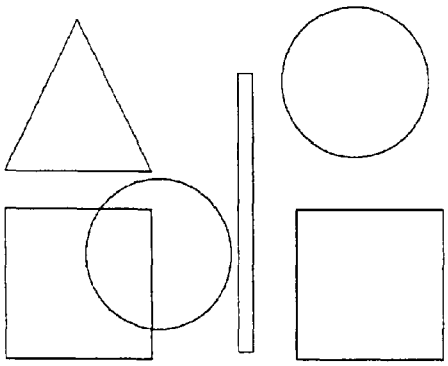


Figure 6.1: Artificial Test Data, *circle, triangle, rectangle, square+circle* and *square*

The test data was chosen, to see how the shape identification routines perform on primitive shapes. Also, the chosen artificial shapes bear some resemblance to the real data represented by blue-green algae, since there exist algae which are more curved (like a circle) or elongated (like a rectangle). In addition the occluded shape clearly highlights the problems for shape routines of overlaying objects.

An important point when discussing object identification is invariance against rotation and scale. All the routines were tested on this bases as the size and orientation of blue-green algae can not be predefined when analysing them under a light microscope. To test this on artificial data also, the objects shown in Figure 6.1 were rotated by 5° and scaled to 75%. This data is always referred to as *modified* whereas the unmodified data is called *original.*

Methods which consider object identification by means of texture are described in Chapter 7, where object identification with the aid of textural algorithms is discussed.

6.1 Fourier Descriptors

One means of classification with the aid of the boundary is Fourier Descriptors (FD), which categorise the shape by means of fourier coefficients. The method has been used for example by Granlund [28] to distinguish between hand print characters and by Zahn and Roskies [108] for closed, mainly artificial curves. The best and most complete introduction can be found in the paper by Wallace and Wintz [102].

Firstly the boundary must be scanned and stored in a counter clockwise direction. For most applications the boundary is then divided into $N = 2^n$ parts, which gives N boundary points. Each point (a, b) will be addressed as a complex number $z = a + jb$ and the sequence of these numbers $(z(0), z(1), \ldots, z(N-1))$ as a complex function $f(z)$. The one-dimensional Fourier Transform (Section 5.1, Equation (5.3)) is now applied to this complex function. The resulting transform $F(u)$ gives the Fourier description $(a(0), a(1), \ldots, a(N-1))$. These values need to be normalized for location and size. $a(0)$ contains the centroid of the boundary. Normalisation for location is done by setting this value to 0. The coefficient $a(1)$ has always the largest magnitude when the boundary was scanned in counter clockwise direction. Normalisation by size is therefore achieved by dividing all Fourier Descriptors $(a(1), a(2), \ldots, a(N-1))$ by that largest magnitude. Sometimes normalisation for orientation and starting point is also required. However in this application only the magnitudes of the descriptors are taken. In such a case a further normalisation is not necessary.

These Fourier Descriptors display great differences between objects containing low frequency information compared to those with high frequencies.

A higher value of N produces more accurate FD which can contain more details of the boundary whereas for a global shape a low value of N is sufficient.

Fourier Descriptors with $N = 16$ were applied to the test data shown in

Figure 6.1. The results are shown in Figure 6.2. The results of the artificial objects are always compared with the results of the modified objects. Figure 6.2 highlights the differences between the various objects and also shows the similarity of the same objects when rotated and scaled.

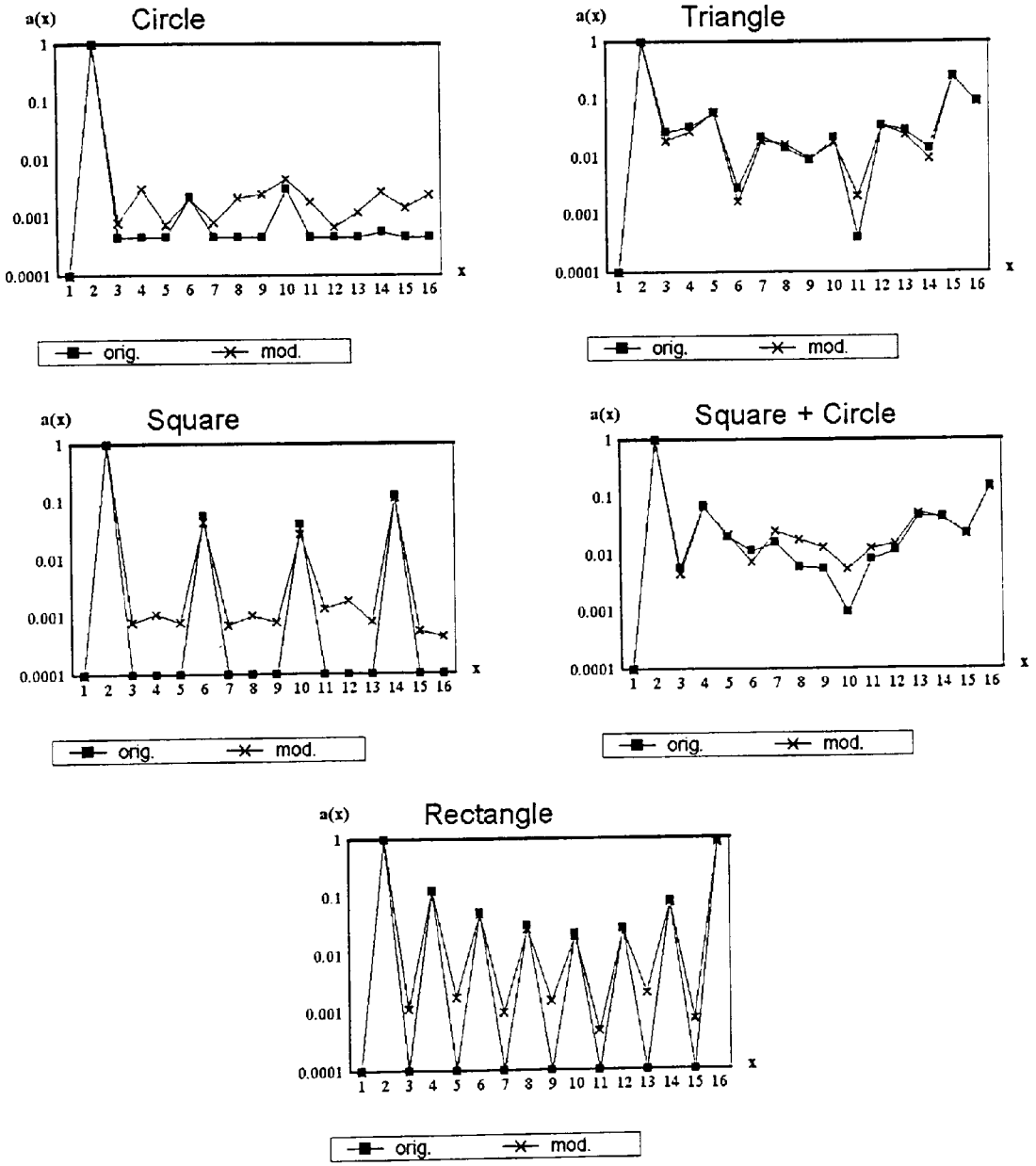


Figure 6.2: The Results of the FD with $N = 16$

In summary it can be said that the Fourier Descriptors work very well,

however they require complete information of the boundary and cannot really be used in case of occlusions. Therefore the descriptors must be treated with care and are not always practicable. In addition, classification by means of FD is very dependent on the shape of the object. This is useful for fixed shapes, but in biological applications the shape may vary, even if the organisms belong to the same species.

For this reason FD can only be used to detect the global shape of a species, omitting the details. Therefore they can be used to distinguish between very straight and very curved organisms. They are not able to detect details between different but similar looking biological organisms. As the FD are therefore only used to describe the overall shape without the need for details the value for N was set to 16 as in the above example.

The results of the FD with $N = 16$ and $N = 256$ were fed into the object classifier. A further discussion of the results given by FD follows later (Section 9.1.1).

6.2 Moment Invariants

Another way of describing and possibly identifying an object is by means of moments and they are defined by (Gonzalez and Woods [27], Hu [38] or Maitra [57]):

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \tag{6.1}$$

which have the discrete form:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{6.2}$$

If the moments of the shape are calculated, the image grey level do not affect the equation. All boundary points and also all the points inside the boundary will be set to 1 whereas all the other points remain 0. In this case the equation

will be applied not only to the boundary, but to the whole shape of the organism. This simplifies Equation (6.2) to:

$$m_{pq} = \sum_x \sum_y x^p y^q \tag{6.3}$$

The centre point of all the pixels may be found according to:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \qquad \bar{y} = \frac{m_{01}}{m_{00}} \tag{6.4}$$

This is easy to see as m_{10} gives the accumulated distance with the x-axis and m_{00} the number of points.

An alternative description of Equation (6.3) is by central moments which may be defined by:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \tag{6.5}$$

The central moments are invariant to the location of the images, unlike the moments calculated by Equation (6.1). However the central moments can be calculated easily with the moments defined by Equation (6.1), or for practical implementation by Equation (6.3). This is very helpful as most image processing packages have pre-programmed functions for that purpose. The equations for the central moments μ_{ij} to be calculated from the moments m_{ij} are (Hu [38]):

$$\mu_{00} = m_{00} \tag{6.6}$$

$$\mu_{10} = 0 \tag{6.7}$$

$$\mu_{01} = 0 \tag{6.8}$$

$$\mu_{11} = m_{11} - \bar{y} m_{10} \tag{6.9}$$

$$\mu_{20} = m_{20} - \bar{x} m_{10} \tag{6.10}$$

$$\mu_{02} = m_{02} - \bar{y} m_{01} \tag{6.11}$$

$$\mu_{12} = m_{12} - 2\bar{y} m_{11} - \bar{x} m_{02} + 2\bar{y}^2 m_{10} \tag{6.12}$$

$$\mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01} \quad (6.13)$$

$$\mu_{30} = m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10} \quad (6.14)$$

$$\mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01} \quad (6.15)$$

Commonly, the Central Moments are calculated up to the third order. They can be normalised for size as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad (6.16)$$

where

$$\gamma = \frac{p+q}{2} + 1$$

for $p + q = 2, 3, \ldots$. This normalisation however does not necessarily mean that all the values of the moments are within the range $0 \leq \eta_{pq} \leq 1$.

Now seven Moment Invariants (MI) can be derived, which are invariant to rotation and changes in scale as is shown by Hu [38]:

$$\phi_1 = \eta_{20} + \eta_{02} \quad (6.17)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11} \quad (6.18)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (6.19)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (6.20)$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \quad (6.21)$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned} \quad (6.22)$$

$$\begin{aligned}\phi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \quad (6.23)$$

With these seven MI, classification can take place unaffected by the orientation and scaling of the objects in an image. A complete overview about applications of Moment Invariants is given by Illing [39] or Dudani et al. [20].

These MI are normally very small as can be seen in Table 6.1. Feeding the actual value into a classifier is therefore difficult. There are several ways to overcome this problem, for example:

- taking the log-value of the MI.
- Further normalize it as shown by Dudani et al. [20].
- Further normalize it as shown by Maitra [57].
- Further normalize it as suggested in this thesis.

The normalisation by Dudani et al. [20] leaves the values small and there seems to be no advantage of taking these values in comparison with the original Moment Invariants. With the method suggested by Maitra [57] small values get multiplied and then divided and the result is somehow unpredictable. In this thesis it is suggested to take roots of the values to compensate for the multiplications in the Equations (6.17)-(6.23). This results in bringing the data range back to a better scale. For example the root for ϕ_2 was already suggested by Hu [38]. Therefore the values for ϕ are further normalized as follows:

$$\psi_1 = \phi_1 \tag{6.24}$$

$$\psi_2 = \sqrt{\phi_2} \tag{6.25}$$

$$\psi_3 = \sqrt{\phi_3} \tag{6.26}$$

$$\psi_4 = \sqrt{\phi_4} \tag{6.27}$$

$$\psi_5 = \sqrt[4]{\phi_5} \tag{6.28}$$

$$\psi_6 = \sqrt[3]{\phi_6} \tag{6.29}$$

$$\psi_7 = \sqrt[4]{\phi_7} \tag{6.30}$$

The roots are based on the powers in the Equations (6.17)-(6.23). Table 6.1 is the example of the MI of a rectangle. The justification of further normalisation of the suggested method can be seen.

	ϕ	log | ϕ |	as in [20]	ψ
1	0.1666622758	-0.7781626921	0.4082429111	0.1666622758
2	0.0000000000	-12.5853218077	0.0000000000	0.0000005097
3	0.0000000001	-10.2104321828	0.0000000133	0.0000078484
4	0.0000000000	-10.5913697546	0.0000000055	0.0000050619
5	0.0000000000	-21.0008232918	0.0000000000	0.0000056207
6	-0.0000000000	-16.8907414061	0.0000000000	0.0000023429
7	-0.0000000000	-21.6988611952	0.0000000000	0.0000037608

Table 6.1: The MI of a Rectangle

The Moments were applied to the test data shown in Figure 6.1 and the resulting MI (ψ) are shown in Figure 6.3. The Moments of the original and modified triangle and occluded square+circle coincide very well. The differences in the MI were often to small to be detected. However the MI of the circle and square do not contain enough information to discriminate between them.

In summary it can be said that the MI give less promising results than the FD. This was also stated in the literature, for example Illing [39] or Reeves [75]. In particular the method breaks down for noisy or occluded images, as can also be seen in Figure 6.3 with the result of the occluded square and circle. However, the method was used to calculate the centre point and the seven MI were put into the object classifier. Further results are given later (Section 9.1.2).

6.2.1 Circularity

When disregarding the specific features of object shape, but considering the overall shape, for example the smoothness or roughness of the overall boundary, the circularity of an object is a good feature to investigate. The circularity was used in connection with three other descriptors by Tomaru et al. [97] and is

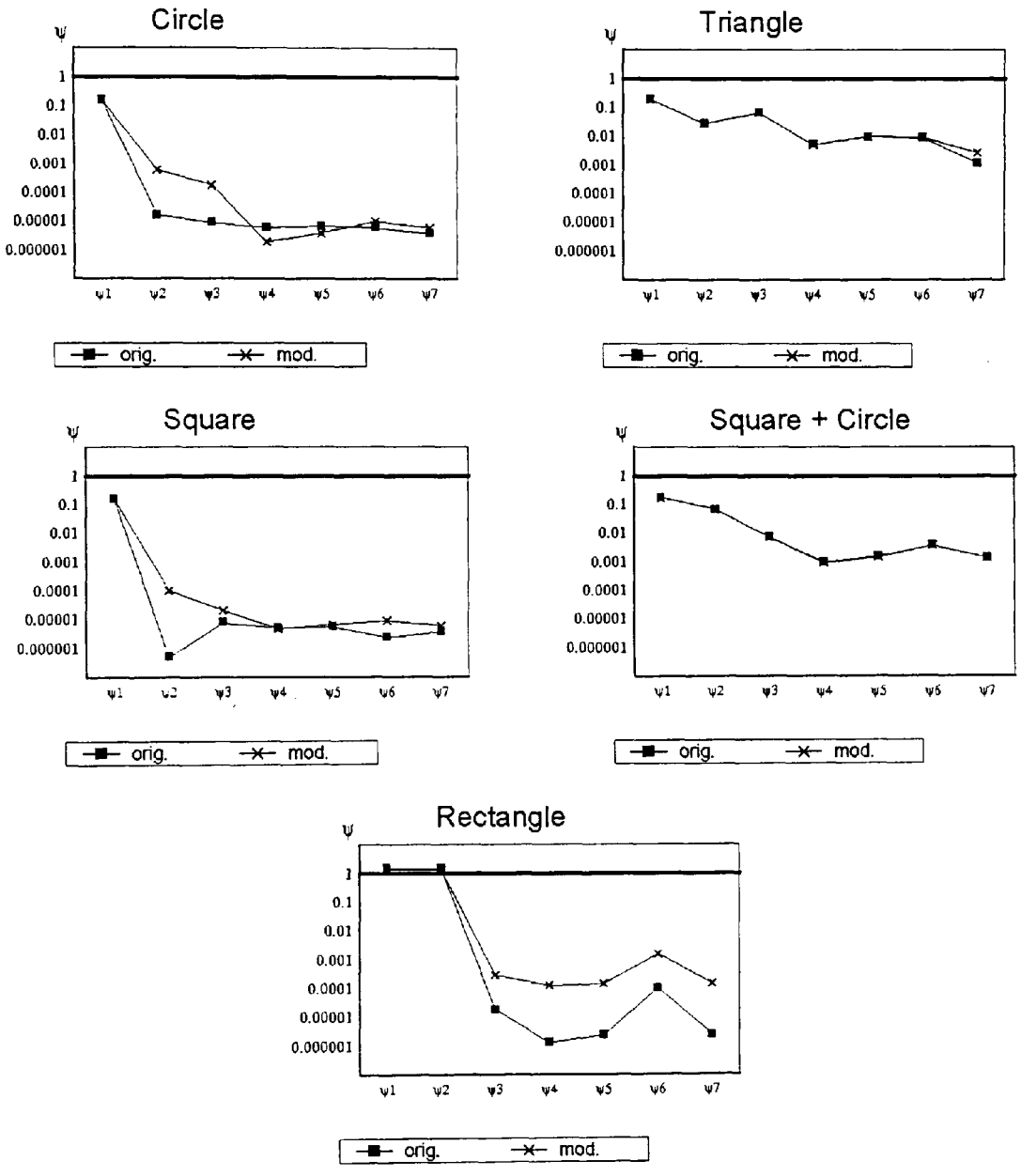


Figure 6.3: The Results of the MI

defined as follows:

$$c = 4\pi \times \frac{\text{area}}{\text{boundary length}^2} \tag{6.31}$$

For a circle $c = 1$, since:

$$c = 4\pi \times \frac{\pi r^2}{(2\pi r)^2} = 1$$

For all other objects the value of c is smaller than 1, but the higher the value of c, the more circular an object is.

The circularity is very easy to calculate, as both, the area and the boundary length of the objects are already known. The area is the moment m_{00} and the boundary length is found while tracing the boundary (Section 4.5.2). However there appears to be a quite significant error due to the digitisation. The circularity of a circle with the radius of 100 pixels was calculated as:

$$c = 4\pi \times \frac{31813}{664^2} = 0.907$$

The area is almost correct with a theoretical area of 31416 pixels. However the length of the boundary is very high, 568 boundary pixels gave a length of 664 compared to the theoretical length of 624. The reason for this was already mentioned in Section 4.5.2, Figure 4.8. Keeping the digitisation error in mind, the circularity distinguishes quite clearly between the artificial test data (Figure 6.1) as shown in Table 6.2.

The feature of circularity was fed into the classifier.

6.3 Hough Transform

The Hough Transform (HT) uses a totally different approach to object identification than the above two methods of FD and MI. The HT can be used on

	original	modified
Circle	0.906	0.891
Square	0.793	0.751
Square+Circle	0.742	0.712
Triangle	0.547	0.548
Rectangle	0.161	0.166

Table 6.2: The Circularity Tested on Artificial Data

complete or interrupted boundaries. Efficiently the HT uses a binary image and converts it into an accumulator space. The input image is either a thresholded image or an image which consists only of the boundary. In the HT a mathematical representation of the given features is analyzed, for example a straight line or a curved part of the boundary. The HT algorithm then identifies which of the given image points conform to the mathematical representation. Good correspondence results in peaks in the Hough space.

Consider the case of the detection of a straight line where the Hough space consists of two parameters, i.e. is two-dimensional. The equation of a line is in polar coordinates:

$$x \cos\theta + y \cos\theta = r \tag{6.32}$$

where θ is the angle ($0° < \theta < 180°$) and r the distance from the origin, ($\mid r \mid \leq$ maximum distance of any image point from the origin).

The Hough space (r, θ) is a plot of possibilities of r and θ. In general r and θ are real numbers and so there are infinite combinations of numbers of r and θ. Clearly, for computation these parameter needs to be discretized so that the Hough space will be an array in which errors may occur due to the process of discretization.

The algorithm works by examining every image point (x, y) and each time

the line (r, θ) passes through that point, one is added to the discrete accumulator at the value (r, θ).

High peaks in the Hough space accumulator give a high possibility of the occurrence of this line in the input image.

The problems with the HT are as follows:

- The method by which the parameters r and θ may be discretized in order to minimise the classification error. This problem is considered by for example Duda and Hart [18] and Murakami et al. [62].
- The size of the Hough space remains large even after discretization and a considerable amount of computer memory is required to store the accumulator and much computer time is required in the processing.
- The start and the end of a detected line are in general unknown.
- The number of parameters for more complicated curves increases and therefore the dimension of the Hough space increases. For example a circle needs 3 parameters whereas an ellipse needs 5 parameters.
- The generalisation of the HT (Ballard [4]) to any shape is possible, but the amount of time and memory needed for this remains very high.

Figure 6.4 shows some results after applying a simple straight line HT algorithm to the test data shown in Figure 6.1. The input parameters to this particular function are the minimum size l of a detectable line and the amount of different values of θ. The HT detects clearly all horizontal and vertical lines. The obvious problem however is to detect the slanted lines of the triangle without detecting parts of the circle. It must be noted that actually some parts of the boundary of a circle are straight lines due to the digitisation error.

The HT has been implemented also on images of blue-green algae. However,

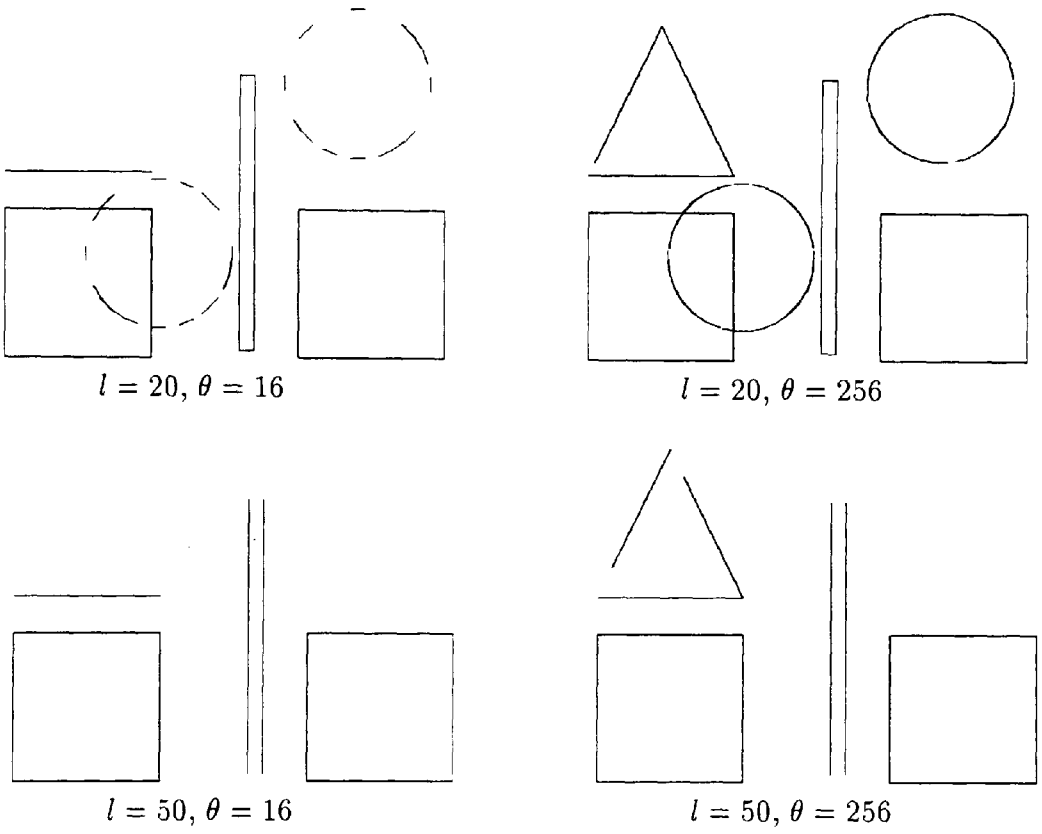


Figure 6.4: Example of the HT with Varying Parameters

the ability of the HT to detect some internal features was not sufficient and reliable enough.

In addition, the results when applied to the boundary did not provide more information. Clearly the boundary of the objects are already known and applying the HT to detect the boundary again does not make sense. On the other hand applying the HT in order to figure out how straight the boundary is, can be very difficult and clearly there are other ways to produce the same results. One simple solution of this is introduced in the next section.

6.4 Straightness Algorithm

This algorithm was developed after applying the straight line Hough Transform to the boundary. However, since biological images seldom have straight edges it was necessary to design and implement a new algorithm with a more flexible approach. The algorithm should associate values to certain parts of the boundary concerning how straight or curved they are. This actually results in a close resemblance with the biological key, where also the smoothness of the shape is a criteria for classification. In the following the newly developed algorithm is explained.

Firstly it is important to define a value, which contains some information about straightness. The equation of a line in its normal form is:

$$0 = ax + by + c \tag{6.33}$$

Hence the distance d_0 of a point (x_0, y_0) from a line is given by:

$$d_0 = \frac{ax_0 + by_0 + c}{\sqrt{a^2 + b^2}} \tag{6.34}$$

Now the straightness sd of a part of a boundary can be defined. Let the first point of the boundary be (x_0, y_0) with (x_n, y_n) the last point. Then all the

distances d_i from all $n-1$ points along this part of the boundary to the line $[(x_0, y_0); (x_n, y_n)]$ are calculated. The straightness sd is then the sum of all the squared distances d_i.

$$sd = \sum_{i=1}^{n-1} {d_i}^2 \tag{6.35}$$

The squared distances are necessary because the distance d_i can be both positive and negative.

The values of sd are small when the tested part of a boundary is straight and high when they are curved. Furthermore it can be seen that the only parameter required for such a function is the length of the boundary part to be examined. A boundary part length of $len = 50$ gave satisfactory results. For the straightness of a whole boundary sd was calculated successively, starting always with the point, which was the first to have a distance of more then $len/5$ to the old starting point.

The resulting sums of the squared distances had to be normalized, in order to receive output values between 0 and 1. This was achieved by dividing each sum by a possible maximum. With $len = 50$ the maximum number of points within the part of the boundary is 49 and these 49 points could sum up to a possible maximum of:

$$\begin{aligned} \text{possible maximum} &= 1^2 + 2^2 + 3^2 + \ldots + 24^2 + 25^2 + 24^2 + \ldots + 1^2 \\ &= 2\sum_{i=1}^{2} 4i^2 + 25^2 \\ &= 10425 \end{aligned}$$

The following measures were taken:

- The minimum value $sd-$ of all normalized sd.

  This represents the straightest part found of the boundary.

- The maximum value $sd+$ of all normalized sd.

  This represents the most curved part found of the boundary.

- The mean *sdm* of the normalized values *sd*.

  This is a value which states the curvedness of the whole boundary.

- The standard deviation *sds* of the normalized values *sd*.

  This is a value which states how the curvedness of the whole boundary varies.

The results when applying the algorithm on the test objects shown in Figure 6.1 are displayed in Table 6.3.

		sd-	*sd+*	*sdm*	*sds*
Circle	orig.	0.01180	0.02832	0.01842	0.00338
	mod.	0.02145	0.04654	0.03128	0.00563
Triangle	orig.	0.00000	0.61562	0.04224	0.13065
	mod.	0.00029	0.55175	0.05469	0.13940
Square	orig.	0.00000	0.44309	0.04687	0.12286
	mod.	0.00041	0.45638	0.05916	0.12725
Rectangle	orig.	0.00000	0.58878	0.03998	0.12903
	mod.	0.00042	0.63263	0.04029	0.12381
Square+Circle	orig.	0.00000	0.49761	0.05768	0.12173
	mod.	0.00041	0.44920	0.06988	0.12196

Table 6.3: The Results of the Straight Line Algorithm

The results clearly distinguish the circle from all the other objects as the boundary does not have any straight parts nor any corners. The differences from the minimum ($sd-$) to the maximum value ($sd+$) are due to the digitization error. However the standard deviation is very low compared to the other data. In addition the circle shows that a smaller circle has a higher circularity.

As expected the triangle has the highest value *sd+* of all objects and therefore *sds* is higher. The similarity of the results of the square, the rectangle and the occlusion of circle and square needs to be highlighted. The mean value *sdm* of the occluded object is higher because of the circular part of the boundary.

The four parameters *sd−*, *sd+*, *sdm* and *sds* were fed into the object classifier with three different values for $len = 50$, 100 and 200. If the boundary was too small for the straightness with $len = 200$, the results for $len = 100$ were taken instead. In this context *too small* means not being able to calculate the straightness at least seven times. With only a low number of values of *sd* the statistical measures would loose their significance.

This especially for this task developed algorithm, although straight forward, is effective and takes measures into account, which biologists use to apply the biological key. Such measures do not appear to have been used in previous work.

Further results are given later (Section 9.1.3).

6.5 Important Boundary Points

The methods of Fourier Descriptors and Moment Invariants were tried and tested, but did not give sufficiently good results for this study. A more successful method has to consider boundary points with high curvature. The curvature C is defined by Rosenfeld and Johnston [78] by the k-cosine:

$$C_{ik} = \frac{\vec{a_{ik}} \cdot \vec{b_{ik}}}{\mid \vec{a_{ik}} \mid \mid \vec{b_{ik}} \mid} \tag{6.36}$$

where the vectors $\vec{a_{ik}}$ and $\vec{b_{ik}}$ are defined by:

$$\vec{a_{ik}} = (x_i - x_{i+k}, y_i - y_{i+k}) \tag{6.37}$$

$$\vec{b_{ik}} = (x_i - x_{i-k}, y_i - y_{i-k}) \tag{6.38}$$

and (x_i, y_i) is the i^{th} point of the boundary. This is illustrated in Figure 6.5.

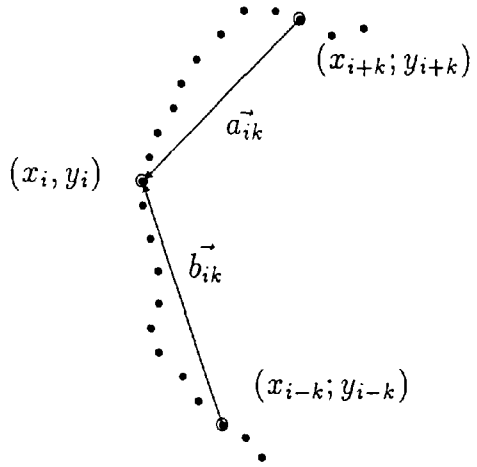


Figure 6.5: The k-cosine

The curvature information is calculated such that $-1 \leq C_{ik} \leq +1$ with $C_{ik} = -1$ for a straight line and $C_{ik} = +1$ for an angle of zero degrees.

The value of k, which is the smoothing factor in this method can be evaluated for each point with the formula for C_{ih}:

$$C_{im} < C_{i,m-1} < \cdots < C_{ih} \not< C_{i,h-1} \tag{6.39}$$

where:

$$1 \leq k \leq m$$

and $m \approx \frac{n}{10}$, with n the number of boundary points. The curvature C_{ih} will then be assigned to the point (x_i, y_i) as its cosine C_i.

It is also possible to set the value of k to a fixed number. After the calculation of each C_i has taken place, local maxima in C_i represent points of high curvature on the boundary. However investigations with this or similar methods, for example Rosenfeld and Weszka [79] showed, that it detects more points than required. If the boundary contains noise, these methods are very unreliable, as shown for example by Illing [39].

In the following a novel approach for the detection of certain boundary points is presented. To emphasise the different methodology, these points are called *important points*. The important points obtained by this method are clearly

different to the points of high curvature. The definition of *important points* as presented within this thesis is novel and has not been used before.

In this work the important points are needed to separate the boundary into different parts, in order to be able to work with occluded objects as illustrated in Figure 6.6. Clearly local details on the boundary are of no interest, but the start and end points of the algae needs to be localised. This specialised problem required the development of a new problem specific algorithm.

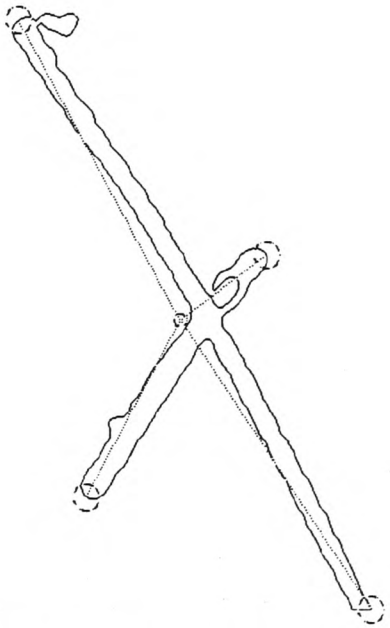


Figure 6.6: The Important Points

The algorithm is as follows: Firstly, the centre point must be calculated. This was already done using moments (Equation (6.4)). The squared distance from the centre point to each point on the boundary is evaluated. The squared distance is taken simply because it is easier to calculate than the actual distance. By comparing this distance for one pixel relative to the neighbourhood pixels enables the important points to be located. For an important point the distance to the centre is larger than the distance the immediate neighbours have. The

squared distances of Figure 6.6 are shown in Figure 6.7 and the result of the algorithm is displayed in Figure 6.8.

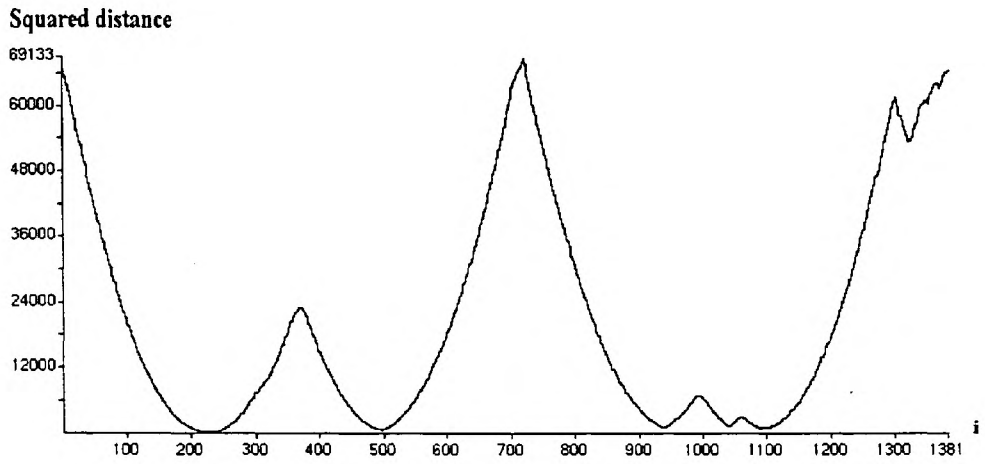


Figure 6.7: The Squared Distance of the Boundary Points to the Centre Point

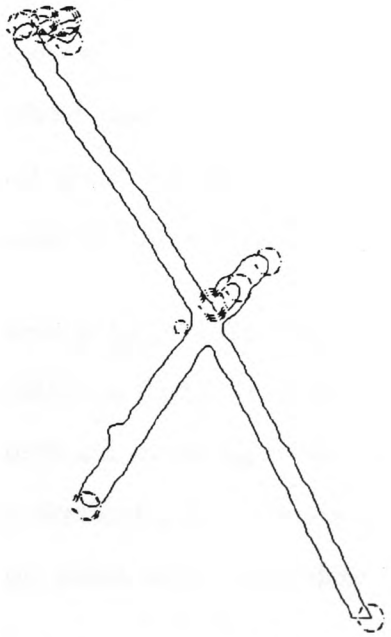


Figure 6.8: The Important Points Found by the Algorithm

It can be seen from Figure 6.8 that several unwanted points are located. This is mainly because of the noise on the boundary. The filtering of these unwanted points can be done in the following way.

Firstly, not only the important points are located but also the minimum points, which have a smaller distance to the centre point than their immediate neighbours. This is similar to a continuous function, in which local maxima and minima occur in alternating order. Important points and minimum points also alternate.

The differences in distance to the centre point which occurs between the important point and the following minimum point are compared. If the difference is less than a threshold, then the minimum point is eliminated. As there are now two important points following each other, one of them should also be eliminated. The important point with the larger distance is kept, and the other point is rejected. Now the next minimum point is taken and probably rejected. In this elimination method a minimum point will finally be found. This minimum point is now compared with the following important point and the process continues.

It can be seen that the threshold in this method influences the result obtained after filtering. The threshold used in this study is set relative to the squared distance of the important point to the centre point. The threshold is either a fifth of the squared distance or 5000, depending on which is larger.

Another point to mention here is the starting point of the algorithm. It is advisable to start with a point which is likely to be an important point. This starting point is determined beforehand in an algorithm which detects the two points on the boundary, which are furthest away from each other. Either of these points is likely to be an important point and is therefore suitable as a starting point.

This algorithm does not work on boundaries with slowly varying curvature. However this should not cause any difficulty since in this case only two important

points are necessary to separate the boundary into two different parts and the already detected possible starting points are taken as important points.

Figure 6.9 is the result after the algorithm was applied on the artificial objects. It can be seen that the algorithm locates clearly all important points and that for the circle simply the starting points are taken.

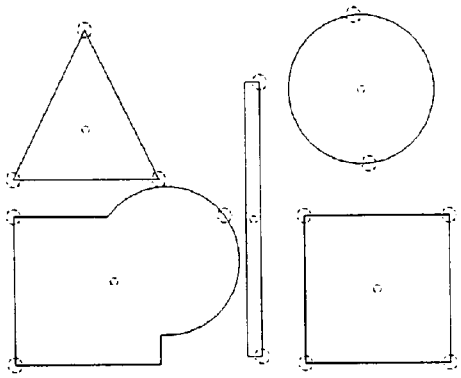


Figure 6.9: The Important Points of the Artificial Data

Figure 6.6 and Plate A.15 show resulting images of blue-green algae after the important point algorithm was applied. The circles show the important points found. The centre point is also marked with a tiny circle.

One unsolved question however is how the important point information can be used to detect the different organisms. It is obvious, that a square has four and a triangle has tree important points. Both the circle and the rectangle have however two points. Just to count the number of important points and to take their relative location with respect to each other into account can not be enough. The important point information can however be used in combination with further identification routines (Section 7.1.1).

Chapter 7

Object Identification using Texture

In Chapter 6 some descriptors of the object boundary were introduced, but when applied to biological images they do not prove to be very effective. In order to distinguish between biological organisms, the texture may be very important. The texture describes image properties inside the boundary, for example smoothness, coarseness and regularity (Gonzalez and Woods [27]). A good introduction of textural analysis is provided by Ballard and Brown [5] and textural approaches are summarised by Haralick [30].

It can be seen, that the frequency of occurrence of the grey levels inside the object is of great importance for texture analysis. There are generally three approaches for textural measures.

- spectral approach
- structural (or syntactic) approach
- statistical approach

The spectral approach is located in the frequency domain. This approach requires the two dimensional Fourier Transform as the Low and High Pass Filter. Problems with the Fourier Transform have already been mentioned in Section 5.1. Furthermore it does not seem to be very promising as the spectral approach did not receive much attention in other research projects involving texture. For example Haralick [30] mentions in his texture survey only the other two approaches.

The structural approach and the statistical approach will be examined in this chapter.

7.1 Structural Approach

In the structural approach an attempt is made to apply a grammar to the object required for detection. In simple terms this is done by defining a grammar based upon some structural rules and then establishing a relationship between the grammar and object (Gonzalez and Woods [27]).

In the following section it is thought that the cells, which are a characteristic feature for some species of blue-green algae, form a valid grammar for the description of blue-green algae.

7.1.1 Cell Finding Algorithm

A detail, which distinguishes certain algae, for example *Anabaena* or *Nostoc*, from others is the existence of filaments, the visibility of cells. The cells are connected to each other in a sequence of blocks. The connecting wall between the cells is darker than the cell itself. Hence, an identification algorithm needs to detect the cells, i.e. the start and the end of each cell. The problem of identification of connected cells for biological organisms does not appear to have been addressed in the literature and it was therefore necessary to develop a new algorithm. The newly developed algorithm has the following steps:

- Determining the important points on the boundary (Section 6.5). Every important point separates the boundary into different segments as is seen in Plate A.15.
- After this step it can be seen that there are always two boundaries opposite to each other. These opposite boundaries are then linked together.
- The next step is to move along the linked boundaries, in such a way that opposite points are linked to each other. Two opposite points are defined

to be those points which are closest together when moving along linked boundaries.

- The average grey level of the line which connects these points is then calculated and stored in a list. These grey levels are taken from the best enhanced image, which is the histogram equalised image.

- If the calculated average grey levels are used directly to determine cell boundaries then experience has shown that many spurious edges are found. The reason for this is that the average grey levels themselves contain noise which must be smoothed away. This can most effectively be done with the Hamming Window (Lynn [55]) as shown in Figure 7.1. The Hamming Window $w[n]$ is defined by:

$$w[n] = \begin{cases} a + (1-a)\cos\left(\frac{n\pi}{N}\right) & -N \leq n \leq N \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

  The chosen value of a is 0.54 as is often the case in signal processing and the range N is set to 10. n is the average grey level to be smoothed.

- From the smoothed list of average grey levels, a minimum represents the darker connection between boundaries and in this way the connection of two cells is determined.

Plate A.15 shows the resulting connections for the example of *Anabaena*. In general, the algorithm works extremely well and there are few difficulties. On some occasions cells could not be located simply because the input image had extremely low contrast, almost impossible to detect.

After the cells have all been located, the gained structural information needs to be investigated. The structural texture information is, that cells are always arranged after each other. Therefore, the number of cells, which belong together is important. In addition the following measures were derived from the cell finding algorithm.

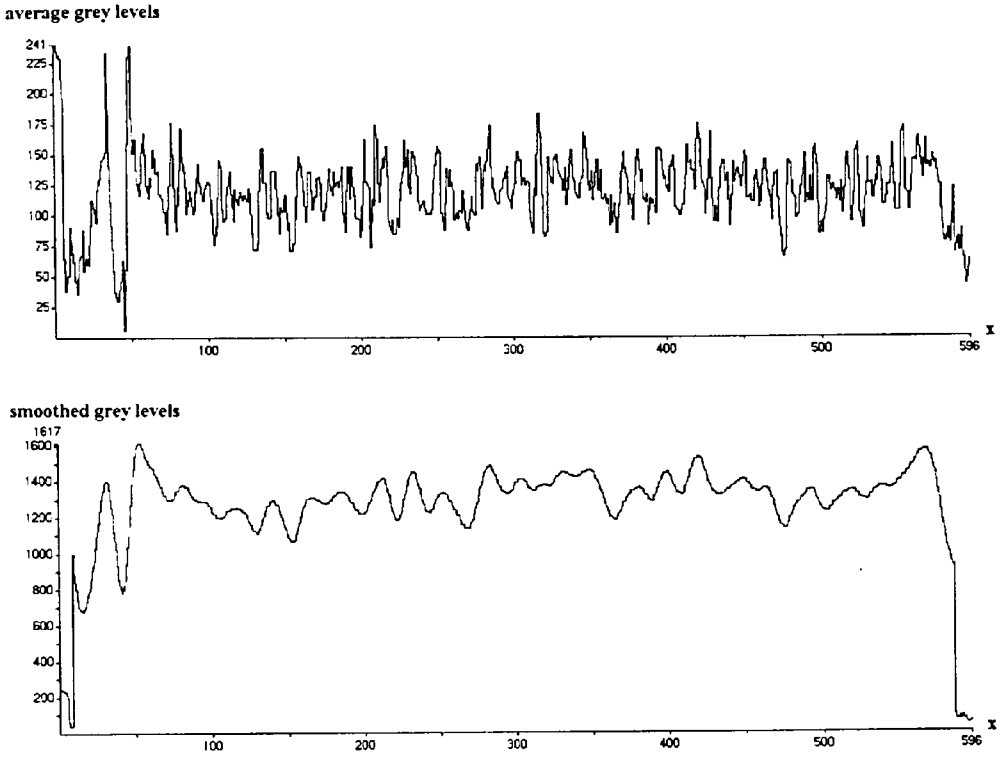


Figure 7.1: The Average Grey Levels before and after Smoothing with the Hamming Window

- the average width of the cells
- the average length of the cells
- the standard deviation of the width
- the standard deviation of the length
- the relative length of the cell

However two of these measures, the average length and width of the cells, need to be treated with great care, since these measures are obviously not invariant against changes in scale.

The cell finding algorithm was applied not only to filamentous algae, but to all nine selected algae. This had to be done, as *a priori* knowledge about algal

type was not available and the algorithm had to be robust enough to cope with that problem. When no cells were found all the measures were set to zero.

Finally it can be said that the newly developed cell finding algorithm works extremely well, detecting the cells within the filamentous algae with great accuracy and reliability. The discrimination result of the cell finding algorithm is shown in Section 9.1.7.

7.2 Statistical Approach

All the statistical measures of texture use direct the grey level values of each pixel and apply then some statistical formulae to them in order to calculate descriptors.

Statistical approaches can be divided into two different classes:

- First order texture
- Second order texture

First order texture contains no information about the pixel location. They are purely based on the information provided by the histogram. First order textures are generally called Grey Level Distribution Moments.

Second order texture takes the specific position of a pixel relative to another into account, for example Moment Invariants or the Spatial Grey Level Distribution Matrix.

All statistical approaches were also tested on certain artificial texture, which are displayed in Plate A.10. The first texture is a white line in the middle which becomes darker towards the outside to achieve a 3-D effect. The second and fourth textures are artificial patterns, whereas the third pattern was generated with a random number generator.

7.2.1 Grey Level Distribution Moments

Consider first order textures, which are based exclusively on the grey level histogram of the object. Those contain no information about pixel location. These descriptors are called Grey Level Distribution Moments (Gonzalez and Woods [27]). This is a statistical approach to texture similar to the one introduced in Section 6.2, however in this case the grey level values take a significant part in the equation. The Grey Level Distribution Moments are defined by:

$$\mu_n = \sum_{i=1}^{L} (z_i - m)^n p(z_i) \tag{7.2}$$

where L is the total number of grey levels in the histogram (here: $L = 256$), z_i are the discrete grey levels, $p(z_i)$ the occurrence of each grey level and m is the average grey level.

Clearly, it can be seen that $\mu_0 = 1$ represents the area under a normalized histogram, whilst $\mu_1 = 0$. The second moment represents the variance of the histogram, whilst the third moment displays the skewness and the fourth moment the flatness of the histogram (Gonzalez and Woods [27]). Due to the power in Equation (7.2) the second and higher moments can have very high values. To obtain values within the desired range from 0 to 1 it is possible to divide each moment μ_n by L^n.

It is obvious that such an algorithm, which is highly related to the histogram has to be implemented before the Histogram Equalization is carried out, because HE can destroy all first order texture information.

The histograms of the four artificial textures from Plate A.10 are shown in Figure 7.2. The Grey Level Distribution Moments are shown in Table 7.1.

It can be seen that these measures describe the differences of the histograms quite well. The drawback however can also be seen for example in the high variances of texture 1 and 3. The high variance is caused because only every 4th distinct grey level is used to build the texture. Such a problem however does not

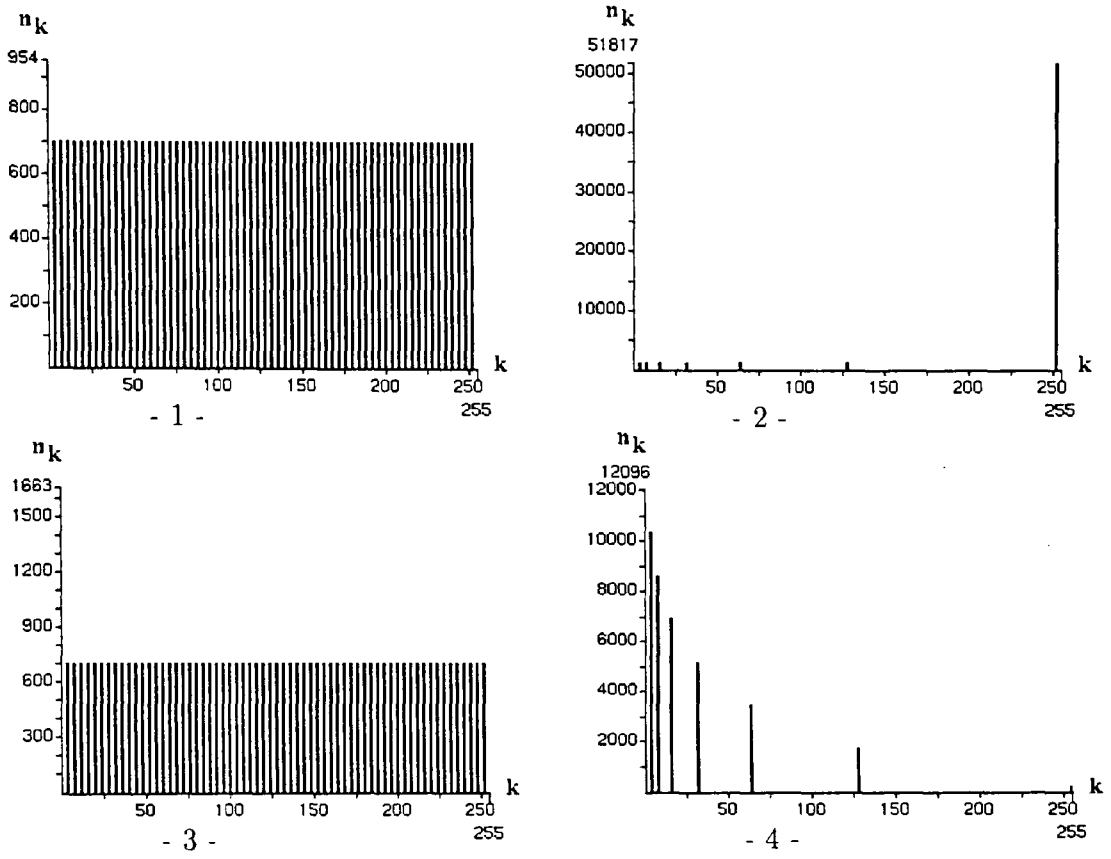


Figure 7.2: The Histograms of the Textures of Plate A.10

	no. of pixels	mean	variance	skewness	flatness
1	44928	125	0.08487	0.00003	0.01295
2	59926	222	0.09012	-0.06156	0.05150
3	45981	123	0.08719	0.00018	0.01361
4	48600	18	0.01526	0.00669	0.00461

Table 7.1: Grey Level Distribution Moments Applied to Artificial Data

occur in the real data and can be ignored.

The Grey Level Distribution Moments have been implemented in the algorithm. In addition the mean value was also given to the classifier. Since the mean of the overall images were already normalized, the individual mean of the objects describe a measure on how much darker they are than the background. Further results are shown in Section 9.1.4.

7.2.2 Moment Invariants

Another way of describing texture is also based on moments and is in fact the same than the technique presented in Section 6.2. In that section the central moments were defined and the centre point was calculated. It was shown that the central moments when applied to an organism with all organism pixel values set to 1 and all the others set to 0 can describe the shape of an object. However, more commonly the central moments are applied to grey levels also, rather than the constant value 1. Equation (6.2) stated already the discrete form of the equation of moments:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{7.3}$$

However now, in contrast to Section 6.2, the grey levels $f(x,y)$ will now not be omitted. Therefore, the central moment description is now given by:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{7.4}$$

Note also that Equation (7.4) is the two-dimensional version of Equation (7.2).

The central moments are also calculated up to the third order. Normalisation and derivation of the seven MI is the same as already shown in Section 6.2.

When applied to the artificial texture of Plate A.10 the seven MI seem to provide different results as can be seen in Figure 7.3. This however does not show their performance when applied to real data. In addition even though the grey

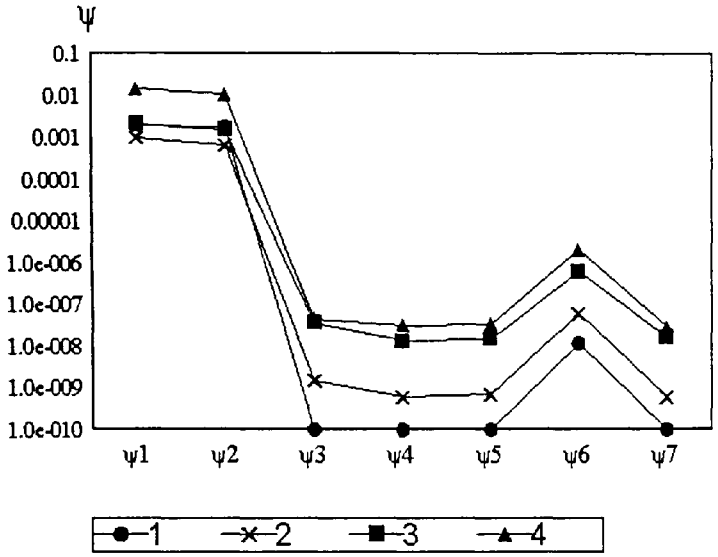


Figure 7.3: The MI applied to the Artificial Texture

levels are taken for calculating the moments, the location of the pixels and the shape of the objects are still significant. In the Equations (7.3) and (7.4) the shape takes as much part as within the boundary moments (Equations (6.3) and (6.5)). The difference is just an adding of the importance of the texture, without omitting the influence of the shape.

The discrimination results of the seven texture MI can be found in Section 9.1.5.

7.2.3 Spatial Grey Level Dependency Matrix

In the case of images of blue-green algae it will be necessary to describe the location of a pixel more accurately, for example the position of pixel with certain grey level intensities relative to the grey level intensity of neighbouring pixels. In order to compare all pixel intensities a two-dimensional matrix C has to be formed. This matrix C is either called a *Spatial Grey Level Dependency Matrix* (SGLDM) as by Haralick et al. [29] or a *Grey Level Co-Occurrence Matrix* as by Gonzalez and Woods [27]. As Haralick et al. [29] is frequently cited in most

articles considering texture, the term SGLDM will be used within this thesis.

Since in this application the images are always Histogram Equalised with a maximum of 64 distinct grey levels, the SGLDM will have a size of 64×64.

In addition a position operator P is defined. For example P can be *one pixel to the right and one pixel below* as in Gonzalez and Woods [27] or simply *next to* used by Thomas [93]. The operator P is now passed over the image. For each object pixel the position operator needs to be evaluated. If the pixel intensity is i and the pixel intensity to which the operator points is j, the matrix element c_{ij} of C will be incremented by one. Hence C is a function of P and P is a function of distance d and angle θ, where the angle could be one specific direction, or a set of directions. For example:

- $P(1,45)$ is the operator which points one to the right and one below.
  $(d = 1, \ \theta = 45°)$
- $P(5,90)$ is the operator which points five below.
  $(d = 5, \ \theta = 90°)$
- $P(1,0,45,90,135)$ is the sum of the results with $P(1,0)$, $P(1,45)$, $P(1,90)$ and $P(1,135)$.
  $(d = 1, \ \theta = 0°, \ 45°, \ 90°, \ 135°)$

There exists a little inconsistency in the literature on how to calculate the SGLDM. The above definition produces a SGLDM which is not normally symmetric and if the orientation of an object is important this type of calculation must be used. As Weszka et al. [104] is the most important paper using this non symmetric matrix, it will be called W-SGLDM. If the orientation of the object can not be controlled, or is of no interest, the way of definition can be slightly modified to produce a symmetric matrix. The only difference to the W-SGLDM is that when the position operator $P(d,\theta)$ is passed over the image the operator $P(d,\theta+180°)$ is simultaneously passed over the image. This matrix will be called

H-SGLDM throughout this thesis, since Haralick et al. [29, 31, ...] always uses the symmetric SGLDM. It follows that:

$$\text{H-SGLDM} = \text{W-SGLDM} + \text{W-SGLDM}^T \tag{7.5}$$

When the term SGLDM is used it applies to both SGLDM's. An example of how to calculate the SGLDM is given in Figure 7.4.

Image:

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

$P(1,0)$

$j \backslash i$	0	1	2	3
0	2	2	1	0
1	0	2	0	0
2	0	0	3	1
3	0	0	0	1

W-SGLDM

$j \backslash i$	0	1	2	3
0	4	2	1	0
1	2	4	0	0
2	1	0	6	1
3	0	0	1	2

H-SGLDM

Figure 7.4: Example for Calculating the SGLDM's

The SGLDM is an accumulator, but in comparison to the Hough Transform accumulator, all entries are important, not just peaks are of interest. Since all

elements are of interest, a statistical analysis of the accumulator needs to be undertaken. Before this can be done, it is necessary to normalize the SGLDM. From now on c_{ij} represents the normalized matrix element of the SGLDM, this means that:

$$\sum_{i,j} c_{ij} = 1$$

The following statistical textural measures were applied and the description of each measure contains conclusions for the SGLDM under point 1 and for the image under point 2 (see below).

- **Energy**

$$U = \sum_i \sum_j c_{ij}^2 \tag{7.6}$$

This descriptor is also called **Uniformity** or **Angular Second Moment**.

1. This descriptor is small, when all matrix elements are almost equal, therefore the higher the value, the more irregular the SGLDM.
2. A constant SGLDM is for example caused by an image where there is no conclusion possible about neighbouring pixels.

- **Entropy**

$$\hat{E} = -\sum_i \sum_j c_{ij} \log c_{ij} \tag{7.7}$$

This measure will have the maximum value of

$$\hat{E}_{\max} = -\sum_i \sum_j \frac{1}{n^2} \log \frac{1}{n^2} = -\frac{n^2}{n^2} \log \frac{1}{n^2} = 2 \log n$$

with n the size of the matrix and all matrix elements the same. To avoid a result which is larger than 1 all results were divided by $2 \log n$.

$$E = \frac{\hat{E}}{2 \log n}$$

Entropy is the opposite measure to Energy.

1. This measure of randomness produces a low value for an irregular SGLDM and, as shown, its highest value $E = 1$ when the SGLDM is uniform.
2. An irregular SGLDM is produced by a regular image, that is for example an image with a certain pattern.

- **Inertia**

$$\hat{I} = \sum_i \sum_j (i-j)^2 c_{ij} \tag{7.8}$$

This measure, in the extreme case, with only one c_{ij} equal to 1 and all the others 0, has the value of $(n-1)^2$. As with the Entropy this is normalized by:

$$I = \frac{\hat{I}}{(n-1)^2}$$

Sometimes this descriptor is referred to as **Element-Difference Moment** or **Contrast**.

1. This descriptor measures the difference moment of the SGLDM.
2. If the input image has high local variation then this descriptors will be high.

- **Local Homogeneity**

$$H = \sum_i \sum_j \frac{1}{1+(i-j)^2} c_{ij} \tag{7.9}$$

This **Inverse Element-Difference Moment** is clearly the opposite measure to Inertia.

1. High values are caused by a SGLDM with high values at the diagonal.
2. The Local Homogeneity is high, when similar grey levels are next to each other in the input image.

- **Correlation**

$$C = \frac{\sum_i \sum_j (i-\mu_x)(j-\mu_y) c_{ij}}{\sigma_x \cdot \sigma_y} \tag{7.10}$$

where μ_x, μ_y, σ_x and σ_y are the means and standard deviations of the marginal distributions of the normalized SGLDM:

$$\mu_x = \sum_i i \sum_j c_{ij} \tag{7.11}$$

$$\mu_y = \sum_j j \sum_i c_{ij} \tag{7.12}$$

$$\sigma_x^2 = \sum_i (i - \mu_x)^2 \sum_j c_{ij} \tag{7.13}$$

$$\sigma_y^2 = \sum_j (j - \mu_y)^2 \sum_i c_{ij} \tag{7.14}$$

For calculation purposes the formulae for the variance and correlation can be made easier:

$$\sigma_x^2 = \sum_i \sum_j i^2 \cdot c_{ij} - \mu_x^2 \tag{7.15}$$

$$\sigma_y^2 = \sum_i \sum_j j^2 \cdot c_{ij} - \mu_y^2 \tag{7.16}$$

$$C = \frac{\sum_i \sum_j i \cdot j \cdot c_{ij} - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y} \tag{7.17}$$

1. The correlation measures the linear dependency of the grey level values in the SGLDM.
2. There is no immediate conclusion for the input image.

As an example all values were calculated based on Figure 7.4. The result is shown in Table 7.2. It can be seen that there is not a significant difference between the results of the W-SGLDM and H-SGLDM, but it must be stressed that they are not the same.

A further example shows the result when applying the five textural measures with the position operator $P(1, 0, 14, 90, 135)$ to the artificial texture of Plate A.10. Only the results of the W-SGLDM are shown in Figure 7.5. It is especially interesting to examine the results for the random texture (No. 3). Especially a low Energy U, high Entropy E and the low Correlation C are noticeable.

	W-SGLDM	H-SGLDM
U	0.1666666865	0.1458333284
E	0.6721802354	0.7555134892
I	0.0648148134	0.0648148209
H	0.8083333373	0.8083332777
C	0.7969884872	0.7195326090

Table 7.2: Example Values of the Descriptors Based on Figure 7.4

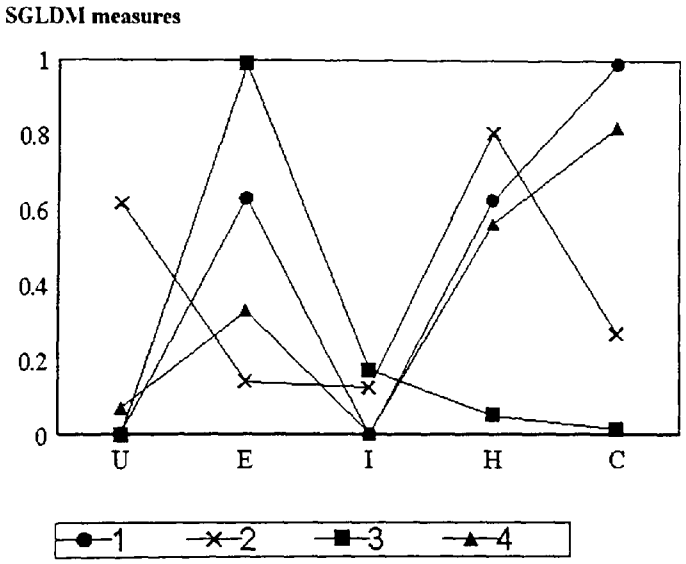


Figure 7.5: The Five Measures Applied to the Artificial Texture

The five above defined measures were fed into the classifier of the SGLDM calculated with the following position operator:

- P(1,0,45,90,135)
- P(2,0,45,90,135)
- P(5,0,45,90,135)
- P(10,0,45,90,135)

That is the offsets are 1, 2, 5, 10 and the four directions (0,45,90,135) are added together. Both the H-SGLDM and W-SGLDM results were taken. The detection results are shown later (Section 9.1.6).

Chapter 8

The Classification of Algae

After an object has been segmented and enhanced, some identification routines have been applied, to calculate various features. These features need to be evaluated for their ability to discriminate. With the help of these features it should be possible to determine the probable species, to which an object belongs. In summary, the steps are:

- to establish, to which of the several groups (in this case species) the currently observed case (object) belongs.
- to select the features which best distinguish the groups.
- to evaluate the accuracy of the classification
- to reject objects which belong to unknown groups and to reject background clutter

Sometimes the features are referred to as pattern and the process to evaluate the correct group is then called pattern recognition.

There are generally two different approaches for pattern recognition.

- Cluster Analysis
- Classification Analysis

However, the algorithms used in both approaches have one thing in common, they need to be taught. This is done by feeding a training set of data to the algorithm, which then develops a means of separating the different groups. If at a later stage new observations are presented to the algorithm, it can then decide, based on the experience of the training set, to which group the new observation belongs.

The difference between cluster analysis and classification analysis is that cluster analysis is based on the idea of unsupervised learning. Sometimes it is

not possible to obtain a training set and link the correct group to each observation. The algorithm has then, with the features as the only available information, to try to establish different groups. The most common algorithm for cluster analysis is the K-means algorithm (Duda and Hart [19], Niblack [64] and Therrien [92]).

Classification algorithms however are trained with the knowledge of the group for all observations in the training set, hence the term supervised learning. Classification algorithms should generally be used if it is possible to provide the group for the training set. This was possible in this application of blue-green algae, and the following classification algorithms will now be explained in detail.

- Minimum Distance Classifier
- Discriminant Analysis
- Neural Networks

The first two algorithms are based on statistical methods, whereas neural networks use artificial intelligence.

In mathematical terms classification can be described as follows. The n-feature vector $\vec{x}$ of an observation will be evaluated with the aid of a classification function and the result is one of the groups g_i:

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \longrightarrow \text{classification function} \longrightarrow g_i \tag{8.1}$$

8.1 Minimum Distance Classifier

The minimum distance classifier is very fast and simple. The Minimum Distance Classifier (Duda and Hart [19], Gonzalez and Woods [27] and Niblack [64]) calculates the means $\vec{m}_i$ of all the different group samples.

$$\vec{m}_i = \bar{\vec{x}}_i = \frac{1}{k_i} \sum_{j=1}^{k_i} \vec{x}_j \qquad \forall \vec{x}_j \in \text{group}_i \tag{8.2}$$

with k_i the number of observations in group i.

A new observation is classed into a group g_i if the euclidian distance between the group mean $\vec{m}_i$ and the observation $\vec{x}$ is smaller than the distances obtained using the other group means. The euclidian distance d_i is:

$$d_i = \sqrt{(m_{i1} - x_1)^2 + (m_{i2} - x_2)^2 + \cdots + (m_{in} - x_n)^2} \tag{8.3}$$

Therefore:

$$\vec{x} \in g_i \qquad \text{if} \quad d_i < d_j \quad \forall j \neq i \tag{8.4}$$

The Minimum Distance Classifier provides optimal statistical results when the variances of the samples of each group are the same (Duda and Hart [19]). If this is not the case the classification with discriminant analysis provides a better classification result. The Minimum Distance Classifier was not implemented in this study, as the condition on the variance could not be met and a better classification algorithm was needed.

8.2 Discriminant Analysis

In addition to the Minimum Distance Classifier the classification with discriminant analysis takes the individual variance of the group samples into account. Discriminant analysis therefore provides an optimal statistical classification (Duda and Hart [19], Gonzalez and Woods [27] and James [41]). The only real restriction

upon discriminant analysis is the requirement on *linear* data, that means that the groups create a cluster, which is formed around a single group mean vector.

8.2.1 Bayes Rule

Consider Bayes Rule applied to the case of discriminating different groups. Firstly the so called *a priori* probability $\Pr(g_i)$ needs to be introduced. This is the probability that an observation belongs to a certain group, however before the evaluation of the individual features takes place. In the general case the *a priori* probability is therefore:

$$\Pr(g_i) = 1/g \tag{8.5}$$

where g is the number of groups.

Sometimes it might be useful to change this probability, for example, if it is known that one sample is more likely to occur than another sample. However in the case of the detection of blue-green algae this could not be assumed.

What really is of interest however is the probability that an observation belongs to a certain group, under the condition of its feature vector. This conditional probability $\Pr(g_i|\vec{x})$ can be expressed with Bayes Rule:

$$\Pr(g_i|\vec{x}) = \frac{\Pr(\vec{x}|g_i)\Pr(g_i)}{\Pr(\vec{x})} \tag{8.6}$$

where:

$$\Pr(\vec{x}) = \sum_{j=1}^{n} \Pr(\vec{x}|g_i)\Pr(g_i)$$

and $\Pr(\vec{x}|g_i)$ is the *state-conditional probability density function* for $\vec{x}$, the probability for $\vec{x}$, given the state g_i.

Based upon this rule the risk of misclassification, that is the probability that the classifier assigns one observation into the wrong class, can be calculated and hence minimised. Because of the possibility of obtaining a minimised risk,

a classifier based upon Bayes Rule is also called an *optimum statistical classifier.* In contrast to the minimum distance classifier the classifiers based on Bayes rule take not only the means of the feature vectors, but also the variances into account. Discriminant analysis is based on Bayes Rule.

8.2.2 Mahalanobis Distance

The Mahalanobis Distance or squared distance of the observation $\vec{x}$ to the mean of the group i, $\vec{m_i}$ is defined by (Therrien [92]):

$$d_i^2(\vec{x}) = (\vec{x} - \vec{m_i})' \, S_P^{-1} \, (\vec{x} - \vec{m_i}) \tag{8.7}$$

where S_P is the pooled covariance matrix. The pooled covariance matrix is calculated as follows:

$$S_P = \frac{1}{k-g} \sum_{i=1}^{g} S_i \cdot k_i \tag{8.8}$$

with g the number of groups, $k = \sum k_i$ the number of samples, k_i the number of observations per group i and S_i the covariance matrix for group i.

The Mahalanobis Distance provides a method for classifying as the smallest value of d_i^2 yields the group, classifying the sample. It is hence quite similar to the Minimum Distance Classifier. In addition it takes the covariance into account. The Mahalanobis distance is therefore the distance in discriminant space.

It can be seen from Equation (8.7) that the same pooled covariance matrix is taken to calculate the distances for each group. The requirement therefore to obtain the optimum result is to have equal covariance matrices for each group. This requirement can not be guaranteed with such data. However Lachenbruch [50] investigates the robustness of discriminant analysis for unequal covariance matrices and states that results can still be satisfactory. The results within this thesis also support this robustness.

Discriminant analysis is also deficient in another way. It classifies an observation into one of several mutually exclusive groups. Sometimes, for example in

this investigation, it is necessary to reject an observation, because it is noise or some other algae. The Mahalanobis Distance was used for this case. If the distance was larger than a set threshold, then the found organism was rejected. The idea to treat the Mahalanobis Distance as euclidian distance is not new (Therrien [92]), but to use the Mahalanobis Distance to reject objects by means of a linear discriminant classifier appears not to have been used before. This reject mechanism and its performance is further described in Section 9.2.

The Mahalanobis Distance was programmed on the SUN to acquire a routine which could directly classify a test image. This required a program which calculated the inverse pooled covariance matrix and the mean vectors for each group from the training data. These values were saved in a file and could then be used in the classification phase to classify the organisms in an image. Statistical packages could not be used for this purpose as their output was not accurate enough.

However statistical packages were used to estimate the error rates and to select the most meaningful descriptors.

8.2.3 Simple Discriminant Function

There are different ways to represent pattern classifiers and one method yields a set of discriminant functions $df_i(\vec{x})$. Simple discriminant functions are synonymous to the Mahalanobis Distance and provide exactly the same classification result. They are defined as:

$$df(\vec{x}) = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots + a_n \cdot x_n \tag{8.9}$$

One function represents one group and the coefficients a_{ij} for the g discriminant functions need to be calculated.

$$df_i(\vec{x}) = a_{i0} + a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + \cdots + a_{in} \cdot x_n \tag{8.10}$$

with

$$a_{i0} = \vec{m}_i{}' \, S_P^{-1} \, \vec{m}_i$$

$$\vec{a}_i{}' = (a_{i1}, a_{i2}, \ldots, a_{in}) = 2\vec{m}_i{}' \, S_P^{-1}$$

for $i = 1, 2, \ldots, g$.

To make a decision about how to assign an observation, all discriminant functions for the different groups are evaluated and the group which maximises its function is chosen:

$$\vec{x}_i \in g_i \quad \text{if} \quad df_i > df_j \quad \forall j \neq i \tag{8.11}$$

This is a straight forward process.

The Statistical Package MINITAB uses these simple discriminant functions. MINITAB was used in this study to estimate the error rates. For the estimation of the error rates it is important not only to consider the error rates when classifying the training set. This would result in a too optimistic outcome. A realistic estimation can not be based on the training set, but must include a testing set. It is often common to divide the available data set into a training set and a testing set, where the training set is used for the calculation of the discriminant functions and the testing set is then classed with these functions to obtain a realistic error estimation. This is not always the best method as only a part of the available data is used to calculate the discriminant functions. If the set is separated for the testing of the accuracy of the classification, however for the final calculation of the discriminant functions all data is used, then the testing could be quite meaningless as the discriminant functions could change significantly. The other reason, why the data set was not divided is that sometimes such a division is not possible due to limitations in the test data. This was the case in this application also, as the sample size was limited.

Lachenbruch and Mickey [49] investigated different methods for the estimation of error rates in discriminant analysis without a necessary division of the data. One method which they examined, sequentially omitted one observation

from the data set. The discriminant functions were calculated with the remaining data set and the omitted observation is then classified. They concluded, that this method should be used if *normality is questionable and the sample size is small relative to the number of variables* (Lachenbruch and Mickey [49]). MINITAB implemented this method and calls it *cross-validation* (XVAL). It was used here to estimate the error rate.

8.2.4 Canonical Discriminant Function

If there are only two groups, two simple discriminant functions would be calculated. If one is subtracted from the other function a decision boundary is obtained. This boundary can also be used for discrimination. Depending on whether a feature vector produces a positive or negative result it belongs to the one or the other group. This decision boundary describes not one individual group, but the differences in the mean and variance between groups. Such discriminant functions are called canonical discriminant functions.

For a classification into g groups with an n-dimensional feature vector the number of canonical discriminant functions is either $g - 1$ or n, whichever is smaller. The reason for this is that for the calculation of the canonical discriminant functions a system of linear equations needs to be solved and for $n < g - 1$ some trivial solutions would occur.

Once all the canonical discriminant functions are calculated and one observation needs to be classified, the decision, to which group the observation belongs is made as follows. All the functions are evaluated and produce a point in the canonical discriminant space. This point is compared with the mean result of the test set by a Minimum Distance Classifier.

The big advantage of canonical discriminant functions in comparison with simple discriminant functions is that through these calculations the usefulness of

the individual features can also be obtained.

Canonical discriminant functions are implemented in the statistical package SPSS, which was used here to select the descriptors which best separate the data. The condition upon which the descriptors were selected was a maximisation of the Mahalanobis Distance between the groups.

The descriptors were selected stepwise and always the most meaningful descriptor based upon the selection rule was included. Sometimes, an already included descriptor was disregarded, because with the inclusion of other descriptors it lost its ability to discriminate. After feeding the descriptors into MINITAB, it was possible that further descriptors were subtracted, because of a similarity with other descriptors. In general, in a selection process there are always less descriptors than the actual number of steps.

A more detailed introduction about canonical discriminant functions and how to select meaningful descriptors is presented by Klecka [46] or SPSS manuals.

8.3 Neural Networks

Neural networks have been used extensively for classification purposes. Neural networks consist of a number of interconnected nodes and can *learn* from experience. Learning is achieved by training on a test set of data.

A good introduction to neural networks can be found in Carling [10]. In the following, two different designs of neural networks will be briefly introduced.

8.3.1 Back-Error Propagation

Back-error propagation is the classic design for a neural network. Input nodes are connected to a hidden layer, which is in turn connected to an output layer

(Pinada [73]). A back-error propagation network is shown in Figure 8.1.

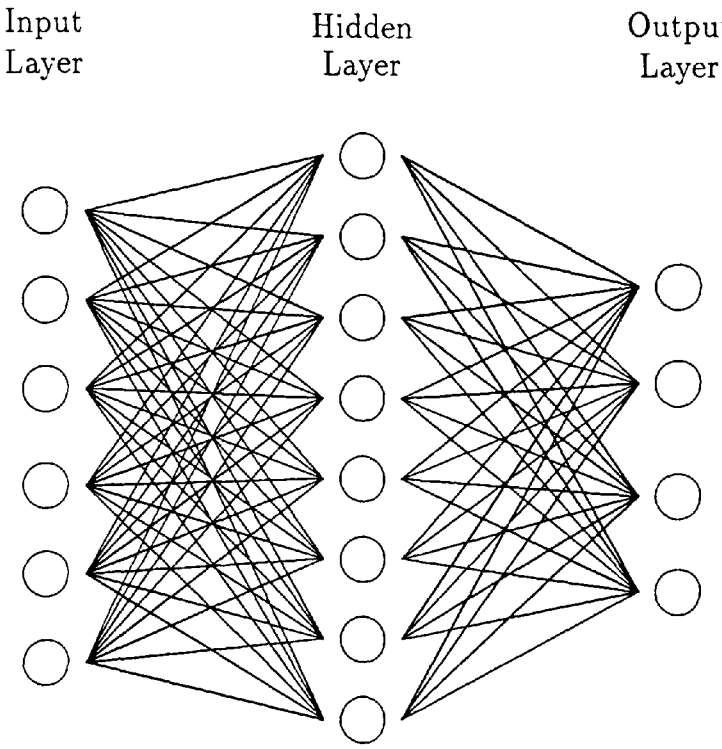


Figure 8.1: Back-Error Propagation Network

The input nodes represent the features and forward the information to the hidden layer. The output nodes represent the result of the classification. The actual *learning* process takes place in the hidden layer, and is achieved in the following way. Every node has several input connections and also several output connections. All the inputs values will be multiplied by a connection specific weight and then summed up. The resulting sum will be passed to the nodes connected to the output connections. It is this weight, which contains the learning information and which needs to be adjusted.

In order to learn, a neural network has to be trained. A test observation will be fed into the neural network and the result compared with the desired or correct result. The error between the desired and obtained result will then be used to change the weights of the neural network, in a mathematical process which moves

backwards, from the output nodes to the input nodes. If the same observation were classed again, the error would be less. Once all test observations have passed several times through the network, the neural network will have learned the characteristics of the set and the classification error will be less.

In this work, a back-error propagation neural network was considered, but no satisfying working network was available. Back-error propagation has to be investigated in further work.

8.3.2 Kohonen Maps

In contrast to the supervised learning of the back-error propagation, a neural network based upon the Kohonen map design, introduced by Kohonen [47], is a method of unsupervised learning.

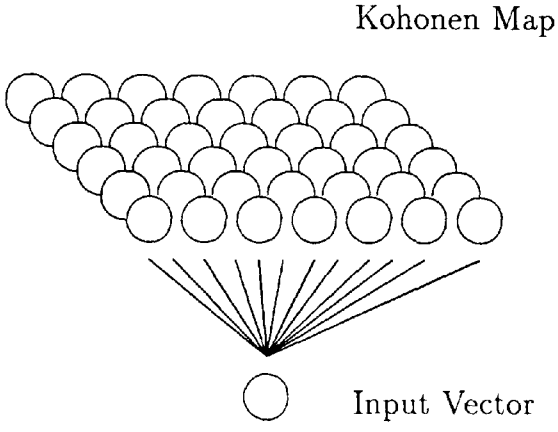


Figure 8.2: Kohonen Network

A Kohonen map generally consists of nodes arranged in two dimensions (Figure 8.2). Each node consists of a vector of the same size as the input feature vector. In the learning process the nodes compete against each other and the node with the smallest euclidian distance of its vector to the input vector *wins*. The vector of the winning node will then be changed, so that in a next pass, it should

win more clearly. Sometimes, neighbouring nodes are adjusted also, to achieve a region which represents the object group. When the learning process has ended, the nodes are assigned to the class, they represent most. In the classification process, the class assigned to the winning node possesses the classification result.

A Kohonen network was used in this investigation on five input features to test the results and compare them with the statistical classification. The results can be found in Section 9.1.6.

8.4 The Major Differences between Discriminant Analysis and Neural Networks

In comparison to the well researched and reliable classification with discriminant analysis, neural networks provide a much newer approach. The main benefit of neural networks is that they have generally no restrictions and can in theory work on any data that has some form of intrinsic structure. Once the learning has taken place, the actual classification should be faster than discriminant analysis, which is very important in real time applications. However the learning itself takes considerably more time than the standard calculations of discriminant analysis.

Discriminant analysis has also some benefits in the selection of meaningful features. In addition, there are various ways to estimate the error rate, whereas in neural networks generally a separation in learn and test data is inevitable.

Chapter 9

Results

In this chapter the results of the investigation are documented. This chapter is mainly concerned about the accuracy of the final classification based upon the feature descriptors. However, the result which should be mentioned first is the excellent segmentation and enhancement achieved. Without such extremely reliable pre-processing further identification would be impossible. The segmentation has little problems in isolating the single organisms within an image, irrespective of their size, shape or appearance. The enhancement not only assists the segmentation in stressing the outlines of the organisms, but also enhances the internal features very well. This is crucial for the reliability of the textural algorithms.

In the next two sections the identification results are summarised. The next section investigates the classification results of the individual features, whereas Section 9.2 displays the results of the features combined. For the individual features it is important to see if these features contain the strength to discriminate. However, the result with the combined features is the crucial and the more important result. In any application the final version should be based always upon a combination of several features from several routines.

To obtain meaningful classification rates, the displayed results are always the results obtained through discriminant analysis with cross-validation (XVAL, Section 8.2.3).

The tables of the results (Table 9.3-9.20) are read in the following way: The table header displays the correct group number of the observations *True Group*, and then. line by line, it is shown, into which group the observations are classified. The last two lines give the number of samples per group and how many were classified correctly. The numbers in the tables are related to the algal species as displayed in Table 9.1.

Table 9.1 also shows the number of samples for each algal group. The omitted numbers 3 and 5 in the tables are reserved for future classification of *Aphanocapsa* and *Gloeotrichia*, which were excluded from this study.

number	species	number of samples (k_i)
0	Background Clutter	20
1	Anabaena	19
2	Aphanizomenon	18
4	Coelosphaerium	17
6	Merismopedia	12
7	Microcystis	18
8	Nostoc	20
9	Oscillatoria	17
10	Eudorina	19
11	Pandorina	18
	$k = \sum k_i$	178

Table 9.1: The Species used for the Results

It can be seen that the background clutter was also treated as one algal group. This was due to the fact that within the test data a considerable amount of background clutter was segmented. To treat background clutter as one species also made the reject mechanism more reliable.

Preliminary results were already published (Thiel and Wiltshire [94]). This paper was primarily concerned with segmentation and enhancement. In addition it showed the possible direction of further study, for example the use of shape and textural features. The second paper (Thiel and Wiltshire [95]) extended these ideas and applied the features to classify two algal species, *Anabaena* and *Oscillatoria*. Finally Thiel et al. [96] classified most species with a detection rate of 98%. However this was done without including any background clutter and without any possibility to reject unknown algae. In addition, the detection rate was calculated directly from the training set without cross-validation and hence the result was very optimistic.

In total 350 features were used for discrimination. The features are listed in Table 9.2, where each feature and its origin is described.

Feature name	Parameters	Number
Circularity		1
Moment Invariants (Boundary)		2 - 8
Fourier Descriptors	$N = 16$	9 - 24
	$N = 256$	25 - 280
Straightness	$len = 50$	281 - 284
	$len = 100$	285 - 288
	$len = 200$	289 - 292
Number of Pixels (not used)		293
Mean and 1st Order Moments		294 - 297
Moment Invariants (Texture)		298 - 304
W- SGLDM Measures	$P(1,0,45,90,135)$	305 - 309
H-	$P(1,0,45,90,135)$	310 - 314
W-	$P(2,0,45,90,135)$	315 - 319
H-	$P(2,0,45,90,135)$	320 - 324
W-	$P(5,0,45,90,135)$	325 - 329
H-	$P(5,0,45,90,135)$	330 - 334
W-	$P(10,0,45,90,135)$	335 - 339
H-	$P(10,0,45,90,135)$	340 - 344
Cell Features		345 - 350

Table 9.2: Summary of the Features used for Classification

9.1 The individual Classification Results

9.1.1 Fourier Descriptors

The results for the Fourier Descriptors with $N = 16$ can be seen in Table 9.3. The detection rate is not very high, but the Fourier Descriptors discriminate extremely well between the curved and filamentous algae. If just such a classification is desired they clearly work very accurately (Table 9.4). Filamentous algae are *Anabaena* (1), *Aphanizomenon* (2), *Nostoc* (8) and *Oscillatoria* (9). Due to the nature of the data a different result can not be expected.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	6	0	0	2	0	4	1	0	0	0
1	0	6	2	0	0	0	2	2	0	0
2	3	5	13	0	0	0	10	10	0	0
4	0	0	0	7	1	1	0	0	2	3
6	1	0	0	1	1	3	0	0	3	0
7	3	0	1	1	4	8	0	1	0	0
8	1	4	2	0	0	0	7	2	0	0
9	0	4	0	0	0	0	0	2	0	0
10	2	0	0	0	2	2	0	0	10	2
11	4	0	0	6	4	0	0	0	4	13
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	6	6	13	7	1	8	7	2	10	13

Percent of *grouped* cases correctly classified: 41.0%

Table 9.3: The Results of the Fourier Descriptors with $N = 16$

Put into Group	... True Group ... clutter	filamentous	curved
clutter	6	1	6
filamentous	4	71	0
curved	10	2	78
Total k_i	20	74	84
Correct	6	71	78

Table 9.4: The Results when Separating Filamentous from Curved Algae

	Percent
first 30	39.3%
first 128	53.4%
last 128	51.7%

Table 9.5: The Results of the Fourier Descriptors with $N = 256$

The results of Fourier Descriptors with a higher N was also investigated. Table 9.5 shows a summary of the results with $N = 256$. In the table *first 30* means that only the first 30 descriptors were taken for the classification. Similarly the meaning of *first 128* and *last 128*, however only the 30 most meaningful descriptors were chosen. A maximum number of 30 was chosen, to compare the results in an unbiased way. It can be seen that the classification results seem to be better than with $N = 16$. However the discrimination between curved and filamentous algae was surprisingly worse. For this reason only the Fourier Descriptors with $N = 16$ were considered later, when all the features were combined.

9.1.2 Moment Invariants (Boundary) and Circularity

The results are shown in Table 9.6. The Moment Invariants do not provide satisfactory results. When comparing them with the Fourier Descriptors they are less successful. This is especially true when separating the algae into two groups, filamentous and curved, as with the Fourier Descriptors (Table 9.7).

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	0	2	1	1	0	0	1	1	0	0
1	0	7	3	0	0	0	6	2	0	0
2	0	3	4	0	0	0	5	3	0	0
4	9	0	0	9	4	5	2	2	1	2
6	4	0	0	3	3	3	0	0	2	1
7	2	0	0	1	0	4	1	2	1	0
8	1	3	4	0	0	0	3	1	0	0
9	1	4	6	0	0	0	2	6	0	0
10	3	0	0	0	2	5	0	0	10	8
11	0	0	0	3	3	1	0	0	5	7
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	0	7	4	9	3	4	3	6	10	7

Percent of *grouped* cases correctly classified: 29.8%

Table 9.6: The Results of the Moment Invariants (Boundary)

Put into Group	... True Group ... clutter	filamentous	curved
clutter	0	5	1
filamentous	2	62	0
curved	18	7	83
Total k_i	20	74	84
Correct	0	62	78

Table 9.7: The Results when Separating Filamentous from Curved Algae

The result of the one feature of circularity is shown in Table 9.8. Even though the classification rate is not very high, the result seems promising. The detection rate is almost as good as with the Moment Invariants, but here, only one single feature was taken. Clearly the measure of circularity is on its own unable to classify sufficiently, it shows however a good potential for combining this feature with some other features.

9.1.3 Straightness Algorithm

The results for the newly developed measures of straightness are shown in Table 9.9. For this investigation the four measures *sd−*, *sd+*, *sdm* and *sds* for all three lengths were taken together to form 12 descriptors. Again, as these measures are solely based on the boundary, the distinction between curved and filamentous algae is important. Table 9.10 clearly highlights this particular ability of the statistical features.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	3	0	0	2	1	1	2	0	3	1
1	1	12	9	0	0	3	6	7	0	0
2	0	3	4	0	0	2	5	2	0	0
4	2	0	0	3	2	0	0	0	7	3
6	1	0	0	2	0	1	0	0	1	1
7	6	2	1	1	4	6	1	6	1	0
8	0	1	3	0	0	1	4	1	0	0
9	1	1	1	0	0	4	2	1	0	0
10	3	0	0	2	4	0	0	0	3	0
11	3	0	0	7	1	0	0	0	4	13
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	3	12	4	3	0	6	4	1	3	13

Percent of *grouped* cases correctly classified: 27.5%

Table 9.8: The Results of the Measure of Circularity

Put into Group	... True Group ...									
	0	1	2	4	6	7	8	9	10	11
0	6	0	0	1	2	1	0	0	3	2
1	0	8	3	0	0	0	6	2	0	0
2	1	5	6	1	0	0	2	6	0	0
4	1	0	0	13	1	1	0	0	0	0
6	4	0	0	1	5	1	0	0	3	2
7	1	0	0	1	3	14	0	0	1	0
8	0	4	5	0	0	0	10	4	0	0
9	3	2	4	0	0	0	2	5	0	0
10	2	0	0	0	1	0	0	0	10	2
11	2	0	0	0	0	1	0	0	2	12
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	6	8	6	13	5	14	10	5	10	12

Percent of *grouped* cases correctly classified: 50.0%

Table 9.9: The Results of the Straightness Algorithm

Put into Group	... True Group ...		
	clutter	filamentous	curved
clutter	6	0	9
filamentous	4	74	1
curved	10	0	74
Total k_i	20	74	84
Correct	6	74	74

Table 9.10: The Results when Separating Filamentous from Curved Algae

9.1.4 Grey Level Distribution Moments and Mean

The results of the mean value and the three first order textures can be found in Table 9.11. These very simple descriptors show a higher detection rate than all of the descriptors of the boundary. In addition only four descriptors achieved this result. From this it can be said already, that much more information is contained within the texture than in the boundary.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	9	2	3	0	3	2	0	0	0	0
1	1	11	4	0	0	0	2	4	0	0
2	3	2	2	0	1	1	6	1	0	3
4	0	0	0	14	0	0	0	1	0	0
6	1	0	0	0	7	2	0	0	1	2
7	2	0	1	0	0	13	1	0	2	0
8	1	0	4	0	0	0	8	1	0	3
9	1	4	2	2	0	0	1	10	0	0
10	0	0	0	1	0	0	0	0	15	3
11	2	0	2	0	1	0	2	0	1	7
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	9	11	2	14	7	13	8	10	15	7

Percent of *grouped* cases correctly classified: 53.9%

Table 9.11: The Results of the Grey Level Distribution Moments and Mean

9.1.5 Moment Invariants (Texture)

Again, the result of the Moment Invariants, now with textural information, is not very encouraging. The results in Table 9.12 show only a little improvement compared with the Moment Invariants of the shape alone.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	3	2	2	1	0	1	3	4	0	0
1	0	7	3	0	0	0	4	2	0	0
2	1	4	8	0	0	0	4	1	0	0
4	7	0	0	8	3	4	0	0	1	2
6	1	0	0	5	6	1	0	0	0	3
7	2	0	0	0	0	2	0	0	0	0
8	0	3	1	0	0	0	5	0	0	0
9	0	3	4	0	0	0	3	9	0	0
10	3	0	0	0	1	6	1	1	13	8
11	3	0	0	3	2	4	0	0	5	5
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	3	7	8	8	6	2	5	9	13	5

Percent of *grouped* cases correctly classified: 37.1%

Table 9.12: The Results of the Moment Invariants (Texture)

9.1.6 Spatial Grey Level Dependency Matrix

A summary of the results acquired with the SGLDM (both the H-SGLDM and W-SGLDM with the four offsets 1, 2, 5 and 10) is given in Table 9.13. Best results were achieved with a distance of 2. In addition there is only an insignificant difference between the W-SGLDM and H-SGLDM results. This was also true, when the results were combined as can be seen at the end of the table. The last line shows a combination of all SGLDM features.

			Percent
1	W-SGLDM	$P(1,0,45,90,135)$	52.8%
2	H-SGLDM	$P(1,0,45,90,135)$	52.2%
3	W-SGLDM	$P(2,0,45,90,135)$	65.2%
4	H-SGLDM	$P(2,0,45,90,135)$	65.7%
5	W-SGLDM	$P(5,0,45,90,135)$	55.6%
6	H-SGLDM	$P(5,0,45,90,135)$	53.9%
7	W-SGLDM	$P(10,0,45,90,135)$	38.2%
8	H-SGLDM	$P(10,0,45,90,135)$	38.8%
9	1+3+5+7		77.5%
10	2+4+6+8		75.3%
11	9+10		78.1%

Table 9.13: The Results of the different SGLDM

The best result for uncombined features was achieved using H-SGLDM with (2,0,45,90,135), that is with a offset of 2 and the angles of 0, 45, 90 and 135, as can be seen in Table 9.14. These five features were also used to test the classification with a Kohonen network. The 10 × 10 Kohonen network used, is displayed in Figure 9.15. The numbers in the Kohonen network state, how often the nodes

were winning when classifying all 179 cases. If a node has the number **0**, it means that this node never won, and can therefore be used to reject unknown organisms. The classification result with the Kohonen network was 68.2%, compared to the 65.5% accuracy of the statistical classifier with cross-validation and 68.0% without cross-validation.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	8	0	0	0	0	0	0	0	0	0
1	1	13	1	0	0	0	4	0	0	0
2	0	2	11	0	0	0	3	2	0	0
4	1	1	1	13	1	0	0	0	0	1
6	4	0	0	1	10	0	0	0	0	2
7	1	0	0	0	0	18	1	0	0	0
8	0	1	0	1	0	0	7	1	2	3
9	0	1	2	0	0	0	1	11	0	1
10	4	1	0	0	0	0	3	1	17	2
11	1	0	3	2	1	0	1	2	0	9
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	8	13	11	13	10	18	7	11	17	9

Percent of *grouped* cases correctly classified: 65.7%

Table 9.14: The Results of the H-SGLDM with $P(2, 0, 45, 90, 135)$

Kohonen Map:

0	0	1	0	0	12	5	7	5	3
0	0	0	10	8	5	8	7	7	7
0	1	1	8	5	0	1	7	1	5
0	0	0	0	0	0	0	0	1	12
0	0	0	0	12	11	9	2	11	9
0	0	0	1	1	3	8	8	10	10
0	1	1	8	2	2	11	10	1	1
0	0	0	9	3	3	2	0	0	0
0	0	0	2	9	2	9	0	0	0
0	0	1	3	2	3	2	9	9	10

Percent of *grouped* cases correctly classified: 68.2%

Table 9.15: The Kohonen Map after Training with the H-SGLDM, $P(2, 0, 45, 90, 135)$.

The best overall result using discriminant analysis with the SGLDM is shown in Table 9.16. The detection rate of 0.781 was the best detection rate for a single method.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	11	0	0	0	0	0	0	0	0	0
1	0	14	2	0	0	0	0	0	1	0
2	0	3	10	0	0	0	0	1	0	0
4	3	0	1	16	0	0	0	0	0	1
6	1	0	0	0	12	0	0	0	0	0
7	1	0	0	0	0	17	3	0	0	0
8	0	1	3	0	0	1	16	0	0	3
9	0	0	1	0	0	0	0	13	0	0
10	3	1	0	0	0	0	1	2	18	2
11	1	0	1	1	0	0	0	1	0	12
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	11	14	10	16	12	17	16	13	18	12

Percent of *grouped* cases correctly classified: 78.1%

Table 9.16: The Results of the SGLDM Combined

9.1.7 Cell Finding Algorithm

The features calculated with the cell finding algorithm produced the classification result shown in Table 9.17. Of course this algorithm has difficulties to distinguish between algae which do not have any cells at all. This explains the high output values of *Pandorina* (11). With filamentous algae it shows some potential in discriminating them, but again, on its own this feature is not sufficient.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	5	0	2	0	3	2	2	2	3	1
1	0	8	4	0	0	0	0	2	0	0
2	0	4	6	0	0	1	1	4	0	0
4	0	0	0	0	1	0	0	0	0	0
6	2	0	0	3	0	2	0	0	1	1
7	4	1	0	1	2	10	0	2	1	1
8	0	3	3	1	0	1	16	0	1	0
9	2	3	3	0	0	2	1	7	0	0
10	0	0	0	0	0	0	0	0	0	0
11	7	0	0	12	6	0	0	0	13	15
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	5	8	6	0	0	10	16	7	0	15

Percent of *grouped* cases correctly classified: 37.6%

Table 9.17: The Results of the Cell Finding Algorithm

9.2 The combined Classification Results

To obtain an optimum classification result the features had to be combined. As mentioned earlier (Section 8.2.4), all features were taken and then a certain rule based on Mahalanobis Distance was used to select the most meaningful descriptors. The selection was done in 30 steps which produced 27 descriptors. These 27 descriptors are shown in Table 9.18.

Feature name	Parameters	Chosen Features
Circularity		1
Moment Invariants (Boundary)		4
Fourier Descriptors	$N = 16$	13, 16, 19,24
Straightness	$len = 50$	282, 283,284
	$len = 100$	
	$len = 200$	292
Mean and 1st Order Moments		294, 295, 296,297
Moment Invariants (Texture)		303,304
W- SGLDM Measures	$P(1,0,45,90,135)$	305, 307,308
H-	$P(1,0,45,90,135)$	
W-	$P(2,0,45,90,135)$	317,318
H-	$P(2,0,45,90,135)$	321
W-	$P(5,0,45,90,135)$	
H-	$P(5,0,45,90,135)$	332
W-	$P(10,0,45,90,135)$	339
H-	$P(10,0,45,90,135)$	343
Cell Features		347,350

Table 9.18: The Selected Descriptors

The result with these 27 descriptors is shown in Table 9.19, however without any reject mechanism, as discriminant analysis has the property to classify one observation into one of mutually exclusive groups.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	12	0	0	1	0	0	0	0	0	1
1	1	15	2	0	0	0	0	0	0	0
2	0	2	14	0	0	0	0	1	0	0
4	0	0	0	16	0	0	0	0	0	0
6	3	0	0	0	12	0	0	0	0	0
7	2	0	0	0	0	18	0	0	0	0
8	0	0	2	0	0	0	20	0	0	0
9	0	2	0	0	0	0	0	16	0	0
10	1	0	0	0	0	0	0	0	18	0
11	1	0	0	0	0	0	0	0	1	17
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	12	15	14	16	12	18	20	16	18	17

Percent of *grouped* cases correctly classified: 88.8%

Table 9.19: The Results of the Combined Features

As already explained in Section 8.2.2 the Mahalanobis Distance was used to try to reject certain observations. The Mahalanobis Distance is a distance in feature or discriminant space and therefore should have some properties which allow certain descriptors to be accepted or rejected. A Mahalanobis Distance larger than 100 was used to reject observations. The *best* observation, that is, the observation which had in its group the smallest Mahalanobis Distance, had a distance of less than 10. It should be mentioned however that the size of the

distance to reject an organism is purely based on experience. The only conclusion that can be made is that for a larger dimensioned feature space the reject distance should be larger. However the reject distance is also based on the covariance of the data. If data has a smaller covariance the reject distance should be smaller.

Table 9.20 shows the final result of the combined features with the reject mechanism in place. Obviously the classification accuracy is now smaller than without a reject.

Put into	... True Group ...									
Group	0	1	2	4	6	7	8	9	10	11
0	10	0	0	0	0	0	0	0	0	1
1	0	15	2	0	0	0	0	0	0	0
2	0	2	13	0	0	0	0	1	0	0
4	0	0	0	15	0	0	0	0	0	0
6	3	0	0	0	12	0	0	0	0	0
7	2	0	0	0	0	17	0	0	0	0
8	0	0	2	0	0	0	20	0	0	0
9	0	2	0	0	0	0	0	16	0	0
10	1	0	0	0	0	0	0	0	18	0
11	1	0	0	0	0	0	0	0	1	17
reject	3	0	1	2	0	1	0	0	0	0
Total k_i	20	19	18	17	12	18	20	17	19	18
Correct	13	15	13	15	12	17	20	16	18	17

Percent of *grouped* cases correctly classified: 87.6%

Table 9.20: The Results of the Combined Features with *Reject*

If in the data some background clutter was rejected this was still taken as

a positive result. This is sensible as background clutter is not a specific organism nor does clutter have any special properties and description.

The algorithm was also tested on some unreal data to check for robustness. An example of such data is given in Plate A.11. This video picture was taken on the campus of the University and shows a construction crane. It was found, that the reject mechanism, although not foolproof, worked on most occasions.

Chapter 10

Algal Identification of Larger Populations

It has been shown how a single object within an image can be identified. For this purpose, the first step was to isolate the object from the surrounding area, that is to segment the object. However what will happen, if there are many objects scattered over the whole image making segmentation impossible. In this chapter a short investigation into such cases is made.

If it is not possible to obtain an image which contains only a limited number of algae, it should always be possible to focus on an area where the field of view is almost completely covered by algae. Examples of such images can be found in the Appendix (Plate A.16-A.19).

As it is not possible to isolate certain algae, the whole image area has to be used for classification. Obviously, only textural algorithms can be used for feature extraction.

The algorithm was developed and tested with the following images of populations of algal species.

- Plate A.16, Anabaena
- Plate A.17, Aphanizomenon
- Plate A.18, Microcystis
- Plate A.19, Nostoc
- Plate A.20, empty background

It must be said, that the image of *Aphanizomenon* is not a very good example of the species. The algae are broken apart and do not really resemble a living scum. The other images, especially the image of *Anabaena*, are good examples of populations of algae. An image containing only background and no algae was also taken, to distinguish between scummy images and an empty image.

To test the methodology, the images were divided into grids of a size of approximately 90 × 90 pixels. This had the effect of dividing each image into 48 parts. Features were calculated for each of the 48 parts of the 5 input images to obtain a sample size of 240.

10.1 Methodology

Firstly, the input images were taken without any enhancement. For each grid the following textural measures were calculated:

- Grey Level Distribution Moments (3 descriptors)
- Moment Invariants (7 descriptors)

The images were then Contrast Stretched and only 64 distinct grey levels were taken. This was necessary for the application of the SGLDM. The five textural measures of Energy, Entropy, Inertia, Local Homogeneity and Correlation were taken with the following offsets:

1, 2, 5, 10, 25.

Again, all four directions (0,45,90,135) were taken together, and both, the W-SGLDM and H-SGLDM were calculated. These 50 measures increased the number of descriptors to 60.

The descriptors were then fed into the discriminant analysis classifier. As in Chapter 9, the results were calculated for the separate descriptors and again after the features were combined.

10.2 Results

Table 10.1 displays the individual results of the textural features. It can be seen that the results are already excellent. The Moment Invariants produce the worst result and the SGLDM discriminate best. This is similar to the results of the classifications of the individual species (Section 9.1).

When the features were combined, a selection process of 15 steps produced 13 features. The 13 features were as follows:

		Percent
Grey Level Distribution Moments		79.2%
Moment Invariants		66.3%
W- SGLDM Measures	$P(1,0,45,90,135)$	90.8%
H-	$P(1,0,45,90,135)$	90.8%
W-	$P(2,0,45,90,135)$	92.9%
H-	$P(2,0,45,90,135)$	92.9%
W-	$P(5,0,45,90,135)$	95.0%
H-	$P(5,0,45,90,135)$	95.0%
W-	$P(10,0,45,90,135)$	92.5%
H-	$P(10,0,45,90,135)$	92.5%
W-	$P(25,0,45,90,135)$	89.6%
H-	$P(25,0,45,90,135)$	89.2%

Table 10.1: The Individual Results of the Classification of Scum

- all three Grey Level Distribution Moments
- one Moment Invariant (ψ_1)
- nine SGLDM textural measures

With the combined features only one out of 240 sample grids was misclassified.

Firstly, this excellent result is based upon the good relation of the number of samples to the number of groups. Secondly, the textural descriptors are used here to classify grids of a rectangular shape, and this shape can not have any negative effect on the classification. In addition a reject mechanism was not implemented, due to the nature of the samples. For a reject, a certain amount of misclassification is needed, to acknowledge when a classification problem occurs. However, when applying the algorithm to for example the image of *Coelosphaerium*, Plate A.3, it was found that the shortest Mahalanobis Distance was excessively

high, compared to the very low values which occurred with the training images. It is sensible to test the results on Plate A.3, because it contains a large number of species and the background does not occupy much space.

In this chapter it was demonstrated, that textural features can also be used efficiently to identify the algal species in populations of algae. However, for a further investigation a database of more images is needed.

Chapter 11

Conclusions

11.1 Discussion

11.1.1 Segmentation

For segmentation purposes, which is the isolation of the individual objects within an image, Thresholding is in most cases superior to Edge Detection. However, in this application neither of them could fulfil the required task sufficiently. It is very often the case, that the segmentation has to be especially developed to fit the special needs involved. This was also the case in this application.

The excellent and very reliable segmentation technique LoG Thresholding was newly developed. This thresholded an image, by taking the pixel neighbourhood convolved with the LoG Operator. This method works irrespective of any object shape or size and only the average grey level of the image has to be adjusted beforehand. Normally, the segmentation is always purpose built to the certain requirements of a particular problem. This is the case also in this application. However it is hoped, that due to its high potential and reliability, LoG Thresholding will also be used for other applications. A disadvantage when using LoG Thresholding is the processing time. Of all the single techniques used for the final algorithm, LoG Thresholding took more processing time than any other technique, due to the convolution of the 9×9 LoG mask with the image. However, the result justifies the increase in processing time for this new developed algorithm.

Edge Detection was very useful in assisting LoG Thresholding by enhancing the appearance of edges. The Morphological Operations proved to be invaluable to smooth the boundary of the object and close some discontinuities. If there were segmentation errors due to the LoG Thresholding, the Morphological Close operation filtered most of them out. A simple Boundary Tracing algorithm was found to be the easiest way to extract the object boundary.

11.1.2 Enhancement

Object Enhancement in the Frequency Domain was ruled out due to the problems with the Fourier Transform. Low Pass Filtering and also Morphological Operations on grey scale images produced an unwanted blurring.

High Pass Filtering is a method to sharpen an image and to bring out more details. Both, the enhancement using convolution and the enhancement with Neighbourhood Averaging produced satisfactory results. Finally the latter method was chosen to enhance the overall appearance of the image, because the cell finding algorithm produced better results with a neighbourhood averaged image than with a convolved image.

For the enhancement of the texture of the individual objects a Contrast Stretch combined with Histogram Equalisation dramatically improves the appearance. However this method destroys first order texture and some textural algorithms have to be used before Histogram Equalisation.

As can be seen the object enhancement is done only with a combination of already established routines.

11.1.3 Object Identification using Boundary Pixel

The Fourier Descriptors showed once again their deterministic strength. The optimal value for N depends on the application. With low values for N, for example $N = 16$ the Fourier Descriptors describe a more overall appearance of the shape. This was sufficient in this investigation. With higher values, for example $N = 256$ high frequency changes of the boundary can also be picked up.

The Moment Invariants were disappointing in this investigation. Compared with the Fourier Descriptors they are not as effective. However the single feature of Circularity produced promising results, even though, when considered in

isolation it is not sufficient for discriminating,

The Hough Transform could not be used in this investigation, however a novel Straightness Algorithm was especially developed and was very useful in the identification process. The Straightness Algorithm extracts features with are close to features used in human perception, and generally it is expected for these features to perform well within the classification.

Naturally, shape descriptors have difficulties in identifying different species, where the shape of same species may vary considerably. However they are still useful in describing the overall appearance of the object boundary, whether the object is more *long* or *curved.* In addition shape description can also be used to assist textural analysis.

11.1.4 Object Identification using Texture

The use of structural approaches for textural object identification is increasing. The classification with the aid of such features is however not straightforward, as most classifiers use statistical data for decision making. In addition, structural approaches need always to be tailor made for a problem, which hinders their general acceptance. In this application, a certain amount of time was used to develop a new problem specific cell finding algorithm. The algorithm then provided information on the number of cells found and also statistical information based upon the size of the cells.

In the statistical approach, the simple Grey Level Distribution Moments were surprising with relatively good results. Compared with the much more complex Moment Invariants, they discriminated the different species more accurately.

The best features in this application were supplied by the five statistical features based upon the Spatial Grey Level Dependency Matrix. All four direc-

tions were taken simultaneously as there was no control on the orientation of the organism. An offset of 2 provided the best uncombined result.

11.1.5 Classification

Firstly, it should be mentioned that, however good the classification algorithm can perform, the results are not only limited by the choice of classifier, but by the strength of the obtained features.

For the classification of blue-green algae discriminant analysis was used. Even though the requirement of equal covariance matrices could not be met, discriminant analysis was able to classify the samples of blue-green algae. Features were selected with the SPSS package and the estimation of error rates with the aid of cross-validation, provided by MINITAB, was chosen. A reject was implemented using Mahalanobis Distance and proved to be reliable. The possibility to reject in discriminant analysis has not been used before.

A short investigation into the possibility of using neural networks, in particular Kohonen maps was undertaken. The use of neural networks is definitely feasible, however further work needs to be undertaken before a firm conclusion can be drawn.

11.2 The Detection of Blue-Green Algae

It has been shown that the automatic detection of blue-green algae is both important and practicable. Blue-green algae play an important role in water quality. Because of their toxicity, the appearance of blue-green algae is of great interest. In addition other algae, which could be included into the investigation, can be used for the indirect measurement of pollutants within freshwater.

Certainly, the algae must be put under a microscope and focused correctly.

But then the automated mechanism can be used to detect and identify the species. The algorithm takes also more than one object per image frame into account and can count the number of species per group.

To add more species of algae to the detection algorithm, test samples of the species need to be obtained and with supervised classification, the inverse covariance matrix and the group mean vectors can be updated for future classification. The segmentation process is already robust enough, to cope with any shape and appearance of algae, as long as the pre-conditions are met. The algae has to appear darker than the background, and the image should have high contrast.

11.2.1 The Biological Key

In this study, the identification process was built in a similar way a biologist uses to detect algae. This included the use of the biological key for automated classification. This is the first time that the biological key was used to assist an automated process.

The establishment of a close connection between the biological key and the identification routines is not trivial, as for example in the cases of the Fourier Descriptors or the five statistical measures of the Spatial Grey Level Dependency Matrix, which are mainly based on pure mathematical routines. However with the Fourier Descriptors or the measure of Circularity it can be established whether the organism found is *long* or *curved.* The straightness algorithm, which was especially developed for this task, is clearly based on the biological key, to establish how straight the algae are at certain points.

The newly developed Cell Finding Algorithm was implemented with respect to the biological key, in order to be able to detect filamentous algae. This would not have been possible otherwise.

The first order textures and the measures based upon the Spatial Grey

Level Dependency Matrix can not be directly linked to the biological key. In a way they work similar to an expert biologist, who because of using his experience can detect species, without any reference to particular features.

This demonstrates, that the biological key is not clearly defined like a mathematical formula. The biological key is a combination of the features and methods used by a biologist to identify algae. In that respect it can be said that the biological key was used, but it was modified to cater for the special capabilities of the automated system, based on already established and newly developed image processing routines.

11.3 Future Work

From this project, several aspects of further work can be derived.

Firstly, different species of algae can be added to the algorithm, in order to be able to identify more algae. If this is done together with an increased number of samples of both new, and already present species, the identification process will be more reliable.

The algorithm has clearly a wide variety of possible applications. Apart from the very problem specific Cell Finding Algorithm the complete methodology can be used to identify any other objects, which need to be classified by means of shape and texture. Due to the general ability of the algorithm to segment an image and the various features which were calculated, little work is necessary to adapt the system for other problems of automated recognition.

An example of such applications is the detection and classification of minute organisms, as their manual classification provides problems due to their size. However, if it is possible to magnify such organisms with a light microscope to a sufficient size (for example a minimum boundary length of 200 as in this work),

then the proposed algorithm should be able to work on them. The restriction to a minimum size of an organism in an image is necessary because of the digital character. Once the image is grabbed, enlargements do not increase the information contained within the object and clearly a minimum amount of information is needed for the calculations of shape and textural features. Other possible examples of applications could be found in the plant world.

In addition to further work which derives from the overall algorithm, several interesting topics can also be found in certain isolated parts of the automated system. For example, the identification with structural texture algorithms also deserves further investigation. It was found, that there exist already a lot of theory about the subject, however practical applications are rare. Structural texture could become invaluable for textural analysis as it is close to human perception. The textural measures already available are mainly based on a statistical approach and lack this closeness.

In this investigation, a reject mechanism had to be implemented in the classification process with discriminant analysis. For this purpose the Mahalanobis Distance was used. This appears to be a novel approach and it could be an interesting project to show, how this distance is influenced by the number of features used and also by the covariance matrix, which describes the shape of the data cluster in the feature space.

A study which is already under way is the further analysis of the classification with neural networks. A Kohonen network and the combination of different network theories will be investigated. The results should be compared with respect to

- the possibility to select meaningful features
- the ability to estimate the error rate realistically
- the reliability of the actual classification

- the presence of any pre-conditions on the data
- the ability to cope with missing or corrupted data
- the possibility to reject unknowns
- the speed of the training process
- the speed of the classification process

It is a very challenging task for neural networks to try to beat the statistical classifier, which has already achieved excellent results.

Finally it is hoped that future image processing packages will provide a built-in classifier, and that it is then not necessary to transfer data to a statistical package for discriminant analysis or to a separate neural network.

Appendix A

Images

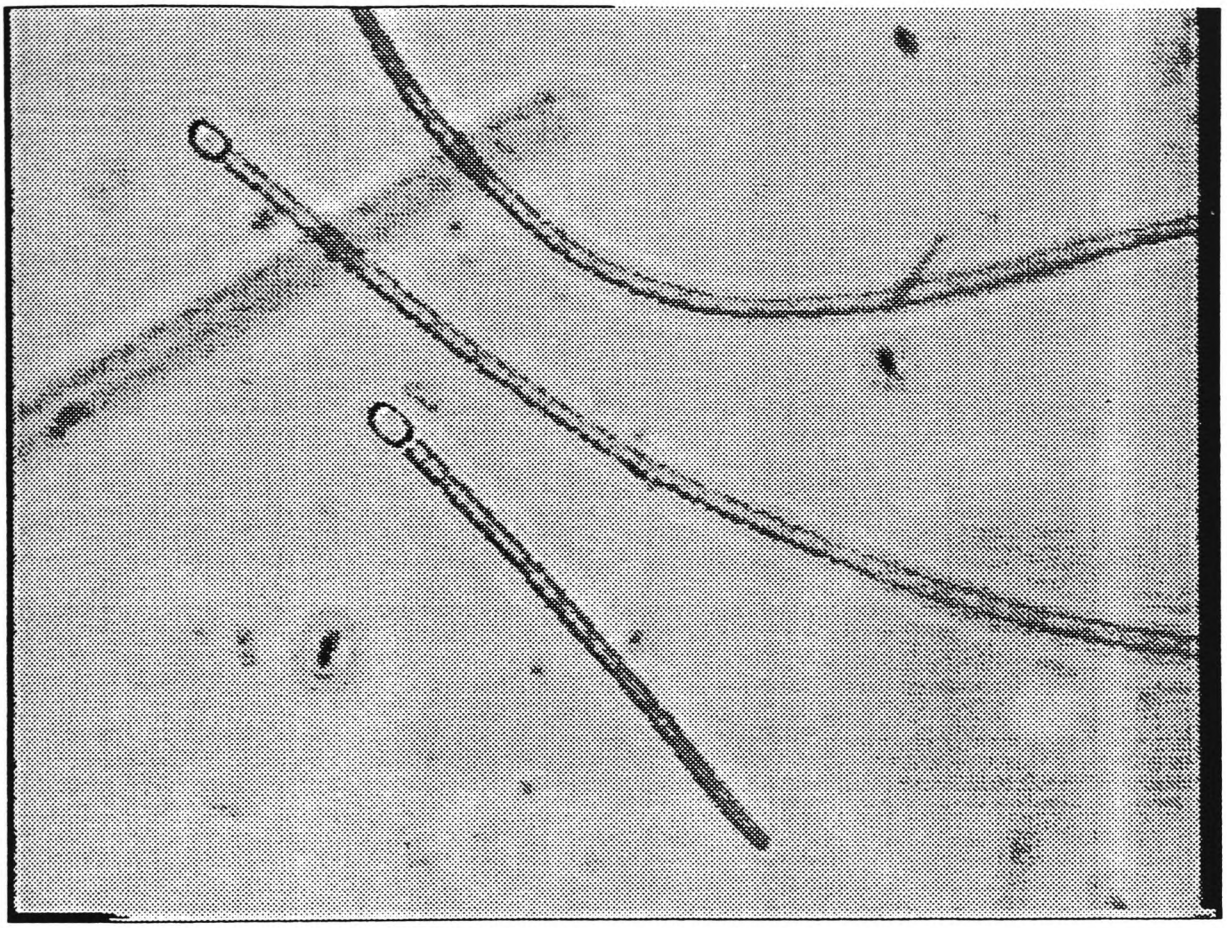


Plate A.1: Anabaena

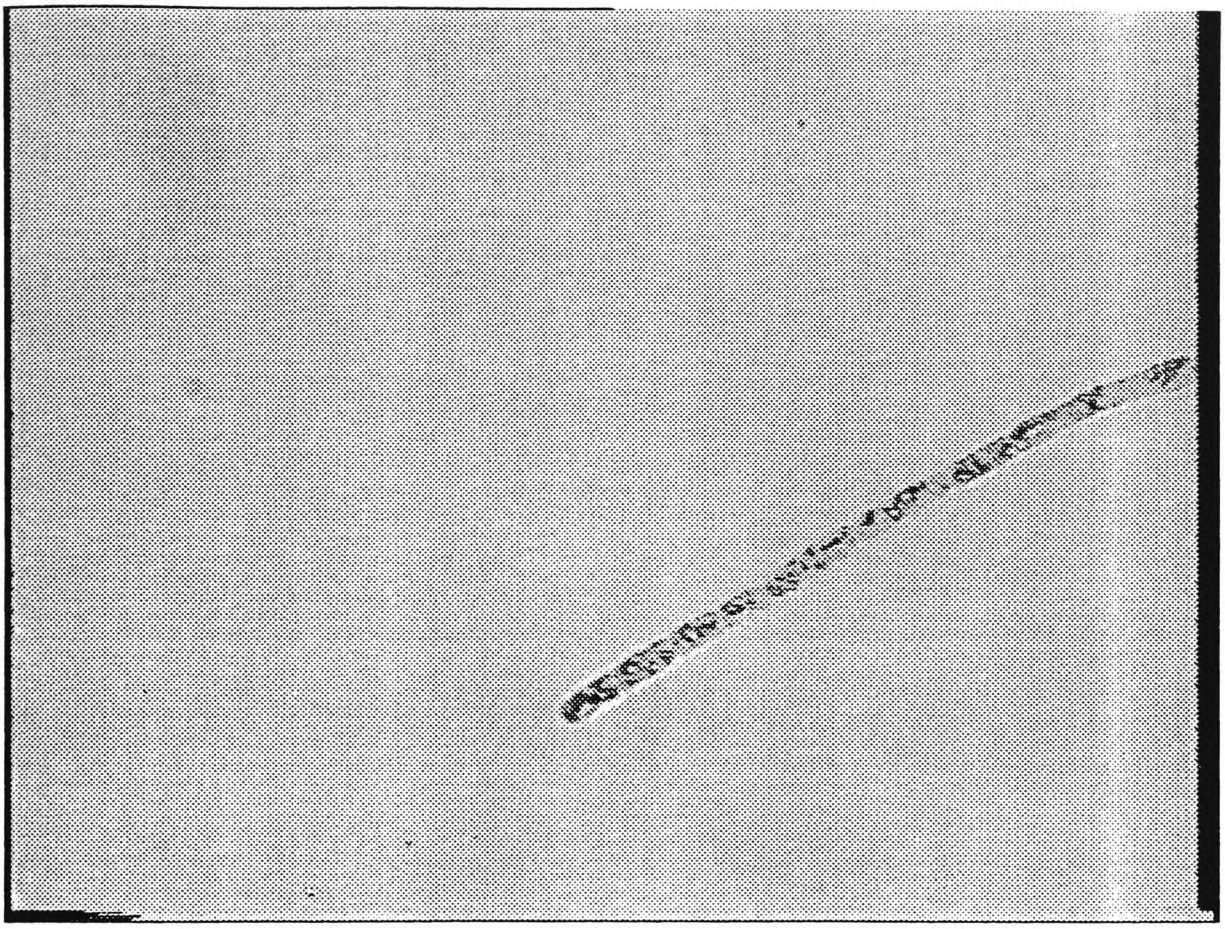


Plate A.2: Aphanizomenon

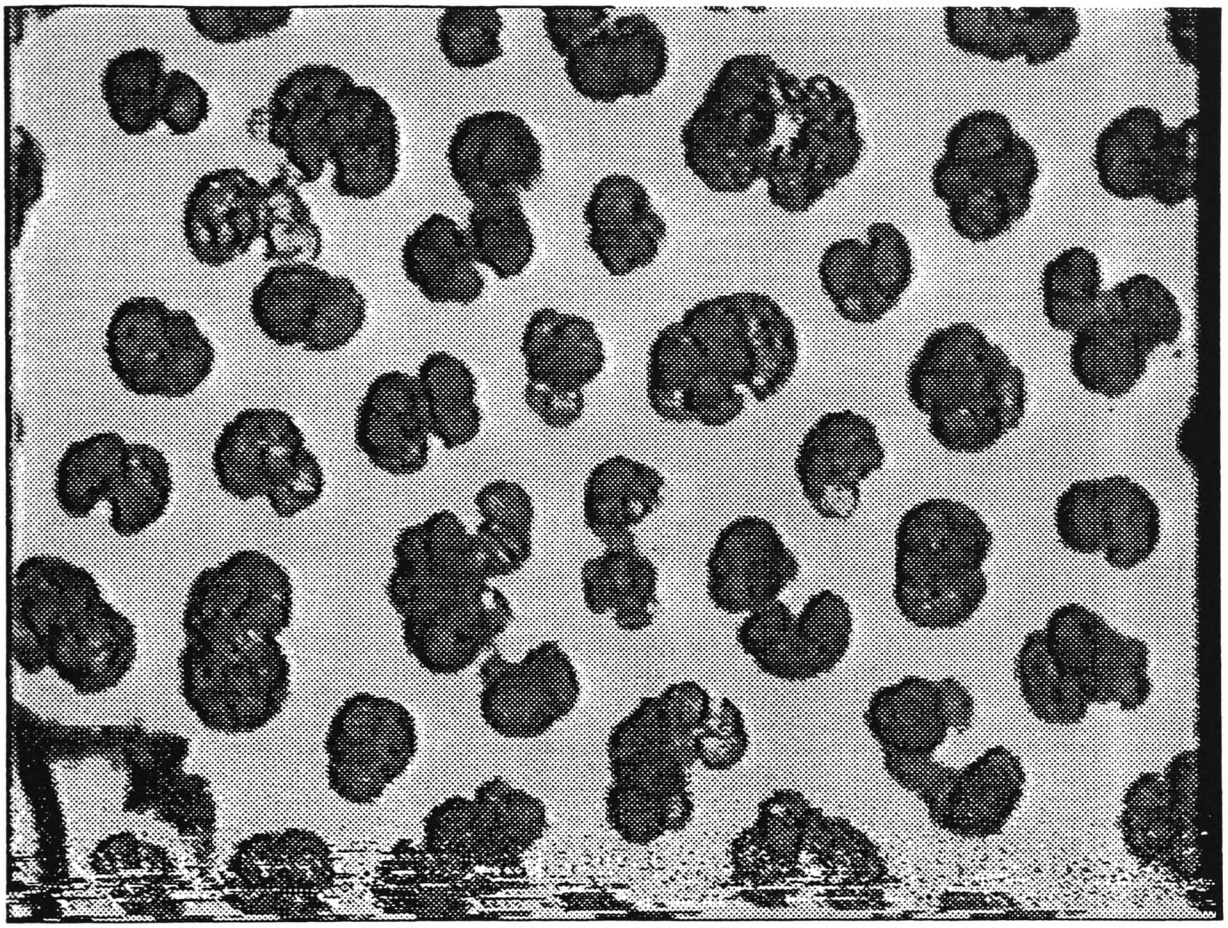


Plate A.3: Coelosphaerium

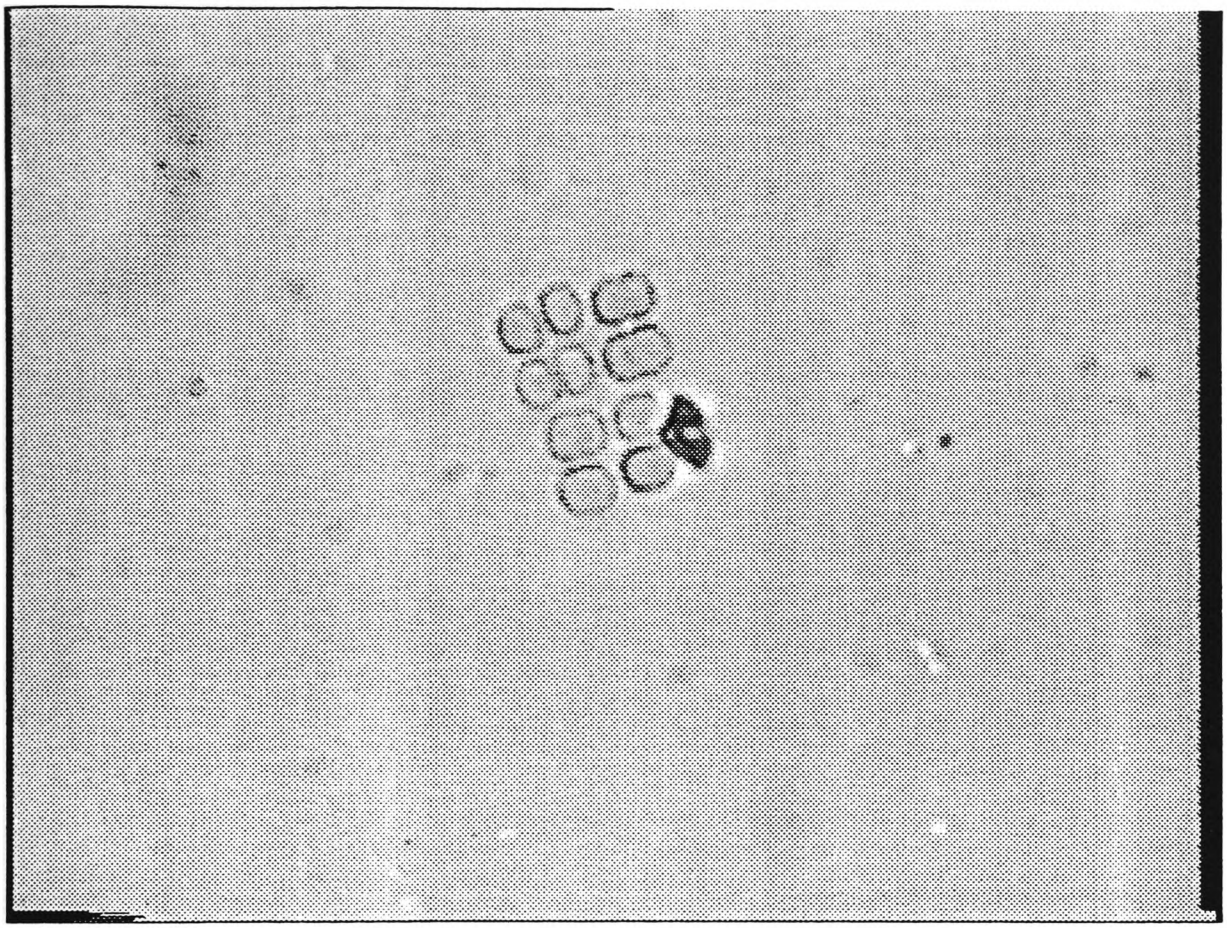


Plate A.1: Merismopedia

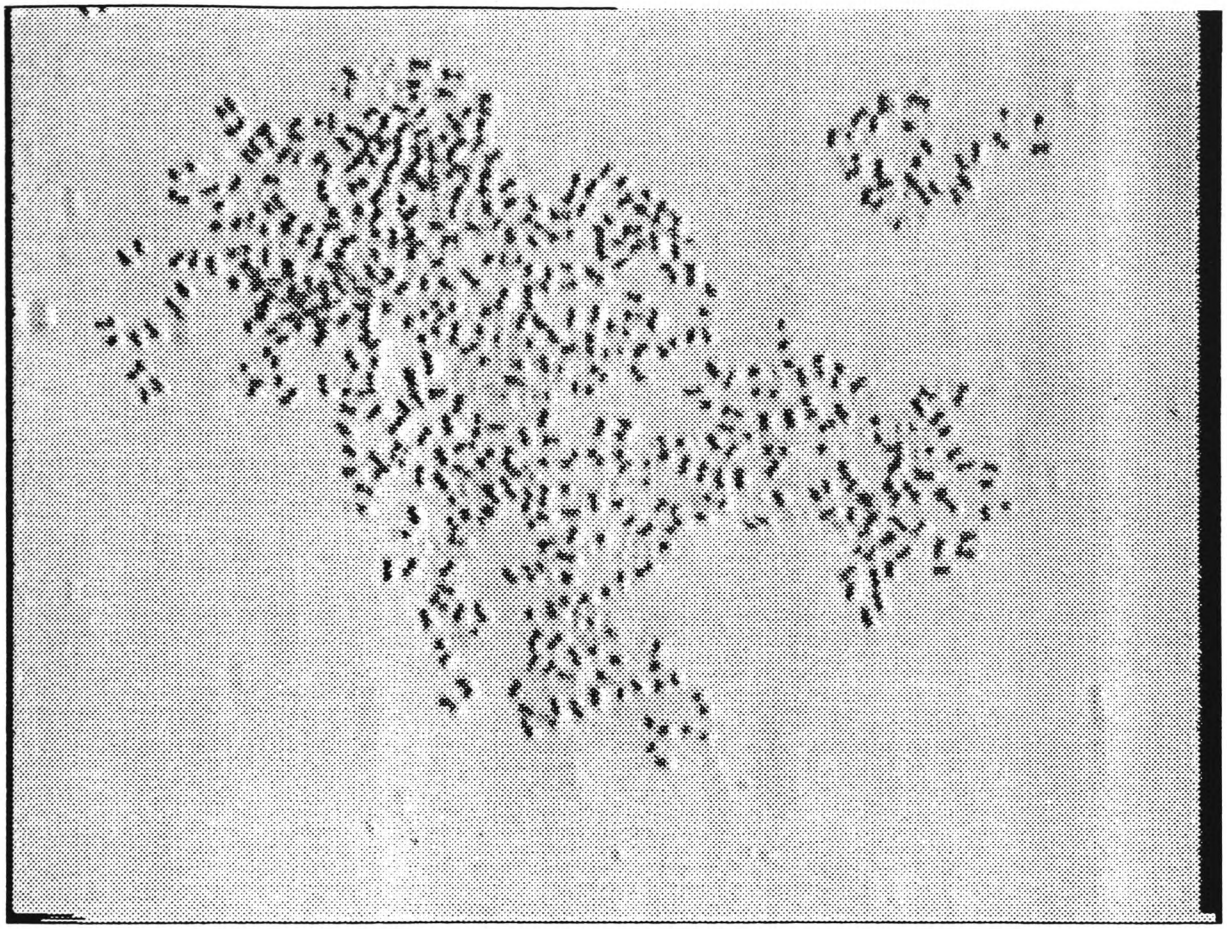


Plate A.5: Microcystis

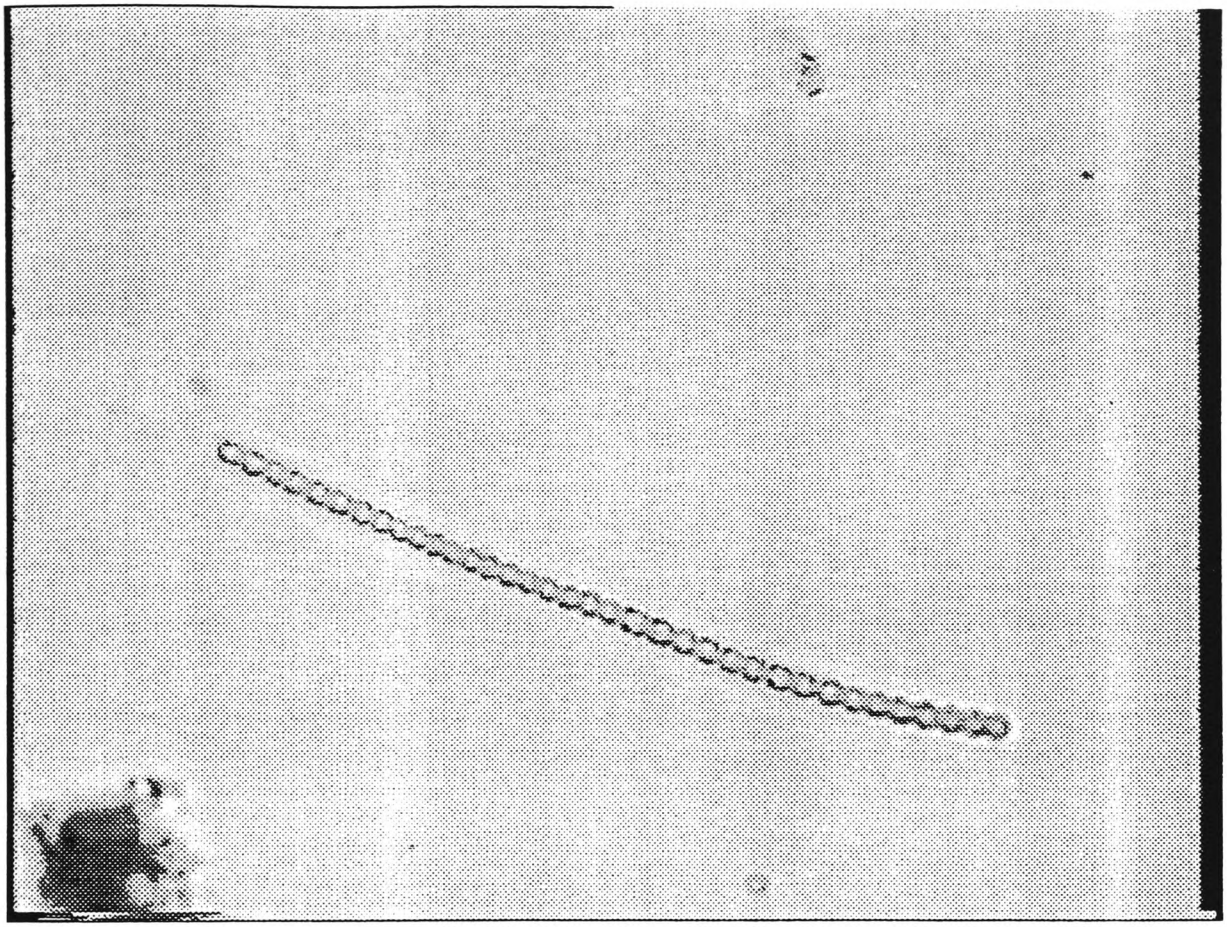


Plate A.6: Nostoc

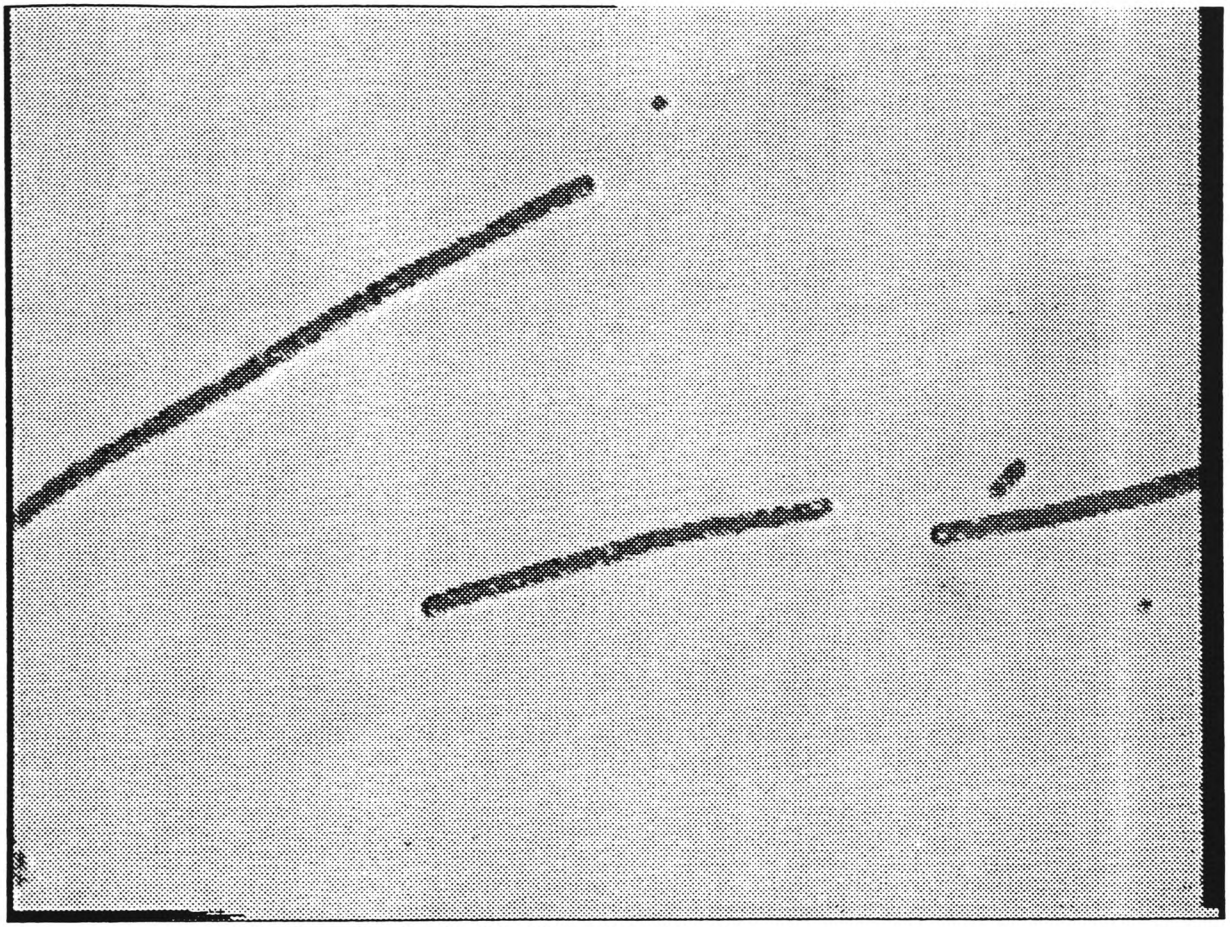


Plate A.7: Oscillatoria

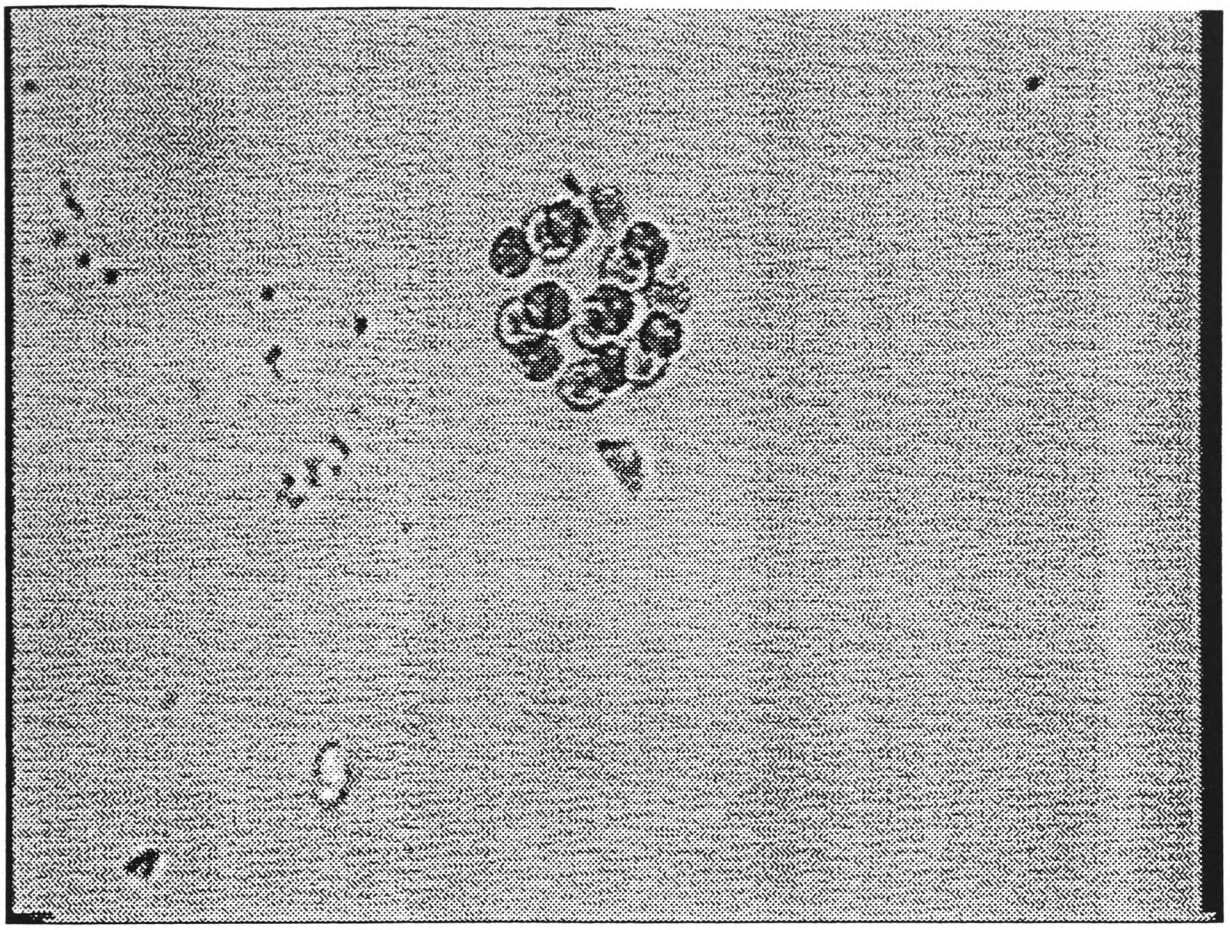


Plate A.8: Eudorina

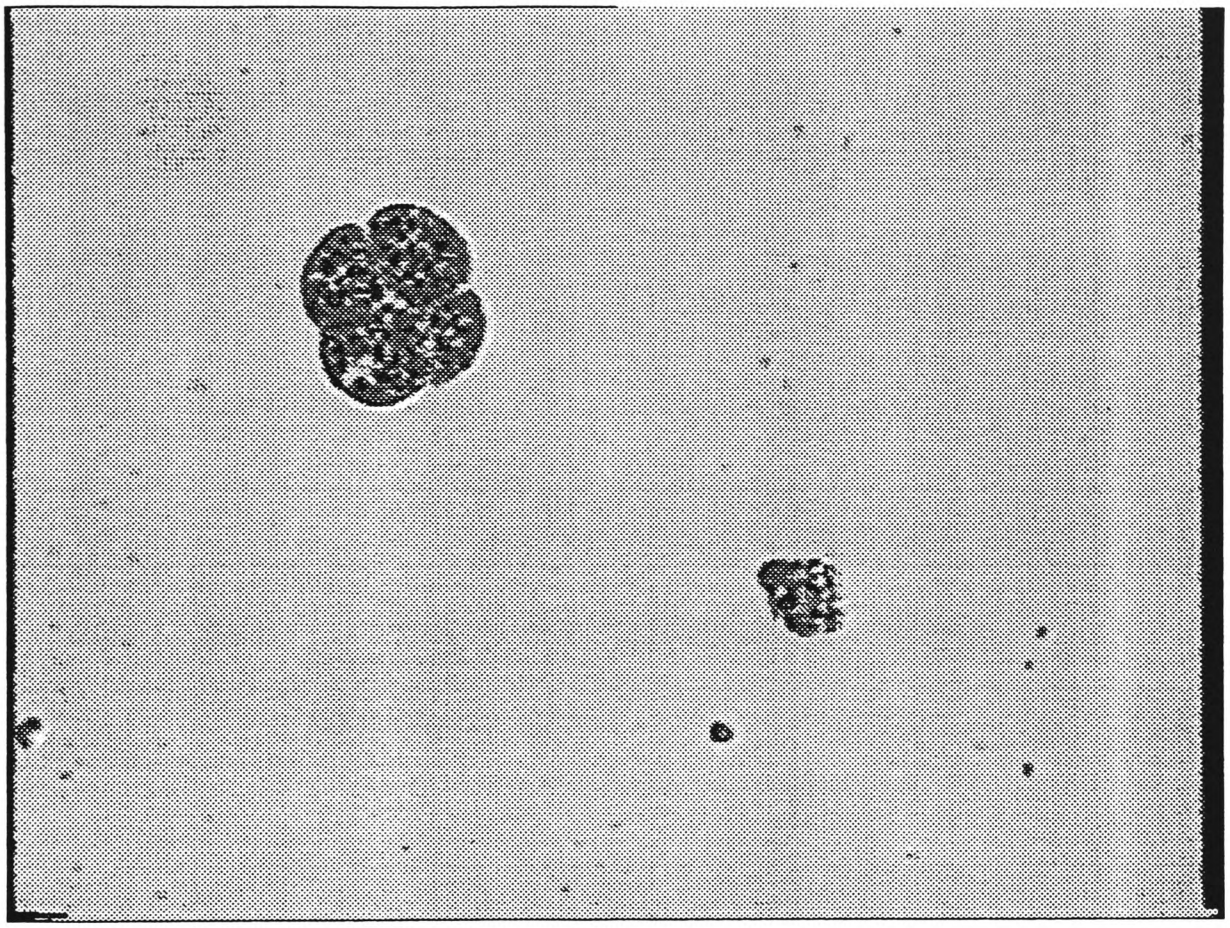
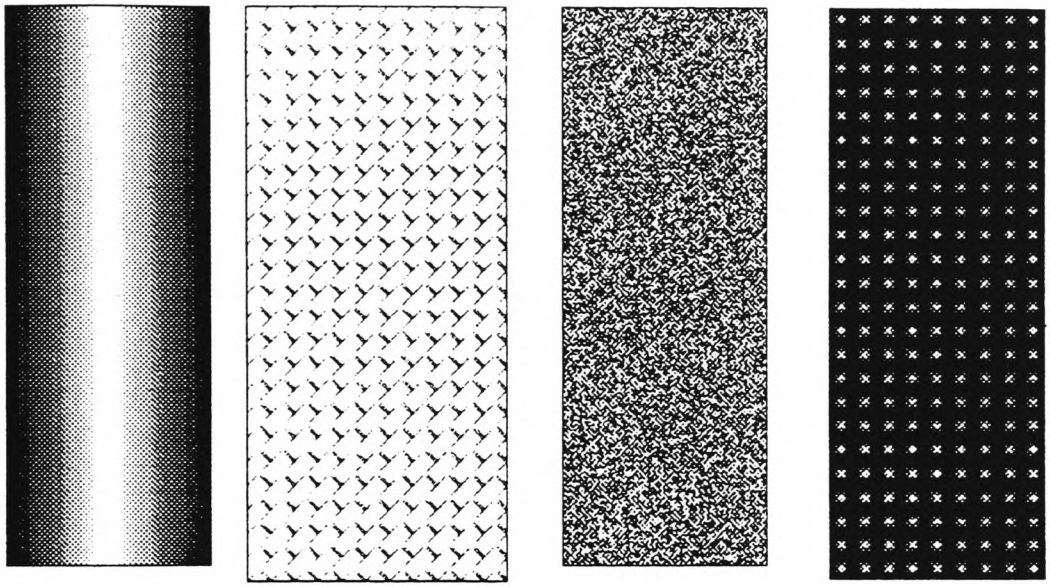


Plate A.9: Pandorina



(1) (2) (3) (4)

Plate A.10: Artificial Texture

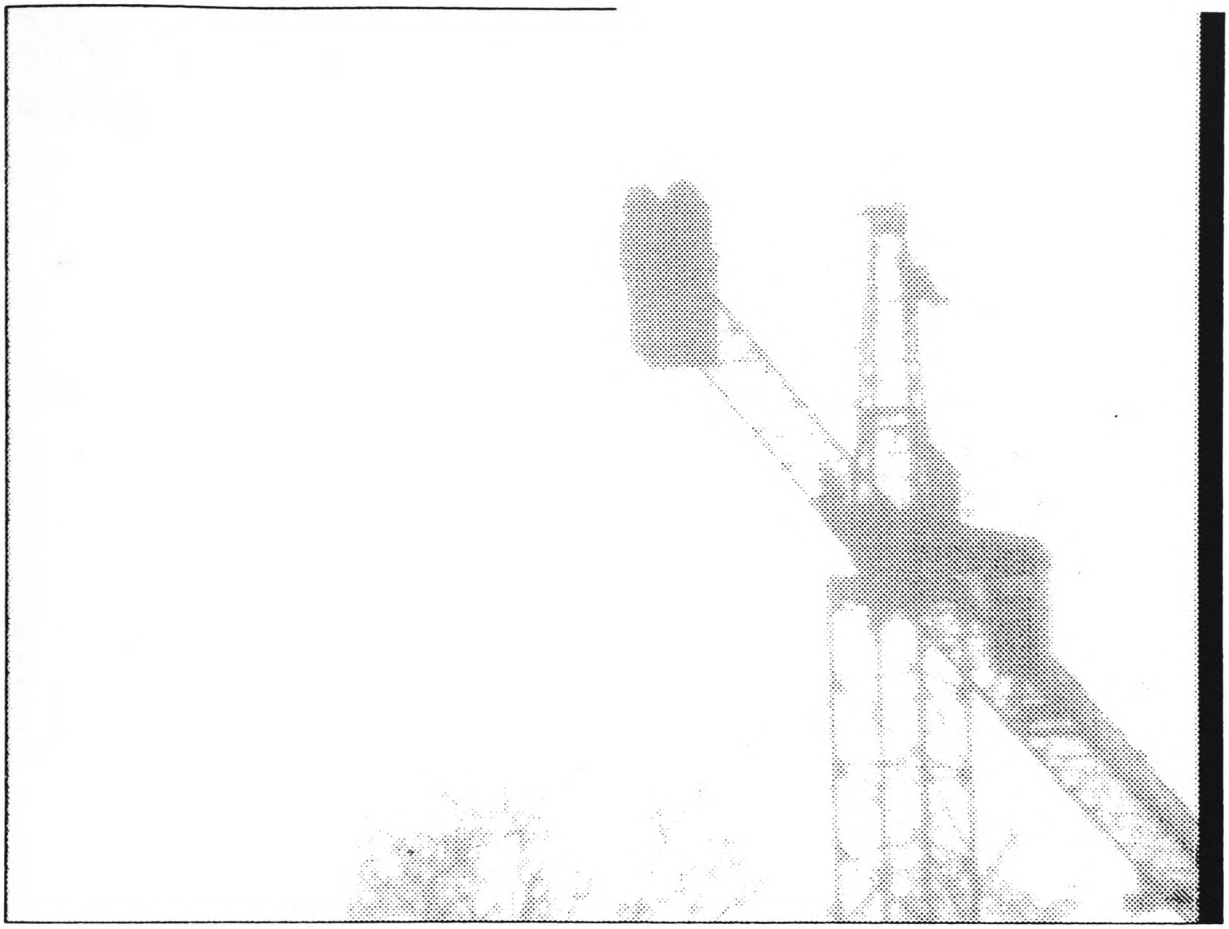


Plate A.11: Test Picture (Crane on Campus)

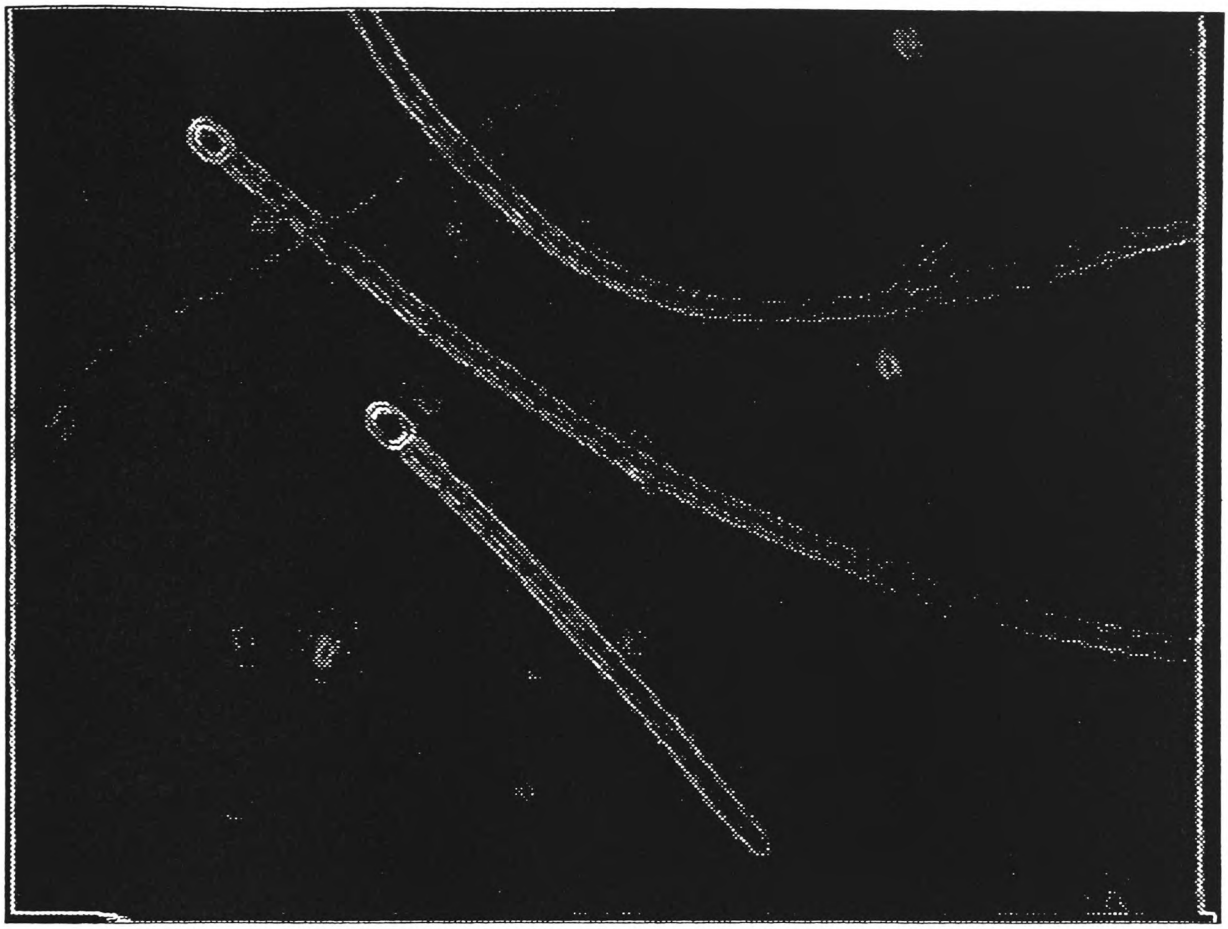


Plate A.12: Plate A.1 after Edge Detection

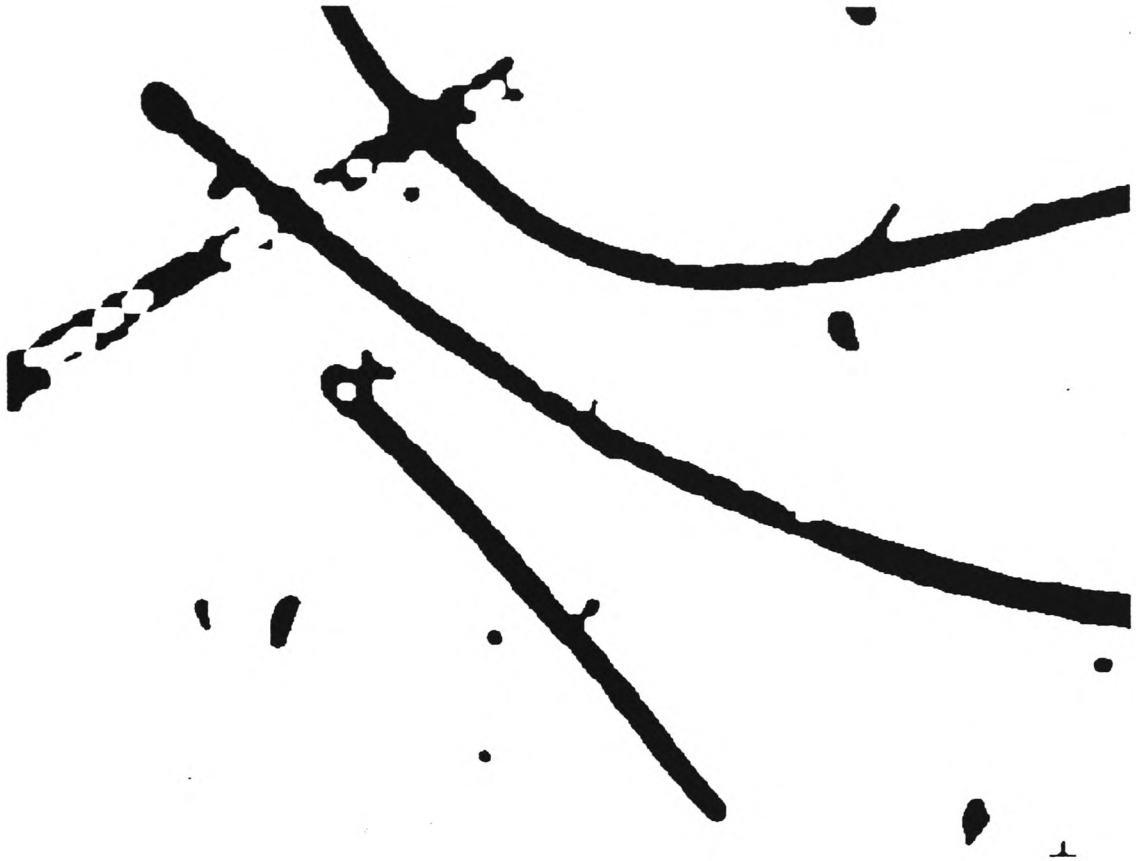


Plate A.13: Plate A.1 after LoG Thresholding and Morphological Close

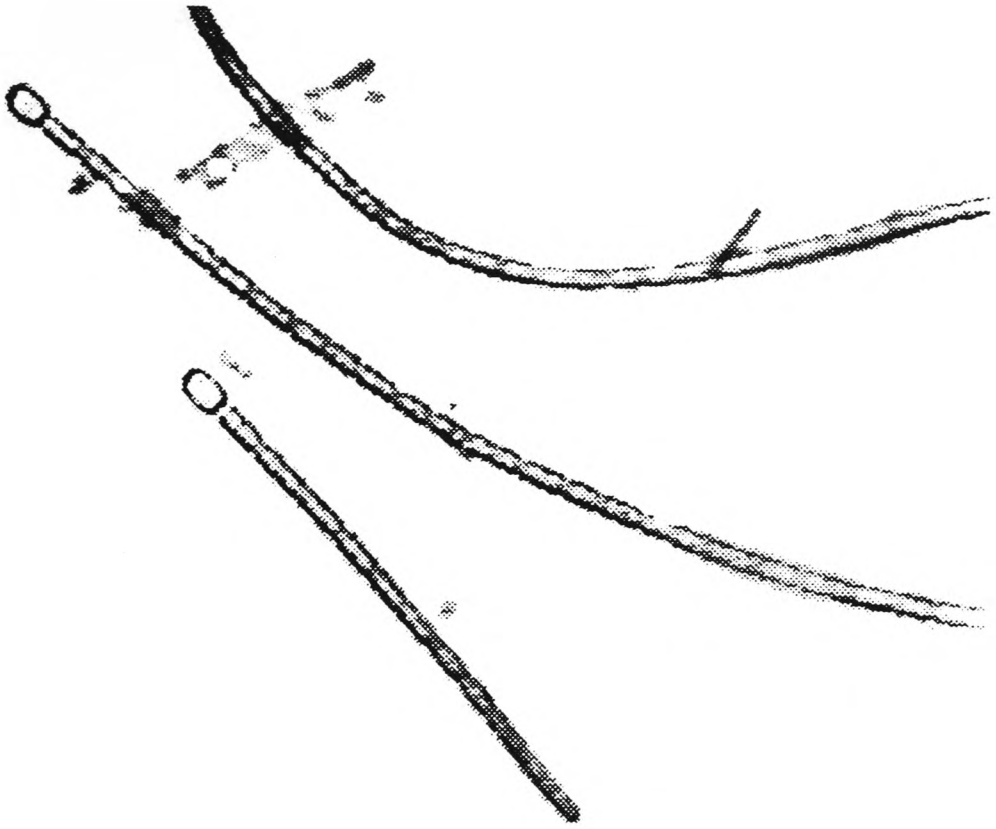


Plate A.14: The Histogram Equalised Objects of Plate A.1

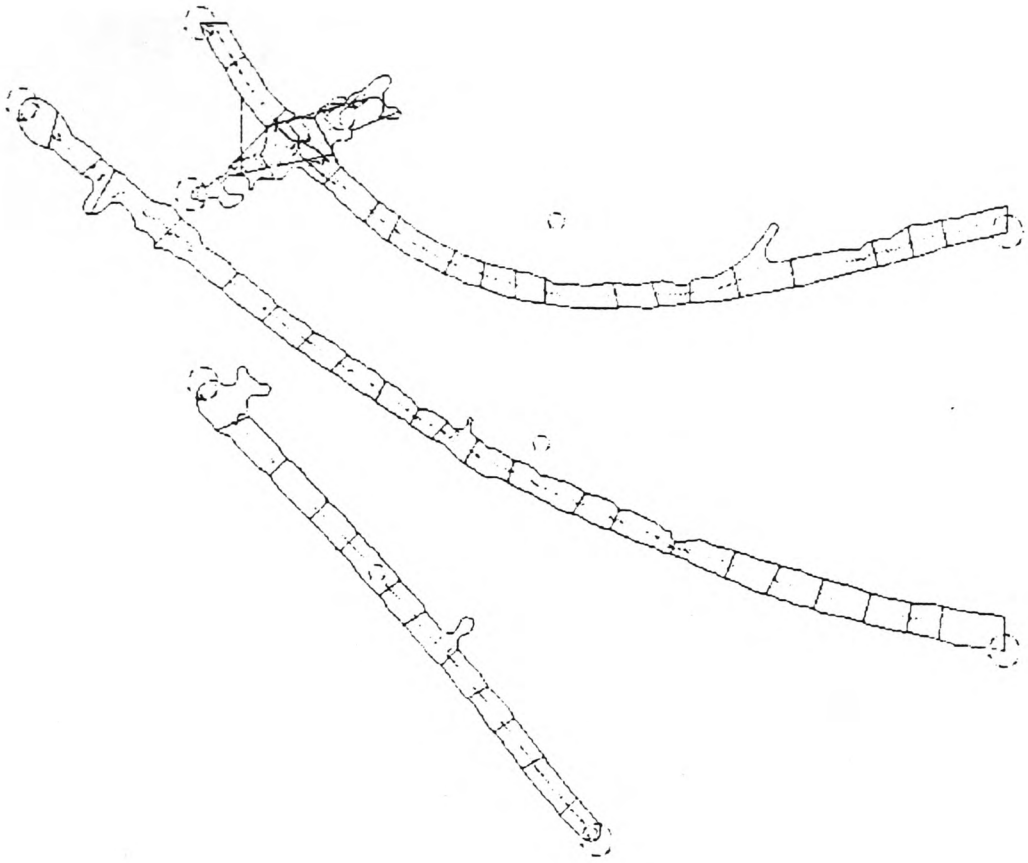


Plate A.15: The Boundary with Important Points and Detected Cells

Plate A.16: Scum of Anabaena

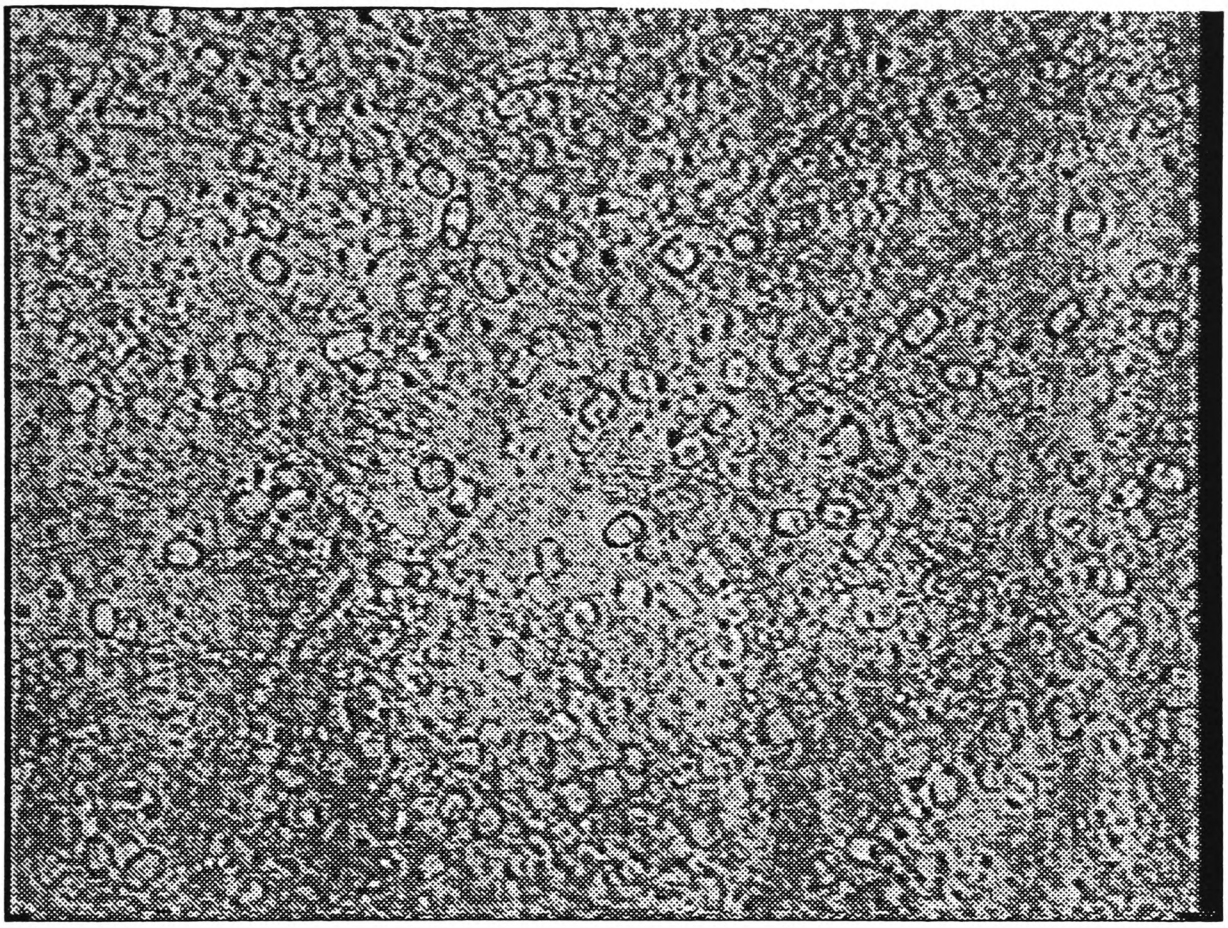


Plate A.17: Scum of Aphanizomenon

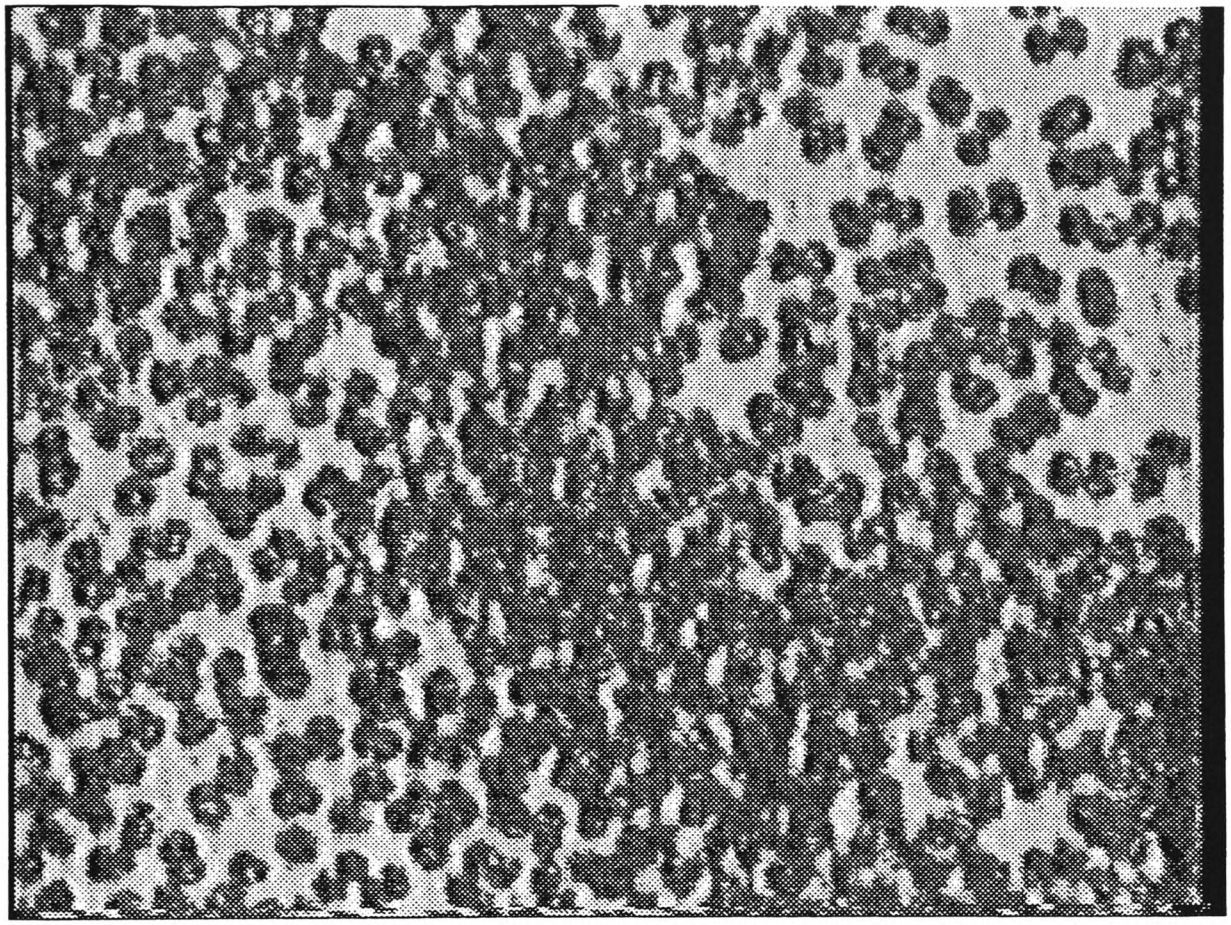


Plate A.18: Scum of Microcystis

Plate A.19: Scum of Nostoc

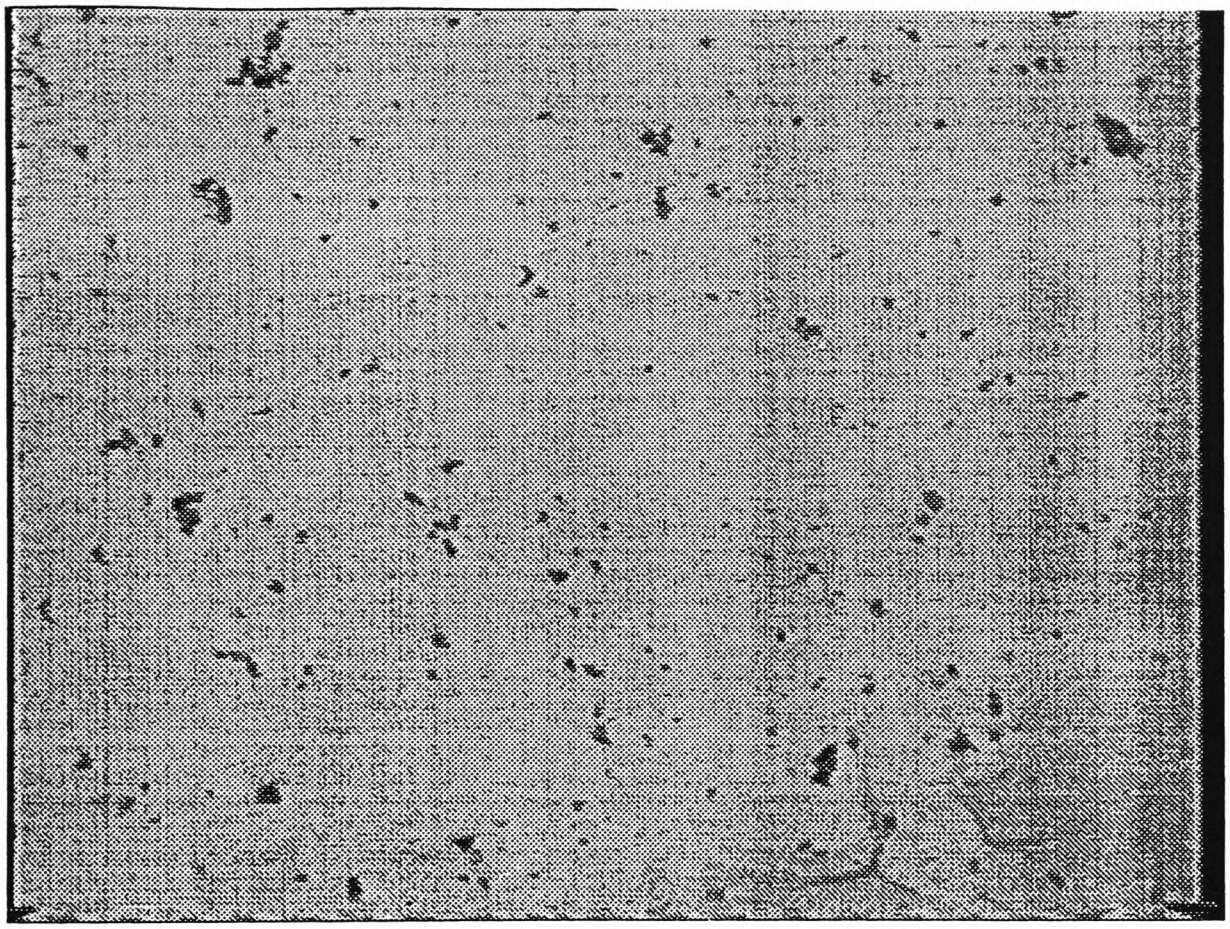


Plate A.20: Empty Microscope Image

Appendix B

The Computer System

B.1 Manual of the System

In this section the computer program and its usage is briefly described. In order to use the whole package it is necessary to read and understand the whole section.

There are two programs for the automated detection of algae:

- hunt.bin
- final.bin

The main difference between them is that *final.bin* incorporates a graphical user interface and displays the images on the screen. *final.bin* must be used when direct work on a video image is required.

There is one program for calculating the data necessary for discriminant analysis:

- make_da.bin

B.1.1 How to use *hunt.bin* and what it does

With no arguments supplied, the program asks for the path of the image file to be processed and also for the name of the image file. This information can also be given in the command line.

For example:

hunt.bin path name the image path/name will be processed

hunt.bin name the image name in the default path will be processed currently the default path is 'set'

While running, the program outputs the length of the segmented objects and four numbers resembling the coordinates of a box around the objects. After each object was processed a message appears.

The information on the run are stored in several files:

path/name-dat: contains the features of all the objects found

path/name_x: image of the segmented objects, where x = *a* first object and so on

path/name_x_end: image of the object containing boundary and the filaments found

path/name_x_fd: contains the 256 fourier descriptors

path/name_x_log: contains textural information

path/name_thr: image of LoG thresholded input file

hunt.bin is used normally only to test the methodology or to see the results of some subroutines. There are two more options to use *hunt.bin.*

hunt.bin name 3

The option **3** must be specified in the command line after the name of the image to be processed. The changes to above are:

- Only images in the default path can be processed.
- In order to save disk space the images (*path/name_x, path/name_x_end, path/name_thr*) are not saved.
- The features of the objects found are appended to the file *algae.dat* and hence not written to *path/name-dat.*

- In order for *algae.dat* to be used further, at the end of the feature list a x is printed which can then be changed to the correct species number.

This option is very powerful when used in a batch file and several images are processed sequentially. Later on, the x in *algae.dat* can be changed to the correct species number for calculation of the discriminant information.

hunt.bin name -3

As above, the option must be specified in the command line. This option means that the image is analyzed and the objects are classified using discriminant analysis. The stored information is the same as without option, but the features are not stored in a single file *path/name-dat* but in individual files *path/name_x.dat.*

B.1.2 How to use *final.bin* and what it does

The main program *final.bin* works direct with images supplied through the video link. A video connection should be established and *final.bin* can start to run[1].

When starting a window will pop up. It has three options.

train This is used to train the data. It means that the features are collected in the file *algae.dat* for further calculation.

test This is used to determine what species the algae in the examined image are.

exit This exits the program

When *train* or *test* are selected the first two options change.

load The image to investigate is to be loaded from the hard disk

[1]In the IP Lab it can only run on Server2.

preview The video image is previewed

When *load* is selected the program asks for the path and the name of the image and processing begins. When *preview* is selected, the image is previewed and can then be grabbed (selection of the *grab* button) and the processing begins. Alternatively it is still possible to load an image from the hard disk.

Training with an image from the hard disk

Firstly the thresholded image is shown in the window. After that the segmented objects are displayed (currently in pseudo colour) and the user is asked to type in the correct species number in the command tool. The features together with the image file name and typed in species number are appended to the file *algae.dat*. Further information can be found in the files starting with **vfc**.

Training with a video image

As 'Training with an image from the hard disk', but instead of the file name, which is now not available, a number is written into the file *algae.dat*. This number is the count of the seconds starting at the beginning of 1970 to the moment when the video image was grabbed. The number can be made understandable with the program *when*.
For example:

```
when 780756647
is Wed Sep 28 12:50:47 1994
```

Testing with an image from the hard disk

As 'training', but at the end of processing the result of the processing is displayed. In addition the calculated features are not saved in *algae.dat* but in

path/name_x.dat.

Testing with a video image

As 'testing with an image from the hard disk', but the calculated features are saved in *number_x.dat*, where number is the count of the seconds. Testing with a video image represents the final use of the PhD project. A video image is loaded and without any user interaction the computer investigates the image and classifies the objects found, printing the result on the screen.

B.1.3 How to use *make_da.bin* and what it does

Firstly it is necessary to understand the file *algae.dat*. *algae.dat* is the file into which the calculated features of each investigated object are written, together with the correct number corresponding the species.

The programs *final.bin* with the option *training* and *hunt.bin* with the option **3** will update the file *algae.dat*. When *hunt.bin* is used, the species number has to be inserted into *algae.dat* manually.

Once the information in *algae.dat* is complete and contains all the desired test results, the file needs to be reformatted. This is done by typing:

dat2nn.bin algae algae 350 d

The program *dat2nn.bin* reads the file *algae.dat* containing 350 features and creates the second data file *algae.da*.

make_da.bin can read the information in *algae.da* and provides then the necessary calculations for discriminant analysis. The results are than stored in the file *dr.dat*, which is read when *hunt.bin* and *final.bin* classify the objects.

The features which are used for discriminant analysis are hard-coded into *make_da.bin* but can be changed in the source without any problems

A change of features is normally only required, when there are new species included. The then updated file *algae.dat* should be loaded into MINITAB and SPSS and statistically evaluated for the most meaningful features. The program *make_da.bin* can then be changed accordingly.

In order to load the features from the file *algae.dat* into any neural network package it is necessary to reformat this file also. The reformat is performed by typing:

dat2nn.bin algae algae 350 n

The file *algae.nn* is created, which is similar to *algae.da*. However it does not contain the species number at the end of each observation, but twelve **0**'s and one **1**, where the position of the **1** indicates the species number. This resembles the desired output which a neural network should provide, when classifying this particular observation.

B.2 The Program Structure

As can be seen in Section B.1, the two programs *hunt.bin* and *final.bin* perform generally the same operations. They represent only the user interface. Both main programs call the same subroutine, *huntsub.c*, to perform the necessary steps for the automated detection, identification and classification of algae.

Figure B.1 is a simplified display of the program structure of the developed algorithm and resembles the subroutine *huntsub.c*. It can be seen how the different routines work sequentially and how the algorithms basic structure is designed.

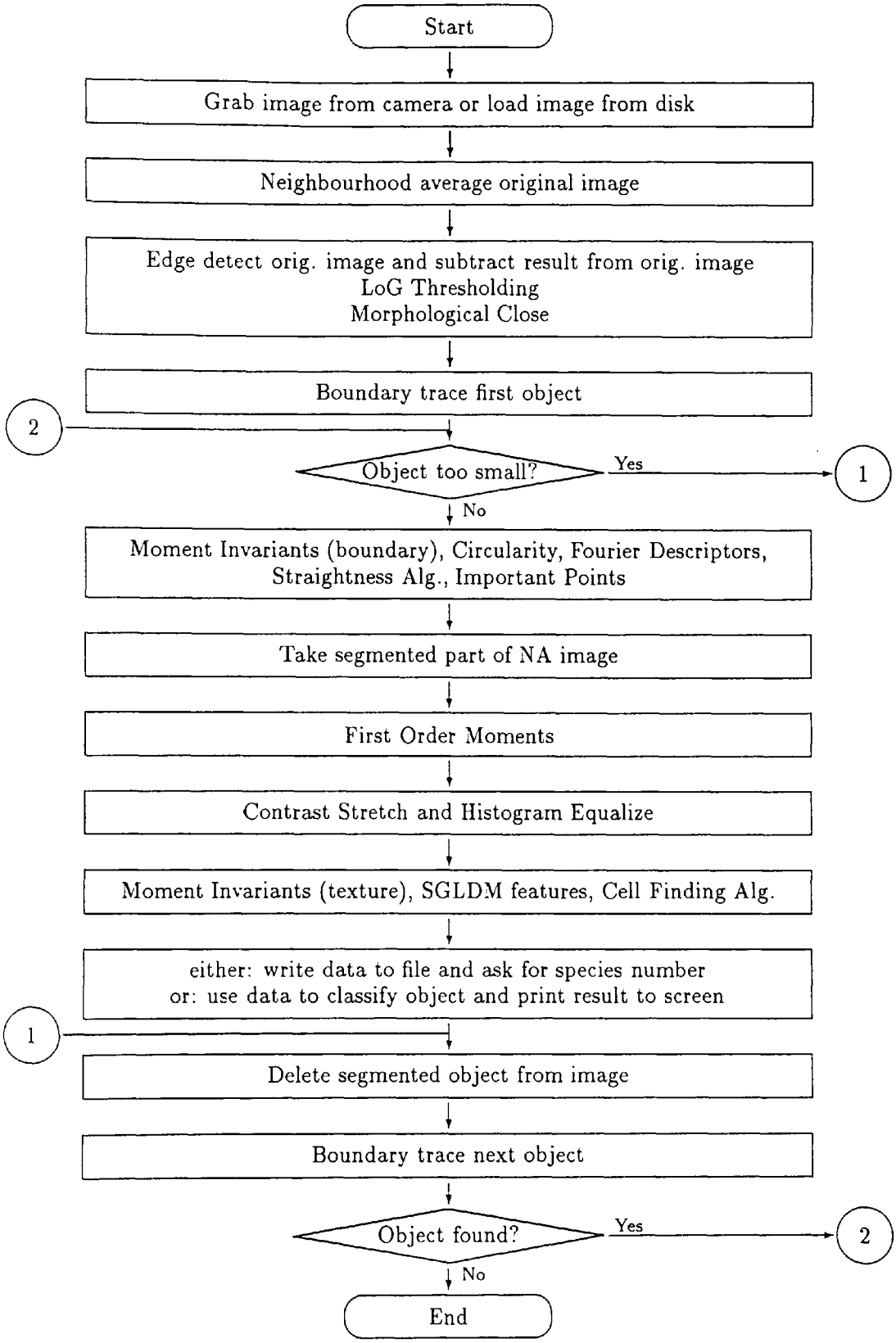


Figure B.1: The Program Structure

Bibliography

[A1] Belcher, H. and Swale, E.
A Beginners Guide to Freshwater Algae
Institute of Terrestrial Ecology, NERC, 1976

[A2] Bellinger, E.G.
A Key to Common British Algae
Institution of Water and Environmental Management, 1992

[A3] Boney, A.D.
Phytoplankton
New Studies in Biology, Edward Arnold, 1989

[A4] Carr, L.G. and Whitton, B.A.
The Biology of Cyanobacteria
Blackwell Scientific Publications, 1982

[A5] Fay, P.
The Blue-Greens
Studies in Biology No. 160, Edward Arnold, 1983

[A6] Fogg, G.E., Stewart, W.D.P., Fay, P. and Walsby, A.E.
The Blue-Green Algae
Academic Press, 1973

[A7] Harding, J.P.C. and Hawley, G.R.W.
Use of Algae for Monitoring Rivers in the United Kingdom
In: *Use of Algae for Monitoring Rivers*, Institut für Botanik, Universität Innsbruck, 1991

[A8] Hynes, H.B.N.
The Biology of Polluted Water
Liverpool University Press, 1960

[A9] Mason, C.F.
Biology of Freshwater Pollution
John Wiley & Sons, Second Edition, 1991

[A10] Morris, I.
An Introduction to the Algae
Hutchinson, 1967

[A11] National Rivers Authority
Toxic Blue-Green Algae
National Rivers Authority, 1990

[A12] Prescott, G.W.
How to Know the Freshwater Algae
Wm. C. Brown Company Publishers, 1970

[A13] Smith, G.M.
Freshwater Algae of the US
McGraw-Hill, 1950

[A14] VanLandingham, S.L.
Guide to the Identification, Environmental Requirements and Pollution Tolerance of Freshwater Blue-Green Algae
S.L. VanLandingham, Cincinnati, Oh., 1982

[A15] Whitton, B.A.
Aims of Monitoring
In: *Use of Algae for Monitoring Rivers*, Institut für Botanik, Universität Innsbruck, 1991

[A16] Whitton, B.A.
Diversity, Ecology, and Taxonomy of the Cyanobacteria
In: *Photosynthetic Prokaryotes*, Plenum Press, 1992

[1] Ahuya, N. and Rosenfeld, A.
Mosaic Models for Textures
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 1, pp. 1-11, 1981

[2] Arthurs, M.J.
An Investigation into the Current Edge Detection Algorithms in Image Processing
Final Year Project, Unpublished, Polytechnic of Wales, 1991

[3] Atiquzzaman, M.
Multiresolution Hough Transform–An Efficient Method of Detecting Patterns in an Image
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 11, pp. 1090-1095, 1992

[4] Ballard, D.H.
Generalizing the Hough Transform to Detect Arbitrary Shapes
Pattern Recognition, Vol. 13, No. 2, pp. 111-122, 1981

[5] Ballard, D.H. and Brown, C.M.
Computer Vision
Prentice Hall, 1982

[6] Berzins, V.
Accuracy of Laplacian Edge Detector
Computer Vision, Graphics, and Image Processing, Vol. 27, pp. 195-210, 1984

[7] Boddy, L. and Morris, C.W.
Analysis of Flow Cytometry Data - A Neural Network Approach
Binary, Vol. 5, pp. 17-22, 1993

[8] Bouman, C. and Liu, B.
Multiple Resolution Segmentation of Textured Images
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 13, No. 2, pp. 99-113, 1991

[9] Bovik, A.C., Kim, N.H., Aggarwal, S.J. and Diller K.R.
Automatic Measurement of Cellular Area Using A PC Based Vision System
In: *Software and Hardware Applications of Microcomputers*, Acta Press, pp. 151-155, 1986

[10] Carling, A.
Introducing Neural Networks
John Wiley & Sons, 1992

[11] Chen, J.S., Huertas, A. and Medoni, G.
Fast Convolution With Laplacian-of-Gaussian Masks
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, pp. 584-590, 1987

[12] Chow, C.K. and Kaneko, T.
Automatic Boundary Detection of the Left Ventricle from Cineangiograms
Computer and Biomedical Research, Vol. 5, pp. 388-410, 1972

[13] Conners, R.W. and Harlow, C.A.
A Theoretical Comparison of Texture Algorithms
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 3, pp. 204-222, 1980

[14] Courtney, P., Beck, M.S. and Martin, W.J.
A Vision Guided Life-Science Laboratory Robot
Measurement Science and Technology, Vol. 2, No. 2, pp. 97-101, 1991

[15] Dapeng, Z. and Zhongrong, L.
Digital Image Texture Analysis Using Grey Level Energy Cooccurrence

Proc. of the Spie, Applications of Artificial Intelligence, Vol. 657, pp. 152-156, 1986

[16] Dayhoff, R.E. and Dayhoff, J.E.
Neural Networks for Medical Image Processing; A Study of Feature Identification
12th Annual Symp. on Computer Applications in Medical Care, pp. 271-275, 1988

[17] Duda, R.O.
Elements of Pattern Recognition
In: *Adaptive, Learning and Pattern Recognition Systems*, Mendel, J.M. and Fu, K.S, Academic Press, 1970

[18] Duda, R.O. and Hart, P.E.
Use of the Hough Transformation to Detect Lines and Curves in Pictures
Communications of the ACM, Vol. 15, pp. 11-15, 1972

[19] Duda, R.O. and Hart, P.E.
Pattern Classification and Scene Analysis
John Wiley & Sons, 1973

[20] Dudani, S.A., Breeding, K.J. and McGhee, R.B.
Aircraft Identification by Moment Invariants
IEEE Trans. on Computers, Vol. C-26, pp. 39-46, 1977

[21] Dwyer, S.J., Chang, J.K., McLaren, R.W. and Lodwick, G.S.
Learning Texture Information from Singular Photographs and its Application in Digital Classification
In: *Learning Systems and Intelligent Robots*, Plenum Press, pp. 423-435, 1974

[22] Edwards, C., Porter, J., Saunders, J.R., Diaper, J., Morgan, J.A.W. and Pickup, R.W.
Flow Cytometry and Microbiology
SGM Quarterly, Vol. 19, pp. 105-108, 1992

[23] Fisher, R.A.
The Use of Multiple Measurements in Taxonomic Problems
Annals of Eugenics, Vol. 7, pp. 179-188, 1936

[24] Fu, K.S., Gonzalez, R.C. and Lee, C.S.G.
Robotics: Control, Sensing, Vision, and Intelligence
Mc-Graw Hill, 1987

[25] Frankel, D.S., Olson, R.J., Frankel, S.L. and Chisholm, S.W.
Use of a Neural Net Computer System for Analysis of Flow Cytometric Data of Phytoplankton Populations
Cytometry. Vol 10, pp. 540-550, 1989

[26] Goetcherian, V.
From Binary to Grey Tone Image Processing Using Fuzzy Logic Concepts
Pattern Recognition, Vol. 12, pp. 7-12, 1980

[27] Gonzalez, R.C. and Woods, R.E.
Digital Image Processing
Addison-Wesley, 1992

[28] Granlund, G.H.
Fourier Preprocessing for Hand Print Character Recognition
IEEE Trans. on Computers, pp. 195-201, 1972

[29] Haralick, R.M., Shanmugam, K. and Dinstein, I.
Textural Features for Image Classification
IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-3, No. 6, pp. 610-621, 1973

[30] Haralick, R.M.
Statistical and Structural Approaches to Texture
Proceedings of the IEEE, Vol. 67, No. 5, pp. 786-804, 1979

[31] Haralick, R.M.
Image Texture Survey
in: *Handbook of Statistics, Vol. 2*, North Holland Pub. Co., pp. 399-415, 1982

[32] Haralick, R.M., Sternberg, S.R. and Zhuang, X.
Image Analysis Using Mathematical Morphology
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, pp. 532-550, 1987

[33] Harris, P.D. and Deskoy, A.
Analysis of Microvascular Systems Using Image Processing
Proc. of the IEEE, pp. 690-693, 1988

[34] He, D.C. and Wang, L.
Textural Features Based on Texture Spectrum
Pattern Recognition, Vol. 24, No. 5, pp. 391-399, 1991

[35] Hills, S.J.
Outline Extraction of Microfossils in Reflected Light Images
Computers and Geosciences, Vol. 14, No. 4, pp. 481-488, 1988

[36] Höhne, K-H., Bomans, M., Pommert, A., Riemer, M. and Tiede, U.
3D-Segmentation and Display of Tomographic Imagery
9th International Conference on Pattern Recognition, pp. 1271-1276, 1988

[37] Horn, B.K.P.
Robot Vision
The MIT Press, McGraw-Hill, 1986

[38] Hu, M.K.
Visual Pattern Recognition by Moment Invariants
IRE Trans. on Information Theory, Vol. IT-8, pp. 179-187, 1962

[39] Illing, D.
Orientation and Recognition of Both Noisy and Partially Occluded 3-D Objects from Single 2-D Images
PhD Thesis, Polytechnic of Wales, 1990

[40] Illingworth. J. and Kittler, J.
The Adaptive Hough Transform
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 5, pp. 690-698, 1987

[41] James, M.
Pattern Recognition
BSP Professional Books, 1987

[42] Jiang, T., Merickel, M.B. and Parrish, E.A.
Automated Threshold Detection Using a Pyramid Data Structure
9th International Conference on Pattern Recognition, pp. 689-692, 1988

[43] Katsinis, C., Poularikas, A.D. and Jeffries, H.P.
Image Processing and Pattern Recognition with Applications to Marine Biological Images
Proc. of SPIE, International Society for Optical Engineering, Vol. 504, pp. 324-329. 1984

[44] Katsinis, C. and Poularikas, A.D.
Analysis of a Sampling Technique Applied to Biological Images
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 6, pp. 832-835, 1987

[45] Kiryati, N and Bruckstein, A.M.
What's in a Set of Points
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 4, pp. 497-500. 1992

[46] Klecka, W.R.
Discriminant Analysis
Series: Quantum Applications in the Social Sciences, Sage Publications, Inc., 1980.

[47] Kohonen, T.
Self-Organization and Associative Memory
Second Edition, Springer Verlag, 1988

[48] Krahe, J.L. and Pousset, P.
The Detection of Parallel Straight Lines with the Application of the Hough Transform
9th International Conference on Pattern Recognition, pp. 939-941, 1988

[49] Lachenbruch, P.A. and Mickey, M.R.
Estimation of Error Rates in Discriminant Analysis
Technometrics, Vol. 10, No. 1, pp. 1-11, 1968

[50] Lachenbruch, P.A.
Discriminant Analysis
Hafner Press, 1975

[51] Lachenbruch, P.A. and Goldstein, M.
Discriminant Analysis
Biometrics, Vol. 35, pp. 69-85, 1979

[52] Laine, A. and Fan, J.
Texture Classification by Wavelet Packet Signatures
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1186-1191, 1993

[53] Low, A.
Introductory Computer Vision and Image Processing
McGraw-Hill, 1991

[54] Lu, S.Y. and Fu, K.S.
A Syntactic Approach to Texture Analysis
Computer Graphics and Image Processing, Vol. 7, pp. 303-330, 1978

[55] Lynn, P.A.
Introductory Digital Signal Processing with Computer Applications
John Wiley & Sons, 1989

[56] Ma, S.D. and Chen, X.
Hough Transform Using Slope and Curvature as Local Properties to Detect Arbitrary 2D Shapes
9th International Conference on Pattern Recognition, pp. 511-513, 1988

[57] Maitra, S.
Moment Invariants
Proc. of the IEEE, Vol. 67, No. 4, pp. 697-699, 1979

[58] Mardia, K.V. and Hainsworth, T.J.
A Spatial Thresholding Method for Image Segmentation
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, pp. 919-927, 1988

[59] Marr, D. and Hildreth, E.
Theory of Edge Detection
Proc. Royal Society London, Vol. B207, pp. 187-217, 1980

[60] Matheron, G.
Random Sets and Integral Geometry
John Wiley & Sons, 1975

[61] Morris, C.W., Lynne, B. and Allman, R.
Identification of Basidiomycete Spores by a Neural Network Analysis of Flow Cytometry Data
Mycological Research, Vol. 96, No. 8, pp. 697-701, 1992

[62] Murakami, K., Koshimizu, H. and Hasegawa, K.
An Algorithm to Extract Convex Hull on θ-ρ Hough Transform Space
9th International Conference on Pattern Recognition, pp. 500-503, 1988

[63] Nakagawa, Y. and Rosenfeld, A.
Some Experiments on Variable Thresholding
Pattern Recognition, Vol. 11, pp. 191-204, 1979

[64] Niblack, W.
An Introduction to Digital Image Processing
Prentice Hall, 1986

[65] Nishihara, H.K. and Crossley, P.A.
Measuring Photolithographic Overlay Accuracy and Critical Dimensions by Correlating Binarized Laplacian of Gaussian Convolution
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.10, No. 1, pp. 17-30, 1988

[66] Ohta, H., Miyauchi, M. and Takata, H.
Computer Classification of Rosette-Forming cells from Microscope Images
Systems and Computers in Japan, Vol. 18, No. 4, pp. 64-76, 1987

[67] *The Mariner 9 TV Experiment: A Case Study*
The Open University, Unit 12, 1982

[68] Pao, D.C.W., Li, H.F. and Jayakumar, R.
Shapes Recognition Using the Straight Line Hough Transform: Theory and Generalization
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 11, pp. 1076-1089, 1992

[69] Parthenis, K., Metaxaki-Kossionides, C. and Dimitriadis, B.
An Automatic Computer Vision System for Blood Analysis
Microprocessing and Microprogramming, Vol. 28, Part 1-5, pp. 243-246, 1990

[70] Pavlidis, T.
Structural Pattern Recognition
Springer Verlag, 1977

[71] Pavlidis, T.
Algorithms for Shape Analysis of Contours and Waveforms
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 4, pp. 301-312, 1980

[72] Pavlidis, T.
Graphics and Image Processing
Springer Verlag, 1982

[73] Pineda, F.J.
Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation
Neural Computation, Vol. 1, pp. 161-172, 1989

[74] Persoon, E. and Fu, K.
Shape Discrimination Using Fourier Descriptors
IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-7, No. 3, pp. 170-178. 1977

[75] Reeves, A.P., Prokop, R.J., Andrews, S.E. and Kuhl, F.P.
Three-dimensional Shape Analysis Using Moments and Fourier Descriptors
Proc. 7th Int. Conf. Pattern Recognition, Canada, pp. 447-450, 1984

[76] Ridler, T. and Calvard, S.
Picture Thresholding Using an Interactive Selection Method
IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-8, No. 8, pp. 630-632. 1978

[77] Rösler, R., Schneider, H.A. and Schuberth, R.
Relation Between Particle Shape and Profile Fourier Coefficients
Powder Technology, Vol. 49, pp. 255-260, 1987

[78] Rosenfeld, A. and Johnston, E.
Angle Detection on Digital Curves
IEEE Trans. on Computers, Vol. C-22, pp. 875-878, 1973

[79] Rosenfeld, A. and Weszka, J.S.
An Improved Method of Angle Detection on Digital Curves
IEEE Trans. on Computers, Vol. C-24, pp. 940-941, 1975

[80] Rosenfeld, A. and Kak, C.
Digital Picture Processing
Academic Press, Second Edition, 1982

[81] Sahoo, P.K., Soltani, S. and Wong, A.K.C.
A Survey of Thresholding Techniques
Computer Vision, Graphics, and Image Processing, Vol. 41, pp. 233-260. 1988

[82] Schachter, B.J., Rosenfeld, A. and Davis, L.S.
Random Mosaic Models for Textures
IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-8, No. 9, pp. 694-702, 1978

[83] Serra, J.
Image Analysis and Mathematical Morphology
Academic Press, 1982

[84] Siew, L.H., Hodgson, R.M. and Wood, E.J.
Texture Measures for Carpet Wear Assessment
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 10, No. 1, pp. 92-105, 1988

[85] Sklansky, J.
On the Hough Technique for Curve Detection
IEEE Trans. on Computers, Vol. C-27, No. 10, pp. 923-926, 1978

[86] Sotak, G.E. and Boyer, K.
Comments on "Fast Convolution With Laplacian-of-Gaussian Masks"
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 11, No. 12, pp. 1329-1332, 1989

[87] Spinrad, R.W. and Yentsch, C.M.
Observations on the Intra- and Interspecific Single Cell Optical Variability of Marine Phytoplancton
Applied Optics, Vol. 26, No. 2, pp. 357-362, 1987

[88] Taxt, T. and Flynn, P.J.
Segmentation of Document Images
IEEE International Conference on Systems, Man, and Cybernetics, Vol. 3, pp. 1062-1067, 1989

[89] Tazelaar, J.M.
Neural Networks
BYTE, August, p. 214, 1989

[90] Teague, M.R.
Image Analysis via the General Theory of Moments
J. Opt. Soc. Am., Vol. 70, No. 8, pp. 920-930, 1980

[91] Teh, C. and Chin, R.T.
On the Detection of Dominant Points on Digital Curves
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 11, No. 8, pp. 859-872, 1989

[92] Therrien, C.W.
Decision Estimation and Classification
John Wiley & Sons, 1989

[93] Thomas, M.C.
Shape and Texture in Quartz Sand Grains: A Quantitative Assessment
PhD Thesis, Polytechnic of Wales, 1992

[94] Thiel, S. and Wiltshire, R.J.
An Investigation in the Automated Detection of Blue-Green Algae
Water Pollution II, Modelling, Measurement and Prediction, Computational Mechanics Publications, pp. 461-468, 1993

[95] Thiel, S. and Wiltshire, R.J.
The Detection of Blue-Green Algae for Measuring Water Quality using Image Processing Techniques
Inland and Coastal Water Quality '93, Measurement and Modelling, Stevenage, 1993 (Conference proceedings to appear in Environment International, 1995).

[96] Thiel, S., Wiltshire, R.J. and Davies, L.J.
Automated Object Recognition of Blue-Green Algae for Measuring Water Quality
submitted to Water Research, July 1994

[97] Tomaru, M., Kumaki, Y., Minamitani, H. and Horikoshi, T.
Feature Extraction and Classification of Cell Morphology in Response to the Cell Cycle
Systems and Computers in Japan, Vol. 20, No. 3, pp. 20-31, 1989

[98] Tomita, F., Shirai, Y. and Tsuji, S.
Description of Textures by a Structural Analysis
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 2, pp. 183-191, 1982

[99] Turney, J.L., Mudge, T.N. and Volz, R.A.
Recognizing Partially Occluded Parts
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 4, pp. 410-421, 1985

[100] Vickers, A.L. and Modestino, J.W.
A Maximum Likelihood Approach to Texture Classification
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, No. 1, pp. 61-68, 1982

[101] Wang, R., Hanson, A.R. and Riseman, E.M.
Fast Extraction of Ellipses
9th International Conference on Pattern Recognition, pp. 508-510, 1988

[102] Wallace, T.P. and Wintz, P.A.
An Efficient Three-Dimensional Aircraft Recognition Algorithm Using Normalized Fourier Descriptors
Computer Graphics and Image Processing, Vol. 13, pp. 99-126, 1980

[103] Werman, M. and Peleg, S.
Min-Max Operators in Texture Analysis
IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 6, pp. 730-733, 1985

[104] Weszka, J.S., Dyer, C.R. and Rosenfeld, A.
A Comparative Study of Texture Measures for Terrain Classification
IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-6, No. 4, pp. 269-285, 1976

[105] Weszka, J.S.
A Survey of Thresholding Techniques
Computer Graphics and Image Processing, Vol. 7, pp. 259-265, 1978

[106] Wiejak, J.S., Buxton, H. and Buxton, B.F.
Convolution with Separable Masks for Early Image Processing
Computer Vision, Graphics and Image Processing, Vol. 32, pp. 279-290, 1985

[107] Wilkins, M.F., Boddy, L. and Morris, C.W.
Kohonen Maps and Learning Vector Quantization Neural Networks for Analysis of Multivariate Biological Data
Binary, Vol. 6, pp. 64-72, 1994

[108] Zahn, C.T. and Roskies, R.Z.
Fourier Descriptors for Plane Closed Curves
IEEE Trans. on Computers, Vol. 21, No. 3, pp. 269-281, 1972