Probabilistic Segmentation of Remotely Sensed Images

Ben Gorte

Publication
Number 63

Promotor:	Dr. Ir. M. Molenaar Professor of Spatial Information Production from Photogrammetry and Remote Sensing
Co-promotor:	Dr. Ir. A. Stein Visiting Professor of Spatial Statistics International Institute for Aerospace Survey and Earth Sciences (ITC), Enschede

Probabilistic Segmentation of Remotely Sensed Images

Ben Gorte

Proefschrift
ter verkrijging van de graad van doctor
op gezag van de rector magnificus
van de Landbouwuniversiteit Wageningen,
Dr. C.M. Karssen,
in het openbaar te verdedigen
op maandag 12 oktober 1998
des namiddags te vier uur in de Aula.

ITC Publication Series
No. 63
The research presented in this thesis was conducted at the
International Institute for Aerospace Survey and Earth Sciences (ITC)
P.O.Box 6, 7500 AA Enschede, the Netherlands.

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

ISBN 90 6164 157 8

Probabilistic Segmentation of Remotely Sensed Images
Thesis Wageningen Agricultural University and ITC,
with ref. Summary in Dutch.

NN08201, 24 99

Propositions

1. Class areas that are estimated as sums of posterior probabilities are more accurate than those obtained by counting pixels after classification.
   This thesis

2. Avoiding spectral overlap during training data selection is hiding the head in the sand. Avoiding spectral overlap during test data selection is cheating.
   This thesis

3. Often, satellite image classification is considered less valuable when more knowledge about the area is already available. Apparently it is difficult to exploit this knowledge for classification improvement.
   This thesis

4. Accurate estimation of probability densities in Bayesian classification is only helpful if prior probabilities are accurately estimated as well.
   This thesis, Table 3.11.

5. Without considering the relation between mapping scale and thematic class definition, classification evaluation becomes unacceptably subjective.
   This thesis

6. The development of segmentation algorithms progresses slowly, because the matter is too complex to be fully understood in the course of one Ph.D. project [Pavlidis, 1986].

7. Quadtree data structures allow to implement the raster data model with vector accuracy.
   This thesis

8. The linguistic advantage of *maximum likelihood* over *k-nearest neighbor* is difficult to compensate with statistical arguments.

9. Certain hobby's, such as knitting and riding motorcycles, are partly based on the charm of being able to master inadequate technology. This principle often helps to explain the popularity of software.

10. Increasingly certain *skills* are considered, in addition to education, when judging an employee's professional quality. This does not imply that education institutes should try to teach those skills.

11. Advertisers pay commercial TV-stations to obtain, through attractive programs, a large audience for their commercials. Too much advertisement on a publically sponsored network makes the viewer think that he supports the advertisers.

12. According to probabilistics, some things that can go wrong might not. What cannot go wrong, will not. Statistical views allow for more optimism than deterministic ones.

Propositions related to the dissertation
Probabilistic Segmentation of Remotely Sensed Images
Ben Gorte, Wageningen, October 12, 1998.

Probabilistic Segmentation of Remotely Sensed Images

Ben Gorte

Thesis
to fulfill the requirements for the degree of doctor
on the authority of the rector magnificus
of Wageningen Agricultural University,
Dr. C.M. Karssen,
to be publicly defended
in the Auditorium on Monday, 12 October 1998, at 16:00 hours.

Acknowledgements

The research described in this thesis was executed under the guidance and supervision of prof. Martien Molenaar, to whom I want to express my sincere gratitude. He greatly helped me to put my story in the Geoinformatics context. I am also very grateful to prof. Alfred Stein, my co-promotor, who supported me in many different ways, and much more than I could ever expect.

Many other people helped me to perform the research and complete the thesis by discussing ideas, conducting experiments, co-authoring papers, relieving me of other duties, showing confidence and providing technical, administrative and moral support. Aneeqa Syed, Anneke Homan, Arun Shresta, Benjamin Aquette, Frank, Frans van de Wel, Gerard Reinink, Gerrit Huurneman, Gueye Lat, Krishna Talukdar, Linda van der Gaag, Maartje, Nanno Mulder, Natasha Kroupnova, Rolf de By, Roshanak Darvishzadeh, Sisi, Wan Bakx, Wim Bakker and dear Father and Mother, thank you very much!

Abstract

For information extraction from image data to create or update geographic information systems, objects are identified and labeled using an integration of segmentation and classification. This yields geometric and thematic information, respectively.

Bayesian image classifiers calculate class posterior probabilities on the basis of estimated class probability densities and prior probabilities. This thesis presents refined probability estimates, which are local, *i.e* pertain to image regions, rather than to the entire image. Local class probability densities are estimated in a non-parametric way with an extended k-Nearest Neighbor method. Iterative estimation of class mixing proportions in arbitrary image regions yields local prior probabilities.

The improved estimates of prior probabilities and probability densities increase the reliability of posterior probabilities and enhance subsequent decision making, such as maximum posterior probability class selection. Moreover, class areas are estimated more accurately, compared to standard Maximum Likelihood classification.

Two sources of image regionalization are distinguished. Ancillary data in geographic information systems often divide the image area into regions with different class mixing proportions, in which probabilities are estimated. Otherwise, a regionalization can be obtained by image segmentation. A region based method is presented, being a generalization of connected component labeling in the quadtree domain. It recursively merges leaves in a quadtree representation of a multi-spectral image into segments with arbitrary shapes and sizes. Order dependency is avoided by applying the procedure iteratively with slowly relaxing homogeneity criteria.

Region fragmentation and region merging, caused by spectral variation within objects and spectral similarity between adjacent objects, are avoided by regarding class homogeneity in addition to spectral homogeneity. As expected, most terrain objects correspond to image segments. These, however, reside at different levels in a segmentation pyramid. Therefore, class mixing proportions are estimated in all segments of such a pyramid to distinguish between pure and mixed ones. Pure segments are selected at the highest possible level, which may vary over the image. They form a non-overlapping set of labeled objects without fragmentation or merging. In image areas where classes cannot be separated, because of spatial or spectral

resolution limitations, mixed segments are selected from the pyramid. They form uncertain objects, to which a mixture of classes with known proportion is assigned.

Subsequently, remotely sensed data are used for taking decisions in geographical information systems. These decisions are usually based on crisp classifications and, therefore, influenced by classification errors and uncertainties. Moreover, when processing spatial data for decision making, the objectives and preferences of the decision maker are crucial to deal with. This thesis proposes to exploit mathematical decision analysis for integrating uncertainties and preferences, on the basis of carefully estimated probabilistic class information. It aims to solve complex decision problems on the basis of remotely sensed data.

Contents

List of symbols

The following list contains symbols that are used throughout the thesis, or at least within a number of successive sections. Symbols, that are only used in derivations are defined locally.

A	:	area, measured as a number of pixels
$A(I), A(s)$	:	a function returning the area of an image I or of a subset $s \subset I$
$C_i, i \in [1..N]$	:	classes in a supervised classification, *i.e.* the probabilistic event that the true class of a pixel is C_i
C_0	:	The *unknown* class, the probabilitic event that a pixel should not be classified as belonging to any class $C_i (i \in [1..N])$
D	:	$2 \times A$ matrix of class probability densities d_{ij} for class C_i $(i \in [1,2])$ at pixel j $(j \in [1..A])$
$G \subset \mathcal{N}^2$	:	grid space with rows $r \in [1..r_{\max}]$ and columns $c \in [1..c_{\max}]$
$I : G \rightarrow \mathbf{X}$	:	an image, which maps grid cell coordinates $(r, c) \in G$ into feature vectors $\mathbf{x} \in \mathbf{X}$
L	:	$2 \times A$ matrix of class *a posteriori* probabilities d_{ij} for class C_i $(i \in [1,2])$ at pixel j $(j \in [1..A])$
M	:	the number of bands of a multi-spectral image = the number of features in a classification = the dimensionality of the feature space
N	:	the number of classes $(C_1, \ldots, C_N)$ in a classification, excluding the eventual *unknown* class C_0
$\mathrm{P}(C_i)$	:	the probability that a pixel belongs to class C_i, irrespective of the feature (prior probability)
$\mathrm{P}(\mathbf{x}\|C_i)$	:	the probability that a class C_i pixel has feature vector $\mathbf{x}$ (class probability density)
$\mathrm{P}(C_i\|\mathbf{x})$	:	the probability that a pixel with feature vector $\mathbf{x}$ belongs to class C_i

$\mathrm{P}(\mathbf{x})$	:	the probability that a pixel has feature vector $\mathbf{x}$
$\mathrm{P}(C_i\|\mathbf{x})\rfloor_s$	:	$\mathrm{P}(C_i\|\mathbf{x})$ for a pixel in a subset $s \subset I$
$p = (r, c) \in G$	:	grid coordinate of a pixel
$S = \{s_j\}$	:	subdivision (stratification, segmentation) of I. $I = \bigcup s_j$ (complete) or $I \supset \bigcup s_j$ (incomplete)
$\#S$	:	the number of elements of a set S, the number of segments in a segmentation S
$s, s_j \subset I$	:	a subset (stratum, segment, region) of image I
T	:	Training samples, a set $\{T_j\}$ of tuples $T_j =< j, \mathbf{x}, C_i >$ relating feature vectors to classes. The training sample number j is often omitted.
$T_i \subset T$	:	Training samples for class C_i
T_i^s, T_i^j	:	Training samples $\{< \mathbf{x}_p, C_i, n >\}$ for class C_i, sampled at pixels $p \in s$ or $p \in s_j$
$\mathbf{X}$	:	the M dimensional feature space
$\mathbf{x} \in \mathbf{X}$	:	a feature vector $(x_1, \ldots, x_M)$, for example reflections in M spectral bands
$\mathbf{x}_p$	:	feature vector of pixel at grid coordinate p

Chapter 1

Introduction

To monitor, analyze and interpret developments in our changing environment, up-to-date spatial data are periodically collected and processed. Increasingly, remote sensing is used as a valuable source for this purpose. It yields data that can be subjected to further analysis in a geographical information system (GIS) at advantageous average cost. By systematic application of spatial operations and visualization, a GIS is able to generate, on request, derivative data sets contributing to making decisions that involve characteristics of spatially-related phenomena of the environment.

Earth observation satellite sensors measure electro-magnetic radiation emitted or reflected by the earth surface. Measured radiation depends on local earth surface characteristics. The relationship between measurement values and land cover allows to extract terrain information from image data. An effective method is visual image interpretation. Human vision, combined with terrain knowledge and understanding of imaging processes, has unsurpassed image interpretation capabilities. Computer vision is steadily advancing and has reached production level maturity in applications concerning, for example, industrial control, medical diagnosis and hand-written text recognition. However, the earth surface, as recorded in remotely sensed imagery, is too complex to be analyzed automatically.

Nevertheless, during the last decades considerable research effort has been given to computer-assisted and (semi-)automatic interpretation of remotely sensed imagery of the earth surface. Image processing and image analysis have entered lecture rooms and geoinformation production organizations.

A distinction can be made between *digital image processing* and *information extraction*. The purpose of digital image processing is to obtain transformed images that are more suitable for subsequent analysis. For example,

Geometric transformations are used to make an image geometrically adhere to a cartographic projection, or to make different images of the same area

exactly coincide with each other.

Image enhancements support image interpretation by improving image contrast and sharpness, or by reducing image noise.

Color transformations identify different aspects of reflection, such as *hue*, *saturation* and *intensity*, and allow for example to separate terrain characteristics from illumination effects.

Feature extraction yields quantified terrain characteristics, such as relative vegetation cover.

Image transformations create data with an 'image' data structure. The values in the data structure represent estimates of quantified terrain characteristics on an ordinal scale.

Information extraction converts measurements of earth surface characteristics into a delineation of entities (objects) in the terrain with attached *attributes* describing their properties. A distinction can be made between *segmentation* and *classification*, where the former is mainly concerned with the spatial distributions of reflection measurements to obtain object delineation, and the latter with spectral characteristics to provide object characterization.

Classification

Classification of remotely sensed data into qualitative information classes is useful to extract information from the spectral attributes of these data, yielding an insightful representation of the real world. Such a representation can be exploited directly as a thematic map or as part of a time series in a change detection application.

Classification procedures apply statistical pattern recognition of image measurement vectors to label each pixel with an appropriate class from a set of information classes, concerning for example land-cover or land-use.

Statistical or probabilistic approaches to classification are motivated by the circumstance that the relation between information classes and measurement vectors is not one-to-one. Different measurements are observed for each class (grass can be green or yellow) and, more important during classification, similar values can be measured for different classes (grass and wheat can both be yellow). Consequently, the decision upon a class given a measurement vector has an element of uncertainty, which can be modeled probabilistically: For the variety of measurements within each class *probability densities* are estimated and *a posteriori* probabilities are used to decide upon a class, given a measurement vector.

When information about an area is available, the purpose of image classification is, for example, to update this information or to improve its quality by refining thematic class definitions or spatial accuracy. In such cases, accuracy and reliability of classification results should be at least comparable to those of existing information. For this reason, the advancement of GIS technology stimulates research

into pattern recognition and classification methods for remotely sensed imagery. Many improvements to standard maximum likelihood techniques have been proposed. Overviews have been published, for example [Argialas and Harlow, 1990] and [Janssen and van der Wel, 1994], and can also be found in digital image processing textbooks [Richards, 1993].

Segmentation

The purpose of image segmentation is to subdivide an image into different parts (segments) that correspond to objects in the terrain.

Whereas image classification, despite imperfections, has become a routinely applied method to analyze imagery, segmentation never became very popular in earth observation applications. Problems still associated with image segmentation are summarized by [Acton, 1996] as: object merging, poor object boundary localization, object boundary ambiguity, object fragmentation and sensitivity to noise. Moreover, segmentation algorithms typically require large computer memory capacities and long processing times. Finally, spatial resolutions of satellite date are considered the prime limiting factor for many applications. Satellite images are often used as substitutes for aerial photography, because of cheaper or faster data acquisition. However, users are reluctant to sacrifice spatial accuracy. It is common to squeeze the largest possible map scale out of the image, which in turn requires to map objects that are covered by only a few image pixels. In such cases, grouping pixels into segments does not significantly contribute to information extraction.

A new chance for image segmentation is justified by ever-increasing computer hardware performances and by the order of magnitude at which spatial resolutions of satellite imagery are expected in improve in the near future. Digital (soft-copy) photogrammetry, using scanned aerial photography, as well as advancements in development of airborne sensors and digital cameras will also increase availability of high resolution digital image data. Generally, the increasing area covered by digital images, combined with the increasing data volume per unit area, motivates further research in automated information extraction.

Integration

To combine the complementary information from segmentation and classification is not straightforward. The results are usually incompatible, because both procedures take decisions independently, using different data characteristics.

This thesis aims at integrating segmentation and classification, by first gathering evidence concerning spatial and class-membership characteristics, and then combining these to delineate and identify relevant terrain objects. During the combination, user requirements play a role. The relevance of objects depends on, for example,

the application domain, the project scale (in relation to the image resolution) and the available knowledge about the terrain. Therefore, a flexible integration method is needed.

Structure of the thesis

Following this introduction, Chapter 2 is about image classification. It reviews common classification methods and puts these in a Bayesian framework. The advantage of non-parametric methods in certain circumstances motivates a further exploration, leading to an estimate of mixing proportions of a set of classes that includes the *unknown* class.

Chapter 3 introduces local statistics in non-parametric Bayesian classification, to exploit available data and terrain knowledge for the improvement of classification results. The main idea is that refined statistics can be obtained if the image area is subdived according to units in ancillary maps in a Geografic Information System (GIS). Since the probability densities of measurements in a class may vary over these units, em local probability densities are introduced and an algorithm for their estimation is presented. Furthermore, a method is described to obtain class area information inside each GIS map unit. This information by itself may be useful for many applications, but also enables further classification improvement after normalizing the area estimates into *a priori* class probabilities.

Spatial image characteristics are investigated in Chapter 4, which describes a novel region-based image segmentation method. The resulting algorithm is a generalized version of a connected component labeling in quadtree GIS. Segments are created for spectrally homogeneous regions with arbitrary shapes and sizes in a multi-spectral image. Later, the algorithm is expanded to produce a segmentation pyramid, rather than a single segmentation.

In chapter 5, classification and segmentation are combined into segmentation pyramid classification to recognize terrain objects from image data. Basically, the proposed method estimates probability densities and prior probabilities in each segment of an entire segmentation pyramid, and selects segments according to class homogeneity from different pyramid levels until complete coverage of the area is obtained. Because selection is expressed in relational database queries, it can be easily tailored to user requirements.

Chapter 6 shows how statistical information that becomes available during classification can be directly applied in application-oriented decision analysis.

Conclusions and recommendations for further research are formulated in the final chapter.

Chapter 2

Classification

Classification is a common technique to extract information from remote sensing image data. It converts *measurements* of earth surface characteristics into *thematic maps* that suit user requirements.

After describing satellite imagery, emphasizing characteristics that are relevant for classification, this Chapter reviews popular classification methods in a Bayesian framework. A non-parametric method is further explored and developed towards a solution of the *unknown*-class problem.

2.1 Satellite imagery

Earth observation satellite sensors measure electro-magnetic radiation that is emitted or reflected by the earth surface. *Active* sensors send radiation pulses in the microwave range of the electro-magnetic spectrum to the earth and measure the returned amounts in successive time-intervals, according to radar principles. *Passive* sensors measure thermal infrared radiation which is emitted by the earth surface, or visible and near-infrared radiation which originates from the sun and is reflected by the earth surface (Figure 2.1).

2.1.1 Image radiometry

Multi-spectral sensors measure reflection in different parts of the electro-magnetic spectrum separately, but at the same time. The number of bands and their locations in the spectrum, expressed as wavelengths or wavelength intervals, specify the *spectral resolution* of a sensor.

Measurement principles vary according to sensor type, but, eventually after pre-preprocessing, each measurements corresponds to a location in the terrain and is

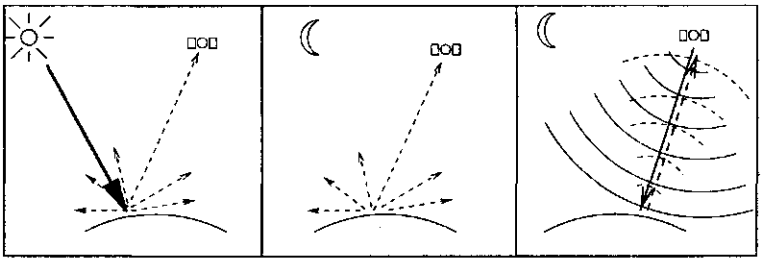

Figure 2.1: Sensor types

presented as a pixel in an image. A sensible geometrical correspondence exists between terrain locations and image pixels, such that the image can be regarded as a projection of the earth surface on an image plane. The distance between locations that correspond to adjacent image pixels is a sensor characteristic, which determines the *spatial resolution* of an image.

It is quite common to pretend that the terrain is subdivided in rectangles or squares, the *terrain elements*, whose size determines the spatial resolution, such that a pixel's measurement value is representative for the entire terrain element. However:

- The reflection may be not uniform within the terrain element, for instance at the boundary between two land cover types, or when details are smaller than the resolution.
- The satellite measures a reflection in an area called *instantaneous field of view (IFOV)*, which is rather circular than square. The measured value is a weighted average of the different reflections within that circular area, where the weight is larger in the center than towards the outside of the circle. Sensors are designed with such a spatial resolution that the circular IFOV's slightly overlap each other. This ensures that the terrain is entirely covered by measurements, without creating too much data redundancy.

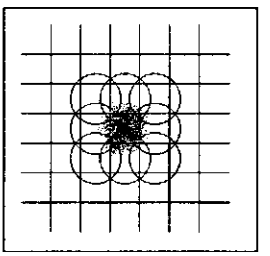

Figure 2.2: Terrain elements and instantaneous field of view

2.1.2 Image geometry

Due to earth curvature, earth rotation, orbit parameters, satellite movements, terrain relief etc., the projection of the earth surface into a satellite image is not a cartographic one. To make an image coincide pixel by pixel with a raster map in a Geographic Information System (GIS), such that pixels with the same row and column position correspond to the same location in the terrain, extensive geometrical corrections have to be performed. In this study, these corrections will be not elaborated upon, although they are a prerequisite for many of the operations to be described.

The examples and case studies in this thesis analyze imagery of multi-spectral SPOT (XS) and Landsat Thematic Mapper (TM) sensors. Both are passive, optical sensors that measure reflected sunlight in the visible and near-infrared regions of the electro-magnetic spectrum [1].

Their spatial and spectral resolutions are given in Table 2.1.

Satellite / Sensor	Spatial resolution		Spectral bands	Image area (km)
	east-west	north-south		
Spot pan	10-12	10	1	60x60
Spot XS	20-25	20	3	60x60
Landsat TM	30	30	6(+1)	185x185
Landsat MSS	57	76	4	185x185

Table 2.1: Resolutions of SPOT and Landsat sensors

2.1.3 Definition

An image I is a function $I : G \rightarrow \mathbf{X}$ that maps grid cells (coordinates) $(r, c) \in G$ into feature vectors $\mathbf{x} \in \mathbf{X}$. The vector space $\mathbf{X}$ is called the feature space.

A grid cell (r, c) is the element at row r and column c of the grid space $G = \{1, \ldots, r_{\max}\} \times \{1, \ldots, c_{\max}\}$, which is a subset of the discrete two-dimensional space $\mathbb{N}^2$.

A pixel $p_{r,c} \in I$ links a grid cell (r, c) with a feature vector $\mathbf{x}_p$, which is usually a vector of measurements that originates from image acquisition. In case of multi-spectral imagery, a feature vector $\mathbf{x}$ consists of reflection measurements $(x_1, \ldots, x_M)$ in M spectral bands. The notation $A = A(I) = r_{\max} \times c_{\max}$ will be used to denote the total number of pixels in image I.

When the positions of pixels in G are irrelevant, it may be convenient to denote pixels in (a subset s of) the image as $p_i, (i \in \{1, \ldots, A\})$ and the corresponding

1. The thermal infrared band of Landsat TM, containing measurements of emitted radiation with 120m resolution, is not used in this study.

feature vectors as $\mathbf{x}_i$. This yields a collection, rather than a set, since duplicates may occur.

2.2 Pattern recognition

This thesis is mainly concerned with supervised classification, which involves automatic recognition of patterns in spectral measurements. In the description by [Ripley, 1996], of a pattern recognition machine

> " ... we are given a set of N pre-determined classes, and assume (in theory) the existence of an oracle that would correctly label each example which might be presented to us. When we receive an example, some measurements are made, known as *features*, and these data are fed into the pattern recognition machine, known as the *classifier*. This is allowed to report
> 'this example is from class C_i'
> 'this example is from none of these classes' or
> 'this example is too hard for me'
> The second category are called *outliers* and the third *rejects* or *'doubt'* reports. Both can have great importance in applications."

Applied to satellite image classification, the presented examples are pixels in an image. Each pixel should be labeled as belonging to a class from a user-defined set of, for example, land-use of land-cover classes. The measurements are available as feature vectors and usually concern reflection in different spectral bands, although feature transformations may have been applied. Examples of so-called *derived features* are principal components, vegetation indices, hue-saturation-intensity and measures of texture.

The oracle in Ripley's description, which is capable to label every pixel correctly, is given by the terrain. After georeferencing the image, for each pixel the corresponding location on the earth surface (the terrain element), is known and the class label can be determined, for example by field inspection. However, there are a few difficulties. Class definition should, strictly speaking, include how to label a pixel, when its terrain element covers several objects of different classes. For example, the label of the largest object (within the terrain element) could be chosen, or the label of the object at center of the element. The reflection measurements, however, do not represent an exact square or a point, but rather a weighted average in a circular IFOV (Figure 2.2). Moreover, such definitions assume extremely precise georeferencing, whereas in practice a better geometric accuracy than (say) half the terrain element size is hard to achieve. Therefore, *mixed pixels* pose a problem. They are due to small or narrow (linear) objects, compared to the spatial resolution, but they also occur at boundaries between large objects. At best, 'doubt reports' are generated for those pixels — otherwise, labeling errors must be expected.

Other 'doubt reports' may be caused by spectral overlap between classes. *Reflectance* is a local earth surface characteristic, and can be considered a class property. Generally, a class cannot be associated with a single reflectance. For a certain crop species, for example, reflectance is influenced by crop variety, growing stage (even within a single image), cultivation, moisture, soil type etc. Moreover, the observations concern *reflections*, which are influenced by atmospheric conditions and illumination. The former may vary over an image, the latter almost certainly does, especially in hilly terrain. As a consequence, a variety of reflections is associated with each class and must be handled by the pattern recognition machine. Moreover, reflections of different classes should be different enough — whether this is the case depends on class definitions on one side and available data on the other. Often between-class differences are small compared to within-class variations, such that some class C_i pixels have the same reflection as some other class C_j pixels. It is difficult to judge image information content in general, independent of applications. Finally, it the user's responsibility to acquire image data that meet his requirements. Nevertheless, in most cases he will try to get as much information from his data as possible, and sometimes a little bit more. Then, the above-mentioned problem appears as *spectral overlap* between classes. The pattern recognition machine may generate errors, produce 'doubt reports', or accompany the selected label with a certainty indication. It is worthwhile to investigate which information can be extracted from which kind of imagery, and what is the quality of the extracted information.

The 'outliers' in Ripley's description present an interesting problem in satellite image classification. Usually, outliers are not caused by unfortunate coincidences in the measurement process, but by the presence of an *unknown* class in the terrain. For instance, when making a crop inventory reflection characteristics of various crops should be be determined accurately. But to characterize reflections of other classes in the image, such as villages, roads and forests, might involve much additional effort. For the purpose of the inventory, a classification into a set of classes that consists of the different crops plus a class *other* would be perfectly adequate. The outliers themselves are not a problem, but their detection is (section 2.4).

The above allows to formulate the goal of this study more precisely:

- to minimize the errors that are caused by spectral overlap
- to quantify the remaining classification uncertainty
- to used the uncertainty information in subsequent decision making
- to incorporate additional information to reach these goals:
  - from external sources (maps, expert knowledge)
  - from additional (*i.e.* spatial) image characteristics.

2.3 Supervised classification methods

Supervised classification labels a feature vector $\mathbf{x}$ with class C_i when $\mathbf{x}$ is more similar to the reflection characteristics of C_i than to those of other classes. Therefore, a measure of similarity between feature vectors and class reflection characteristics has to be established.

It would be ideal to have a universal database, describing class reflection characteristics that are valid for any image of a given satellite/sensor system. After following a standardized pre-processing procedure, the user would have to select a set of candidate classes for a particular image from the database, according to the theme of his interest. Subsequently, the classifier could compare each feature vector in the image with the reflection characteristics of candidate classes and assign a label after maximum similarity has been established.

However, the database needs to take spectral variability across images into account, caused, for example, by different atmospheric conditions and sun angles, and by seasonal influences (soil moisture, crop growing stage etc.). This causes excessive spectral overlap between different classes. Correction for these influence can be attempted. Some of them, such as atmospheric conditions and sun angle, can be accounted for quite reliably [Mulder, 1976], but others require large amounts of additional data, as well as extremely complicated reflection models.

Therefore, in practice class reflection characteristics are established for each image to be classified. Still, the reflection characteristics are influenced by, for example, differences in soil type and soil moisture, but within an image these differences are smaller than across images. Consequently, supervised classification involves a *training* stage, in which the user gives examples of each class to the classifier. This means that for a number of pixels in the image the class of the terrain elements must be known beforehand.

On the basis of the examples the classifier determines class reflection characteristics in such a form that the image feature vectors can be compared to them during the *decision* stage of supervised classification.

2.3.1 Training stage

Training a classifier for a particular image yields a set of <grid coordinate, class label> pairs. Since the image defines a feature vector at each grid coordinate and since the coordinates are irrelevant, this set can be transformed into a collection of tuples $< \mathbf{x}_p, C_i >$ having a feature vector and a class label as components. The training samples form a collection, rather than a set, since duplicates may occur, which are relevant for class characterization. To allow for set notation, a training sample number can be added to the tuples. Therefore, the elements t_j of the set

$T = \{t_j\}$ of training samples have the form

$$t_j = < j, \mathbf{x}_p, C_i > .$$

The training sample number will be omitted when this causes no confusion.

When $\mathbf{x}_p$ is located in a part of the feature space where classes C_i and C_j have spectral overlap, it may happen that some pixels with this feature vector are indicated as C_i and others as C_j. Then

$$t_j = < \mathbf{x}_p, C_i > \text{ and } t_k = < \mathbf{x}_p, C_j > \tag{2.1}$$

may both be members of T.

In this definition, training samples are taken inside the image to be classified. This is not strictly necessary. Training samples could be taken from another image, if circumstances like atmospheric conditions and sun angle were the same during acquisition of both images (for example with two adjacent images recorded during the same satellite orbit), or if the differences were corrected for.

Classification can benefit from availability of (digital) data, for example during the — usually expensive — task of finding sufficient numbers of training samples. As a rule, with M spectral bands between 10 M and 100 M samples are needed for each class to estimate parameters for class distributions in a maximum likelihood classification [Swain and Davis, 1978]. When using non-parametric methods to estimate class probability density, larger numbers of training samples are required, because in such a case the shape of the distribution must be estimated, not only the parameters.

In remote sensing applications, it is often difficult to find sufficient numbers of reliable training samples. Image interpretation easily leads to selecting obvious, spectrally distinct ones only, thereby reducing representativeness for the entire population.

Moreover, even after extensive fieldwork training samples may appear doubtful when the corresponding feature vectors in the image are examined. A good training sample selection should include deviating pixels due to within-class spectral variability, but exclude those that are caused by irregularities in the terrain. This involves precise class definition, which, in turn is also related to image resolution and map scale, when image date is going to be combined with map data. Crisp cover classes may become fuzzy for aggregated objects. During training sample selection, the question is when cover classes are crisp: Is a pixel with a clear space of sub-pixel size still a *forest* pixel? Is a pixel with a few trees still an *urban* pixel? If so, is the user prepared to accept the resulting spectral overlap? If not, can he accept that such pixels become *outliers*?

These choices, which are related to the necessity to provide the classifier with *representative samples* on one hand, while avoiding errors on the other hand, still hamper routinely application of supervised classification in production environments.

Other cases show a potential abundance of training samples. Many regions in the (developed) world were already mapped numerous times, and the purpose of classification is to update information. With an existing map and a limited amount of fieldwork it may be possible to identify large, unchanged areas for each class, from which training samples for classification of a new image can be selected at will. This allows to apply refined classification techniques. At the same time, application of such refined methods is necessary to prevent that classification gives lower quality than the existing information has.

2.3.2 Decision stage

In addition to the above-mentioned class definition issues, concerning cover classes being crisp or fuzzy, depending on aggregation level, one of the main problems in classification is spectral overlap between classes. It was already mentioned that two very similar feature vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, or even different occurrences of the same feature vector $\mathbf{x}_1$, may once get trained as class C_i and once as class C_j. Consequently, the classifier cannot decide "for sure", whether a feature vector $\mathbf{x}$ in the neighborhood of $\mathbf{x}_1$ and $\mathbf{x}_2$ (in the feature space) belongs to class C_i or to class C_j. For any given feature vector $\mathbf{x}$ a decision in favor of one of the classes will be made, and it will be applied to all occurrences of $\mathbf{x}$ in the entire image. It is very likely, however, that in a number of cases this decision is the wrong one, and the classifier cannot identify these cases.

Statistical classification methods attempt to take the best decisions in a statistical sense: those with the highest probability of being correct. Still, a single decision will be applied to all occurrences of a particular feature vector, still this will sometimes be wrong and still there is no way to tell when, but the point is exactly to maximize the number or correct decisions for any given feature vector.

Statistical classifiers apply a maximum *a posteriori probability* decision rule. The algorithms calculate for each C_i the *a posteriori* probability $\mathrm{P}(C_i|\mathbf{x}_p)$ that a pixel p with feature vector $\mathbf{x}_p$ belongs to class C_i, and select the class where this is maximum.

When all elements from the right hand side of Bayes formula

$$\mathrm{P}(C_i|\mathbf{x}_p) = \frac{\mathrm{P}(\mathbf{x}_p|C_i)\ \mathrm{P}(C_i)}{\mathrm{P}(\mathbf{x}_p)} \tag{2.2}$$

are known, correct *a posteriori* probabilities are obtained. The maximum *a posteriori* probability classification yields the optimal result, providing an upper limit for the

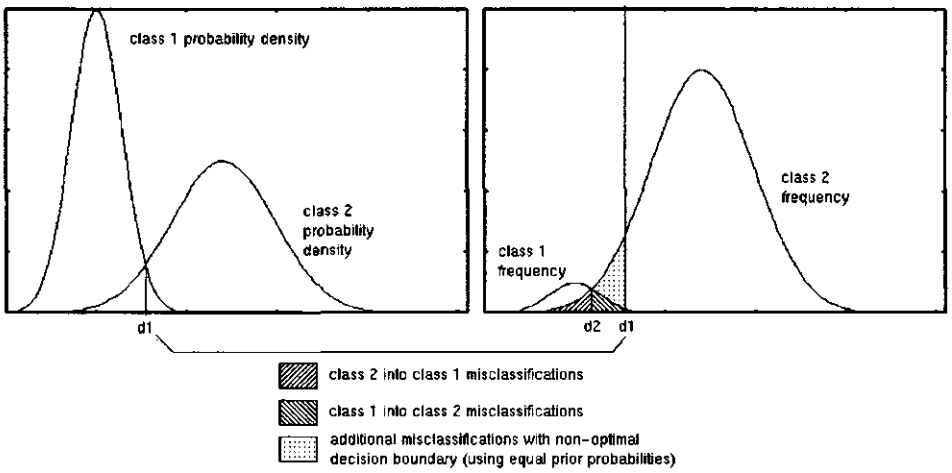

Figure 2.3: Decision boundaries between two normally distributed classes, using class probability densities $P(\mathbf{x}|C_i)$ with equal prior probabilities, (d1) and actual class frequencies $P(C_i|\mathbf{x})$, which can be obtained by applying prior probabilities (d2).

overall classification accuracy: No other classifier can be expected to give a better result [Ripley, 1996], if all misclassifications are considered equally unfavorable[2].

In practice, the elements at the right hand side of (2.2) are not known, but have to be estimated. Class probability densities $P(\mathbf{x}_p|C_i)$ are estimated on the basis of training samples, and classification algorithms differ in the way this is accomplished. In addition, some methods allow the user to specify class *a priori probabilities* $P(C_i)$ as the proportions of the different classes in the total area. Correct estimation of *a priori* probabilities, combined with correct probability densities, maximizes the expected overall classification accuracy (Fig. 2.3).

The unconditional feature probability density $P(\mathbf{x}_p)$ is often neglected, since it is equal for all classes at a given $\mathbf{x}_p$, and therefore does not influence the decision. Consequently, no "proper" values for $P(C_i|\mathbf{x}_p)$ are calculated. This is of no concern if only a labeled map is output and probability values are not presented to the user, as it is the case in commonly used image processing software. If the resulting (maximum) probabilities are to be output as well, normalization has to take place, which substitutes

2. If this is not the case, that is when an $M \times M$ matrix associates different *costs* c_{ij} to misclassifications of class C_i into C_j (with $c_{ii} = 0$), the theory expands into *minimum cost classification*, where the decision minimizes the expected cost $K_i = \sum_{j=1}^{N} P(C_i|\mathbf{x}_p)\, c_{ij}$. Chapter 6 describes *utility* based decision making, which covers this situation.

$$P(\mathbf{x}_p) = \sum_{j=1}^{M} P(\mathbf{x}_p|C_j)\ P(C_j) \tag{2.3}$$

in eq. (2.2).

Estimating $P(\mathbf{x}_p)$ independently from $P(\mathbf{x}_p|C_i)$ yields *a posteriori* probabilities without normalization, such that their sum may be less than 1. Then, under the assumption that all classes together partition the space [Molenaar, 1998], the probability that a pixel belongs to an *unknown* class can be estimated as

$$P(C_0|\mathbf{x}_p) = 1 \ - \ \sum_{i=1}^{M} P(C_i|\mathbf{x}_p).$$

This enables identification of *outliers* in the terminology of [Ripley, 1996], as will be demonstrated in section 2.4.

2.3.3 Review of classification algorithms

Besides statistical classification methods, there are classifiers whose decision mechanisms algorithms are usually described non-statistically. The following section is inspired by [Duda and Hart, 1973] and places representatives of both categories in a Bayesian framework.

Maximum Likelihood

In most satellite image classifications, the class probability density $P(\mathbf{x}_p|C_i)$ is modeled by a multivariate normal (Gaussian) distribution function

$$P(\mathbf{x}_p|C_i) = (2\pi)^{-M/2}|V_i|^{-\frac{1}{2}}e^{-\frac{1}{2}(\mathbf{y}^T V_i^{-1}\mathbf{y})} \tag{2.4}$$

with:

M : the number of features
V_i : the $M \times M$ variance-covariance matrix of class C_i, with elements v_{jk}^i
$|V_i|$: the determinant of V_i
V_i^{-1} : the inverse of V_i
$\mathbf{y} : \mathbf{x}_p - \mathbf{m}_i$ ($\mathbf{m}_i$ is the class mean vector), as a column vector with M components
$\mathbf{y}^T$: the transposed of $\mathbf{y}$ (a row vector).

For each class C_i, the training samples $T_i = \{< \mathbf{x}, C_i >\} \in T$ are analyzed. The feature vectors $\mathbf{x}$ in T_i are put into a sequence. The average of the vectors in this sequence yields the mean vector $\mathbf{m}_i$. Looking at the j-th component of each feature vector, the variance v_{jj}^i can be obtained. With $j \in [1..M]$, this gives the diagonal

elements of V_i. Similarly, between the j-th and k-th components of all feature vectors in the training samples of C_i, the covariance $v_{jk}^i, (j,k \in [1..M], j \neq k)$ can be determined, to yield the off-diagonal elements of V_i.

Strictly speaking, the *values* of $\mathbf{m}_i$ and V_i, as calculated from the training samples, are used as estimates for the *parameters* μ_i and Σ_i of a multi-variate normal distribution $N(\mu_i, \Sigma_i)$ for the entire C_i population. This distinction opens a variety of statistical methods for *parameter estimation*, which is ignored in this study. A major problem in maximum likelihood is that the Gaussian distribution model is not always suitable, and that subjectivity is involved in training sample selection — both problems are related to class definition schemes. Only little improvement can be expected from advanced parameter estimation methods.

The algorithms implement a *decision function*, by attributing to each $\mathbf{x}_p$ the class with the maximum $\mathrm{P}(C_i|\mathbf{x}_p)$.

In the remote sensing community, it is common to use the term *maximum likelihood* (or *full maximum likelihood*) for classifiers which:

1. use Bayes formula to calculate *a posteriori* probabilities
2. normalize these (eq. 2.3)
3. assume Gaussian class probability densities

According to definitions in statistics, the second and third property are not necessary for maximum likelihood. Independent estimation of unconditional feature densities $\mathrm{P}(\mathbf{x}_p)$ enables to bring the class *unknown* into the maximum likelihood framework. Moreover, estimators exist for parameters of other distribution functions, as well as direct (non-parametric) class probability density estimators, after which maximum likelihood can still be applied.

Maximum probability density

A simplification is obtained by disregarding the *class prior probabilities* $\mathrm{P}(C_i)$, thereby reducing the decision rule to *maximum probability density*. Overall classification accuracy is expected to decrease, compared to maximum likelihood (Figure 2.3). On the other hand, the result does not suffer from the bias that maximum likelihood tends to have in favor of classes with large prior probabilities [Conese and Maselli, 1992].

Consider the task to find roads in a forested area, where the proportion of forest pixels (say, 99 %) is used as prior probability for the class *forest*, leaving 1 % for *road*. In the result, *forest* may be assigned to all image pixels, giving the user a useless map with an overall accuracy of 99 %. If priors are not considered, the user can expect some forest to become classified as road, but at least most of the road will be found, too, The overall accuracy is expected to be lower, for example, 97 %. The user has to decide which of the classified *road* pixels are really road, but this is probably easy when looking at the spatial arrangement of these pixels.

Minimum Mahalanobis distance

For an efficient implementation, it is allowed to omit the constant factor $(2\pi)^{-M/2}$ from (2.4) and to take -2 times the logarithm of the remaining part. This yields for each class a value

$$D_i(\mathbf{x}_p) = \ln|V_i| + \mathbf{y}^{\mathrm{T}} V_i^{-1} \mathbf{y}. \tag{2.5}$$

For each class, $\ln|V_i|$ and V_i^{-1} are calculated at the beginning of the classification, and no time-consuming exponentiations are necessary at each pixel. The decision function selects the class with the minimum D_i value.

Class prior probabilities $\mathrm{P}(C_i)$ can be included by minimizing over

$$D_i'(\mathbf{x}_p) = D_i(\mathbf{x}_p) - 2\ln(\mathrm{P}(C_i)) \tag{2.6}$$

The decisions are the same as from maximum likelihood. Actual *a posteriori* probability values can be obtained, applying

$$\mathrm{P}(C_i|\mathbf{x}_p) \sim e^{-\frac{1}{2}D_i'(\mathbf{x}_p)}, \tag{2.7}$$

followed by normalization (but the efficiency gain is lost).

A further reduction omits $\ln|V_i|$ and minimizes over the squared *Mahalanobis distances* M_i:

$$M_i^2(\mathbf{x}_p) = \mathbf{y}^{\mathrm{T}} V_i^{-1} \mathbf{y} \tag{2.8}$$

between a feature vector $\mathbf{x}_p$ and a class mean vector $\mathbf{m}_i$.

Sometimes, V_i is replaced by V, the matrix of covariances between the M features over the entire image. Then, the differences between within-class variabilities are neglected, but feature space anisotropy is still considered.

Minimum Distance

If also V_i is omitted from the calculation (set to the $M \times M$ unity matrix), the decision is based on the minimum squared Euclidean distance E_i:

$$E_i^2(\mathbf{x}_p) = \mathbf{y}^{\mathrm{T}} \mathbf{y}. \tag{2.9}$$

Compared to the Euclidean distance $E_i(\mathbf{x}_p)$, the Mahalanobis distance $M_i(\mathbf{x}_p)$ between a feature vector $\mathbf{x}_p$ and a class mean vector $\mathbf{m}_i$ is weighed by the inverse of the variance-covariance matrix V_i. Therefore, wide-spread classes seem 'nearer' than compact ones.

Similar to (2.7), probabilities could be re-constructed, now under the assumption that the variances are equal in all bands for all classes, while covariances are 0.

A final simplification might be obtained by using city-block distances

$$B_i(\mathbf{x}_p) = \sum_{j=1}^{M} |x_{p,j} - m_{i,j}| \ , \tag{2.10}$$

where $x_{p,j}$ and $m_{i,j}$ are the components of $\mathbf{x}_p$ and $\mathbf{m}_i$, respectively.

Box classifier

Outside the above family of classifiers is the *box classifier* (or *parallelepiped classifier*). For each class, a box, *i.e.* a rectangle, block or hyper-block, according to the number of features, is created in the feature space on the basis of the training samples. The decision rule attributes a class to a pixel according to which box contains the pixel's feature vector. According to the *min-max* method, boxes are chosen that fit exactly around the training samples. Alternatively, the *mean-standard deviation* method positions a box in such a way that the class mean vector is at the center, whereas the size in each dimension is determined by the standard deviation of the corresponding feature. A statistical model would be that for each class the features are uniformly distributed over an interval, and that the features values are independent — especially the second assumption is not very realistic, considering, for example, that illumination effects affect all image bands equally. Under the model, with mean $m_{i,j}$ and standard deviation $s_{i,j}$ for feature j in class C_i, the interval should be chosen as $[m_{i,j} - s_{i,j}\sqrt{3}, m_{i,j} + s_{i,j}\sqrt{3}]$, since the standard deviation of such interval equals $s_{i,j}$. If the box B_i of class C_i covers b_i feature space cells, the class probability density $\mathrm{P}(\mathbf{x}_p|C_i)$ is estimated as:

$$\mathrm{P}(\mathbf{x}_p|C_i) = \begin{cases} \frac{1}{b_i} & \text{if } \mathbf{x}_p \in B_i \\ 0 & \text{otherwise} \end{cases}$$

When $\mathbf{x}_p$ is inside a feature space area where two boxes overlap, the class with the smallest box should be chosen (in the absence of prior probabilities), since it yields the highest probability density.

k-Nearest Neighbor

Non-parametric classifiers, such as k-Nearest Neighbor, implement decision rules that do not assume parameterized class probability distribution functions. Instead, they consider a (small) subset of the training samples around the feature vector $\mathbf{x}_p$ to be classified, and assign the class label of to the majority of these samples.

The k-nearest neighbor method selects those k training samples that are nearest to $\mathbf{x}_p$ in the feature space. Usually, Euclidean distance is used, but Mahalanobis distance could be preferred for anisotropic feature spaces.

The relationship between the k-Nearest Neighbor decision function and class probability density estimates can be explained intuitively. If many C_i pixels are found near to $\mathbf{x}_p$, apparently C_i has a high density in that part of the feature space. In addition, the total numbers of training samples should be considered. If the training set contains many more samples for C_i than for C_j, then also relatively many C_i samples, compared to C_j, will be found near to $\mathbf{x}_p$.

Let k_i ($i \in [1..N]$) be the number of C_i samples among the k nearest neighbors of $\mathbf{x}_p$. Therefore, $\sum k_i = k$. Two cases can be distinguished:

Equal sampling: If equal numbers of samples were taken for each class, the values of k_i for $\mathbf{x}_p$ are proportional to the class probability densities $\mathrm{P}(C_i|\mathbf{x}_p)$. The majority vote yields a non-parametric maximum class probability density classifier.

Proportional sampling: If the (expected) class proportions are reflected in the numbers of training samples per class, for example when sample locations were randomly chosen, the values of k_i are proportional to the enumerator $\mathrm{P}(\mathbf{x}_p|C_i)\ \mathrm{P}(C_i)$ of Bayes formula. With normalized $\mathrm{P}(C_i|\mathbf{x}_p)$ ($i \in [1..N]$),

$$\mathrm{P}(C_i|\mathbf{x}_p) = \frac{k_i}{k}$$

A non-parametric *maximum likelihood* decision is taken by the majority vote [Duda and Hart, 1973], [Mulder and Middelkoop, 1990].

2.3.4 Comparison

Compared to *full maximum likelihood*, its simplified derivates, such as *minimum Mahalanobis distance* and *minimum Euclidean distance*, are not inferior in all cases. It may be worse to use poorly estimated class covariance matrices, caused by insufficient training samples, than to ignore them entirely.

Non-parametric classifiers are able to model irregularly-shaped class probability densities, which often occur in satellite images. Certain land-use classes, such as *built-up* and *agricultural* areas, may consist of several land covers with different spectral signatures in different (unknown) proportions, for which Gaussian class probability densities may be unrealistic. Also signatures of land-cover classes, influenced by soil type, soil moisture, sun incidence angle (on slopes) etc. may be inadequately modeled by Gaussian densities. To obtain accurate non-parametric probability density estimates, many training samples are required. Where parametric methods only need to estimate a few distribution function parameters, non-parametric ones must be able to produce many density estimates independently of each other in different, preferably small parts of the feature space.

Therefore, the k-Nearest Neighbor classifier is suitable when there already is information available about the area, for example when the goal is to update existing

information. To use this existing information is the main objective of this study, and k-Nearest Neighbor is a promising tool. The next section further explores the statistical estimation capabilities of this method. It will be the primary class probability estimation tool throughout the subsequent Chapters.

The parameter k in k-Nearest Neighbor, the number of neighbors to be searched for in the feature space around each $\mathbf{x}_p$ to be classified, has to be specified by the user. It is obvious that it depends on the training set size — k should certainly not be larger than the number of training samples of the smallest class in the set, otherwise a pure pixel of that class can never be found. On the other hand, when many training samples are available for each class, the results of k-Nearest Neighbor appear to be quite stable under different k-values.

As an example, results of a series of straightforward k-NN classification with different k of the "Twente" data set (see section 3.2.3), using training samples according to Table 2.2, are shown in Table 2.3.

class	samples
agric.	12164
ind.	522
city	868
resid.	2177
water	125
natveg	3086

Table 2.2: Number of samples per class

k	average accuracy	average reliability	overall accuracy
1	72.83	65.74	84.14
3	74.55	67.21	84.92
5	76.00	68.78	85.62
7	75.54	69.26	85.98
9	75.43	69.36	86.12
11	75.31	69.85	86.23
13	75.47	69.90	86.33
17	75.92	70.27	86.67
25	76.03	70.63	86.82
31	75.87	70.78	86.89
39	75.50	70.80	86.88
49	75.40	70.70	86.75

Table 2.3: k-NN classifications with different k

Feature space partitioning

Since classification labeling is irrespective of the position of the pixel p in the image, a decision function provides a partitioning of the feature space $\mathbf{X}$. Figure 2.4 illustrates the above-mentioned classification methods in a two dimensional feature space of a the "Flevo" example (see sections 2.4 and 3.2.2).

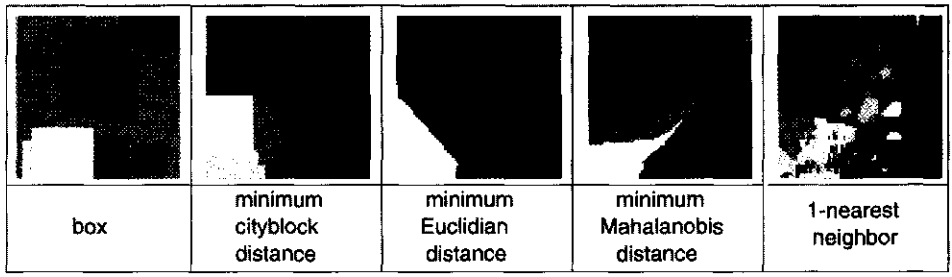

Figure 2.4: Partioning of a two-dimensional feature space (Thematic Mapper bands 3 and 5) by five common classification methods, using Flevo data set.

2.4 Estimation of prior and conditional probabilities per class

To implement a classifier, prior probabilities and conditional probability densities of all classes must be either known or estimated from training data. However, in some applications, this information is not available for all classes. There can be an unknown class in the image, and information about the number of pixels that belong to it is usually not present. For instance, in remote sensing images used for estimation of different vegetation types, classes of plants can be sampled very accurately. But there can be other classes in the image, such as built-up areas, with unknown probability distributions. They distort classification.

In these cases, classification with *rejection* can be used.

In section 2.2, following [Ripley, 1996], a distinction was made between rejects (or doubt reports) and outliers, where the first refer to the problems of spectral overlap and mixed pixels, and the second to pixels that cannot be attributed to any class. [Dubuisson and Masson, 1993] make the same distinction, but use the terms *ambiguity reject* and *distance reject*, respectively. I have a slight preference for the latter terminology, although it might be argued that 'distance reject' already refers too much to the implementation.

Ambiguity reject indicates that there is not enough information in the training set to classify a pixel. The pixel belongs to a region between different classes in the feature space. In case of nearest neighbor classification, a pixel is not attributed to any class (rejected) if the number of neighbors of each class is less than a qualifying majority level [Hellman, 1970], [Dasarathy, 1980]. This type of rejection is not sufficient when an unknown class is present. In this case distance reject must be used, which indicates that a pixel is located too far from all known classes to be attributed to one of them.

In terms of probabilities, we would like to reject the pixels for which the max-

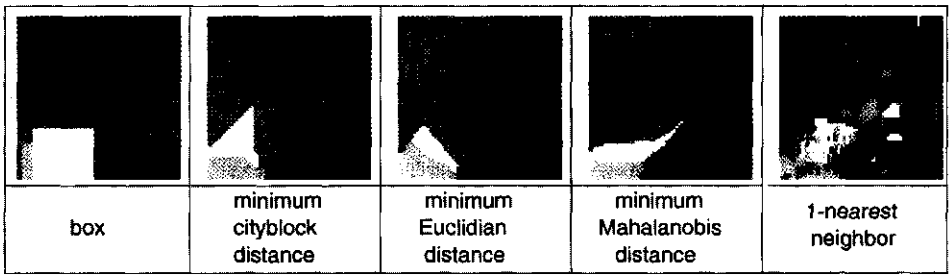

Figure 2.5: Partioning of a two-dimensional feature space (Thematic Mapper bands 3 and 5) by five common classification methods, using Flevo data set with *unknown* class.

imum a posteriori probability belongs to the unknown class. In case of nearest neighbor the search radius [Dasarathy, 1980] or the mean distance to the neighbors [Dubuisson and Masson, 1993] can be thresholded, but this takes neither the unconditional probability density, which is different in every point of the feature space, nor the class priors into account. This leads to classification errors.

When a maximum *a posteriori* probability classifier is used, it would be an advantage to be able to threshold the *a posteriori* probabilities of known classes to find the unknown class, for which, however, the prior probabilities and the unconditional probability density must be known. Commercially available image processing packages approximate this by thresholding Mahalanobis, Euclidean or city-block distance [the ERDAS Field Guide, 1993]. Also in this case, unconditional probability density and class priors are not considered. Another problem is that in both cases the threshold is not known and has to be guessed interactively or estimated from the training set.

Feature space partitionings, obtained by the above methods, are shown in Figure 2.5 and can be compared with Figure 2.4.

In section 2.3.2 it was mentioned that knowing the *a posteriori* class probabilities $P(C_i|\mathbf{x})$ $(i \in [1..N])$, the *a posteriori* probability that a pixel belongs to a different (*unknown*) class C_0 equals $P(C_0|\mathbf{x}) = 1 - \sum P(C_i|\mathbf{x})$. However, usually $\sum P(C_i|\mathbf{x}) = 1$, because $P(C_i|\mathbf{x})$ is normalized.

To find the probability for the unknown class, $P(\mathbf{x})$ has to be estimated independently, in addition to class probability densities.

Estimation of conditional and unconditional densities

Whereas the (conditional) class distributions are often assumed to be Normal, this assumption can certainly not be made for the unconditional distribution of feature vectors. To simply take this distribution from the frequency of occurrence in the

image (the multi-dimensional histogram) introduces too much noise, since the number of possible feature vectors in a multi-dimensional feature space, compared to the number of pixels in the image, is very large. The feature space if only sparsely filled with actual image pixels.

An extended k-NN algorithm is able to estimate the conditional feature probability density $\mathrm{P}(\mathbf{x}|C_i)$ and the unconditional density $\mathrm{P}(\mathbf{x})$ at the same time.

The way to obtain the first is described in [Fukunaga & Hummels, 1987] and [Therrien, 1989], and was already suggested by [Duda and Hart, 1973]. While looking in the feature space for k training samples around a feature vector $\mathbf{x}$, meanwhile counting k_i per class, the algorithm keeps track of the size of the volume $V_\mathbf{x} \subset X$ in the feature space neighborhood around $\mathbf{x}$ that is traversed. $V_\mathbf{x}$ is a sphere in a 3-dimensional space, generally it is a hyper-sphere. At the center of $V_\mathbf{x}$ is $\mathbf{x}$; its radius is such that $V_\mathbf{x}$ contains k training samples. The size $v_\mathbf{x}$ of the volume is, therefore, the discrete number of inspected feature space cells.

Using the symbol $N_i = A(T_i)$ for the total number of training samples of class C_i, an estimate for the probability that such a training sample $\mathbf{x}_0$ is inside $V_\mathbf{x}$ is

$$\mathrm{P}(\mathbf{x}_0 \in V_\mathbf{x}|C_i) = \frac{k_i}{N_i} \tag{2.11}$$

Assuming that the conditional feature density $\mathrm{P}(\mathbf{x}|C_i)$ is constant inside the volume, it can be estimated as

$$\mathrm{P}(\mathbf{x}|C_i) = \frac{1}{v_\mathbf{x}} \frac{k_i}{N_i} \tag{2.12}$$

As mentioned before, the assumption implies that the volume must not be too large. To fill a large number of small volumes in the feature space, many training samples are needed (Figure 2.6). Moreover, declaring the estimate in (2.12) valid for all C_i pixels with feature vector $\mathbf{x}$ in the image requires representative sampling.

While scanning the feature space, the algorithm also counts the total number $A_\mathbf{x}$ of image pixels that have their feature vectors inside the volume $V_\mathbf{x}$. Note that this is a subset of the image, whereas $V_\mathbf{x}$ is a subset of the feature space.

$$A_\mathbf{x} = A(\{p \in I : \mathbf{x}_p \in V_\mathbf{x}\}).$$

Knowing how many pixels out of the total number $A = A(I)$ of pixels in the entire image are similar to $\mathbf{x}$, *i.e.* are near to $\mathbf{x}$ in the feature space, the probability that this happens to a "random" image pixel is estimated as

$$\mathrm{P}(\mathbf{x} \in V_\mathbf{x}) = \frac{A_\mathbf{x}}{A} \tag{2.13}$$

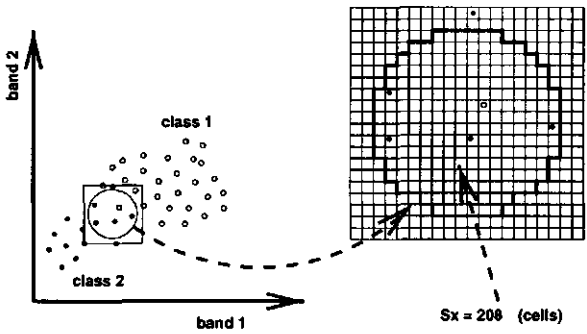

Figure 2.6: k-Nearest Neighbor estimation of probability densities

Assuming that the unconditional density is constant inside the volume, this becomes

$$\mathrm{P}(\mathbf{x}) = \frac{A_{\mathbf{x}}}{A\ v_{\mathbf{x}}} \tag{2.14}$$

If $Q_i(\mathbf{x})$ denotes

$$Q_i(\mathbf{x}) = \frac{\mathrm{P}(\mathbf{x}|C_i)}{\mathrm{P}(\mathbf{x})}, \tag{2.15}$$

Bayes formula (2.2) becomes

$$\mathrm{P}(C_i|\mathbf{x}) = Q_i(\mathbf{x})\mathrm{P}(C_i). \tag{2.16}$$

An estimate for $Q_i(\mathbf{x})$, using (2.12) and (2.14), is

$$Q_i(\mathbf{x}) = \frac{\mathrm{P}(\mathbf{x}|C_i)}{\mathrm{P}(\mathbf{x})} = \frac{k_i A}{N_i A_{\mathbf{x}}}. \tag{2.17}$$

It is advantageous that $v_{\mathbf{x}}$ disappears from the calculation. The variables k_i, $A_{\mathbf{x}}$ and $v_{\mathbf{x}}$ are stochastic and the possibility to eliminate one of them reduces statistical noise.

Estimation of a priori and a posteriori probabilities

For a pixel of class C_i without any spectral overlap with other classes, the a posteriori probability $\mathrm{P}(C_i|\mathbf{x}) = 1$, whereas $\mathrm{P}(C_j|\mathbf{x}) = 0$ for $j \neq i$.

In such a pixel, $Q_i(\mathbf{x})$ is at its maximum value $Q_i^{\max}$, so there we can derive the prior probability $\mathrm{P}(C_i)$ of class C_i from (2.16)

$$1 = Q_i^{\max}\mathrm{P}(C_i) \tag{2.18}$$

Since $\mathrm{P}(C_i)$ does not depend on $\mathbf{x}$ (it is valid for the entire image), we can now also substitute it in (2.16) for unpure pixels:

$$\mathrm{P}(C_i|\mathbf{x}) = Q_i(\mathbf{x})\mathrm{P}(C_i) = \frac{Q_i(\mathbf{x})}{Q_i^{\max}}. \tag{2.19}$$

Because $Q_i(\mathbf{x})$ is stochastic, some care must be taken during the calculation of $Q_i^{\max}$. The global maximum might be an outlier. Instead, assuming that training samples are pure, the average value of Q_i in the training samples of C_i can be used to estimate $Q_i^{\max}$.

It is interesting to observe that the stochastic nature of $Q_i(\mathbf{x})$ is not the only reason for its variability. Also among the pixels that belong to the *unknown* class, there are some that are much more similar to class C_i than to any other. For those pixels, k-NN will find $k_i = k$ and $k_j = 0$ for $j \neq i$, and before knowing the *unknown* class they can only be considered pure C_i pixels. They, however, will get a small Q_i value and, accordingly, $\mathrm{P}(C_i|\mathbf{x})$ is small.

Classification

After the *a posteriori* probabilities of all known classes are estimated, the a posteriori probability of the unknown class, as was already mentioned, can be easily calculated as

$$\mathrm{P}(C_0|\mathbf{x}) = 1 - \sum_{i=1}^{N} \mathrm{P}(C_i|\mathbf{x}).$$

Pixels are then, as usual, assigned to the class with the highest *a posteriori* probability.

Experiment

The experiment concerns a satellite image of an agricultural area in the Netherlands. The image is recorded by the Thematic Mapper sensor of a LANDSAT satellite, which measures reflected sunlight in six spectral bands (visible and infrared) with a spatial resolution of 30 m (Figure 2.7). In the experiment bands 3, 4 and 5 are used. The study area is located around Biddinghuizen in the Flevopolder, the Netherlands.

The purpose is to make a map of agricultural crops in the area. The seven predominant crops define the classes of the classification: grass, wheat, potatoes, sugar

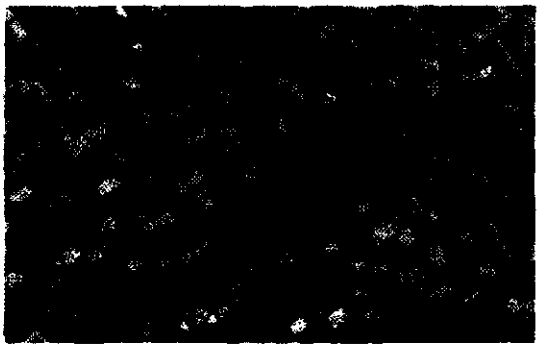

Figure 2.7: Thematic Mapper image bands 4,5,3 of Flevo study area

beets, peas, beans and onions. Agricultural survey data are available, from which 200 training samples are taken for each class. Not part of the survey, although present in the area, are a village, a few canals, some forested areas, roads, farmhouses and orchards. Together these constitute the *unknown* class.

During the experiment, the prior probabilities of the seven classes are estimated as $P(C_i) = 1/Q_i^{\max}$. Their sum equals 0.723, leaving 0.277 for the *unknown* class (Table 2.4).

class	prior
unknown	0.277
grass	0.016
wheat	0.212
potatoes	0.166
sugar beet	0.204
peas	0.016
beans	0.054
onions	0.055

Table 2.4: Estimated class prior probabilities, including *unknown*

This result is reasonably in accordance with the survey data; an exact comparison is not possible since the survey is incomplete (Fig. 2.8).

Next, *a posteriori* probabilities are calculated for each class in each pixel. Subtracting their sum from 1 gives the probabilities for *unknown*. The final classification is obtained by selecting the class with the maximum probability from all classes, including *unknown* (Figure 2.8).

Minimum Mahalanobis distance:

	gr	wh	po	sb	pe	be	on	uncl	ACC
gr	129	11	1	8	2	4	0	0	0.83
wh	197	5633	20	317	16	60	106	0	0.89
po	223	7	5340	2	6	3	16	0	0.95
sb	72	368	69	4561	4	60	48	0	0.88
pe	0	0	0	0	270	0	8	0	0.97
be	20	3	0	40	0	906	37	0	0.90
on	6	9	0	2	11	24	882	0	0.94
REL	0.20	0.93	0.98	0.93	0.87	0.86	0.80		

average accuracy = 91.00 %
average reliability = 79.67 %

overall accuracy = 90.87 %
overall reliability = 90.87 %

k-Nearest Neighbor:

	gr	wh	po	sb	pe	be	on	uncl	ACC
gr	124	10	0	9	1	8	3	0	0.80
wh	169	5406	14	560	16	61	123	0	0.85
po	192	4	5363	8	5	2	23	0	0.96
sb	49	164	78	4802	8	36	45	0	0.93
pe	0	0	0	0	269	0	9	0	0.97
be	13	2	1	30	0	921	39	0	0.92
on	8	4	0	11	20	14	877	0	0.94
REL	0.22	0.97	0.98	0.89	0.84	0.88	0.78		

average accuracy = 90.83 %
average reliability = 79.58 %

overall accuracy = 91.08 %
overall reliability = 91.08 %

k-Nearest Neighbor with estimated priors:

	gr	wh	po	sb	pe	be	on	uncl	ACC
gr	100	27	2	13	1	11	1	0	0.65
wh	32	5528	19	614	11	53	92	0	0.87
po	58	26	5465	24	2	3	19	0	0.98
sb	16	182	85	4843	3	27	26	0	0.93
pe	0	1	0	0	263	0	14	0	0.95
be	0	5	1	55	0	911	34	0	0.91
on	5	9	0	19	5	15	881	0	0.94
REL	0.47	0.96	0.98	0.87	0.92	0.89	0.83		

average accuracy = 88.88 %
average reliability = 84.61 %

overall accuracy = 92.26 %
overall reliability = 92.26 %

k-Nearest Neighbor with unknown class:

	gr	wh	po	sb	pe	be	on	uncl	ACC
gr	90	21	1	11	0	4	0	28	0.58
wh	30	5425	5	474	7	39	41	328	0.85
po	4	1	5261	1	0	1	7	322	0.94
sb	8	166	64	4722	2	21	15	184	0.91
pe	0	0	0	0	231	0	7	40	0.83
be	0	2	0	35	0	861	25	83	0.86
on	5	4	0	4	4	12	802	103	0.86
REL	0.66	0.97	0.99	0.90	0.95	0.92	0.89		

average accuracy = 83.31 %
average reliability = 89.54 %

overall accuracy = 89.19 %
overall reliability = 94.46 %

Table 2.5: Error matrices

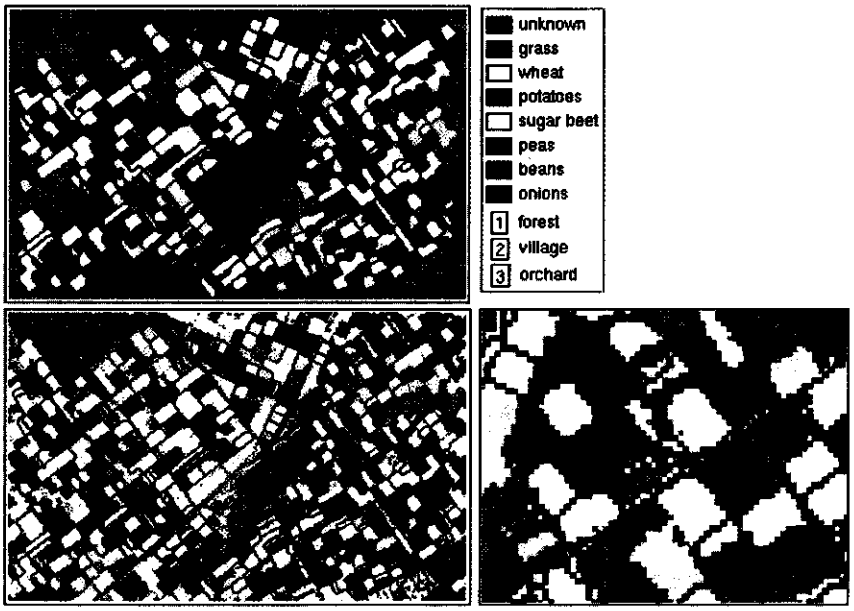

Figure 2.8: Upper: survey map – lower: non-parametric classification with *unknown* class, overlayed with crop map boundaries

2.5 Classification uncertainty

When the maximum *a posteriori* probability of pixels after classification is less than 1, the correctness of the assigned label is uncertain. Chapter 6 describes automated decision support, taking classification uncertainty into account.

For the time being, to support the user to evaluate classifications and to make decisions on the basis of the results, the software should provide an certainty measure, in addition to the classification itself. The per-pixel *a posteriori* probability vector contains the complete information, but for interpretation, a single scalar number is preferable. Several scalar certainty measures can be conceived, which can be visualized as a gray-scale map, or be combined with the classified map, for example by displaying each pixel with a hue and a saturation according to the class label, and an intensity according to the certainty measure. (Figure 2.9).

An obvious certainty measure is the maximum probability value, which (by definition) indicates the probability that the classifier took the correct decision.

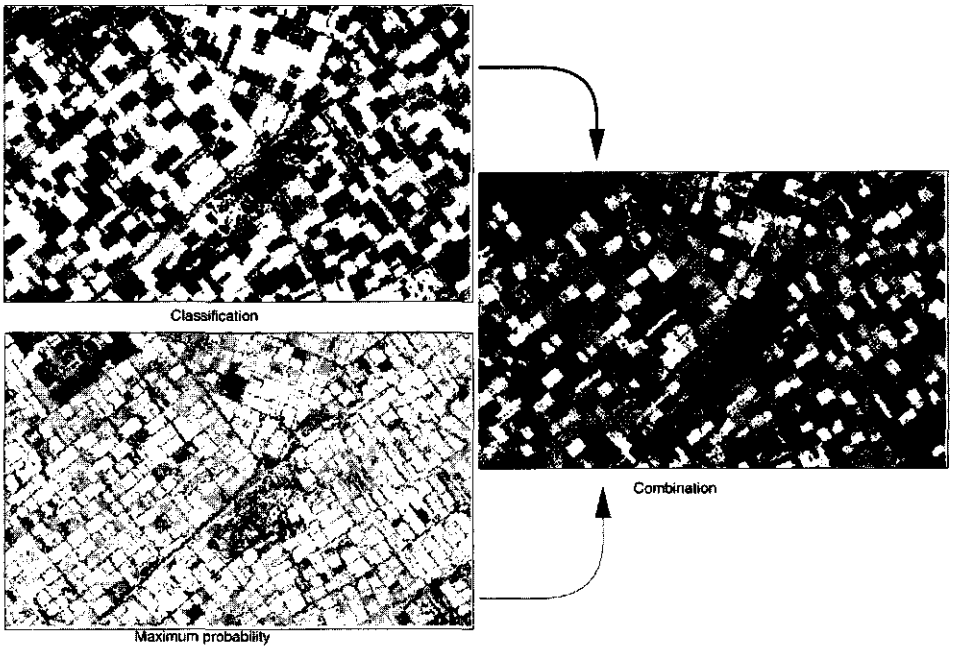

Figure 2.9: Visualization of classification uncertainty by intensity modulation

Entropy

To capture the entire probability vector, instead of only its maximum, *weighted uncertainty* measures can be used, such as the well-known *entropy* measure, originating from information theory [Shannon, 1948], [Kullback, 1954]. The measure pertains to a statistical variable and to the uncertainties in its possible values, expressing the distribution and the extent of these uncertainties in a single number [Goodchild *et al.*, 1994]. In the entropy measure, the uncertainty in a single value of a statistical variable is defined as the *information content* of a piece of information that would reveal this value with perfect accuracy. This quantity is weighted by the probability that the value occurs and summed over all values, which gives

$$e(\mathbf{x}_p) = \sum_{i=0}^{N} -\mathrm{P}(C_i|\mathbf{x}_p)\ \log_2(\mathrm{P}(C_i|\mathbf{x}_p)) \tag{2.20}$$

The Flevo example in the previous section deals with eight classes, including *unknown*. In case of complete certainty concerning class membership, three bits are needed to encode the information in each pixel: a class number between 0 (binary 000) and 7 (binary 111). The entropy measure (eq. 2.20 with N=7) yields a number

between 0 and 3, which specifies how much of these three bits of information is still missing after classification has been completed (Figure 2.10). All *a posteriori* probabilities being equal to $\frac{1}{8}$ means that nothing is known about class membership, and the entropy value equals 3. If, on the other hand, one of the probabilities equals 1 (and the others 0), class membership is completely determined, which is reflected in entropy value 0. Between these two extremes is the situation with two probabilities equal to 0.5, the remaining six being 0. Then the entropy measure yields the value 1: one additional bit of information would be needed to change the complete ambiguity between two classes into a definite choice. Similarly, four times 0.25 and four times 0 gives entropy 2, the number of bits needed to choose one out of four.

Note, that entropy expresses uncertainty **according to** the vector of a posteriori probabilities. It does not involve uncertainty **concerning** these probabilities: they are assumed to be correct. However, to estimate them correctly is exactly the problem in classification. As a consequence, the entropy measure cannot be used to compare classifiers that estimate probabilities in different ways.

Figure 2.10: Entropy measure for classification uncertainty

2.6 Conclusion

This Chapter explored k-NN methods to gather statistical information relevant for satellite image classification. These methods can reveal more information than traditional image classifiers, concerning quantification of classification uncertainty and assignment of pixels to an *unknown* class. The algorithm is mathematically solid, without too many heuristic assumptions. The classification is according to maximum *a posteriori* probability.

It is noticed, that the method is based on the assumption of a representative sampling of known classes. An investigation of the sensitivity of the results with respect to this assumption can be a future research topic, as well as the question how results are influenced by classes being crisp or fuzzy.

Another question to be considered in the future is a further use of the obtained *unknown* class. In fact, we are speaking about a 'remainder-class'. This might consist of several unknown cover classes, for which further subdivision may be desirable. A possibility, suggested in [Dasarathy, 1980] and [Dubuisson and Masson, 1993], is to use unsupervised classification (clustering) to reveal the structure of the unknown class. Another possibility is to use region-based segmentation techniques, such as region merging (Chapter 4), to distinguish areas inside the unknown class.

Chapter 3

Local statistics

Let an area, consisting of a left and a right half s_1 and s_2, be covered by two classes A and B, in proportions 1:3 for s_1 and 3:1 for s_2. (Figure 3.1).

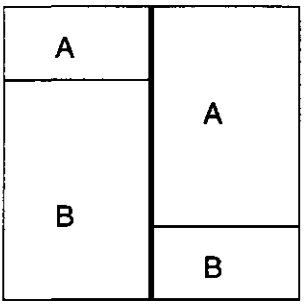

Figure 3.1: Two classes in two mixing proportions

When classifying s_1 with prior probabilities $\mathrm{P}(A) = 0.25$ and $\mathrm{P}(B) = 0.75$, a higher overall classification accuracy can be expected than with equal prior probabilities. The same holds in s_2, when $\mathrm{P}(A) = 0.75$ and $\mathrm{P}(B) = 0.25$ are used.

Therefore, classifying the two halves separately with the correct prior probabilities and combining the results gives a higher overall classification accuracy than classifying the entire image at once with prior probabilities $\mathrm{P}(A) = 0.5$ and $\mathrm{P}(B) = 0.5$. The improvement requires:

1. a subdivision of the area, such that different mixing proportions are obtained
2. estimates of these mixing proportions, to be used as *local prior probabilities*.

Class mixing proportions depend on local terrain conditions, such as soil, elevation, slope, hydrology, infrastructure and socio-economic factors. Therefore, a suitable subdivision can be obtained from available GIS maps concerning these themes. *Expert knowledge* can be brought into play to relate the units in those maps to

expected class mixing proportions. Alternatively, procedures to estimate prior probabilities can be applied locally.

Section 2.4 described a prior probability estimator in the context of exploring k-NN classification. It also enables estimation of probabilities for the *unknown* class. The first section of this Chapter develops a more robust prior probability estimator, which, however, involves normalization and does not allow for an *unknown* class. Secondly, this algorithm is applied locally, and two experiments will be presented. The third section identifies specific circumstances in which also class probability densities must be estimated locally. Both a semi-parametric and a non-parametric method are given.

A comparison of results of the various methods will be presented in the final section.

3.1 Iterative prior probability estimation

Statistical pattern recognition procedures, such as maximum likelihood classification, are applied to (multi-spectral) satellite images, in order to produce thematic maps, mostly concerning land-use or land-cover. Sometimes, the purpose of a classification is not primarily to make a map, but to obtain estimates of the sizes of the areas covered by the different classes, for example to make crop yield predictions.

Area estimates created by counting the number of pixels per class label after a maximum likelihood classification are not reliable, because classifiers tend to be biased. For example, when prior probabilities are chosen according to the expected class areas, classes with high prior probabilities are likely to be over-estimated. In the hypothetical case where we know the correct class areas beforehand and base prior probabilities on these, we obtain an improved classification compared to using equal priors. However, the area estimates which result from making a histogram afterwards will be different from the prior knowledge [Conese and Maselli, 1992].

Maximum *a posteriori* probability classifiers select for a given feature vector $\mathbf{x}$ the class label C_m from a set of classes $\{C_i\}$ $(i \in [1..N])$ with the highest *a posteriori* probability. This requires estimation of class probability densities (from training samples) and of class *a priori* probabilities. The latter have to be specified by the user as the expected relative class areas. In applications as mentioned above, however, this is part of the information that the user hopes to get from the classifier, not what he wants to specify beforehand.

Suppose that 100 image pixels have the same feature vector $\mathbf{x}$ and that the posterior probability $\mathrm{P}(C_i|\mathbf{x}) = 0.77$. This should be interpreted as: 77 out of these 100 pixels can be expected to belong to C_i; the other 23 belong to other classes. Unfortunately, a per-pixel classifier is unable to tell which 23 these are. It will classify all 100 as C_i and, therefore, makes 23 wrong decisions and over-estimates the area of C_i by 23 pixels.

Therefore, an iterative process in which histogram-based area estimates of iteration step n are used as prior probabilities in step $n+1$ will not converge to the correct result. Once one of the priors is over-estimated, it may be more over-estimated in the next iteration. (This does not necessarily imply that one of the classes will go to a prior of 1 and the rest to 0. It only means that the final estimate is not correct.)

The previous chapter described an *a priori* probability estimator by estimating the ratio between *a priori* and *a posteriori* probabilities. In pixels without overlap with other classes, the latter equals 1, and, using the ratio, the former can be established.

The approach, followed in this Chapter, uses the entire vector of posterior probabilities per pixel. [Duda and Hart, 1973] suggests that the sum of these vectors over the entire image yields an estimate for the vector $(A_1, \ldots, A_N)$ of the areas per class, measured in pixels; by normalizing the areas we obtain the vector of prior probabilities. This claim will be validated for non-parametric local prior estimation.

Iteration

The method, described below, iteratively calculates prior probabilities for an image on the basis of class probability densities per pixel. The claim to be proven is that if the initial prior probabilities are correct, the resulting ones are the same. But if they deviate, the resulting ones are different and closer to the correct ones, and will finally converge to those.

3.1.1 Description

Suppose an image has 3 spectral bands and A pixels. The feature vectors look like:

pixel number :	1	2	.....	A
band 1	12			
band 2	47			
band 3	88			
feature vectors :	$\mathbf{x}_1$	$\mathbf{x}_2$	.....	$\mathbf{x}_A$

Note that the feature vectors are not necessarily all different.

We want a classification with N classes. From training data, either using parametric (Gaussian) or non-parametric (k-NN) methods, we get class probability densities as functions of $\mathbf{x}$:

$\mathrm{P}(\mathbf{x}_p|C_i)$:

	$\mathbf{x}_1$	$\mathbf{x}_2$	.....	$\mathbf{x}_A$
C_1	.1			
C_2	.6			
$\vdots$				
C_N	.2			

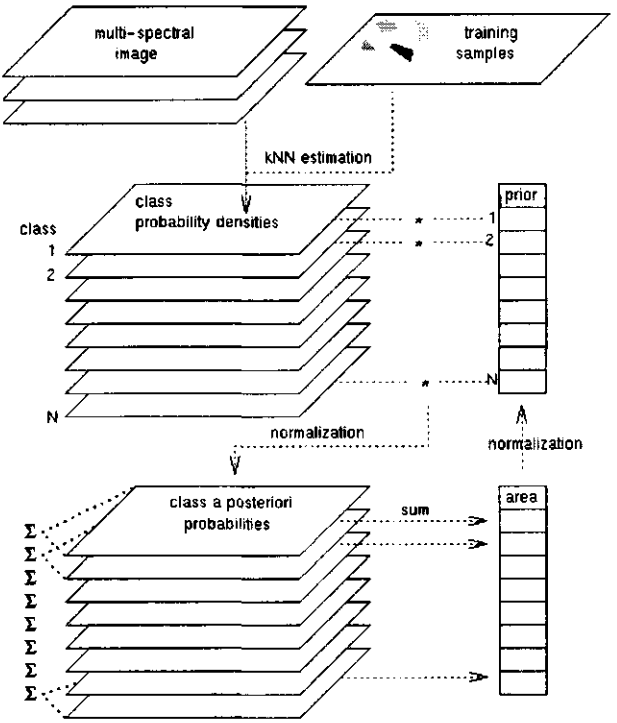

Figure 3.2: Iterative calculation of prior probabilities

Note that with many different feature vectors in an image, the probability to find a paricular one is usually very small. Therefore, $\sum_{i=1}^{N} \mathrm{P}(\mathbf{x}_p|C_i) \neq 1$. Also the sum $\sum_{j=1}^{A} \mathrm{P}(\mathbf{x}_j|C_i) \neq 1$, because of duplicate feature vectors. The sum increases with A.

If class prior probabilities are available:

$$\mathrm{P}^T(C_i) = \boxed{0.1} \boxed{0.2} \boxed{\ldots\ldots} \boxed{0.65}$$

then for each pixel and each class

$$\mathrm{P}(\mathbf{x}_p|C_i)\ \mathrm{P}(C_i)\ :$$

	$\mathbf{x}_1$	$\mathbf{x}_2$	.....	$\mathbf{x}_A$
C_1	.01			
C_2	.12			
⋮				
C_N	.13			

can be calculated. After normalization of every column posterior probabilities

$$\mathrm{P}(C_i|\mathbf{x}_p) = \frac{\mathrm{P}(\mathbf{x}_p|C_i)\ \mathrm{P}(C_i)}{\sum_{j=1}^{N} \mathrm{P}(\mathbf{x}_p|C_j)\mathrm{P}(C_j)}$$

are obtained:

$\mathrm{P}(C_i|\mathbf{x}_p)$:

	$\mathbf{x}_1$	$\mathbf{x}_2$	.....	$\mathbf{x}_A$
C_1	.02			
C_2	.24			
$\vdots$				
C_N	.26			

The interpretation is that if there are 100 pixels with the same feature vector $\mathbf{x}_1$, then 2 are expected to belong to class 1, 24 to class 2, ..., and 26 to class N. These are the contributions of those 100 pixels to the respective class areas. Since it is not possible to differentiate between pixels with the same feature vector, the contributions are spread equally over these 100 pixels. Each of them contributes 0.2 to class 1, 0.24 to class 2, etc. Therefore, the contribution of each pixel with feature $\mathbf{x}_1$ to the total area covered by class C_i is equal to the *a posteriori* probability $\mathrm{P}(C_i|\mathbf{x}_1)$. Of course, this is independent of the number 100.

	1	2	.....	A		**area**
$\sum_j$	.02				$\rightarrow$	A_1
$\sum_j$	.24				$\rightarrow$	A_2
$\vdots$					$\vdots$	
$\sum_j$	.26				$\rightarrow$	A_N

The sum of the areas $= A$, the size of the image in pixels.

We can normalize the array into prior probabilities, for example:

$\mathrm{P}^T(C_i)$ =	0.3	0.09	.....	0.55

But we were already using prior probabilities:

$\mathrm{P}^T(C_i)$ =	0.1	0.2	.....	0.65

If the prior probability values that were specified beforehand were correct, there is a problem: the estimated areas are obviously wrong. But (*i*) if the initial priors are only roughly estimated, (*ii*) the classes are well-defined and (*iii*) there is confidence in the training samples being selected representatively (these requirements exist for any image classification), then we may assume that the calculated priors are more accurate than the initial ones. This opens the possibility to repeat the process,

starting off with the newly calculated priors ..., and so on. The interesting point is that this process stabilizes.

So, the iterative process looks like:

1. initialize priors (for example: all to $\frac{1}{N}$).
2. apply Bayes' formula:
   - multiply probability densities by priors
   - normalize per pixel
3. sum over pixels $\Rightarrow$ class areas
4. normalize $\Rightarrow$ updated priors
5. if priors changed significantly, go to 2
6. select maximum likelihood class per pixel $\Rightarrow$ map

(See Fig. 3.2).

The only data needed for this process is a vector of class probability densities for each pixel, as it can be estimated, for example, by the K-NN method of Chapter 2. In the following Lemma, conditions will be formulated for a set of (probability) vectors, under which the iteration converges to a sensible (non-trivial) solution. Next, it will be shown that a collection of probability density vectors, established during supervised classification, satisfies those conditions, if the training samples are representative for the class populations. After this has been proven for a classification with only two classes, the general case with N classes is considered.

3.1.2 A central Lemma

We consider a two-class (C_1 and C_2) problem with A pixels having feature vectors $\mathbf{x}_1 \ldots \mathbf{x}_A$. Note that not all $\mathbf{x}_i$ are necessarily different. Suppose, the probability densities $d_{1i} = \mathrm{P}(\mathbf{x}_i|C_1)$ and $d_{2i} = \mathrm{P}(\mathbf{x}_i|C_2)$ are known for each pixel. So, we have a collection D of A probability densities vectors

$$D = \begin{pmatrix} d_{11} \\ d_{21} \end{pmatrix}, \begin{pmatrix} d_{12} \\ d_{22} \end{pmatrix}, \cdots, \begin{pmatrix} d_{1A} \\ d_{2A} \end{pmatrix}$$

We will use p_1 and p_2 for the prior probabilities $\mathrm{P}(C_1)$ and $\mathrm{P}(C_2)$, respectively; obviously, $p_1 + p_2 = 1$. Let l_{1i} be the shorthand for the posterior probability $\mathrm{P}(C_1|\mathbf{x}_i)$ that pixel i belongs to class C_1. According to Bayes' formula, it can be calculated as

$$l_{1i} = \frac{d_{1i}p_1}{d_{1i}p_1 + d_{2i}p_2} \tag{3.1}$$

Similarly, $l_{2i} = \mathrm{P}(C_2|\mathbf{x}_i)$

After applying Bayes' rule A times, we obtain a collection L with A posterior probability vectors:

$$L = \begin{pmatrix} l_{11} \\ l_{21} \end{pmatrix}, \begin{pmatrix} l_{12} \\ l_{22} \end{pmatrix}, \cdots, \begin{pmatrix} l_{1A} \\ l_{2A} \end{pmatrix}$$

The sum of the vectors in L equals the vector (A_1, A_2), the total areas covered by C_1 and C_2, respectively. The priors p_1 and p_2 are supposed to be the relative areas, such that $p_1 = \frac{A_1}{A}$ and $p_2 = \frac{A_2}{A}$. Therefore:

$$\sum_{i=1}^{A} l_{1i} = \sum_{i=1}^{A} \frac{d_{1i}p_1}{d_{1i}p_1 + d_{2i}p_2} = A\ p_1 \tag{3.2}$$

and

$$\sum_{i=1}^{A} l_{2i} = \sum_{i=1}^{A} \frac{d_{2i}p_1}{d_{1i}p_1 + d_{2i}p_2} = A\ p_2. \tag{3.3}$$

Note that these equations are equivalent, since $l_{1i} + l_{2i} = 1$ and $p_1 + p_2 = 1$.

Moreover, if the prior probabilities p_1 and p_2 are unknown, they can be obtained by solving equation (3.2). The solution depends only on the matrix D.

Let us look at three small examples of D, D_1, D_2 and D_3, given (omitting the parentheses) as:

$$D_1 = \begin{matrix} 4 & 2 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 3 \end{matrix}, D_2 = \begin{matrix} 4 & 2 & 2 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 2 & 3 \end{matrix}, D_3 = \begin{matrix} 4 & 2 & 2 & 1 & 1 & 4 \\ 1 & 1 & 1 & 1 & 2 & 7 \end{matrix}$$

When dividing all the numbers by 1000 it becomes more easy to imagine them as probability densities, but it will have no effect on the results, because of the normalization in 3.2 and 3.3. We will show below that D_1 gives a solution $0 < p_1 < 1$, D_2 does not give a regular solution, and D_3 yields $p_1 = 1$.

This leads to the following considerations:

- What conditions must D satisfy in order to yield a (unique) solution?
- How to find the solution?
- Does D meet the conditions if it consists of probability densities?

The answers to those three questions will be given in the next three paragraphs.

Conditions on D for a non-trivial solution

The conditions that D must satisfy to yield a non-trivial solution can be found by solving p_1 in equation 3.2. A non-trivial solution $0 < p_1 < 1$ exists under the conditions, formulated in the following Lemma.

Lemma:

Given A observations $\mathbf{x}_1 \ldots \mathbf{x}_A$ of samples belonging to two classes C_1 and C_2, and the collection D of class probability densities $d_{1i} = \mathrm{P}(\mathbf{x}_i|C_1)$ and

$d_{2i} = \mathrm{P}(\mathbf{x}_i|C_2)$. **One and only one non-trivial solution for the population sizes will be found if**

$$\sum_{i=1}^{A} \frac{d_{1i}}{d_{2i}} > A \quad \textbf{and} \quad \sum_{i=1}^{A} \frac{d_{2i}}{d_{1i}} > A.$$

Otherwise, the population sizes are equal to A for one class and to 0 for the other.

Proof:

Consider the equation

$$\sum_{i=1}^{A} l_{1i} = \sum_{i=1}^{A} \frac{d_{1i}p_1}{d_{1i}p_1 + d_{2i}p_2} = A\, p_1 \tag{3.4}$$

(see 3.2) and try to solve p_1. First, we rewrite the equation as

$$\mathcal{F}(p_1) = \sum_{i=1}^{A} f_i(p_1) = 0 \tag{3.5}$$

with

$$f_i(p_1) = \frac{d_{1i}p_1}{d_{1i}p_1 + d_{2i}p_2} - p_1 \tag{3.6}$$

Next, we observe that, because of the normalization in (3.1), in each pixel the posterior probabilities only depend on the ratio between the two densities rather than on their absolute values. This implies that we do not allow 0's to occur in D so that the class probability density for purple grass pixels is positive. Also negative values will not occur.

Therefore, instead of D we can use a collection E:

$$E = \begin{pmatrix} e_1 \\ 1 \end{pmatrix}, \begin{pmatrix} e_2 \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} e_A \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{d_{11}}{d_{21}} \\ 1 \end{pmatrix}, \begin{pmatrix} \frac{d_{12}}{d_{22}} \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} \frac{d_{1q}}{d_{2q}} \\ 1 \end{pmatrix} \tag{3.7}$$

This allows us to change (3.6) into

$$f_i(p_1) = \frac{e_i p_1}{e_i p_1 + p_2} - p_1 \tag{3.8}$$

and since $p_2 = 1 - p_1$:

$$f_i(p_1) = \frac{e_i p_1}{(e_i - 1)p_1 + 1} - p_1 \tag{3.9}$$

Because $f_i(0) = 0$ and $f_i(1) = 0$, it is clear that $p_1 = 0$ and $p_1 = 1$ are solutions of (3.5). We will call these the *trivial* solutions, because they reduce the classification to a one-class problem. Moreover, solutions with $p_1 < 0$ or $p_1 > 1$ may occur. We will not worry about these, since p_1 is a probability. The question is whether there is a (unique) solution for $0 < p_1 < 1$.

Let us have a look at the f_i type of functions. For convenience, we simplify the notation slightly and write (3.9) as

$$f_e(p) = \frac{ep}{(e-1)p+1} - p \tag{3.10}$$

The first, second and third derivatives f'_e, f''_e and f'''_e of f with respect to p will be needed. Let $g_e(p)$ be the denominator of the first term of f_e: $g_e(p) = (e-1)p + 1$, and $g'_e(p) = e - 1$. In the interval $0 \leq p \leq 1$ of our interest, $f_e(p)$ is continuous and $g_e(p) > 0$. It can be shown easily that

$$f_e(p) = \frac{ep}{g_e(p)} - p \tag{3.11}$$

$$f'_e(p) = \frac{e}{g^2(p)} - 1 \tag{3.12}$$

$$f''_e(p) = \frac{-2e(e-1)}{g^3_e(p)} \tag{3.13}$$

$$f'''_e(p) = \frac{6e(e-1)^2}{g^4_e(p)}. \tag{3.14}$$

We observe that in the interval $0 \leq p \leq 1$:

- $f_e(0) = 0$ and $f_e(1) = 0$
- if $e > 1$, $f''_e(p) < 0$, so f_e is convex and $f_e(p) > 0 \; \forall p : 0 < p < 1$
- if $e < 1$, $f''_e(p) > 0$, so f_e is concave and $f_e(p) < 0 \; \forall p : 0 < p < 1$
- if $e = 1$, $g_1(p) \equiv 1$ and $f_1(p) \equiv 0$

Two examples, $f_{0.2}(p)$ and $f_5(p)$ are shown in Fig. 3.3. They have only trivial solutions. Their sum, however, has a non-trivial solution, as the Figure illustrates.

Returning to the problem of solving $\mathcal{F}(p) = 0$, where $\mathcal{F}(p)$ is the sum of A functions $f_e(p)$, the question is now which combinations of f_e's give non-trivial solutions. The answer is given by the derivative of $\mathcal{F}$ at the points $p = 0$ and $p = 1$: $\mathcal{F}'(0)$ and $\mathcal{F}'(1)$. If they have different signs, or if one equals 0, then the function is either entirely positive or entirely negative (within $0 < p < 1$).

Only if both derivative values are positive, the function has a non-trivial solution.

The situation that both derivative values are negative, which would also lead to a non-trivial solution, will not occur, because $\mathcal{F}'''(p) = \sum f'''(p) > 0$. As p increases,

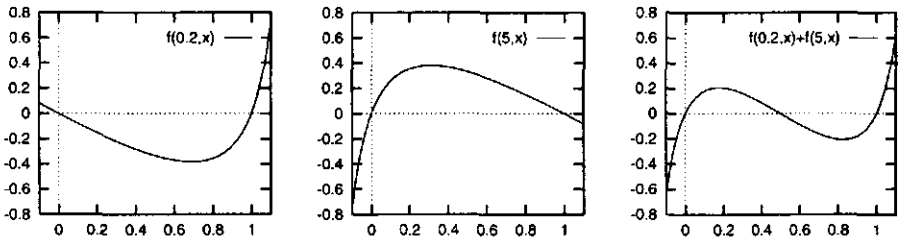

Figure 3.3: Plots of $f_{0.2}(p)$, $f_5(p)$ and $f_{0.2}(p) + f_5(p)$

$\mathcal{F}$ can change from convex into concave, but not the other way round. This also explains why there will never be more than one non-trivial solution.

Consequently,

$$\begin{aligned}\mathcal{F}'(0) &= \sum_{i=1}^{A} f_i'(0) = \sum_{i=1}^{A} (\frac{e_i}{g_i^2(0)} - 1) = \sum_{i=1}^{A} (e_i - 1) = \sum_{i=1}^{A} (\frac{d_{1i}}{d_{2i}} - 1) \\ &= \sum_{i=1}^{A} \frac{d_{1i}}{d_{2i}} - A > 0\end{aligned}$$

and

$$\begin{aligned}\mathcal{F}'(1) &= \sum_{i=1}^{A} f_i'(1) = \sum_{i=1}^{A} (\frac{e_i}{g_i^2(1)} - 1) = \sum_{i=1}^{A} (\frac{e_i}{e_i^2} - 1) = \sum_{i=1}^{A} (\frac{1}{e_i} - 1) \\ &= \sum_{i=1}^{A} \frac{d_{2i}}{d_{1i}} - A > 0\end{aligned}$$

This result is summarized in the lemma.

If we return to the three small examples, we see that all three satisfy the first condition of the Lemma. D_1 also satisfies the second, D_2 does not and D_3 is an example of the limiting case: $\frac{1}{4} + \frac{1}{2} + \frac{1}{2} + \frac{1}{1} + \frac{2}{1} + \frac{7}{4} = \frac{24}{4} = 6 = A$.

Finally, a special case occurs when

$$\sum_{A} \frac{d_{1i}}{d_{2i}} = A \textbf{ and } \sum_{A} \frac{d_{2i}}{d_{1i}} = A$$

This implies that for all i, $d_{1i} = d_{2i}$, because with $\frac{d_{1i}}{d_{2i}} = 1 + \delta_i$ (and therefore $\sum_A \delta_i = \sum_A 1 - A = 0$), from

$$\sum 1 + \delta_i \quad = \quad \sum \frac{1}{1 + \delta_i}$$

follows:

$$\begin{aligned}
\sum \frac{(1+\delta_i)^2}{1+\delta_i} &= \sum \frac{1}{1+\delta_i} \\
\sum (1+\delta_i)^2 &= \sum 1 \\
\sum (1+2\delta_i+\delta_i^2) &= A \\
A+0+\sum \delta_i^2 &= A \\
\delta_i &= 0
\end{aligned}$$

Therefore, everywhere the two probability densities are the same. The image contains no information upon which the two classes could be distinguished, $\mathcal{F}_1(\mathbf{x}) \equiv 0$ for all p_1.

Checking the conditions for a non-trivial solution when D is a collection of probability densities

Suppose that the conditional probability density functions for two classes are $\mathrm{F}(\mathbf{x})$ and $\mathrm{G}(\mathbf{x})$. Let A_k be the (unknown) number of pixels that actually belong to C_k, so that $A_1 + A_2 = A$.

To be proven is that the condition stated by the Central Lemma is satisfied:

$$S_K = \sum_A \frac{F(\mathbf{x})}{G(\mathbf{x})} > A$$

Let $K(\mathbf{x}) = \frac{F(\mathbf{x})}{G(\mathbf{x})}$, so $S_K = \sum_A K$. Since we prefer to work in the feature space, instead of in the image space, we must take the frequencies of occurrence $n_{\mathbf{x}}$ of each $\mathbf{x}$ into account:

$$S_K = \sum_A K(\mathbf{x}) = \sum_{\mathrm{X}} n_{\mathbf{x}} K(\mathbf{x})$$

where X is the set of all possible feature vectors.

Let $n_{1\mathbf{x}}$ and $n_{2\mathbf{x}}$ be the number of pixels with feature vector $\mathbf{x}$ in C_1 and C_2, respectively. They follow from the probability density functions:

$$n_{\boldsymbol{x}} = n_{1\mathbf{x}} + n_{2\mathbf{x}} = A_1 F(\mathbf{x}) + A_2 G(\mathbf{x})$$

and, therefore,

$$S_K = \sum_{\mathrm{X}} A_1 F(\mathbf{x}) K(\mathbf{x}) + A_2 G(\mathbf{x}) K(\mathbf{x}).$$

The sum over $K(\mathbf{x})$ now becomes

$$\begin{aligned}\sum_A \frac{F(\mathbf{x})}{G(\mathbf{x})} &= \sum_X (A_1 F(\mathbf{x})\frac{F(\mathbf{x})}{G(\mathbf{x})} + A_2 G(\mathbf{x})\frac{F(\mathbf{x})}{G(\mathbf{x})}) \\ &= A_1 \sum_X \frac{F^2(\mathbf{x})}{G(\mathbf{x})} + A_2 \sum_X F(\mathbf{x}) \\ &= A_1 \sum_X \frac{F^2(\mathbf{x})}{G(\mathbf{x})} + A_2 \qquad (3.15)\end{aligned}$$

Introducing $\Delta(\mathbf{x})$ as $\Delta = G - F$ and observing that $\sum_X \Delta = \sum_X G - \sum_X F = 0$, we obtain:

$$\begin{aligned}\sum_A \frac{F(\mathbf{x})}{G(\mathbf{x})} &= A_1 \sum_X (\frac{(G+\Delta)^2}{G}) + A_2 \\ &= A_1 \sum_X (\frac{G^2 + 2G\Delta + \Delta^2}{G}) + A_2 \\ &= A_1 (\sum_X G + 2\sum_X \Delta + \sum_X \frac{\Delta^2}{G}) + A_2 \\ &= A_1 (1 + 0 + \sum_X \frac{\Delta^2}{G}) + A_2 \\ &> A_1 + A_2 = A. \qquad (3.16)\end{aligned}$$

We can prove the same for $\sum_A G/F$, and hence the two conditions in the Lemma in the previous section are satisfied.

In the limiting case of $A_1 = 0$, (3.15) reduces to $\sum_A F/G = A_2 = A$; if $A_2 = 0$, we get $\sum_A G/F = 0$. Both sums are equal to A in the special case of $F = G$.

From two to N classes

The case of N classes, instead of two, can be handled by the above lemma by taking one class at a time, say C_1, and group the other $N-1$ classes into C_2 by averaging their probability densities. Note that no assumption was made about probability density functions. As a result, we will get $2 \times N$ conditions.

Note that the limiting case of priors being equal to zero is not just a theoretical one, in particular in case of stratification. In many strata, only a subset of classes may occur.

3.2 Local prior probabilities

Prior probabilities can be a remedy against spectrally overlapping classes. If a feature vector $\mathbf{x}$ has non-zero probability density values for several classes, the pixel potentially belongs to any of these. Selection of classes on the basis of spectral characteristics ($\mathbf{x}$) only, results in a large probability of error. Proper prior probabilities help to make the guess more educated.

Many classifiers allow the use of global prior probabilities, estimated on the expected (relative) class areas. Although improvement of overall classification accuracy is achieved, it is usually, quite limited [Middelkoop and Janssen, 1991] (also see Fig. 2.3).

The previous section described an iterative method to estimate priors, which need no longer to be specified by the user. This can be considered advantageous, but also now no spectacular classification improvement is expected.

However, decisions are made for each pixel, independent of the others, and the decision for a pixel could theoretically be influenced by a vector of prior probabilities that is valid for only that pixel. Unfortunately, we do not know the correct prior probabilities for each individual pixel a priori — there would be no point in making any classification. Subdivision of the image into regions (segments or strata) according to a (GIS) context map and finding a prior probability vector for each region, yields a compromise between global and individual-pixel priors. Using this additional information, significant classification improvements can be achieved [Strahler, 1980] [Middelkoop and Janssen, 1991]. As demonstrated in the introduction of this Chapter, higher accuracies may be expected with a set of *local priors* per region than with (global) priors for the entire image.

Many sources of such spatially-distributed prior probabilities can be conceived, concerning the influence of elevations, slopes and aspects on agricultural and natural vegetation, the expansion of built-up areas around existing urban areas, deforestation around settlements, etc. The user must be able to specify a large number of prior probabilities, one for each class in each stratum [Strahler, 1980] [Middelkoop and Janssen, 1991]. If he is able to do so, a statistically sound method is obtained to integrate *expert knowledge* into classification. Otherwise, a problem remains. Collecting the necessary priors may involve much additional effort.

3.2.1 Iterative local prior probability estimation

Therefore, it becomes interesting to apply iterative class area estimation, not for the entire image at once, but for each region separately. In this way, similar classification accuracy improvements can be obtained as by using user-specified local prior probabilities, but without the need to know these priors. The class area estimates are normalized and used as prior probabilities (a different set in each region) during the next iteration.

The user obtains relative class area estimates (the final set of prior probabilites) for each region at the end of the process, instead of having to enter them at the beginning.

3.2.2 Flevo case study

A very nice example of the use of *knowledge based* local prior probabilities was elaborated by [Middelkoop and Janssen, 1991]. Their study concerned an area of 6.5 $\times$ 10.2 km^2 around the village of Biddinghuizen in the Flevopolder, the Netherlands.

The purpose was to make an image classification based on three spectral bands (3, 4 and 5) of a Thematic Mapper satellite image, recorded during the growing season in 1987. The farmers in the area were requested to indicate (in a map) which crops they were growing in 1985, 1986 and 1987. Seven predominant crops were selected: grass, potato, wheat, sugar beet, peas, beans and onions. The 1987 crop map was used to obtain training samples and to evaluate the classification results. The 1986 crop map was used to subdivide the area into strata, according to the seven classes. For each stratum a vector of prior probabilities for 1987 was made, which results in the transition matrix of Table 3.1, where for example the number 15.3 in the top row, third columns means: 15.3% of last years grass land is expected to be used for wheat this year.

Different methods were investigated by Middelkoop and Jansen to obtain such a matrix:

- By using crop rotation schemes: it turned out that three of those existed, to be used at the individual farmers discretion; in addition, two farmers using the same scheme are usually not synchronized. The information was combined into one matrix using Markov modeling.
- By overlaying the crop map of 1985 and the one of 1986, assuming that the same transitions also occur between 1986 and 1987.
- By interviewing experts, asking them to make an educated guess of the values in the matrix.

\	gr	po	wh	sb	pe	be	on
gr	2.5	0.0	15.3	82.1	0.0	0.0	0.0
po	0.2	1.3	24.1	68.8	1.6	1.5	2.2
wh	5.6	70.3	0.6	12.8	1.9	3.4	5.1
sb	1.4	31.2	6.3	0.7	5.8	27.0	27.2
pe	0.0	14.1	68.2	17.5	0.0	0.0	0.0
be	1.4	1.1	76.6	15.8	1.9	0.0	3.0
on	3.4	3.8	84.2	8.2	0.0	0.1	0.0

Table 3.1: Ideal crop rotation transition matrix, obtained by overlaying crop maps of two successive years

As expected, the results of these methods were not the same. The matrix printed in Table 3.1 is actually the reference matrix: it was obtained by overlaying the crop maps of 1986 and 1987. It was created to evaluate the other matrices and should not be used to improve the classification.

With the same data sets (Thematic Mapper bands 3, 4 and 5, and the crop maps of 1986 and 1987), I repeated the classification, using iteratively estimated instead of knowledge based prior probabilities.

The availability of the 1987 crop map was used also now as a source for training samples: 200 per class, randomly distributed over the area. k-Nearest-Neighbor with $k = 7$ was used to make the probability density maps, one per class.

The iterative process calculates for each crop in 1986 prior probabilities for the 1987 classification. These can be formatted in a transition matrix (Table 3.2), which can be compared with the one in Table 3.1.

\	gr	po	wh	sb	pe	be	on
gr	3.5	1.4	12.2	76.9	0.0	0.4	5.6
po	2.6	2.9	22.0	64.7	0.9	2.0	4.9
wh	5.6	61.2	0.7	18.3	3.4	2.5	8.3
sb	1.3	25.2	8.3	10.3	5.0	22.5	27.4
pe	2.8	12.8	63.4	18.4	0.3	0.5	1.8
be	5.1	1.4	69.4	19.4	1.2	1.2	2.3
on	7.2	3.3	79.6	8.5	0.0	0.4	1.0

Table 3.2: Crop rotation transition matrix, estimated by iterative calculation of prior probabilities, differentiated according to previous years' crop map

Using these priors for a maximum likelihood classification and comparing the results with the 1987 crop map yields an overall accuracy of 82.1 %, compared to 76.3 % with equal prior probabilities. Middelkoop and Jansen, who were using Gaussian maximum likelihood and a larger number of training samples, obtained with their different transition matrices overall accuracies between 79.6 and 81.9 % (and 76.0 % when they used equal prior probabilities).

All those figures seem quite disappointing. One must take into account that a lot of misclassifications occur at the boundaries of the fields, partly because of the mixed pixel effect, but mostly because it is impossible to align the map exactly (with an accuracy of, say, less than half a pixel) with the imagery.

Alternatively, I compared classification and crop map using only those pixels that are not adjacent to a field boundary. In that case, the overall accuracy using equal priors is 91.5 %; iteratively estimated priors give 94.7 % (Middelkoop and Jansen did not make such a comparison.)

To test the method under less favorable spectral circumstances, the experiment was

repeated using only band 4 of the Thematic Mapper image.

The overall classification accuracy increases from 57.0 % (with equal priors) to 78.2 %, using estimated priors. I emphasize that prior estimation and subsequent maximum *a posteriori* probability class selection were both based on a single band. The estimated priors, therefore, are different and can be compared to those obtained before (3.3). Also Middelkoop and Jansen classified on band 4 only and obtained overall accuracies increasing from 61.4 (equal priors) up to between 74.3 and 80.9 %, using different transition matrices.

\	gr	po	wh	sb	pe	be	on
gr	6.0	14.3	5.0	72.8	0.0	1.7	0.3
po	5.7	6.8	16.6	64.0	1.3	4.2	1.3
wh	3.0	76.6	0.3	8.0	4.8	2.0	5.4
sb	5.9	23.4	11.6	8.0	6.3	17.0	27.8
pe	1.0	8.0	66.8	15.9	0.9	4.9	2.5
be	0.8	1.5	71.2	13.7	1.4	10.3	1.2
on	0.9	0.6	88.8	5.0	0.2	4.3	0.2

Table 3.3: Crop rotation transition matrix, estimated by iterative calculation of prior probabilities, differentiated according to previous years' crop map, based on a single band image

3.2.3 Twente case study

The area in the eastern part of the province of Overijssel in the Netherlands, the so-called Twente region, containing the cities of Enschede and Hengelo, is predominantly rural, and covered with grassland, agricultural crops (mainly maize), woods and heather. The image area of approximately 26 by 22 km^2 also contains residential and industrial areas. Landsat TM imagery of 1992 was available (Fig. 3.4) and a classification was carried out using six classes: *city*, *residential* (suburbs and villages), *industrial*, *agriculture* (including grasslands), *natural vegetation* (forest and heather) and *water*.

The class-selection was according to land use. For example, there is only one class *agriculture* , without differentiation according to crops - also in practical cases such differentiation is often difficult to obtain, and perhaps not even required by the application. The class is spectrally heterogeneous. On the other hand, the classes *city*, *residential* and *industrial* are not only heterogeneous, but also have a large spectral overlap, whereas to distinguish them may be a user requirement.

The main purpose was to test the iterative classification method in a controlled experimental setup.

Training samples were selected using area frame sampling (AFS) [Cochran, 1977],

Figure 3.4: TM image of Twente, Netherlands, R: band 4, G: band 5, B: band 3.

which found wide acceptance in agricultural statistics and remote sensing [Meyer-Roux, 1987], [Gallego, 1995].

The area was subdivided in blocks of 10 km $\times$ 10 km. A sampling density of 3 % was chosen and realized using three segments of one square kilometer in each block. The location of the segments with respect to a block is chosen at random, and subsequently applied to each block. From the 18 segments thus identified, 16 were digitized from 1:25000 topographic maps, published in 1992. The remaining two segments are located in Germany, outside the study area. The results were converted into training samples, while eliminating pixels that were found suspicious, according to our knowledge of the terrain and inspection of the feature space (Fig. 3.5 and Table 2.2).

Between map survey and image recording, some land use changes may have taken place.

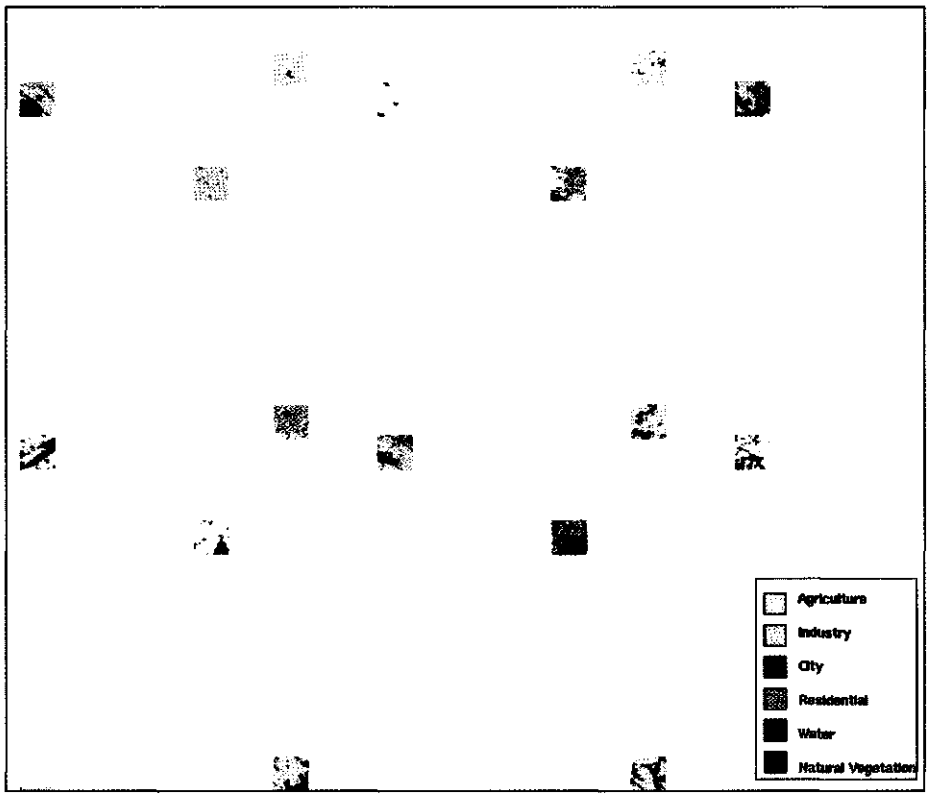

Figure 3.5: Training Samples

Following the same procedure, another set of samples was collected to evaluate the classification results. The training set and the evaluation set have no overlap in the terrain.

The AFS method provides representative sample sets, which is a prerequisite for reliable estimates of conditional probability densities.

From a straightforward Gaussian maximum likelihood classification, not too much was expected: the classes *city, residential* and *industrial* are very inhomogeneous and largely overlap each other. *Agriculture* poses another problem, by being a multi-modal class, which is also the case for *natural vegetation*. From the error matrix of a maximum likelihood classification we conclude, however, that these classes behaved well (Table 3.4).

Classification somewhat improved using non-parametric estimation of probability

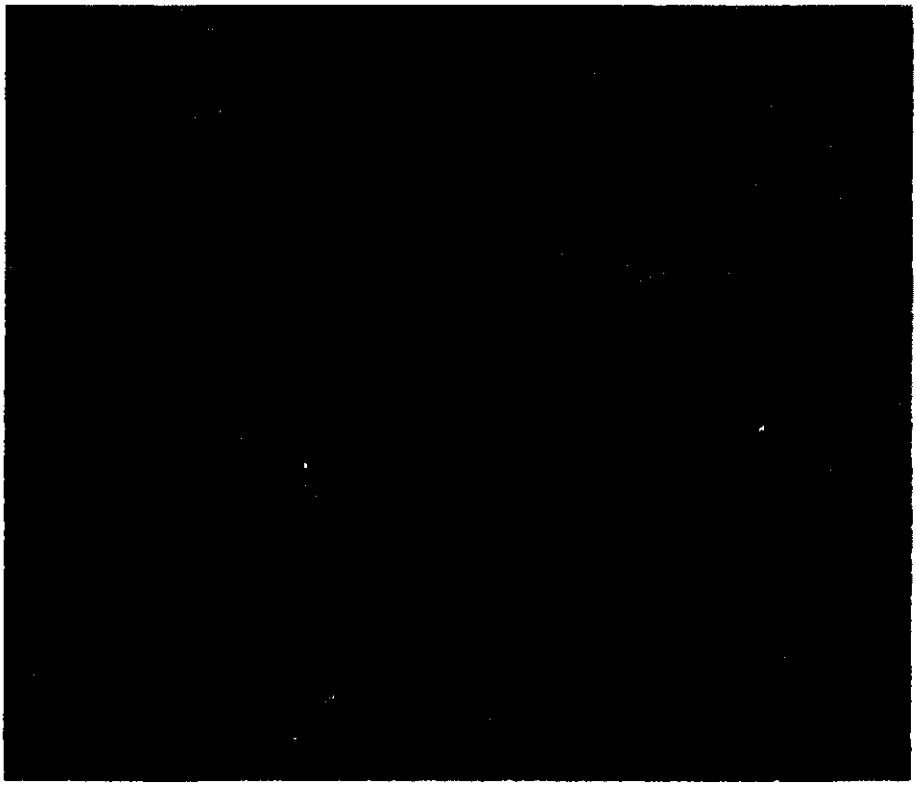

Figure 3.4: TM image of Twente, Netherlands, R: band 4, G: band 5, B: band 3.

which found wide acceptance in agricultural statistics and remote sensing [Meyer-Roux, 1987], [Gallego, 1995].

The area was subdivided in blocks of 10 km $\times$ 10 km. A sampling density of 3 % was chosen and realized using three segments of one square kilometer in each block. The location of the segments with respect to a block is chosen at random, and subsequently applied to each block. From the 18 segments thus identified, 16 were digitized from 1:25000 topographic maps, published in 1992. The remaining two segments are located in Germany, outside the study area. The results were converted into training samples, while eliminating pixels that were found suspicious, according to our knowledge of the terrain and inspection of the feature space (Fig. 3.5 and Table 2.2).

Between map survey and image recording, some land use changes may have taken place.

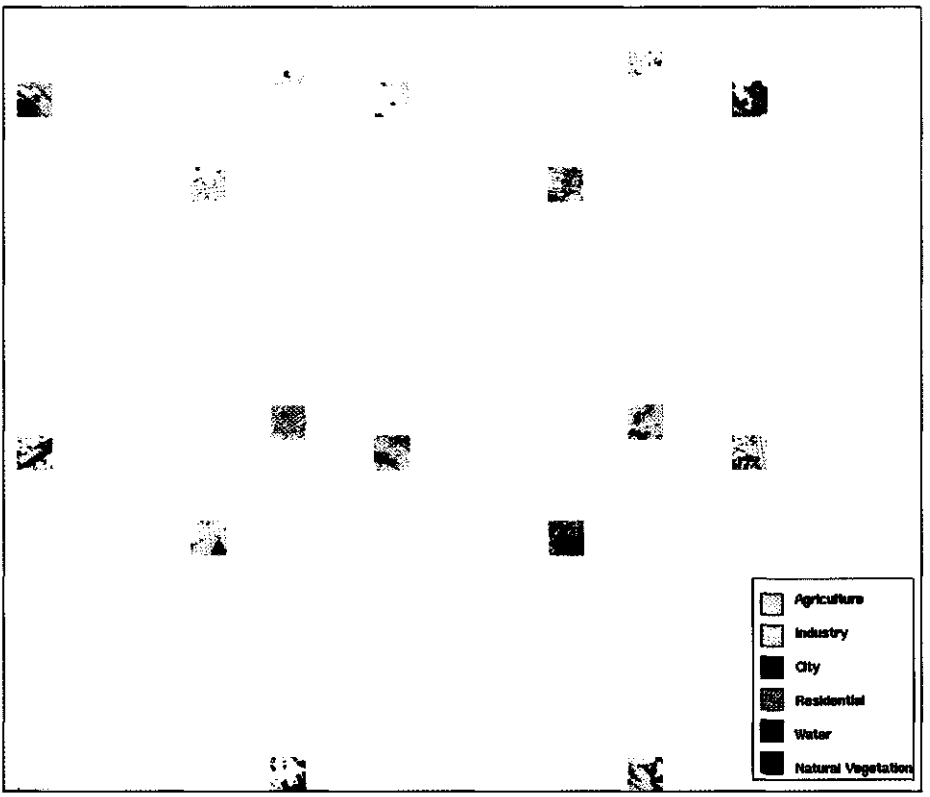

Figure 3.5: Training Samples

Following the same procedure, another set of samples was collected to evaluate the classification results. The training set and the evaluation set have no overlap in the terrain.

The AFS method provides representative sample sets, which is a prerequisite for reliable estimates of conditional probability densities.

From a straightforward Gaussian maximum likelihood classification, not too much was expected: the classes *city*, *residential* and *industrial* are very inhomogeneous and largely overlap each other. *Agriculture* poses another problem, by being a multi-modal class, which is also the case for *natural vegetation*. From the error matrix of a maximum likelihood classification we conclude, however, that these classes behaved well (Table 3.4).

Classification somewhat improved using non-parametric estimation of probability densities, without applying prior probabilities. We used kNN with $k = 13$, followed

	agr	ind	city	res	water	nat	uncl	ACC
agr	8939	424	11	507	54	883	0	0.83
ind	1	487	310	40	0	4	0	0.58
city	0	5	191	6	0	0	0	0.95
res	42	295	555	1817	0	117	0	0.64
water	0	20	3	0	89	0	0	0.79
nat	107	41	10	184	18	2264	0	0.86
REL	0.98	0.38	0.18	0.71	0.55	0.69		

average accuracy = 77.51 %
average reliability = 58.34 %

overall accuracy = 79.13 %

Table 3.4: Error matrix of Gaussian ML classification, using classes Agriculture, Industrial area, City (center) area, Residential area (suburbs and villages), Water and Natural vegetation. The rows of the matrix refer to ground truth, according to an evaluation set. The columns refer to the result of the classification, which, in this case, contains no unclassified pixels (uncl). The column ACC contains the class accuracies: the fraction of evaluation set pixels that was classified correctly. The row REL indicates the reliability per class, or the fraction of pixels of a certain class in the classification result, which indeed belongs to that class according to the evaluation set

by compensation for the different numbers of training samples per class, as described in section 2.4. The overall accuracy increased from 79.1 % to 82.0 %. (Table 3.5).

	agr	ind	city	res	water	nat	uncl	ACC
agr	9229	286	33	628	54	588	0	0.85
ind	19	533	239	31	20	0	0	0.63
city	0	28	167	6	1	0	0	0.83
res	45	434	320	1933	1	93	0	0.68
water	0	3	0	0	109	0	0	0.97
nat	33	40	17	205	16	2313	0	0.88
REL	0.99	0.40	0.22	0.69	0.54	0.77		

average accuracy = 80.86 %
average reliability = 60.20 %

overall accuracy = 81.98 %

Table 3.5: Error matrix of non-parametric classification with equal prior probabilities

In the context of our proposed method, additional information was used to perform

a stratification of the area in the required sense. We used the Dutch *Postal district* map for this purpose. The Netherlands have been subdivided into postal districts, each one having a 4-digit area code. The districts usually coincide with distinct regions, such as industrial zones and town quarters. Therefore, different class area distributions can be expected in each postal district. They can be estimated by applying the iterative process to each district separately. The image area contains 66 districts. (Fig. 3.6).

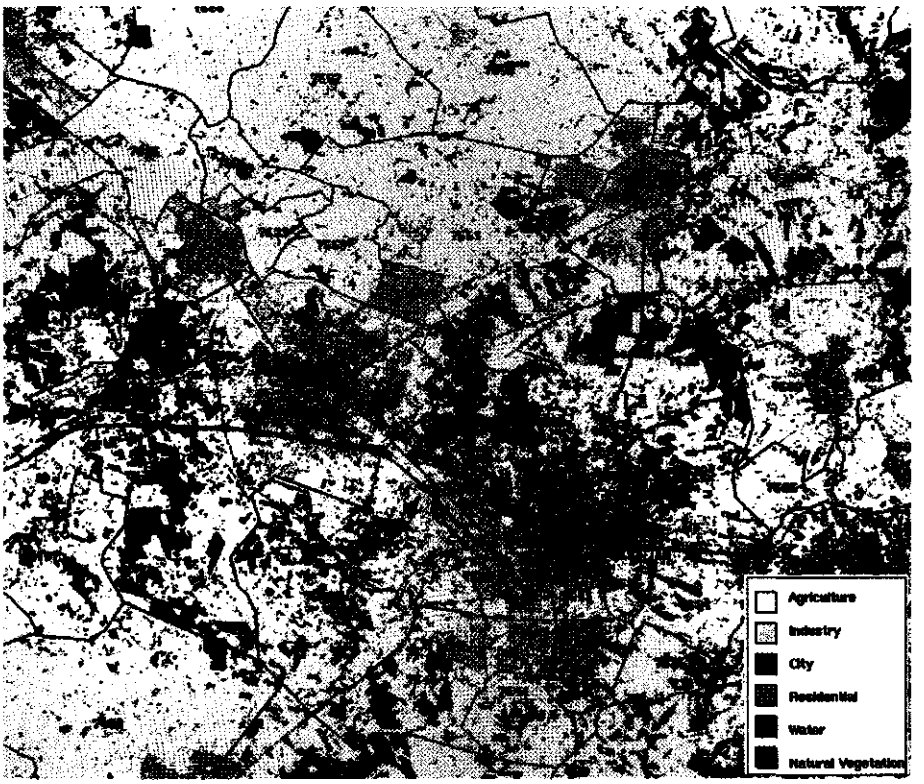

Figure 3.6: Final classification, overlaid with post-code area boundaries.

While performing the iterative process, the (relative) areas covered by the six classes are calculated in each postal district. The result after 10 iterations is presented in Table 3.6, which shows for each district the relative class areas as percentages.

Convergence has been reached in the sense that during the 10th iteration none of the probabilities changed more than 0.05 %.

Using the data in Table 3.6 as prior probabilities, a maximum *a posteriori* probabil-

District	agric.	industry	city	resid.	water	nat.veg.	nr. pixels
7481	64.8	0.1	0.0	5.7	0.0	29.3	12668
7482	86.9	0.2	0.0	6.6	0.0	6.3	32416
7491	36.5	2.2	1.0	38.0	0.1	22.3	6651
7495	59.0	0.1	0.0	3.4	0.2	37.4	35874
7496	96.3	0.0	0.0	1.7	0.0	2.0	933
7497	79.1	0.1	0.0	5.4	0.0	15.3	12183
7511	1.1	12.0	63.7	22.5	0.0	0.7	1240
7512	1.0	23.4	54.0	20.7	0.0	0.9	1073
7513	4.6	15.3	35.8	37.8	0.1	6.4	1174
7514	0.9	13.0	53.7	29.9	0.0	2.5	1253
7521	21.2	9.3	10.5	50.3	0.0	8.7	3236
7522	27.3	0.5	1.3	9.4	1.6	60.0	13791
7523	5.8	6.1	26.7	55.2	0.0	6.2	2302
7524	53.5	0.9	0.3	9.1	1.2	35.1	16204
7525	55.9	0.0	0.0	1.3	0.0	42.8	9976
7531	23.2	3.7	4.4	36.9	0.1	31.7	3948
7532	56.9	2.4	0.4	11.6	0.0	28.7	12923
7533	1.8	7.9	23.8	62.3	0.0	4.3	817
7534	69.0	0.7	0.7	13.6	0.0	16.0	8996
7535	49.1	0.5	4.5	29.0	0.0	16.9	2792
7536	79.5	0.0	0.1	2.1	0.0	18.3	4900
7541	37.7	11.9	3.2	44.3	0.2	2.8	2242
7542	41.0	0.6	0.1	28.1	0.3	30.0	6205
7543	36.0	9.4	6.2	42.3	0.6	5.5	2194
7544	59.5	1.4	0.2	23.4	0.1	15.4	10632
7545	38.5	10.1	7.7	39.9	0.0	3.8	4801
7546	57.6	2.2	0.2	11.3	2.6	26.1	13555
7547	49.0	11.9	2.4	12.2	2.4	22.1	13354
7548	68.9	0.7	0.3	6.0	0.2	23.9	20300
7551	0.0	3.0	73.9	22.7	0.0	0.4	757
7552	35.3	1.0	3.2	42.9	1.1	16.5	6966
7553	4.5	27.9	20.2	44.7	1.3	1.4	3209
7554	59.3	3.4	0.6	5.7	0.6	30.5	28773
7555	49.1	1.9	0.7	22.7	0.1	25.6	11549
7556	39.8	12.9	4.1	34.6	0.6	7.9	6665
7557	26.4	9.7	8.5	53.5	0.0	1.9	3188
7558	13.3	3.0	0.2	77.6	0.0	6.0	3044
7559	88.6	1.9	0.0	3.1	1.5	4.9	7226
7561	73.1	0.8	0.1	4.2	0.6	21.2	30727
7571	0.2	8.4	40.8	48.8	0.0	1.8	466
7572	37.7	5.1	2.0	52.2	0.0	3.0	2345
7573	11.0	7.2	0.7	59.8	0.0	21.3	1573
7574	37.4	3.2	0.3	42.1	0.1	17.0	1990
7575	43.9	24.4	3.3	19.4	0.2	8.8	3789
7576	29.9	0.7	0.4	43.3	4.5	21.2	3436
7577	52.6	4.3	0.1	36.1	0.0	7.0	3285
7581	66.4	1.9	0.4	15.1	1.0	15.2	13412
7582	69.8	0.8	0.1	16.7	0.0	12.6	9600
7585	89.1	0.3	0.0	5.5	0.0	5.2	4697
7586	77.1	0.3	0.0	15.4	0.0	7.3	7277
7587	65.6	0.7	0.0	2.5	0.0	31.3	42993
7588	72.2	0.1	0.0	1.5	0.0	26.2	12756
7595	89.5	0.2	0.0	4.4	0.0	5.9	20653
7596	81.4	0.2	0.0	2.7	0.0	15.7	13177
7597	89.2	0.1	0.0	1.9	0.0	8.8	20760
7601	41.9	13.0	2.1	39.7	0.0	3.3	2802
7606	0.0	3.7	4.9	91.4	0.0	0.0	82
7607	68.6	1.3	1.8	11.8	0.0	16.4	7755
7609	26.9	0.1	0.0	6.5	0.1	66.4	1440
7621	48.1	0.8	0.1	39.1	0.2	11.7	4810
7622	21.0	10.1	3.3	62.0	0.0	3.6	2769
7623	78.8	2.7	0.1	12.8	0.1	5.5	4506
7625	90.0	0.2	0.0	4.3	0.0	5.5	11010
7626	88.0	0.1	0.0	2.8	0.0	9.1	5454
7627	76.3	0.1	0.0	5.3	0.0	18.3	6562
7665	94.7	0.1	0.0	0.8	0.0	4.4	10971

Table 3.6: Relative class areas per postal district

ity classification was performed, resulting in an improvement of the overall accuracy from 82.0 % to 91.3 % (Table 3.7).

Note that, in contrast to [Strahler, 1980] and [Middelkoop and Janssen, 1991], no semantic information about the stratification map is needed. The correspondence between postal districts and thematic classes is established by the iterative process.

	agr	ind	city	res	water	nat	uncl	ACC
agr	10214	57	4	116	50	377	0	0.94
ind	31	626	118	62	3	2	0	0.74
city	0	4	185	12	0	1	0	0.92
res	102	129	93	2387	0	115	0	0.84
water	0	12	0	0	97	3	0	0.87
nat	109	7	0	92	8	2408	0	0.92
REL	0.98	0.75	0.46	0.89	0.61	0.83		

average accuracy = 87.20 %
average reliability = 75.43 %

overall accuracy = 91.35 %

Table 3.7: Error matrix of non-parametric classification with iteratively calculated, spatially distributed prior probabilities

Some doubts remain concerning the final classification. Some of the districts are dominated by a single class, which will consequently be over-represented, at the expense of classes with very small prior probabilities. For example, a few small villages in rural areas almost disappeared, although this may also be caused by the wider spacing between buildings in small villages, when compared to urban areas.

A possible solution is to modify the class selection stage, such that it distributes class labels in accordance with the calculated area estimates, over each postal district. This modification involves sorting the *a posteriori* probabilities in the area. However, such a modification is not straightforward and it is not certain that it will lead to an improvement of accuracies and reliabilities, because it is not maximum *a posteriori* probability classification any more.

3.3 Local probability densities

First, consider a constructed example, involving an image of 400 pixels (Fig. 3.7). The image is divided in 2 regions, s_1 and s_2, covering 153 and 247 pixels respectively. There are 26 pixels in the entire image with feature vector $\mathbf{x}$. Of those, 14 are in s_1 and 12 in s_2. Furthermore, it is assumed that 207 pixels belong to class C_i: of these, 67 C_i pixels are in s_1 and 140 in $C2$.

The image pixels in Fig. 3.7 have been rearranged such that all pixels with feature vector $\mathbf{x}$ occur in one horizontal strip, region s_1 is entirely located to the left of s_2, and the pixels that belong to C_i are located in the center of the image — this rearrangement does not cause any loss of generality.

By counting pixels, various probabilities can be estimated, applying to either the entire image, or to s_1 and s_2 separately. These probabilities are summarized in Table 3.8.

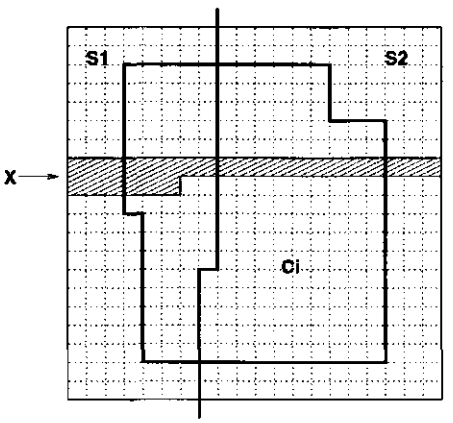

Figure 3.7: Synthetic image with two regions

	Image	s_1	s_2
$\mathrm{P}(\mathbf{x})$	$\frac{26}{400}$	$\frac{14}{153}$	$\frac{12}{247}$
$\mathrm{P}(C_i)$	$\frac{207}{400}$	$\frac{67}{153}$	$\frac{140}{247}$
$\mathrm{P}(\mathbf{x}\|C_i)$	$\frac{17}{207}$	$\frac{8}{67}$	$\frac{9}{140}$
$\mathrm{P}(C_i\|\mathbf{x})$	$\frac{17}{26}$	$\frac{8}{14}$	$\frac{9}{12}$

Table 3.8: Probabilities in synthetic two-region image

In each of the three cases (entire image, region s_1 and region s_2), the probabilities obey Bayes' formula. For example, for s_1 we obtain by substitution in

$$\mathrm{P}(C_i|\mathbf{x}) = \frac{\mathrm{P}(\mathbf{x}|C_i)\ \mathrm{P}(C_i)}{\mathrm{P}(\mathbf{x})}$$

that

$$\frac{8}{14} = \frac{\frac{8}{67}\ \frac{67}{153}}{\frac{14}{153}}.$$

The posterior probabilities $\mathrm{P}(C_i|\mathbf{x})$ are different in the three cases, as are the prior probabilities $\mathrm{P}(C_i)$. A new observation is that there are also three different values for the probability densities $\mathrm{P}(\mathbf{x}|C_i)$ of class C_i. Therefore, probability density estimation of image feature vectors has to be refined, which leads to the distinction between global and local probability density models.

The conclusion is trivial and well-known, but often disregarded: Bayes formula can only be applied if the four probabilities involved ($\mathrm{P}(C_i|\mathbf{x}), \mathrm{P}(\mathbf{x}|C_i), \mathrm{P}(C_i)$ and $\mathrm{P}(\mathbf{x})$) concern the same population of pixels.

3.3.1 Global Probability Density Model

The previous sections used a global probability density model, assuming that class probability densities for the feature vector $\mathbf{x}$ of a given pixel, as derived from the

training samples in the feature space, are valid for all pixels with feature vector $\mathbf{x}$ in the entire image and do not depend on the position of the pixel. A non-parametric estimate for the (global) class probability density $\mathrm{P}(\mathbf{x}|C_i)$ was given in eq. (2.12) as

$$\mathrm{P}(\mathbf{x}|C_i) = \frac{k_i}{N_i v_{\mathbf{x}}}$$

Because $v_{\mathbf{x}}$, the size of the volume in the feature space around $\mathbf{x}$ that contains the k nearest training samples, is class-independent,

$$\mathrm{P}(\mathbf{x}|C_i) \sim \frac{k_i}{N_i}.$$

Note that $\sum \mathrm{P}(\mathbf{x}|C_i) \neq 1$, so normalization is not allowed.

When the area is subdivided into regions, the global probability density model assumes that class reflection characteristics, modeled by probability densities, are region-independent. Three details are to be noted, especially when iterative area estimation is applied:

- If most classes are present in a region, probably the region has so much spectral variation that each class shows heterogeneity similar to its training samples.
- Many regions contain only a limited subset of the total number of classes. For those classes that do not appear and have low spectral overlap with the present classes, we expect that the probability densities are small enough to lead to the 'limiting case' in the central lemma in Chapter 2. Then the area of those classes will rightly be estimated 0. For the classed that are present, the global probability density assumption becomes important. When it is not valid, *i.e.* when a class in a region only partly covers the cluster in the feature space of the training samples, errors in prior probability estimates are to be expected.
- If only one class is present in a region, it is generally irrelevant that the local spectral variation of that class (in the region) is smaller than the global one (over the entire training set). Although probability density estimates are probably incorrect, the class will still dominate the others and force them to the trivial solution, mentioned in the central lemma.

The examples in the previous sections fit these requirements. In the crop rotation case, it is expected that the spectral signature of a crop does not depend on which crop was grown last year. In the land-use classification using postal districts it may occur that certain regions are dominated by one class, while others are absent.

For certain situations, however, the global probability density model appeared to be not strict enough. In general, this is the case when regions are small (for example, individual agricultural fields) and/or spectrally homogeneous, and are in a feature space area with spectral overlap between classes. This situation will often occur

when the regionalization is done automatically by *image segmentation* (Chapter 4), where segments are formed on the basis of spectral homogeneity. Therefore, the spectral variation in almost every segment is much smaller than that of any class in the training samples. Since the method is supposed to estimate class areas in mixed segments, as well as in segments that cover parts of the feature space with spectral overlap, inadequately modeled classes will frequently occur and cause poor area estimates.

Consider, for example, an image from a digital color (RGB) camera, where pixels are displayed with (approximately) the colors of the corresponding terrain elements. Most of the grass pixels are green, but some are yellow. If $\mathbf{x}_y$ is a feature vector in the yellow part of the RGB feature space, the global probability density $\mathrm{P}(\mathbf{x}_y|grass)$ is small.

Image segmentation creates segments of adjacent pixels with similar colors. Segments containing pixels with feature vector $\mathbf{x}_y$ correspond to yellow fields in the terrain. If s is one of these segments and if it contains grass pixels, the probability that a grass pixel in s is yellow equals 1. The local probability density that the feature vector of a grass pixel in this segment equals $\mathbf{x}$ is significantly larger than the global one. This probability does not depend on the presence or the amount of green grass elsewhere in the image. Therefore, to establish the probability, only the yellow grass pixels in the training set are taken into account.

For this reason, a local probability density model was developed.

3.3.2 Local Probability Density Model

The local probability density model aims at estimating class probability densities $\mathrm{P}(\mathbf{x}_p|C_i)$ for pixels p with feature vectors $\mathbf{x}_p$ in a particular image region. It will be derived from the global model. In chapter 2, using the modified k-Nearest Neighbor (k-NN) estimator, $\mathrm{P}(\mathbf{x}_p|C_i)$ was shown to be proportional to the number k_i of class C_i samples in a neighborhood with k samples around x_p, divided by the total number N_i of training samples for class C_i.

Training sample selection

Training samples are usually, but not necessarily, taken inside the image to be classified. When estimating probability densities, pixel positions in the image do not play a role. The training samples could be taken from another image as well, as long as a number of circumstances (atmospheric conditions, sun angle etc) during the acquisition of both images were the same.

Similarly, we are allowed to apply the estimation procedure to a single region, on the basis of training samples that were taken elsewhere in the (same) image. We will treat the entire collection of training samples as a "pool", from which we only use those that are required to classify the region.

Non-parametric estimation

Applying k-NN to a region s, we find for each pixel p those k training samples that are nearest to p's feature vector $\mathbf{x}_p$ in the feature space. They form a set $T^p \in T$ with k elements. This set is split into subsets T_i^p, such that

$$k_i = A(T_i^p) \text{ and } k = A(T^p).$$

The union of the sets T_i^p over the pixels in s gives a set T_i^s for each class:

$$T_i^s = \bigcup_{p \in s} T_i^p,$$

denoting the C_i training samples involved in the classification of s. We define A_i^s as

$$A_i^s = A(T_i^s),$$

the number of training samples for each class, relevant for the classification of s. These are precisely the training samples that we are looking for: the yellow *grass* samples in the example.

For some classes N_i^s may be 0, which means that none of the pixels in s has a class C_i sample in the neighborhood of its feature vector. Then, $\mathrm{P}(\mathbf{x}_p|C_i) = 0$ for all pixels p in s. For the other classes, we estimate for each pixel:

$$\mathrm{P}(\mathbf{x}_p|C_i) \sim \frac{k_i}{A_i^s}. \tag{3.17}$$

Semi-parametric estimation

Although the emphasis in this thesis, as far as classification is concerned, is on non-parametric (k-NN) methods, a semi-parametric local probability density estimator was also implemented. Here, subsets T_i^s of the training data are selected as above, consisting of those samples that *would be* involved in a k-NN classification of a region. Subsequently, class mean vectors $\mathbf{m}_i$ and covariance matrices V_i are based on those subsets and used to calculate Euclidean or Mahalanobis distances or Gaussian class probability densities. Since each segment has its own covariance matrix, determinants and the inverse matrices are calculated for each of them, using the method in [Press *et al*, 1992] and stored in the segment table. Gaussian densities can be used for iterative local prior probability estimation (Table 3.11).

3.3.3 Completely homogeneous regions

Returning to the example of the yellow field in the image, in the limiting case of a completely homogeneous region, where all feature vectors equal $\mathbf{x}$, the local probability density $\mathrm{P}(\mathbf{x}|grass)$ equals 1. If, according to the training data, also yellow

wheat exists, the local probability density $P(\mathbf{x}|wheat)$ is also 1. Moreover, inside the region, the unconditional probability $P(\mathbf{x}) = 1$. Then, according to Bayes formula, the *a posteriori* probabilities are equal to the priors. If additional knowledge, *i.e.* prior probabilities, suggest that this was a *grass* field, then it is probably still like that. The classifier cannot contribute to the information about this yellow field.

In case the priors are iteratively estimated, the prior and posterior probabilities for both classes in this homogeneous region will be equal to $\frac{1}{2}$.

3.3.4 Comparison with stratified classification

A well-known method to improve classifications is *stratified classification* [Meyer-Roux, 1987]. First, the area is subdivided in n strata s_a, $(a \in [1..n])$, on the basis of additional data or image interpretation, to obtain in those strata less spectral variability than in the entire image, thereby reducing spectral overlap. Certain classes may be not present in all strata. Next, the n strata are classified independently, using n training sets, possibly with different sets of classes, and n prior probability vectors, either specified by the user or iteratively estimated. Finally, the results, which are spatially disjoint, are combined into a single classified map.

An obvious drawback of this method is that, depending on the number of strata, very many training samples are needed from all over the area. Requirements concerning numbers of training samples must be met in each stratum. Assuming the availability of these samples, let T_i^a be the collection of training samples for class C_i in stratum s_a and consider stratified classification *vs.* local probability density estimation in s_a:

- If T_i^a has no overlap with any T_i^b $(b \neq a)$, both methods use T_i^a and produce the same estimates. If T_i^a has overlap with some T_i^b $(a \neq b)$, but not with other T_i^c $(a \neq b \neq c)$, stratified classification uses only T_i^a, whereas for local probability density estimation T_i^b is also available, while T_i^c is not considered. If all T_i^b $(b \in [1..n])$ overlap, the area subdivision did not help for class C_i — it looks the same everywhere in the image. Again, local probability density estimation benefits from the larger number of available samples.
- If T_j^a overlaps T_i^b, but not T_i^a, stratified classification is successful for both C_i and C_j, whereas local probability density estimation for C_j in s_a is erroneously based on T_i^b.

Combining both observations, the local probability density method, compared to stratified classification, may give better estimates for C_i, but worse ones for C_j. How this affects the final results needs further investigation. Local probability density estimation was primarily developed for 'automatic stratifications' from image segmentation (Chapter 4), where very many strata (segments) are created, for which no separate training sample sets can be collected.

3.3.5 Implementation

k-Nearest Neighbor classification is expensive. It requires a large memory for storing feature space and training sample data and searching neighbors in the feature space takes considerable time. Especially for local probability density estimations, the algorithm needs to be carefully designed.

According to equation 3.17, the local probability $P(\mathbf{x}|C_i)$ in a segment s of a segmentation is proportional to $\frac{k_i}{A_i^s}$, where k_i is the number of class C_i samples among the k nearest neighbors of $\mathbf{x}$ and A_i^s is the total number of class C_i samples involved in the classification of segment s.

The algorithm consist of four steps:

1. Collecting feature space and training sample data
2. Nearest neighbor search
3. Calculation of A_i^s
4. Calculation of $\frac{k_i}{A_i^s}$ and normalization

The number of different feature vectors in an image is usually much smaller than the number of pixels. The window of the Ameland SPOT image used in Chapter 5, for example, consists of $460 \times 785 = 361100$ pixels, but there are only 10494 different feature vectors. If neighbors are searched only once per feature vector, and only for feature vectors that actually occur, the number of searches is drastically reduced.

To be able to process each feature vector only once, all occurring feature vectors have to be stored, as well as the associated statistics, such as class probability densities. A hash table technique, which provides fast random access to data records with a sparse key, is suitable, since the feature vectors are a (composite) key from a domain of $256^3 = 16,777,216$ elements. Also training data is stored in this data structure.

A hash table is an array of pointers to data records (Fig. 3.8). The array must be somewhat larger than the maximum expected number of records. A hash function maps a key into an index in the hash table, where a pointer to the actual data is found. An overflow mechanism takes care of the inevitable situations that different key values are mapped on the same index. The hash function should distribute the occurring key values as uniformly as possible over the index range, which is not trivial, since the data are not yet known when the hash function is designed [Date, 1981].

In [Mather, 1987], hashing is used to speed up maximum likelihood classification, by classifying occurring feature vectors instead of image pixels. The benefit is even larger for k-NN, where the feature space is accessed very many times when searching for neighboring training samples of feature vectors (Table 3.9).

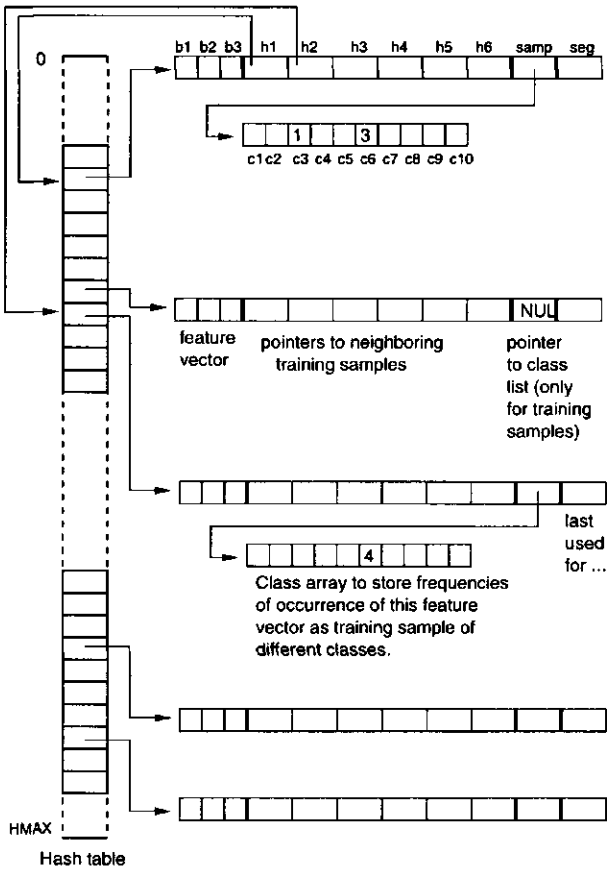

Figure 3.8: Hash table data structure for kNN local probability density algorithm

Collecting feature space and training sample data

First, the algorithm makes a pass through the image and the map of training samples simultaneously to evaluate which feature vectors are present in the image, and which of those feature vectors occur as training samples, with their associated classes. Pixel by pixel, each feature vector is looked up in the list and new ones are inserted. If a pixel is used as training sample, a class array is added to the feature vector, if it did not yet exist. Different training sample pixels, from one or several classes, may have the same feature vector (Eq. 2.1). In the Ameland case (see Chapter 5), there are 1744 pixels in the training set, but these pixels have only 849 distinct feature vectors. When a feature vector occurs in the training set again, the class array is updated.

Nearest neighbor search

In a second step, the program scans through the feature space data structure to find for all feature vectors that are present k neighboring training samples, by inspecting the feature space around the feature vector. Feature space cells are visited in order of increasing Euclidean distance around the current feature vector, until k neighbors are found or a maximum distance (search radius) R is reached. For each feature vector, a list of k pointers to training samples is created, To avoid calculation of countless distances, a list with index offsets with respect to a current feature vector is created once, when the program starts. Traversal of this list gives feature vectors to be inspected, relative to the current one, in increasing distance order.

The average required number of feature space accesses depends on several factors, such as the number of training samples and the ranges of pixels values in the different bands. If the density of training samples in the feature space is low, the search radius R determines how many feature space cells have to be examined (Table 3.9).

R	3 bands	4 bands
1	7	9
2	33	89
3	123	425
4	257	1281
5	515	3121
6	925	6577
7	1419	11833
8	2109	20185
9	3071	32633
10	4169	49689
11	5575	72465
12	7153	102353
13	9171	140945
14	11513	190121
15	14147	250553
16	17077	323721
17	20479	411913
18	24405	519025
19	28671	643441
20	33401	789905

Table 3.9: Number of neighbors within search radius R in feature spaces with 3 or 4 dimensions.

Calculation of A_i^s

The third step, traverses the sorted pixel table, which was created in the previous paragraph. Thus, all image pixels are visited in a segment by segment order. The purpose is to create a list with one row for each segment and one columns per

class, showing how many samples of each class are involved in the classification of a segment (Table 3.10). These are the A_i^s values in 3.17.

For each pixel in the table, the feature vector is retrieved from the image. Via the hash table the list of nearest training samples is accessed, and each training sample is checked for whether it was already used in the current segment. If not, the training sample counts are updated and the the training sample is marked "used".

	Class								
Segment	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	30
2	0	12	202	0	2	13	8	0	202
3	0	0	43	192	0	1	0	0	80
4	0	0	2	0	0	0	0	0	21
5	0	0	0	0	0	0	0	0	22
6	223	202	64	204	194	196	212	109	202
7	0	0	12	0	0	0	0	0	21
8	0	85	58	0	0	1	14	0	54
9	0	21	13	0	0	0	0	0	0
10	0	0	11	0	0	0	0	0	32
11	0	0	0	0	0	0	0	0	31
12	0	0	0	0	0	0	0	0	118
13	0	32	70	0	5	19	13	0	85
14	0	0	6	0	0	0	0	0	38

Table 3.10: A_i^s: Number of training samples per class, involved in the classification of each segment at the highest segmentation level.

Calculation of $\frac{k_i}{A_i^s}$

The fourth and final step traverses the image once more, now in the usual line by line order. It takes the feature vector of each pixel from the image and the segment number from the rasterized segmentation quadtree. Via the hash table and the training sample pointers the k_i values are retrieved. The A_i^s values are fetched from the table that contains the number of samples of each class in each segment. Hence, for each pixel $\frac{k_i}{A_i^s}$ is found. These values are normalized and stored in probability density maps (Figure 5.9).

Semi-parametric methods

For semi-parametric local probability density estimation, for each region s subsets T_n^s of the training data are formed for each class C_n according to a k-nearest neighbor criterion, and distribution parameters $\mathbf{m}_n^s$ and V_n^s are estimated using these subsets. The implementation closely resembles the non-parametric one. Differences

are in the third and fourth step of the algorithm. Instead of only counting the training samples per class (step 3) in s, their feature vectors are now also considered. They are looked-up in the hash table and variables s_i^s and ss_{ij}^s are maintained, where s_i denotes the sum of the i-th component over T_n^s, and ss_{ij} the sum of products of the i-th and j-th component (including ss_{ii}). At the end of a segment, the components m_i of $\mathbf{m}_n^s$ are formed by

$$m_i = \frac{s_i^s}{A_n}$$

and the elements v_{ij} of V_n^s by

$$v_{ij} = \frac{ss_{ij} - \frac{s_i s_j}{A_n}}{A_n - 1}.$$

The class mean vectors $\mathbf{m}_n^s$ are stored with s, as well as the logarithm of the determinant of the covariance matrix $\ln|V_n^s|$ and the inverted matrix $(V_n^s)^{-1}$.

In step four, the feature vectors to be classified are retrieved from the image once more, and class probability density values are calculated with the parameters of the segment containing the pixel.

3.4 Comparison

Different classifiers were applied to the Twente image, used in section 3.2.3. The results are summarized in Table 3.11. Each time the same training set was used. The figures in the table are based on a single evaluation set. Unfortunately, it was different from the one in section 3.2.3. The k-NN classifications were made with $k = 11$. Local prior probabilities (per postal district) were estimated iteratively.

Table 3.11 allows for the following observations:

- Gaussian methods are superior to minimum Euclidean distance methods (row 1 vs. 3 and 2 vs. 4). Due to differences in class variability, covariance matrices must be used and there are enough training samples to estimate them reliably.
- Straightforward k-NN classification (row 6) performs remarkably well. Proportional area frame sampling was applied (Table 2.2), such that a maximum *a posteriori* probability result is obtained.
- After compensating for differences in class training set sizes, k-NN gives maximum probability density classification (row 7) and can be compared with the equal-priors Gaussian method using (row 3).
- Local prior probability estimation improves classification significantly (row 5 vs. 4, 8 vs. 7 and 10 vs. 9).

	probability density	prob. dens model	prior probability	average accuracy	average reliability	overall accuracy
1	min. distance	global	-	70.99	47.79	65.93
2	min. distance	local	-	59.86	43.22	68.88
3	gaussian	global	-	74.59	56.55	77.84
4	gaussian	local	-	71.99	55.21	76.31
5	gaussian	local	local	81.01	68.48	86.25
6	k-NN	global	inherent	75.31	69.85	86.23
7	k-NN	global	-	76.45	59.95	79.44
8	k-NN	global	local	82.40	73.93	89.13
9	k-NN	local	-	74.68	58.83	78.46
10	k-NN	local	local	81.80	73.24	88.87

Table 3.11: Comparison of classifiers for Twente data set

- Local prior probabilities give better results than global ones (row 8 vs. 6) especially in average accuracy, where all classes have the same weight. The underestimation of small classes is less severe with local priors than with global priors Also average reliability and overall accuracy increase.
- Local probability density estimation works, but does not help in this case. There is no reason to apply it without using local priors as well. That one component in Bayes formula is unknown cannot be repaired by changing another (row 2 vs. 1, 4 vs. 3 and 9 vs. 7). Gaussian local probability densities serve the purpose to enable local prior probability estimation (row 5 vs. 4), in order to improve classification results (row 5 vs 3). Although the data do not require local probability density estimation (see section 3.3.1), the results are only slightly inferior than those from global estimation (row 10 vs. 8).
- With the available training samples, k-NN methods outperform Gaussian methods (rows 7 vs. 3, 9 vs. 4 and 10 vs. 5).

3.5 Conclusions

This Chapter explained the benefit of local statistics for classification, differentiated according to regions defined by ancillary GIS data.

Local *a priori* probabilities can be the reflection of the user's knowledge concerning class mixing proportions in different regions. However, in the absence of this knowledge, similar information can be *obtained* by an iterative class area estimation algorithm. It was shown that requirements concerning class probability densities for this method are fulfilled when training samples are representative for class populations.

The crop-rotation experiment confirms that the algorithm serves the purpose to accurately estimate class-areas and to increase classification accuracies to the same extent as with knowledge-based local prior probabilities.

A second algorithm estimates local class probability densities, which is required for homogeneous regions, which, for example, may result from image segmentation. This method can be used as an alternative for stratified classification when it is not possible to collect sufficient training data in all strata. Depending on the available training data, a choice can be made between a non-parametric and a semi-parametric local probability density estimation.

Chapter 4

Segmentation

Although image segmentation has received attention in literature since the seventies [Horovwitz and Pavlidis, 1976], it did not become widely accepted in the field of analysis of remotely sensed imagery. For example, in respected textbooks in this field, such as [Mather, 1987] and [Richards, 1993], image segmentation is not mentioned. Also the major commercial digital image processing software packages do not include image segmentation.

The abstract of "a critical survey of image analysis methods" [Pavlidis, 1986], in which image segmentation is a major issue, says:

> " A survey of the literature of the last fifteen years reveals that in spite of increased understanding of the nature of images we have been very slow in integrating the results into useful image analysis programs."

Ten years later, problems still associated with image segmentation are summarized by [Acton, 1996] as: region merging, poor boundary localization, region boundary ambiguity, region fragmentation and sensitivity to noise.

Although [Pavlidis, 1986] warns against anthropomorphic implications, image segmentation is intuitively appealing. Human image vision generally tends to divide the image into homogeneous areas first, and characterize those areas more carefully later. Applying this approach to digital image analysis software leads to a segmentation step, which divides the image into segments that correspond — in the ideal case — to meaningful objects in the terrain, followed by a supervised classification step, in which each segment is compared with class characteristics that are derived from training data. In contrast to usual classification methods, the comparison does not have to be limited to spectral properties, but can also take spatial characteristics of segments (size, shape and adjacency to other segments) into account.

The remainder of this Chapter focuses on quadtree based segmentation. The advantages of quadtrees in the context of segmentation will be elaborated upon later. The success of any segmentation algorithm depends on the availability of

- High resolution imagery, such that relevant objects are represented by a significant number of pixels; otherwise there is no point in segmentation.
- Powerful hardware: fast and with a large memory capacity.
- An efficient implementation, regarding the sizes of remote sensing images.

Since the first two requirements are increasingly being fulfilled, it is worthwhile to focus on the third and try to re-introduce image segmentation in earth observation image analysis.

A special case, which is typical for earth observation applications, is multi-band imagery. Grey-scale segmentations ([Morris *et al.*, 1986], [Chang and Li, 1995]) of the individual bands do not exploit the full image information content. Each band gives give a different set of segments, which creates additional difficulties when they are to be combined. In this paper a method is presented that segments a multi-spectral image into one unique set of objects.

The purpose of image segmentation is to subdivide an image into regions that are homogeneous according to certain criteria, in such a way that these regions correspond to relevant objects in the terrain. The relevance of objects depends on user requirements.

4.1 Existing methods

The two major approaches in image segmentation are *edge based* and *region based.*

4.1.1 Edge-based segmentation

Edge based segmentation is executed in two steps. The first step is to find segment boundaries in the image by identifying edge pixels, at those places where grey value changes occur. This is a neighborhood operation: to decide whether a pixel is an edge pixel, neighboring pixels have to be examined. Subsequently, each image region that is completely surrounded by edge pixels becomes a segment. A problem is that edge pixels, identified during the first step, do not obey topological constraints for segment boundaries. Therefore, an intermediate step is necessary to remove superfluous edge pixels and fill gaps in boundaries. Edge based segmentation divides the image pixels into two kinds, those belonging to segments and those belonging to boundaries. This corresponds to a model for object representation in the raster domain where 'object pixels' are labeled with the object they belong to, and a separate label is reserved for 'boundary pixels' [Molenaar, 1998].

4.1.2 Region based segmentation

Area based segmentation creates segments by applying homogeneity criteria inside candidate segments. A distinction is made between *region growing* and *split and*

merge algorithms, Region growing can be implemented in different ways, for example as follows. Segments are formed starting from (randomly placed) seed pixels by iteratively augmenting them with surrounding pixels as long as the homogeneity criteria are satisfied. When no more pixels can be attributed to any of the segments, new seeds are placed in the the unsegmented areas and the process is repeated. This continues until the whole image is segmented. Split and merge algorithms start by subdividing the image into squares of a fixed size, usually corresponding to leaves at a certain level in a quadtree. Recursively, leaves are tested for homogeneity and heterogeneous leaves are subdivided into four lower level ones, while homogeneous leaves may be combined with three (homogeneous) neighbors into one leaf at a higher level, provided the homogeneity criteria continue to be satisfied. The recursion stops at the low end at single-pixel leaves (they are homogeneous), and at the high end when no further combinations can be made (the extreme case being an entirely homogeneous image). Subsequently, adjacent leaves at different levels are combined into irregularly shaped, homogeneous segments.

After region-based segmentation, each pixel belongs to a segment. There are no boundary pixels. This corresponds to the raster model which labels a pixel according to the object that has the largest overlap with the cell [Molenaar, 1998]. The advantage of this model, compared to the above-mentioned model that distinguishes between object and boundary pixels, is that the objects form a spatial partitioning: Also the terrain is usually regarded as being completely filled with objects. Moreover, as long as spatial resolutions are still an important limiting factor in satellite image applications, boundary pixels may completely obscure small objects — how to represent a 10 m wide road in a 10 m resolution map, using boundary pixels?

Region-based segmentations generally suffer from *order dependency*. During region growing, a segment could be expanded with any of a subset of neighboring pixels, but not with all of them. Conversely, a pixel can be adjacent to more than one segment and might be added to each of those. The choices made in those cases are, to a certain degree, arbitrary and they are usually influenced by the order in which the data are stored and possible combinations examined. Similar considerations apply to split and merge. In the initial (recursive) phase, the homogeneous regions are restricted by the quadtree structure. They are square, their sizes are powers of two, and they can only be located at a limited set of positions within the image. When merging leaves into segments during the final stage of split and merge, the order in which the combinations are examined plays a role.

4.2 Definitions

Adjacency

Two grid cells (r_a, c_a) and (r_b, c_b) are adjacent in G (they are neighbors) if they are in subsequent rows within the same column ($|r_a - r_b| = 1$ and $c_a = c_b$) or in

subsequent columns within the same row ($|c_a - c_b| = 1$ and $r_a = r_b$).

Therefore, every grid cell that is not at one of the edges of the image has exactly four neighbors. This is called the 4-adjacency model, as opposed to the 8-adjacency model where also diagonally adjacent pixels ($|r_a - r_b| = 1$ and $|c_a - c_b| = 1$) are considered neighbors (Figure 4.1).

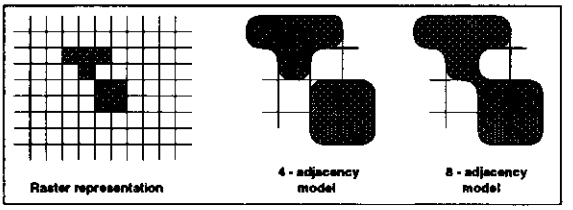

Figure 4.1: 4-adjacency: two segments — 8-adjacency: one segment.

Segment

A segment s is a subset of G in which the grid cells are connected (form a contiguous set). This means that for each pair of grid cells (r_a, c_a) and (r_b, c_b) in s there exists an ordered set of grid cells $\{(r_a, c_a), \ldots, (r_b, c_b)\}$ in which any pair of subsequent elements are adjacent.

We can distinguish between 4 and 8-adjacent segments depending on the adjacency model used. Which model is more suitable depends upon the type of objects and the image resolution. If the objects extend over areas that are typically large in comparison to the area covered by one grid cell in the terrain, 4-adjacency is appropriate. For example, two agricultural fields with the same crop having only a corner in common should be considered two objects. If, on the other hand, we want to model linear objects, such as roads and railways, which have a width smaller than the image resolution, the 8-adjacency model is required to prevent an object to be broken in many small segments. (Figure 4.2).

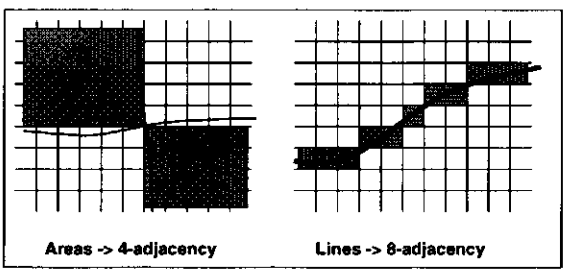

Figure 4.2: Object types, raster representation and preferred adjacency model

The 4 or 8-adjacency issue is of particular importance in binary images $I : G \rightarrow \{0, 1\}$, which represent a collection of objects (segments in which the pixel value equals 1) against a background (0). A relevant observation that can be made in Figure 4.1 is that the 8-adjacency model (for objects) implies that the background is 4-adjacent, and vice versa.

However, here we are developing a model for the situation that the entire terrain is filled with objects and, therefore, the entire grid space will be filled with segments. Then, using 8-adjacency, it will be difficult to imagine what kind of objects are modeled in case of two pairs of diagonally adjacent pixels that meet at one corner, as in $\frac{\text{a}\ |\ \text{b}}{\text{b}\ |\ \text{a}}$, a problem that does not arise when 4-adjacency is assumed.

Two segments s_1 and s_2 are *adjacent* if there are 4-adjacent pixels $p_1 \in s_1$ and $p_2 \in s_2$. Then $s_1 \cup s_2$ is a segment as well.

Segmentations

A segmentation S of the grid space G is a set of non-overlapping segments $\{s_i\}$ in G. Therefore, each $s_i \subset G$ and $\forall i \neq j : s_i \cap s_j = \phi$. S contains $\#S$ segments. It follows that $\bigcup s_i \subset G$. If $\bigcup s_i = G$, we call the segmentation *complete.*

When a segmentation of the grid space G is based on the image itself, *i.e.* on the spatial distribution of feature vectors in I, we will call it an internal segmentation or *image segmentation* of I.

If a segmentation is based on additional data, such as stratifications or context maps (Chapter 2), it is called *external* with respect to I.

The above definition of segmentation is not very strict. Many authors, *e.g.* in the survey of [Fu and Mui, 1981], restrict the notion of segmentation to what is called internal, complete segmentation in our terminology. Moreover, the definition in [Fu and Mui, 1981] involves a homogeneity predicate $Y(s)$, which only depends on the feature vectors in s, such that $Y(s_i) = \text{TRUE}$ for all $s_i \in S$, and $Y(s_i \cup s_j) = \text{FALSE}$ for all adjacent pairs s_i, s_j. This seems unnecessarily restrictive. For example, the merging criterion in section 4.4 includes the spectral distance between adjacent segments, which cannot be expressed in a predicate $Y(s)$ that depends on one segment. On the other hand, including the predicate in the definition is not so helpful, since with a single predicate $Y(s)$ several segmentations of I are still possible.

Hybrids between internal and external segmentation use ancillary data to guide the segmentation process [Ballard and Brown, 1982]. For example, a digital map can indicate where segment boundaries are likely to occur [Janssen, 1994].

Internal segmentations

Region based image segmentation algorithms identify segments in which the pixels satisfy homogeneity criteria in the feature space $\mathbf{X}$. This applies as well to tex-

ture based segmentations, where the reflections in a segment are not necessarily homogeneous. The feature space **X** will contain features where different image textures are mapped into different feature values, according to a texture-detecting filter algorithm, in such a way that a region with a uniform texture will become a homogeneous region in the texture feature [Rosenfeld, 1975], [Haralick, 1979], [Iron and Petersen, 1981]. In an MSc project [Talukdar, 1997], average gray value difference [Cross *et al*, 1988] was used to derive a texture feature from high resolution pan-chromatic imagery, adding a dimension to the feature space of a lower resolution multi spectral image. Several combinations of pan-chromatic SPOT (10 m resolution) or IRS-C (5.8 m) with multi-spectral SPOT (20 m) or Thematic Mapper (30 m) were investigated. The multi resolution approach is valid, since texture-feature extraction implies loss of resolution due to inexact identification of the location of boundaries between differently textured objects [Pratt, 1978].

4.3 Quadtrees — a data structure for integration of GIS and image data

To widen the opportunities to study and investigate spatial data structures in the institute, a modest software system for region quadtrees [Samet, 1990] was gradually developed during the last few years [Gorte, 1995b].

Region quadtrees constitute a spatial data structure, suitable to support implementation of the *raster* spatial data model [Molenaar, 1998].

4.3.1 The raster spatial data model

The raster data model, in accordance with the definition of *data model* [Tsichritzis and Lochovsky, 1982], allows to structure spatial data and to specify operations to be performed on these data. Concerning the structuring part of the definition, the user can organize the data about the study area in different raster layers and attach meanings (semantics) to pixel values in each layer [Burrough, 1986], [Gorte *et al.*, 1988], such as:

- Mapping unit identifier, providing a link to an attribute table
- Class label or nominal attribute identifier, such as soil type or land-use class
- Measurement, such as elevation, slope, soil depth, pH.
- Distance, *e.g* to the nearest road
- Reflection or feature vector in image data.

Spatial analysis and query can be specified in terms of primitive operations, as defined by the data model. These include:

- Overlay, to establish relations between layers
- Adjacency analysis, to establish topological relationships within a layer

- Indexing, to exploit relationships between raster maps and attribute tables
- Classification, to create class labels from measurements
- Connected component labeling, to create *objects* from classes or other attributes
- Window operations, to calculate densities, perform smoothing of values and shapes, etc.
- Distance transforms to calculate distances to objects and to create buffers around objects

The expressive power of these primitive operations is quite high and they offer great flexibility. Many kinds of spatial analysis can be formulated easily [Tomlin, 1990]. Another advantage of using the same (raster) data structure for maps and images is the ease of integration of the two.

To implement the raster data model in a spatial data base system, a data storage structure has to be chosen. Straightforward implementations store the data sequentially in a row-by-row order. The byte-position p of a pixel in this structure can be easily obtained from a raster coordinate (r, c), given the number of columns n_c in the map and the number of bytes b per pixel, as $p = b(n_c r + c)$. More advanced methods divide the area in *tiles* or *patches* to allow random access to the data without excessive input/output load, assuming *locality of reference*, which means that the next pixel to be accessed is usually near to the current one. This increases performance when 'roaming' through a map on the screen, or when performing geometric transformations that involve rotation.

A major drawback of the raster data model is that a trade-off has to be made between spatial resolution on one hand, and storage requirements and execution times on the other. Despite ever-increasing capacities and speeds of personal computers and workstations, certain combinations of spatial resolution and area size are simply not practical.

For example, a mapping scale of 1:25000 requires a spatial resolution in the order of 2.5m — this would correspond to 0.1mm on the map. An area of 200 × 200 km^2, which is not large enough to cover the Netherlands, contains 80,000 × 80,000 pixels. A layer in which one byte is sufficient to code the data occupies is 6.4 Gb. Certain queries and analysis operations traverse the entire data set and (extrapolating from measurements that will be described in section 4.3.2) will need hours or days to execute.

Many general purpose data compression techniques, such as Lempel-Ziv (LZ77), LZW and Huffman coding can be used to greatly relieve the storage requirements. However, they do not improve processing speed at all, since data have to be decompressed prior to any operation (which takes a little bit of extra time, in fact).

Region quadtrees are equivalent to raster maps. They offer the same semantic content as the raster maps on which they were based [Molenaar, 1998]. Raster GIS

analysis operations are also defined in the quadtree domain and most of them can be implemented efficiently.

Therefore, the quadtree data structure offers an alternative storage method for implementations of the raster data model. Data, stored using this method, are semantically valid for the model, and algorithms exist to support the required operations.

Quadtree data structure and software help to decrease the storage and processing time requirements at the same time, especially at high resolutions. Roughly, storage requirements increase linearly with resolution when using quadtrees, and quadratically using rasters. The challenge of quadtrees is to create algorithms that work in the quadtree domain, which means that they do not expand the data to raster format at any stage. In that case, processing times depend on the quadtree data set sizes, which leads to a significant gain at high resolutions.

The advantages of using quadtrees are largest when using high resolution GIS maps, especially when they contain relatively large objects or homogeneous regions. Unfortunately, not much is gained in terms of space and time, when images are processed as quadtrees. However, quadtrees allow to combine data layers with different resolutions without having to re-sample one to the other.

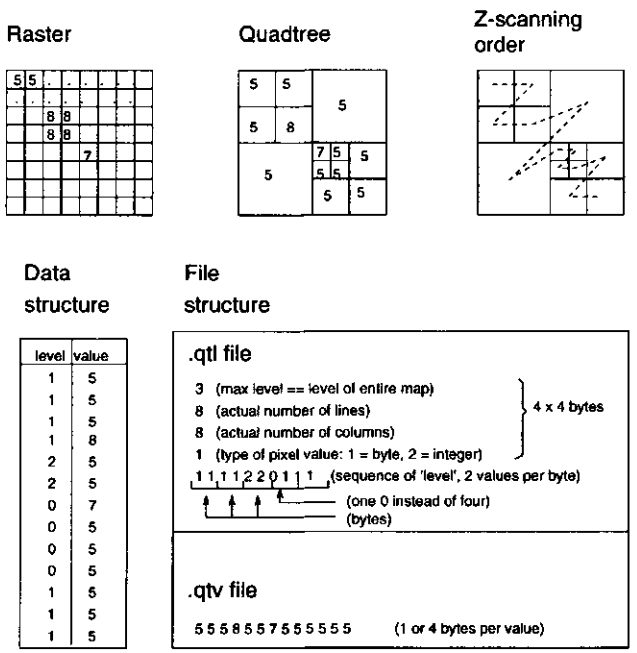

Figure 4.3: Quadtree data structure

The quadtree package in which the segmentation algorithms are embedded is briefly described below. It is based on a linear, sequential quadtree data structure, without indexing (Fig. 4.3). The programs read input and write output streams sequentially and simultaneously, without excessive buffering in internal memory. Therefore, there are no (practical) limitations to the sizes (resolutions) of the data sets to be processed. The entire data set does not have to reside in internal memory at any time.

The following modules are present:

General: raster to quadtree and quadtree to raster conversions, image calculations, statistical analysis (histograms, multi-band statistics), simple map and image generalization, which allows to create levels of detail (LOD) at different representation scales.

Image Analysis: Training data analysis and maximum likelihood classification, principal component transformation, RGB to IHS transforms.

GIS: Map overlay and map calculation, aggregation functions, determination of topology (region adjacency), connected component labeling.

The segmentation algorithm is based on the one for connected component labeling — in fact, the latter will appear to be a special case of the former. Also, the map calculation module will be involved in the segmentation process, as well as the connected component labeling. Therefore, we describe these three modules with somewhat more detail.

Connected component labeling: a program that assigns to each homogeneous region a unique value. The output quadtree values have the type *integer*, which allows over 2.10^9 regions. It is interesting to notice that the structure of the quadtree does not change with this operation.

The program assumes 4-adjacency: only four neighbors of a pixel (above, below, left and right) are taken into account when connectivity is established, instead of 8 neighbors (including the diagonal ones). In case of very high resolutions that can be handled by quadtrees, region pairs that are 8-adjacent without being also 4-adjacent are very unlikely to occur,

Image and map calculations are carried out by a program which allows overlaying data layers by performing arithmetical, mathematical, logical and relational operations on corresponding pixels in different layers.

This program also provides the link between spatial and attribute data. If pixel values have the meaning of *object number*, attribute values can be found at any pixel by indexing the attribute table with the pixel value. See the result of segmentation in Figure 4.5.

Region Adjacency software can be used to establish adjacency between pixel values in a quadtree. The result is a relational table with two columns; if somewhere in the quadtree a pixel with value p is neighboring a pixel with value q, then (p, q) will be a record in the table. The table is sorted in ascending order of (primarily) the first column and (secondarily) the second

column. The value in the second columns is always larger than the one in the first; if p is larger than q, there will be a record (q, p) in the table. Therefore, every combination is listed only once.
The operation makes most sense if the quadtree is filled with regions that have unique numbers, such as the result of an image segmentation. In that case it generates region adjacency information, which can be incorporated in subsequent classification.

4.3.2 Quadtree performance

As an example, quadtree performance was measured using a land-use map of the Razan area in the province of Hamadan, Iran [Sharifi *et al.*, 1996] (Fig. 4.4). The original raster map covers an area of 74.3×78.0 km^2 at a spatial resolution of 20m, giving a data set of 3714 rows and 3900 columns. From this data set versions with lower resolutions (40, 80, 160 and 320 m) were generated by replacing square areas in the original (with sizes of $2 \times 2, 4 \times 4, 8 \times 8$ and 16×16 pixels, respectively) by a single pixels at a lower resolution. For the output pixel value the predominant value in each input square was used.

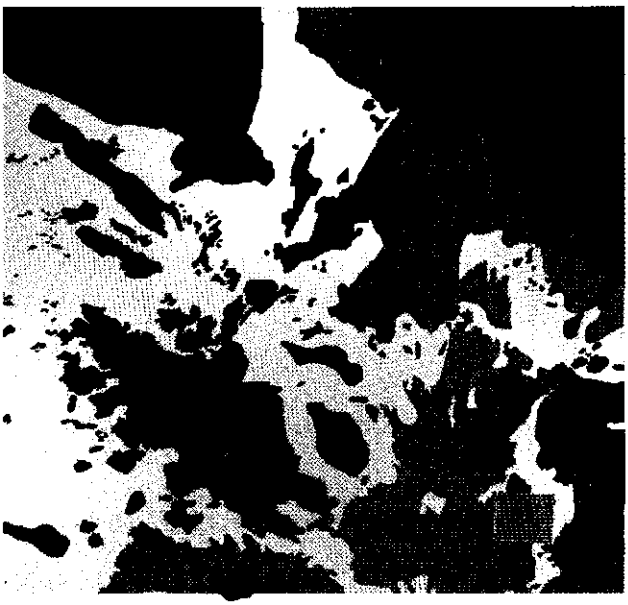

Figure 4.4: Sample landuse map (Iran)

The results were converted to quadtrees. Table 4.1 shows the data volume in raster pixels and in quadtree leaves. Whereas the former increases quadratically with the

res.	size		volume		execution times (s)				
(m)	rows	colums	pixels	leaves	ar(r)	ar(q)	ad(r)	ad(q)	cc(q)
320	232	243	56367	11566	0.21	0.07	0.03	0.07	0.08
160	464	487	225968	26305	0.78	0.16	0.08	0.13	0.22
80	928	975	904800	59374	3.07	0.32	0.32	0.28	0.50
40	1857	1950	3621150	138862	12.34	0.68	1.44	0.70	1.28
20	3714	3900	14484600	245080	50.47	1.23	5.94	1.27	2.78

Table 4.1: Performance of quadtree *vs.* raster at different resolutions. Executions times are given for ar(r) and ar(q) (arithmetic calculation in raster and vector), for ad(r) and ad(q) (adjacency analysis in raster and vector) and for cc(q) (connected component labeling, only in quadtree). The measurements concern CPU time on a Pentium 133 MHz computer with 96 Mb main memory, running the Linux (PC-UNIX) operating system. All algorithms are implemented in the C programming languare.

resolution, the latter shows an approximately linear increase. Note, that to get the actual quadtree storage requirements, the space to store *levels* has to be added to the space for *values*. In the described implementation, the increase is theoretically between 12.5% and 50% (see Fig. 4.3). At the different resolutions of this example it is between 26.1% and 31.4%. The quadtree with 20m resolution occupies 322134 bytes: 245080 for values and 77054 for levels.

For the record, Lempel-Ziv compression reduced the 20m-resolution raster map to 163408 bytes, and the corresponding quadtree to 69720 bytes. Therefore, there is not so much reason to discuss quadtrees *vs.* general purpose compression techniques. They are complementary.

Execution times of the following operations were measured (Table 4.1):

Arithmetic calculation of $r = \sqrt{a+1}$ on an attribute a (columns ar(r) and ar(q)). The time for both raster and quadtree depends on the number of square-root calls involved, although the table shows that for some extra time is caused by quadtree overhead (compare, for example, 1.23s for 245080 leaves with 0.78s for 225986 pixels).

Region adjacency, which determines for each pair of map units whether they are adjacent (using 4-adjacency), shows the advantage of quadtrees at higher resolutions (columns ra(r) and ra(q)). The figures for lower resolutions show that the additional quadtree overhead is quite heavy, but it is (again) proportional to the quadtree size.

Connected component labeling, which determines a unique identifier for each map unit with a homogeneous attribute value (or class number), also shows linear increase of execution time with resolution (column cc(q) — I do not have an equivalent program for raster maps). The algorithm is the basis for region-merging image segmentation (section 4.4).

Let me allow myself to make some extrapolations. A 80,000 × 80,000 pixel map (as

was mentioned above) of this area, therefore with 1m resolution, might contain some $20 \times 250{,}000 = 5$ million leaves to represent 6.4 billion pixels. Region adjacency calculation could be done in $20 \times 1.27\text{s} \approx 25\text{s}$, whereas in a raster implementation it would take $400 \times 5.94\text{s} \approx 40$ minutes. Extrapolated execution times for the arithmetic calculation example are 25 seconds and 5.6 hours, respectively. I must admit that these figures are not completely honest. At a so much higher resolution the map will contain more units, which does not matter in the raster implementation, but has a negative influence on performance when quadtrees are used. In addition I should mention that my implementation cannot handle such large quadtrees. A hard limitation is caused by the four bits that are reserved for *levels*, which puts the maximum level at 16 and the maximum size at 65535×65535. But 'softer' limitations may be the consequence of memory capacity, for example. I did not have the opportunity to test data sets of such sizes, since they would have to imported from raster maps, which do not fit on any computer within my reach.

The last twelve years have shown doubling of computer performance approximately every 18 months, giving an increase of a factor $2^8 = 256$ during this twelve year period. Between 1986 and 1998, main memory sizes went from 512kB to 128MB and hard disk capacities from a typical 20 MB to 5 GB, Pentium-II chips easily reach performance indices of a few hundred times the 80286, only the price for all this luxury is more or less the same. However, to bring hours or days of execution time for raster processing back to acceptable seconds or minutes requires again a performance increase of, say, a factor between 1000 and 4000 ($2^{10} - 2^{12}$), which can be expected in 15 to 18 years — can it?

Twelve years ago, maps with 512×512 pixels were as practical or impractical as 8000×8000 maps are nowadays: certain things can be done quite quickly, but usually the word 'interactive' really does not apply anymore. A simple formula like $r = \sqrt{a+1}$, for example, takes almost four minutes, which is acceptable if you really need it, but annoying when you find out that r should be $\sqrt{a}+1$.

In 1986, quadtrees did not help, since 512×512 quadtrees are more demanding than raster maps of that size. For 8000×8000, however, quadtrees already offer advantages, and they will increasingly do so when hardware allows for larger data sets in the near future. Quadtree implementations might bring very-high resolution raster data models for GIS much nearer.

4.4 Segmentation by region merging [1]

In the course of developing the quadtree system, a stage was reached were quadtree based image segmentation could be implemented without too much additional effort.

1. Region merging as a segmentation method, not as the problem of merging terrain regions into a single segment.

A *region merging* segmentation method resulted, which does not show order dependency problems. It is a hybrid between region growing and split and merge. The algorithm makes a recursive, bottom up quadtree traversal, which starts at single pixels (or larger quadtree leaves in which the pixel values are constant) and recursively merges adjacent regions, forming irregularly shaped segments at all stages. The order dependency problem is solved by performing several iterations, while slowly relaxing the homogeneity criteria until a user defined degree of segmentation is reached.

4.4.1 Description

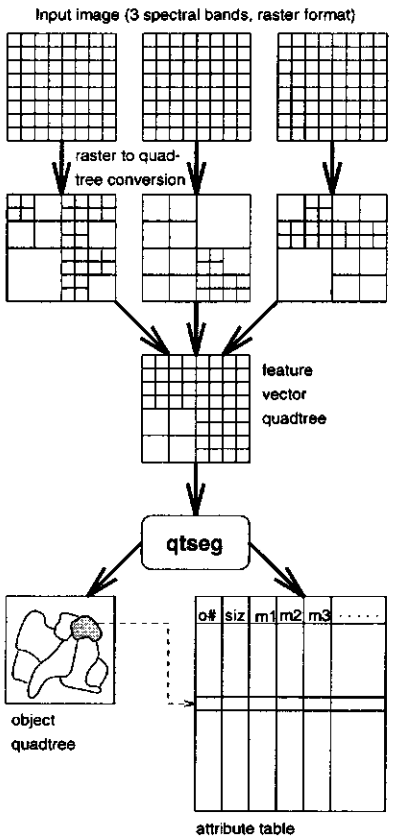

Figure 4.5: Data structures for quadtree segmentation of multi-spectral images

The region merging algorithm uses several image bands, which are first combined into a feature vector quadtree, as input and gives one segmentation as output: a

set of objects, where each object has multi-spectral properties (mean vector and variance-covariance matrix). Moreover, topological (object adjacency) information can be retrieved as well as, of course, object locations, sizes and perimeters.

The algorithm recursively merges leaves and regions of the feature vector. Since all the leaves are used, the resulting segmentation is complete. In each recursion level of the algorithm, adjacency of (sub)segments to be merged is determined along the horizontal and vertical boundaries in quadtree nodes, so that the segmentation uses 4-adjacency.

A merging criterion controls whether adjacent regions can be merged. In the current implementation the criterion is quite simple: With a user-selected *spectral distance* threshold θ, the Euclidean distance between the feature vectors of two candidates may be not larger than 2θ and none of the variances and covariances after merging may exceed θ^2. The algorithm leafs room for more advanced criteria, but in experiments to date, the simple criterion appeared to be satisfactory. Note that *connected component labeling* is a special case with only one band and $\theta = 0$.

Like in the other programs in the quadtree package, the quadtree is scanned sequentially, which implies a single traversal through the image in Z-scan order (Figure 4.3). Therefore, the algorithm is recursive and works bottom-up. It starts trying to combine individual pixels (within quadrants) first, and looks at possibilities to combine adjacent regions in larger quadrants later.

The program relies on a highly dynamic data structure consisting of an **index table** and an **object table**. The object table has one record for each (intermediate) object, in which the object size and spectral attributes are stored. In case of three spectral bands, these attributes are: the sums of the pixel values in band 1, 2 and 3 over the entire object (S_1, S_2, S_3), the sum-of-squares (S_{11}, S_{22}, S_{33}) and the sums of the cross-products (S_{12}, S_{13}, S_{23}). These are used in the calculations of the mean values and the covariance matrix for the object.

An object is entered in the table when a new leaf from the input is read. A new entry in the index table points to the object. When processing a quadrant, the values to either side of the boundaries between the sub-quadrants are taken from a **stack**. Via the index table, the spectral data are retrieved from the object table and used in the merge criterion.

If two objects can be merged, their respective attribute values (sizes and sums) are added and stored in the table entry of the object with the lowest object number. The other object is removed from the table. Also the index table is updated: the higher entry will point to the lower one. Figure 4.6 shows the states of the index and object table before and after processing the quadrant in Figure 4.7.

After the quadrant is finished, the (new) values at the outer boundaries are known. They are stored at the next higher level of the stack, from where they will be retrieved when the next larger quadrant (containing this one) is processed.

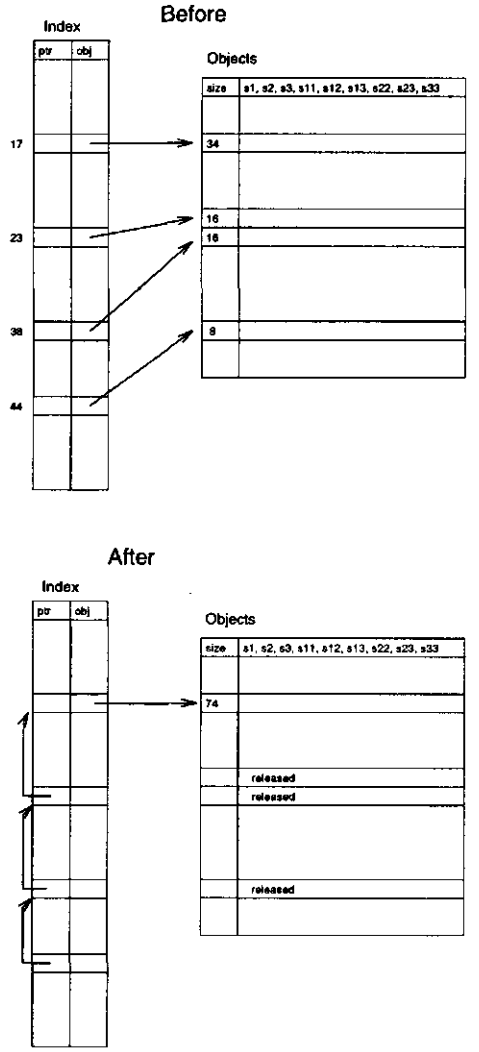

Figure 4.6: Index and Object table before and after processing fig. 4.7

When the entire quadtree has been processed in this way, which is when the program reaches the highest level, the index table is updated: All entries that have an object number associated with them are moved to the top of the table; the pointers of all other entries are updated so that they will point to the end of the chains. Then the input quadtree is read again and the output (segmented) quadtree is produced. Finally, an attribute table is created from the object table, by transforming sums

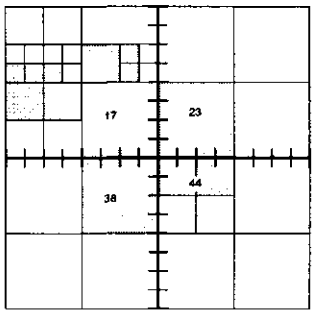

Figure 4.7: Segmentation at intermediate level

and sizes into means and covariances. The attribute table is stored on disk and can be used in subsequent analysis.

4.4.2 Iteration

Due to the recursive z-scan order, the algorithm has a slight tendency to create segments of regular shapes, according to the quadrants (order dependency). At a certain quadtree level, the algorithm attempts to merge regions within quadrants, before examining adjacent quadtrees at the next higher level. This effect could be completely removed by making the process perform a few iterations, with increasing threshold values. When starting with a lower threshold value than the final one, the risk of inadvertedly merging sub-quadrants reduces. Irregular shapes will already be formed, however, and will be the basis for further merging later, when higher threshold values come into effect.

4.4.3 Small objects

Segmenting satellite images creates many small segments (say, less than five pixels in size). One reason may be, of course, that due to the limited resolution of satellite imagery, there are many of such small objects in the terrain.

More important, however, is the effect of *mixed pixels*, especially at the boundaries of objects with quite different spectral signatures. In the feature space, those mixed pixels are too far away from both objects, and therefore they cannot be merged with one of them. The question is what to do with them. From a segmentation point of view, we would like them to be incorporated into larger (neighboring) segments. To achieve this, we can relax the merging criterion, by increasing the threshold value especially for small segments. However, the spectral values of the boundary pixels will contaminate those of the entire segment (unless we don't update the

values of the larger segment when merger is due to criterion relaxation — this was not investigated, however) and influence a later classification. Another possibility is to leave the small segments (mixed pixels) out of the classification procedure and classify only the large ones. The above-described map calculation program can be used to make the selection of large segments, based on the sizes in the attribute table. Under the assumption that objects are relatively large, compared to the pixel size, there is a slight preference for the second option.

4.4.4 Experiment

Segmentation was applied to a Landsat TM image of the Flevopolder in the Netherlands. The area is suitable for demonstrating the segmentation method, because there are large fields. Usually, Landsat TM does not satisfy the previously stated condition that objects should consist of a significant number of pixels. The method will be more useful when higher resolution imagery becomes available.

The results are shown in Figures 4.8 and 4.9. Using map calculation, combining the segment quadtree with the attribute table, only large segments were selected and a random grey value was assigned to them. Small segments were removed.

Figure 4.8: Detail of segmented image. Objects are displayed with random grey values, those that are smaller than five pixels are black

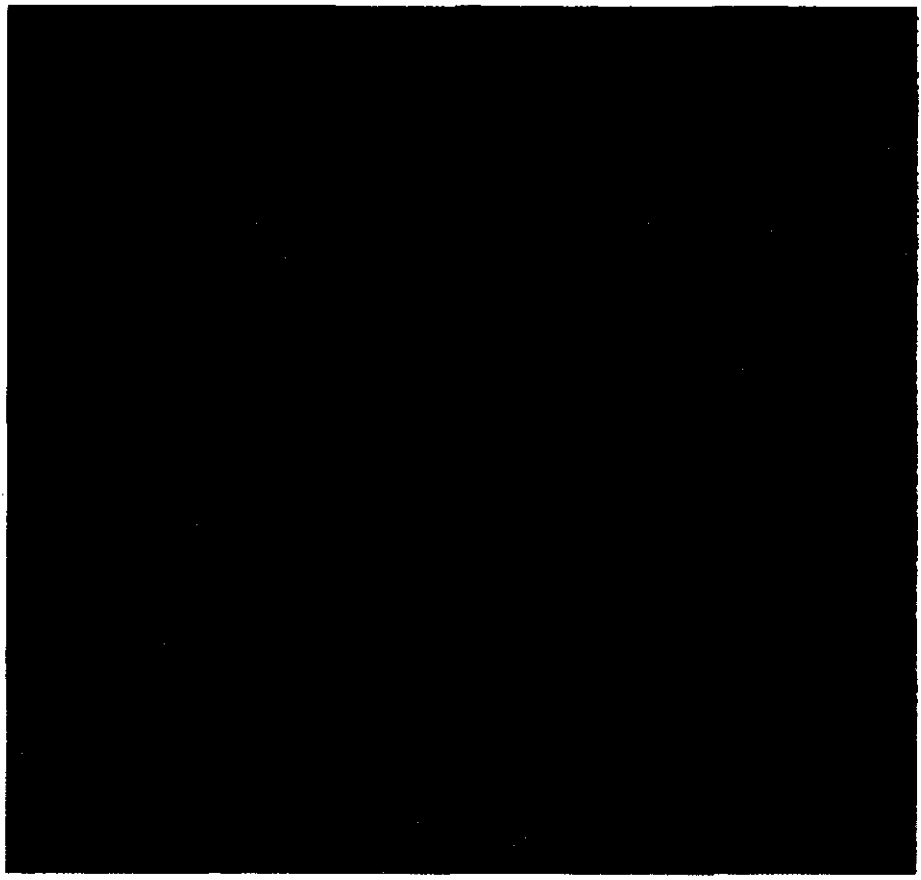

Figure 4.9: Segmented image (1000 * 1000 pixels)

The image consists of 1000 × 1000 pixels. With a final spectral distance threshold value of 6, 180811 segments were created. Despite the large objects in the terrain, many segments are very small: 136870 single pixels and respectively 20053, 5252, 4009 and 2539 segments of two, three, four and five pixels. Figure 4.8 shows small segments in black and reveals that they are mostly boundary (mixed) pixels.

On the other side of the scale, there are four segments with more than ten thousand pixels. They are water bodies (IJsselmeer and Randmeren), with 11149, 33317, 44069 and 111375 pixels, respectively. The distribution of the sizes of the more moderate objects is shown in Figure 4.10

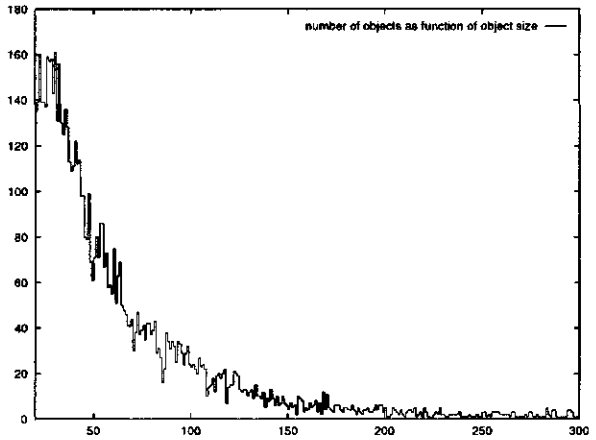

Figure 4.10: Distribution of object sizes

4.4.5 Evaluation

Although the presented region merging algorithms resembles classical *split and merge*, the result are significantly better (Fig. 4.11. Split and merge creates segments that coincide with quadtree leaves, which does not really help in case resolution is a constraint. Most quadtree leaves, except those at very low levels, contain segment boundaries and are, therefore, not suitable parts of segments. A subsequent step is needed to merge quadrants into final segments [Cross *et al*, 1988], but with a resolution constraint this means that almost the entire merging is done in this second step. *Region merging*, as presented, combines both steps throughout the algorithm.

The resulting program is far from trivial. It is embedded in the quadtree system, which was designed with optimization objectives from the beginning, to exploit the potential performance increase by using a quadtree data structure, instead of raster. As a result, the segmentation program is really fast. Whereas [Schoenmakers, 1995] spends considerable effort to optimize a hybrid region growing / split and merge method, he mentions execution times of between 55 and 135 hours for an image of 1.48 million pixels on a Sun Sparc 10-41 workstation with 64 MB of RAM. The segmentation of the 1 million pixel example presented above, using four iterations, takes 52 s on a 133 MHz Pentium with 96 MB.

Order dependency can be checked by slightly shifting the input image in the grid space, for example by removing the first row and column from the image, prior to converting it to a quadtree. The quadtree structure changes quite significantly, but hardly any changes are observed in the resulting segmentation.

The algorithm presents two difficulties:

- The user has to provide a series of threshold values $t_1, \ldots, t_n$. The last

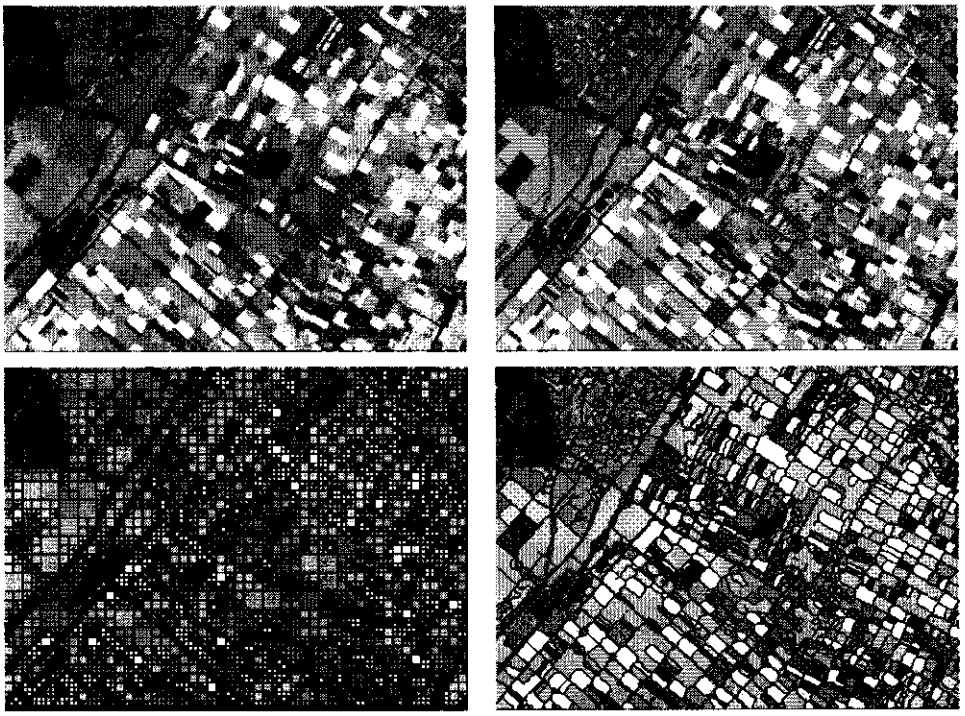

Figure 4.11: Conventional split-and-merge (upper left) vs. proposed method (upper right). Bottom row: with added segment boundaries.

value, t_n determines the "coarseness" of the final result, whereas the preceding values serve to make the merging gradual, to prevent a too large influence of the quadtree-induced subdivision of the space on the shapes of the segments. This solves the order dependency problem, described by [Haralick and Shapiro, 1985], [Tilton, 1989], [Pavlidis, 1986] and [Fu and Mui, 1981]. Unfortunately, the user has to apply trial and error to find suitable threshold values and to visually inspect the segmentation results. If he observes too many small segments, such that many (supposed) terrain objects are still subdivided, the final threshold value t_n should be increased. Conversely, if too many terrain objects are combined into single segments, the t_n threshold value should be lowered. Moreover, those occurrences of *region merging* and *region fragmentation*, in the terminology of [Acton, 1996], happen at the same time in any segmentation that is not extremely fine or extremely coarse. (Figure 4.12.)

- Segmentations contain many segments of only few pixels. This is partly caused by small terrain objects compared to the image resolution. Recogni-

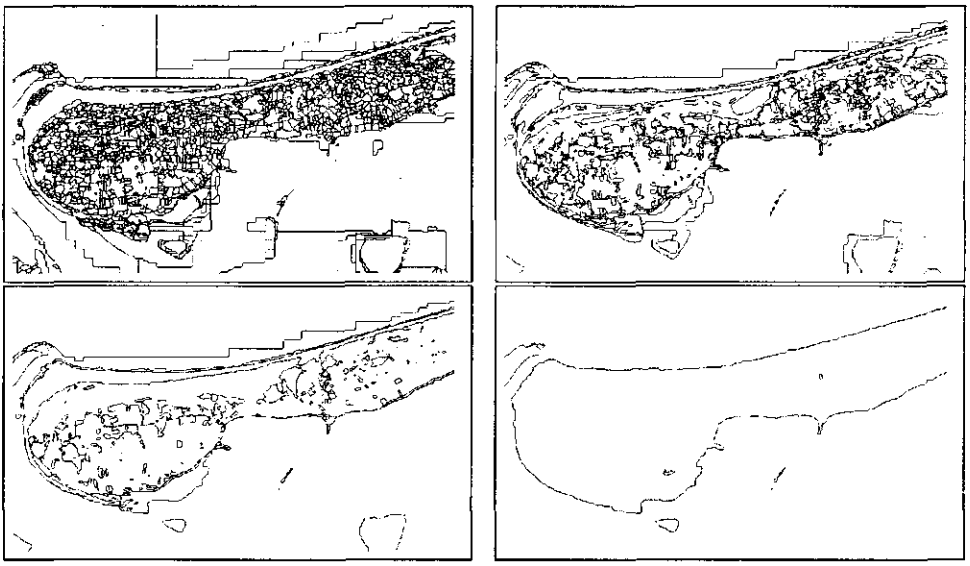

Figure 4.12: Segment boundaries during four segmentation iterations

tion of such objects is inherently difficult — whether it is important depends upon the user's requirements. For the other part, small segments occur at *mixed pixels*, which are pixels that are intersected by boundaries between larger terrain objects.

The first problem is inherent to any image segmentation algorithm. It is exactly the reason to look for *semantic* merging criteria, based on class homogeneity, rather than feature vector homogeneity (Chapter 5). There, an entire segmentation pyramid (section 4.5) is used to select homogeneous segments (objects) at the highest possible pyramid level. In areas where no homogeneous segments can be found at any level, I will select suitable segments at an appropriate level and mark them as "mixture of terrain objects for which the exact location cannot be determined".

The second problem is tricky. It is inherent to region based image segmentation techniques ([Schoenmakers, 1995]), as opposed to edge based techniques. In the latter, most mixed pixels will be within the set of boundary pixels and, therefore, not expected to be included in the segments themselves. However, the question remains whether this is a real solution, since the notion of boundary pixels does not agree with the model that area objects cover the entire image region. To have separate boundary pixels is unfavorable in case spatial resolution is a limiting factor, as in satellite imagery. There are many objects of only a few pixels and, therefore, the fraction of boundary pixels in the total image will be large (Chapter 4). This conclusion is in line with the data model of [Molenaar, 1998], where objects provide

a space partitioning and boundaries are implicit.

[Schoenmakers, 1995], [Haralick and Shapiro, 1985] [Pavlidis and Liow, 1990] and [Le Moigne and Tilton, 1992] propose hybrids between edge and region based segmentation.

4.5 Segmentation Pyramids

A segmentation pyramid $\mathcal{S}$ of G is a collection of D nested segmentations $S^1, \ldots, S^D$, D being the depth of the pyramid. S_i^l is a segment in the segmentation at level l of the pyramid ($1 \leq l \leq D, 1 \leq i \leq \#S^l$). At higher levels in the pyramid the number of segments is smaller and the segmentation becomes coarser, because higher level segments are supersets (aggregates) of lower level ones. Using L^j for the number of segments in S^j:

$$L^m \geq L^n \quad (m, n \in [1, \ldots, D],\ m < n)$$

and

$$\forall_{k \in [1,\ldots,L^m]} \exists_{l \in [1,\ldots,L^n]} : s_k^m \subset s_l^n \quad (m, n \in [1, \ldots, D],\ m < n).$$

When all segmentations in a pyramid are complete, each segment at a certain level $n > 1$ can be composed of a subset of segments from a segmentation at a lower level:

$$\forall_{l \in [1,\ldots,L^n]} \exists_{k^* \subset [1,\ldots,L^m]} : \bigcup_{k^*} s_k^m = s_l^n \quad (m, n \in [1, \ldots, D],\ m < n).$$

The highest level segmentation in a pyramid may consist of only one segment, which is then the root of a segmentation *tree*. Otherwise, a segmentation pyramid becomes a tree when we add a dummy segmentation S^{D+1} of one segment ($S^{D+1} = \{G\}$) at the top. At the lowest level, each pixel may be a separate segment. If required, a segmentation S^0 of single pixel segments may be added otherwise at the bottom of the pyramid.

A parent - child relation exists between segments at two successive levels. Indirect parents (grandparents, grand-grandparents etc.) are called ancestors, indirect children are descendents. The operator anc(s) gives a set that contains all ancestors of a segment s. Likewise, dec(s) yields the descendents. Therefore, $s \cup \text{dec}(s)$ forms a segmentation tree.

An alternative representation is by means of a relational table with $\#S^1$ records (one record for each segment in the lowest level segmentation S^1) and D fields (one for each segmentation level), showing for each segment s_i^1 in which of the higher level segments it is contained. Normalizing the table separates it into $D-1$ tables, each showing the parent-child relationship between two successive segmentation levels.

Figure 4.13 shows both representations. The displayed segmentation pyramid contains three complete segmentations and, therefore, the tree has uniform depth and the table contains no empty fields (fields with vales NULL). In a tree with non-complete segmentations, it is possible that none of the pixels of a segment s at level $l > 1$ are included in lower level segmentations. Then, the node corresponding to s becomes a leaf in the tree at level l. The lower level fields in the segment's record in the table are empty (and the total number of records is larger than $\#S^1$).

Iterative region merging with a sequence of thresholds yields a segmentation pyramid when the result after each iteration is stored.

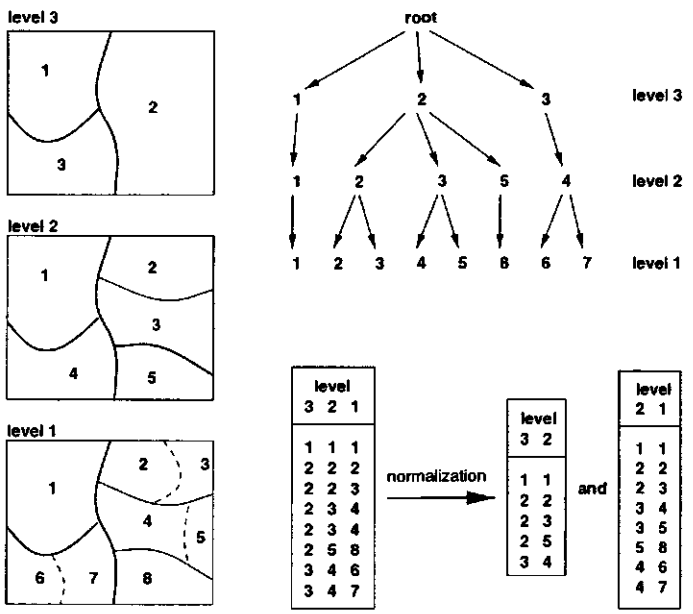

Figure 4.13: Segmentation pyramid with tree and table representations

4.6 Conclusion

This Chapter introduces a multi-spectral image segmentation method, which is embedded in a quadtree based GIS and Image Processing system. Generally, the

system gives the possibility to integrate remote sensing data, map data and attribute data. It processes high resolution raster maps and images, without excessive storage and processing time requirements.

The segmentation algorithm avoids order dependency by making a few iterations in which the *merging criteria*, which involve spectral distance and covariance, are gradually relaxed. Despite iteration, the algorithm is fast. Moreover, it creates a segmentation *pyramid* by outputting a segmentation after each iteration. Whereas region fragmentation and merging are inherent to data-driven segmentation, these problems can be solved by segment pyramid classification, *i.e.* class mixing proportion estimation in all segments within the pyramid. This is addressed in the next Chapter.

Chapter 5

Integration

The previous chapters addressed improvement of procedures for information extraction from multi spectral imagery. A distinction was made between classification and segmentation.

Classification attempts to find the most appropriate class label for each pixel in an image (Chapters 2 and 3). The purpose of image segmentation, on the other hand, is to subdivide an image into regions that are homogeneous according to certain criteria, such that these regions correspond to objects in the terrain (Chapter 4).

In the context of information extraction from imagery to create or update data in a geographic information system, it is required to identify objects and label them in such that they can be stored as entities in a data base. Therefore, we want to combine segmentation and classification. Object identification yields geometric information (where is an object to be found?), whereas the labeling provides thematic information (what type of object is identified?).

Sequential application of segmentation and classification (or of classification and segmentation) is straightforward. The result of a classification can be submitted to connected component labeling, which yields a unique identifier for each connected set of pixels of the same class. The pixels in the resulting raster data set have these identifiers as their values. Simultaneously, a list is produced that links each identifier to a class label. Conversely, multi spectral image segmentation produces an image with a unique identifier for each homogeneous region, together with a list that links each identifier to the spectral properties, *e.g.* mean vector and covariance matrix, of the corresponding region. Subsequently, the spectral property list can be submitted to classification, such that each element of the list is assigned a class label.

Both combination approaches are illustrated in Figure 5.1 . They share the disadvantage that errors and uncertainties from the first stage are carried over to the second. Therefore, erroneous results may be obtained.

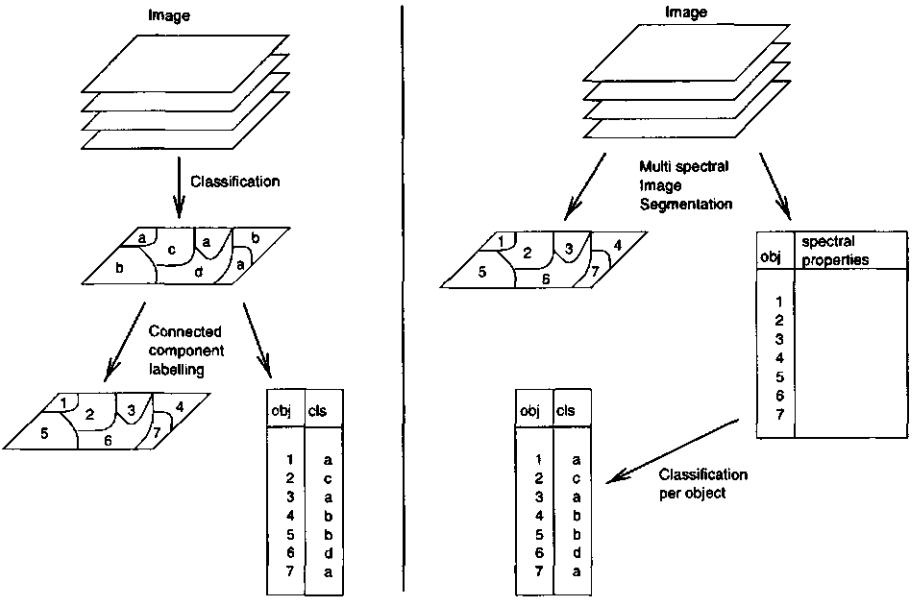

Figure 5.1: Sequential combination of classification and segmentation

When classification is applied first, it has the tendency to give misclassified pixels due to image noise and terrain cover irregularities, such as sandy patches in grassland, open spaces in forests, varying conditions in agricultural fields, etc. Subsequent connected component labeling will treat those as separate segments, which, from the user perspective, usually do not coincide with meaningful terrain objects. Therefore, the results must be edited extensively, before they are suitable for storage in a GIS database. Automatic editing, based on geometric and topological segments properties, is possible, but complex [Abkar, 1994].

When starting off with segmentation, a balance has to be found between region merging and region fragmentation. When several objects merge into one segment, subsequent classification will not split them. Fragmentation is less severe, as far as the resulting segments are classified into one class. Letting the scale tip too far in the direction of fragmentation, however, will produce so many small segments that the point of segmentation is lost. In a procedure called *layered classification* [Franklin and Wilson, 1991] apply conventional split and merge segmentation, restricting the quadrant size to a minimum of 4×4 pixels. They applied a quite large variance threshold to prevent that "an unacceptably large amount of quadrants" would be declared inhomogeneous. Next, F-statistics, followed by a Student's t-test, compare quadrants with class statistics. Since it appeared that many quadrants do not obtain a label, a second stage follows where unlabeled pixels

are submitted to straightforward minimum distance classification. Finally, in areas where the minimum distance exceeds a threshold an ancillary (elevation) "channel" is added to the feature space and the first two stages are repeated for pixels that were not labeled yet. [Franklin and Wilson, 1991] achieved significant improvement, compared to standard maximum likelihood classification. In a recent M.Sc. study [Shresta, 1998], however, the accuracy of a land-use classification in a mountainous area was lower with layered classification than with standard maximum likelihood. After replacing minimum distance in the second step by minimum Mahalanobis distance, accuracy increased from 62% for maximum likelihood to 65% for layered classification.

To improve upon this, an algorithm has been developed to integrate classification and segmentation.

5.1 Class homogeneity criteria

Segmentation, described in the previous chapter, is controlled by merging criteria on homogeneity of feature vectors within segments. A conceptually straightforward approach to include class statistics in segmentation is to design a merging criterion that evaluates class homogeneity, rather than feature vector homogeneity. Such an approach is attractive, because we are ultimately looking for meaningful objects in the terrain, by identifying labeled segments in the image.

Formulating a merging criterion for class homogeneity involves some considerations. First, two segments can be merged if there is sufficient confidence that they belong to the same class. Second, we want to have sufficient confidence that the merged segments belongs to a single class. At this moment, it is not possible to tell whether the first requirement implies the second, because it depends on how confidence will be quantified and called sufficient.

In an MSc research project [Lat, 1996], the student-t test was applied to assess the correspondence between segment and class feature vector distributions. A major problem is that a single object of a certain class is usually much more homogeneous than the entire collection of objects in that class.

Another consideration concerns segment size. We consider three cases:

Two small segments: A segmentation process that only merges starts from single pixels. If two pixels can only be merged when they have the same maximum likelihood class, the result of segmentation will be exactly the same as from connected component labeling of a segmented image. Therefore, this requirement is too strict. Two adjacent pixels that would be classified differently by a conventional maximum likelihood classifier can be merged under certain conditions, which have to be formulated.

Two large segments: It has to be decided, whether adjacent segments that belong to the same class should always be merged. For example, two spectrally different segments, both classified as *agriculture* are most likely two different fields. Whether they are considered one or two objects depends on the definition of "object".

A large and a small segment To an extent that depends on the algorithm, segmentation tends to create small segments for noise and boundary pixels, which may have a different maximum likelihood class than their neighbors. However, in the chosen data model (see Chapter 4) boundaries between objects are no objects themselves.

It is not entirely clear how two pieces of evidence that two adjacent segments belong to a certain class can be combined into evidence that the merged segment belongs to that class [1]. In a probabilistic approach, given the probabilities P_1 and P_2 that segment 1 and segment 2 belong to a certain class, a function f is needed that yields $P_{1,2} = f(P_1, P_2)$, the probability that the combined segments belong to that class.

A second attempt to integrate segmentation and classification was more successful, on the basis of algorithms described in the previous chapters:

- An improved algorithm for segmentation of multi spectral images, which iteratively yields a pyramid of segmentations with different degrees of aggregation
- An algorithm for regionalized class area estimation.

Although class area estimation intents to facilitate incorporation of ancillary data (GIS context maps) in classification, it also allows to incorporate segmentations. In this case, segmentation only provides a subdivision of the area, whereas local class probability densities are still calculated in the original image.

In coarse segmentations of segmentation pyramid, most segments will show a mixture of classes, because they contain several terrain objects. Only objects that are spectrally quite distinct from their neighbors in the terrain will appear as separate segments with single-class coverage. Other objects are expected to be present as segments in lower levels of the pyramid.

This leads to a global description of a recursive procedure, which will be refined in the next sections:

1. Generate a sequence of segmentations, using increasing threshold values.
2. Start from the last result in the series, which is the coarsest segmentation.
3. Estimate class areas for every segment.
4. For each segment s:

1. An interesting question in an aggregation/generalization perspective is whether they might belong to a common superclass, and how evidences should be combined in that case. Compare generalization strategies in Ch. 8 in [Molenaar, 1998]

(a) if the segment s is covered by only one class, mark it as "object", which, together with its unique class label, is to be stored in the result
(b) otherwise, go back in the sequence (*i.e.* down in the pyramid) to a segmentation where s is subdivided in two or more subsegments $s_1, s_2, \ldots$
(c) estimate class areas for each subsegment
(d) if $s_1, s_2, \ldots$ are similarly mixed as s, conclude that s is a single object consisting of a mixture of classes. s is marked as 'mixed object', which will be stored in the result together with the class area vector.
(e) otherwise, repeat 4. with $s_1, s_2, \ldots$

5.2 Selecting segments from a segmentation pyramid

The previous sections gave a pyramid of segmentations with different levels, and the opportunity to define a boolean function $p(s)$, which tells whether a segment s contains a single class or a mixture of classes, by examining relative class areas.

In this section, subsets will be selected from different pyramid levels such that the union of those subsets covers the entire area, *i.e.* a new complete segmentation is created from segments that are selected from different levels of the segmentation pyramid [2]. Preferably, pure segments are selected. If pure segments at different levels of the pyramid coincide, the one at the highest level has priority. At locations where no pure segments exist, mixed segments will be placed. More precisely: selected pure segments have no pure ancestors, and selected mixed segments have no pure ancestors or descendents. Also mixed segments (for which no pure subsegments exist at lower levels) will be selected from the highest segmentation level possible, which means that each of its ancestors should have at least one pure descendent. If a mixed segment has an ancestor with nothing but mixed descendents, this ancestor should get priority.

We use the symbol's P and M for the sets of selected pure and mixed segments, respectively, and the operators $\mathrm{anc}(s)$ and $\mathrm{dec}(s)$ to denote the sets of ancestors and descendents of a segment s. Then, the segment selection scheme can be formulated as

$$P = \{s : \quad p(s) \ \wedge \ (\forall_{t \in \mathrm{anc}(s)} : \neg p(t))\} \tag{5.1}$$

$$\begin{aligned} M = \{s : \quad & \neg p(s) \ \wedge \\ & \forall_{t \in \mathrm{anc}(s)} : \neg p(t) \ \wedge \\ & \forall_{t \in \mathrm{dec}(s)} : \neg p(t) \ \wedge \\ & \forall_{t \in \mathrm{anc}(s)} \exists_{u \in \mathrm{dect}} : p(u)\} \end{aligned} \tag{5.2}$$

P and M are segmentations, as well as $P \cup M$. Moreover, if the segmentation pyramid is complete, $P \cup M$ is a complete

2. If the selection pyramid is not complete, the resulting segmentation may be also not complete

5.3 Detailed description and case study

The entire procedure, which I will call segmentation pyramid classification, consists of five steps:

1. Data preparation, including class definition and collection of training samples
2. Segmentation of the image into a pyramid
3. Area estimation of classes within segments
4. Selection of segments from the pyramid, based on area estimates.
5. Final classification and evaluation.

The procedure is demonstrated with an example of a multi-spectral SPOT image of the island of Ameland in the Netherlands, which will be introduced first.

5.3.1 Data Preparation

Image

The input image is a window of 460×785 pixels from a multi spectral SPOT image of August 9th, 1992. The image area of approximately 9.2×15.7 km covers most of the island of Ameland, to the North of the Netherlands.

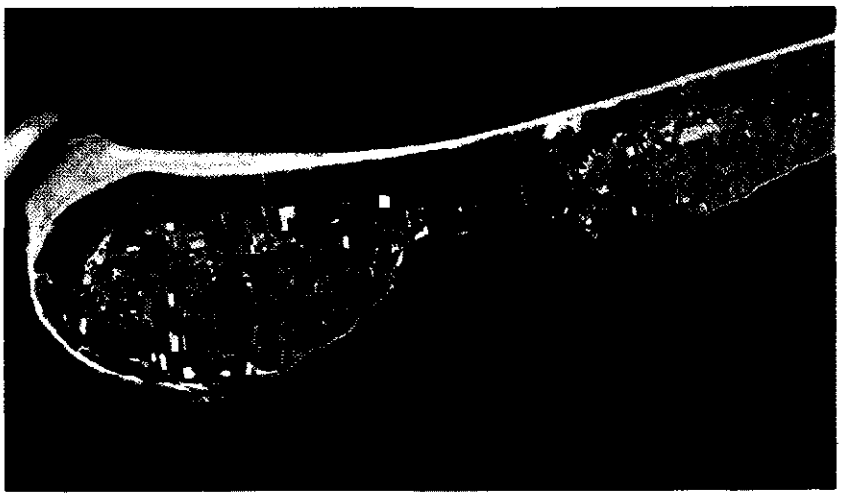

Figure 5.2: SPOT XSS Ameland, the Netherlands, August 9, 1992

No radiometric or geometric preprocessing were performed, other than the standard corrections that are done by the image distributor. Therefore, the image and the results shown do not conform to a standard map projection.

Class Selection

Based on image inspection and familiarity with the terrain, nine classes were selected (Fig. 5.4).

Shallow water was chosen, because large reflection differences were noticed in the sea. Parts of the *Waddenzee*, the sea between the island and the main land, are very shallow or dry during low tide. However, the distinction between *shallow water* and *sea* was considered irrelevant for the application. After the classification, these classes will be combined, such that confusion between them will not influence the classification evaluation.

The class *bare soil* refers to recently ploughed agricultural fields. This class is spectrally similar to *beach*. Also, between *beach* and *dune* spectral overlap can be expected, since the dunes are partly sandy. *Marshland* is densely vegetated and predominantly located in depressions between dunes. The class *built-up* contains the four villages on the island, as well as areas for recreational habitation, such as camping grounds with semi-permanent caravans and summer houses.

A few preliminary image segmentations with different thresholds were executed. From these, a number of segments were selected that, according to visual inspection, have single class coverage (Fig. 5.3).

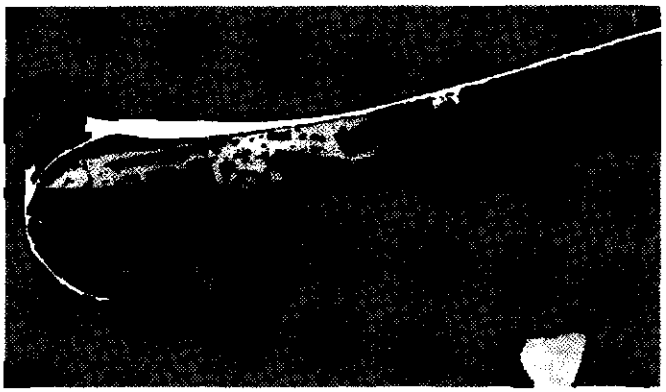

Figure 5.3: Preliminary segmentation with class assignment for training sample selection

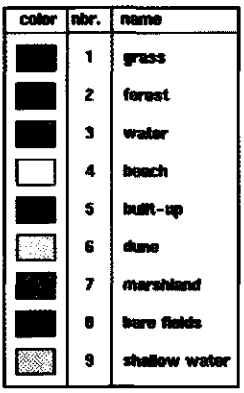

Figure 5.4: Ameland classes

After identifying one or two segments for each class, approximately 200 pixels per class were randomly chosen from these segments, to be used as training samples.

By image interpretation, without using image segmentation, a separate set of pixels was chosen for evaluating classifications.

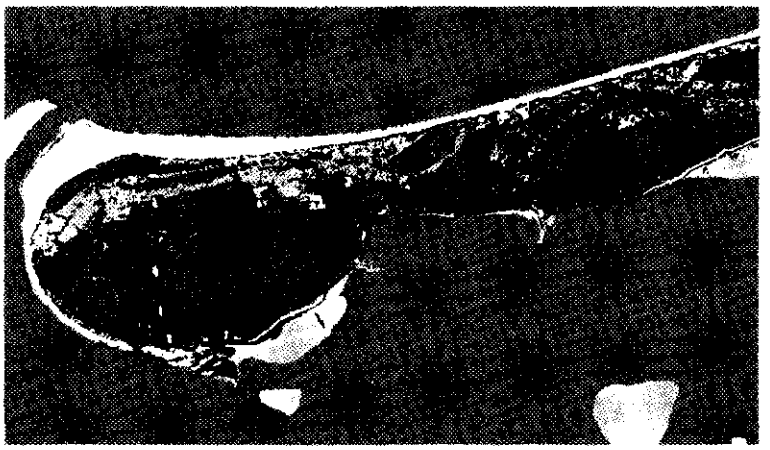

Figure 5.5: Maximum Likelihood Classification

Maximum Likelihood Classification

A standard maximum likelihood classification with equal prior probabilities serves as a yardstick, against which segmentation pyramid classification will be measured. At first sight (Fig. 5.5), many pixels appear to be erroneously classified as *built-up*. This is confirmed by the error matrix (Table 5.1). At the same time, a considerable fraction of the *built-up* evaluation set pixels were misclassified. The remaining classes perform satisfactory, confirming the suitability of the class selection.

	grass	for	water	beach	built	dune	marsh	bare	UNCL	ACC
grass	377	0	0	0	61	7	7	18	0	0.80
forest	0	189	0	0	1	1	26	0	0	0.87
water	0	0	484	0	0	0	0	0	0	1.00
beach	0	0	0	193	0	0	0	0	0	1.00
built-up	0	0	0	0	72	16	0	11	0	0.73
dune	0	0	0	0	78	332	11	0	0	0.79
marshl.	0	39	0	0	17	0	205	0	0	0.79
bare	0	0	0	0	1	0	0	60	0	0.98
REL	1.00	0.83	1.00	1.00	0.31	0.93	0.82	0.67		

average accuracy = 86.98 %
average reliability = 82.15 %

overall accuracy = 86.67 %
overall reliability = 86.67 %

Table 5.1: Error matrix of ML classification of Ameland, SPOT XS

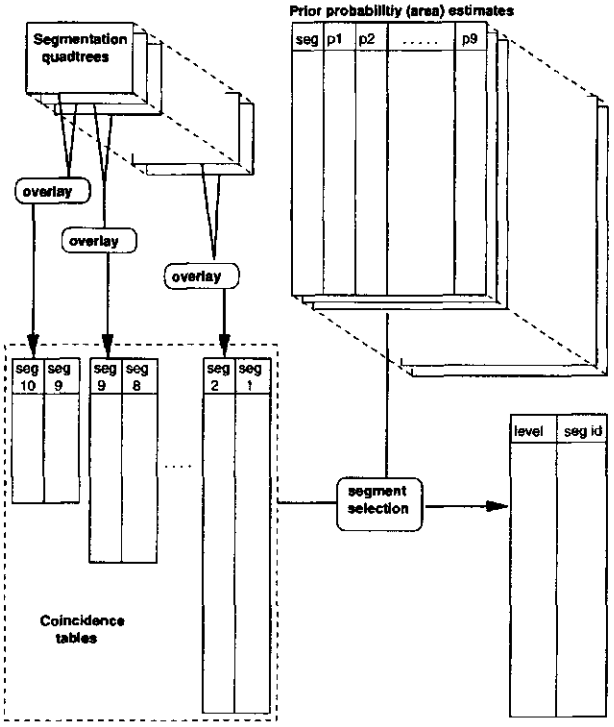

Figure 5.6: Object identification from classified segmentations

5.3.2 Building Segmentation Pyramid

The creation of a segmentation pyramid involves a number of steps, for converting the image bands into a feature vector quadtree, creating the segmentation pyramid and editing it to remove small segments (Fig. 5.7).

Quadtree processing

The three bands of the image are converted one by one from raster into quadtree files and then combined into a single feature vector quadtree. In the current implementation, the value-field of a leaf in a feature vector quadtree occupies four bytes, supporting four bands at most. In case of a SPOT image, one byte is unused.

Image segmentation

The image segmentation method, described in Ch. 4, was slightly adapted to generate a series of segmented quadtrees and attribute tables, one pair for each of a

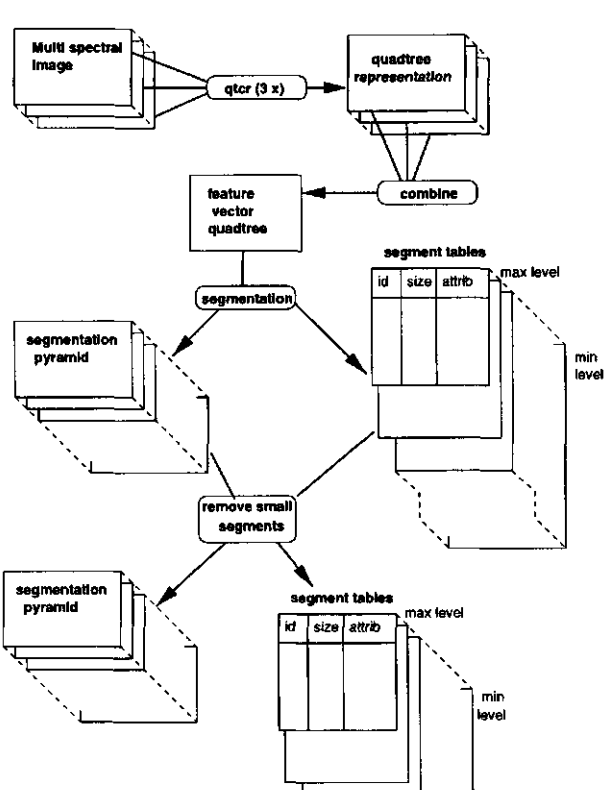

Figure 5.7: Building Segmentation Pyramid

sequence of thresholds. The method performs area based image segmentation via recursive merging, controlled by homogeneity criteria. Two adjacent segments s_1 and s_2 can be merged if their average feature vectors are close enough and if the heterogeneity (in terms of variances and covariances) after merging is not too large (see Ch. 4). The threshold value which is involved has to be specified by the user and controls the coarseness of the segmentation.

To avoid the effect of order dependency, the user may specify a sequence of ascending threshold values, after which the program acts iteratively, making a sequence of segmentations, from fine to coarse. During each iteration, the entire quadtree is recursively traversed.

After each iteration, a quadtree and an attribute table are stored. The values of

the leaves in the quadtree are unique segment identifiers, which serve as indices in the table to link segments with their attribute data (segment size and aggregate spectral properties). The output of the program forms a complete segmentation pyramid.

Whereas the threshold values (especially the final one in the sequence) for an 'optimal' image segmentation in Ch. 4 were difficult to determine, it can now be expected that the final results of segmentation pyramid classification are quite insensitive to these values. At a later stage, suitable segments will be selected from the entire pyramid and all the other segments will be discarded. A few practical considerations still exist, however.

- The first (smallest) value in the sequence determines the initial number of segments and, therefore, the memory requirements for the segment attribute table. In this table, several operations (insertion, combination, re-ordering) are performed that require random access. If the table size exceeds the available physical memory, the entire process slows down dramatically.
- Large threshold values produce a single segment for the entire image. This determines an upper bound for the threshold value.
- The length of the sequence determines the number of levels in the pyramid. Each level will, later in the process, be submitted to non-parametric iterative class area estimation algorithm, which is a quite costly operation.

For the SPOT image in this case study, a sequence of ten spectral distance threshold values was used, ranging from 10 to 105. The intermediate values were chosen in such a way that the number of segments between each pair of successive iterations decreases by approximately 50 % (Table 5.2).

When linking the segmentation quadtree of a certain level with the average feature vector columns of its attribute table, a segmented color composite can be made (Fig. 5.8).

Small Object Removal

Chapter 4 describes that the area based recursive quadtree merging algorithm produces many small segments. In the Ameland case, in all segmentation levels, except level 10, more than 60 % of the segments have a size of 5 pixels or less. They cover only a few percents of the image area, however. (Table 5.2).

In this example, small segments are removed. In general, the following considerations play a role:

- Small segments occur at boundaries between contrasting objects. The boundary pixels are mixed and they may be not similar enough to either of the two segments that correspond to the objects.

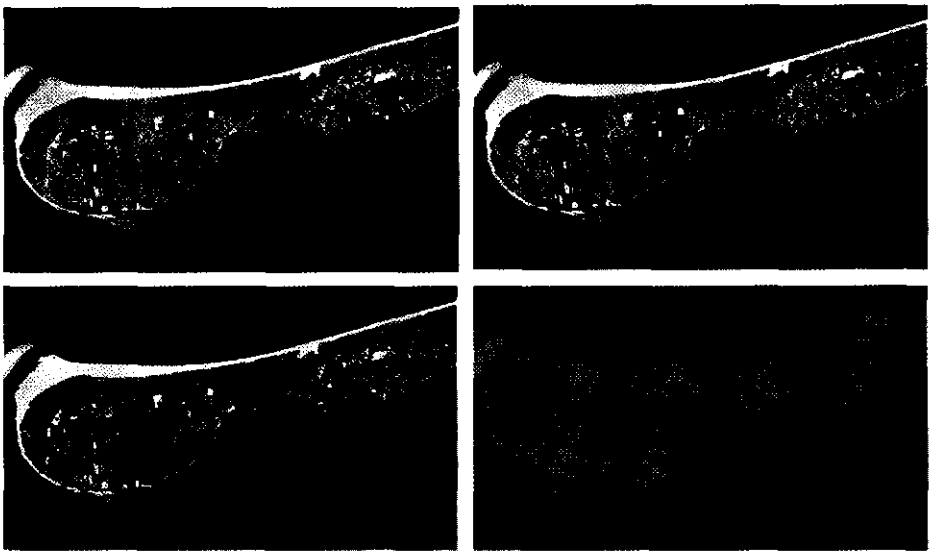

Figure 5.8: Levels 1, 3, 6, and 10 of segmentation pyramid of the Ameland XSS image (Fig. 5.2)

- The statistical area estimates are expected to be unreliable when segments become (very) small, especially if they contain a mixture of classes, as boundary pixels do.
- Whether small segments are expected to represent relevant object depends on the application and on the spatial image resolution.

To remove small segments, a *new_segment_id* column is added to the attribute table. Segments with a value less than 6 in the existing *npix* column get a *new_segment_id* of zero. The larger segments get a unique positive *new_segment_id* value. Next, the leaf values in the segmentation quadtree, which are indices in the attribute table, are replaced by the *new_segment_id* values. Finally, the small segment records (with *new_segment_id* = 0) are removed from the table. Now, the quadtree values can again be used as indices in the table.

The result of small segment removal at all levels is a segmentation pyramid which is not complete, according to the definition in Chapter. 5.

Pyramid creation

The result of the previous section is a collection of segmentations at different levels, with for each segment in every segmentation a set of area estimates for the different classes. However, the spatial coincidence of the different segmentations is not explicitly described in a data structure. It would have been possible to have this

level	threshold	segments	small segments	area (perc).
1	10	30951	27662	11.72
2	15	15320	12931	5.89
3	20	8477	6893	3.25
4	25	4892	3856	1.85
5	30	3021	2272	1.12
6	35	2016	1476	0.72
7	45	996	677	0.34
8	55	551	359	0.19
9	75	251	155	0.09
10	105	22	8	0.01

Table 5.2: Homogeneity criterion threshold values and number of segments in each level of the segmentation pyramid of the Ameland test image. The fourth and fifth columns show the number of small segments (with less than 6 pixels) and the percentages of the total image area covered by these segments.

description be output by the segmentation algorithm, because in this algorithm it is known which segments at a certain level are part of a new one at a higher level. However, this turned out to unduly increase the complexity of the algorithm. It is easier to re-create the coincidence relationships between the segmentations now, by overlaying the segmentation quadtrees of successive levels two by two into coincidence tables (Fig. 5.6).

5.3.3 Classification and Area Estimation

Each level of the 10-level segmentation pyramid, obtained in the previous section, is submitted to iterative class area estimation, to establish which segments are *pure* in terms of class membership and which ones are *mixed*. This section describes the class area estimation process (Fig. 5.9). On the basis of the results, a selection of segments from different pyramid levels will be made to form the final labeled segmentation.

Local class probability density estimation

Because segments are created on the basis of spectral homogeneity, local probability density estimation (section 3.3) has to be used.

Local probability density estimation in a segment needs to know which subset of training samples of each class is used to classify the segment. For a practical implementation the image is processed segment by segment, instead of line by line or leaf by leaf. Therefore, the segmentation quadtree is converted into a table with one record per pixel and three columns *segment_id*, *line_nbr* and *column_nbr*. The

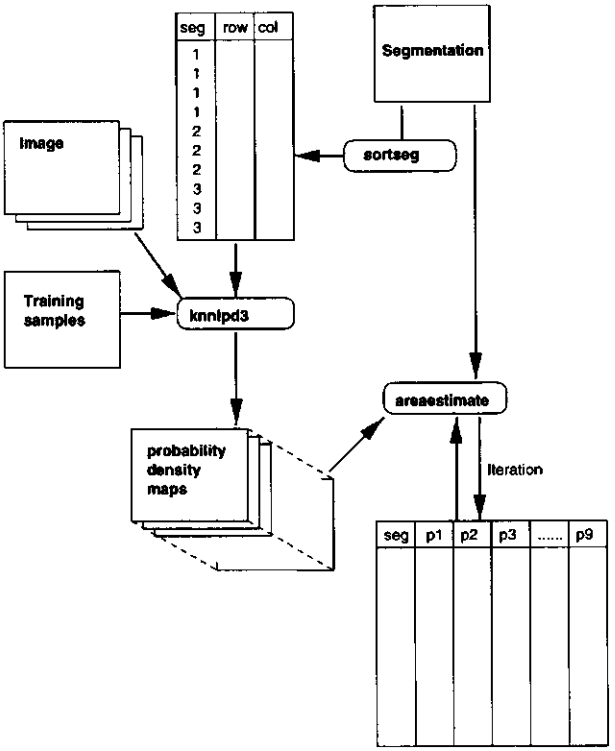

Figure 5.9: Class area estimation in segmentation pyramid

table is sorted on *segment_id*. The sorted table determines the processing order in the next stage.

Iterative local class prior probability estimation

According to sections 3.1 and 3.2, for each image segment the vector of prior probabilities $\mathrm{P}(C_i)$ can found by iteratively solving

$$\mathrm{P}(C_i) = \frac{\sum_{p=1}^{n} \mathrm{P}(C_i|\mathbf{x}_p)}{n} = \frac{\sum_{p=1}^{n} \frac{\mathrm{P}(\mathbf{x}_p|C_i)\mathrm{P}(C_i)}{\mathrm{P}(\mathbf{x}_p)}}{n}, \tag{5.3}$$

where $\mathrm{P}(\mathbf{x}_p|C_i)$ is the local class probability density for class C_i inside the segment, and n is the segment size in pixels.

The program that implements this equation, needs as input the class probability density maps from the previous section, the segmentation map, and an initial table

of prior probabilities. This table has as many rows as there are segments in the segmentation and one column per class. The initial values do not have to be chosen very carefully. Automatically initializing the table with values $\frac{1}{N}$ (with N classes) is adequate.

The output of the program is an updated prior probability table, which can be used as input for the next iteration. Convergence is usually quick: after ten iterations the difference between values in input and output table is negligible.

Implementation of 5.3 is straightforward. Inside the algorithm, the *a posteriori* probabilities $P(C_i|\mathbf{x}_p)$ are calculated. Optionally, using these probabilities, the program outputs a maximum *a posteriori probability* classification.

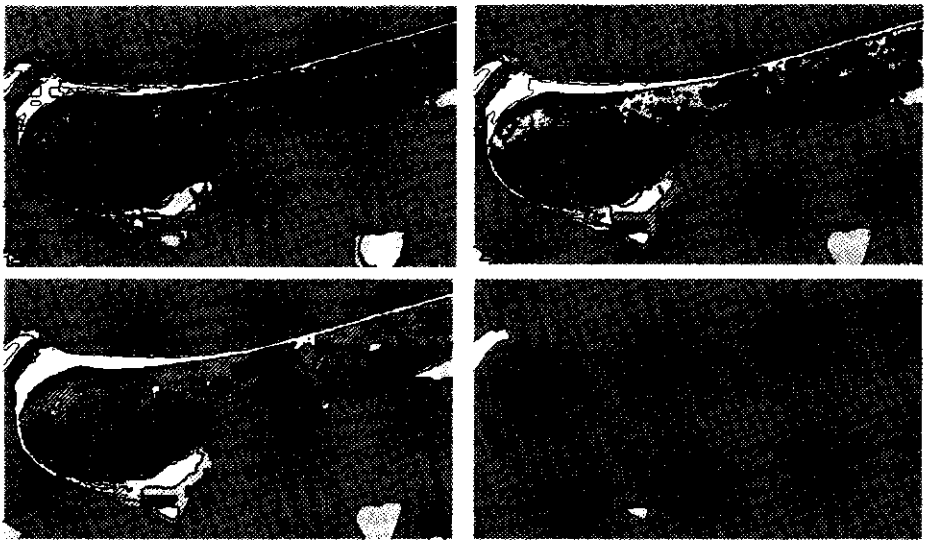

Figure 5.10: Classified Levels 1, 3, 6, and 10 of segmentation pyramid, where colors are according to the class with the highest prior probability in each segment. Hatching indicates segments containing class mixtures

5.3.4 Segment selection

This section combines segmentations and per-segment class area estimates, by selecting segments from the pyramid, such that the entire area is covered. Where possible, *pure* segments are selected, which are dominated by a single class. If, at a certain location, pure segments exist at different levels of the segmentation pyramid, the one at the highest level is selected. At locations where no pure segments exist, *mixed* segments will be placed, again from the highest possible level in the tree.

Two approaches were tested to implement segment selection: recursive tree traversal and relational algebra query. Although they are equivalent implementations of the current segment selection model, it is difficult to predict which of them is more suitable when the model is refined. Therefore, both solutions are described below.

Recursive tree traversal

Tree traversal is implemented by a recursive function, which processes segment s at level l of the segmentation pyramid. The function handles the subtree that may exist below s by invoking itself for each child of s. The function return value can be PURE, MIXED or DONE.

PURE is returned when the contribution of a single class to the area of s exceeds a threshold t. The function ignores descendents of s and outputs a data tuple $[l, s]$.

MIXED is returned when s (being not PURE) has no children, or when all its children appear to be MIXED after recursive function invocation.

DONE is returned otherwise: s is not PURE, has children s_i at level $l-1$ and not all of them are MIXED. Now, the function has to output data tuples $[l-1, s_i]$ for all PURE and MIXED children.

After transforming the segmentation pyramid into a tree, by adding a dummy segmentation of one segment above the highest level, the function is invoked once, with this dummy segment as argument.

Segment selection by recursive tree traversal yields a list of [level, segment pairs]. From the prior probability tables, on the other hand, predominant classes can be derived, which are the classes with the highest and the second highest probability in each segment. The locations of the selected segments, finally, are stored in the segment quadtrees. The data can be combined in a quadtree map calculation, overlaying quadtrees that are linked with predominant class tables, to visualize the selected objects (Fig. 5.11).

The *level* field in the selected segment list points to a segmentation quadtree and the *segment* field corresponds to the values of the quadtree. Together they identify the spatial extent of a segment. The selected segment list also identifies a record in one of the predominant class tables. It contains one or two classes, for pure and mixed segments, respectively, which determine the color of each pixel for visualization (Fig. 5.12).

Relational tuple calculus

Since the relevance of objects depends on user requirements, a more flexible selection mechanism was implemented as well. Selection can be regarded as a query on a collection of relational tables:

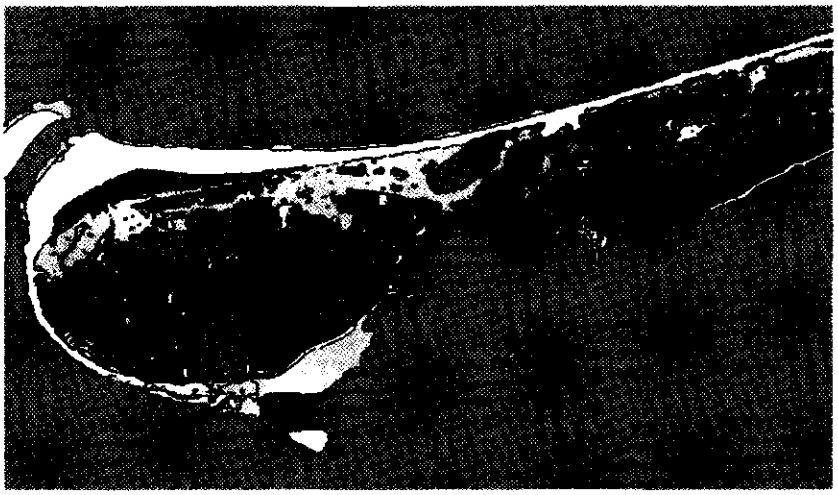

Figure 5.11: Selected segments, colored according to predominant classes, with added object boundaries

Segment tables belonging to the different segmentation levels in the pyramid

Coincidence tables which define the pyramid structure by describing coincidence relations between successive levels

Area estimate tables or prior probability tables, containing relative class areas in each segment of the entire segmentation tree.

Such queries can be formulated by users and executed in a relational database management system environment. For the relatively simple selection scheme of 5.2 and 5.2, the queries are given using relational tuple calculus.

Relational tuple calculus is a database language based on first-order predicate logic, tuple variables and relational tables (sets of k-tuples). Its most important expression form is the predicative set generation $\{t \in T \mid \varphi(t) \bullet f(t)\}$, which can informally be defined as "for those tuples t in the table T that satisfy $\varphi(t)$, evaluate the tuple generator function $f(t)$ and put the result in the result table". It allows to experts a large number of queries, which can all be translated into SOL expressions.

The segmentation process generates segments, which are collected in the set S. At the same time, we maintain a parent-child relation $A \subseteq S \times S$, which records the segment s at pyramid level i and its direct descendent s' at level $i-1$ as a pair (s, s'). From A, we may obtain its (non-reflexive) transitive closure A^+, which records the full ancestry lineage (Fig. 5.13).

We note in passing that $^+$ is not a standard relational algebra operator, but that many of the newer database systems are now starting to incorporate it. (It is well known that $^+$ cannot be expressed in terms of the standard operators.)

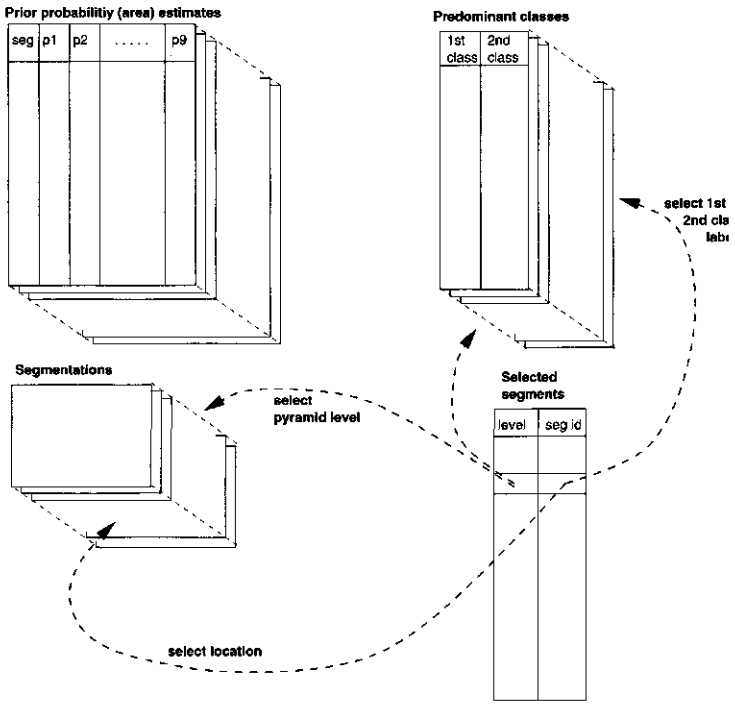

Figure 5.12: Linking selected segments, segmentation quadtrees and prior probability tables for visualization

From A^+, we derive two parameterized subsets of S:

$$\text{dec}(s) \stackrel{\text{def}}{=} \{t \in A^+ \mid t.1 = s \bullet t.2\}$$

$$\text{anc}(s) \stackrel{\text{def}}{=} \{t \in A^+ \mid t.2 = s \bullet t.1\},$$

where $t.1$ and $t.2$ are there first, respectively second field of tuple t in the ancestry relation A^+ (Fig. 5.13).

Two relational tuple calculus queries select P and M, the sets of selected pure and mixed segments, according to equations 5.2 and 5.2. (Fig. 5.14). The queries contain a predicate pure(s) that indicates the conditions for which a segment s is considered pure, by inspecting the class area tables.

$$\begin{aligned} P &\stackrel{\text{def}}{=} \{s \in S \mid \text{pure}(s) \wedge \\ &\qquad \forall_{t \in \text{anc}(s)} : \neg\text{pure}(t) \bullet s\} \\ M &\stackrel{\text{def}}{=} \{s \in S \mid \neg\text{pure}(s) \wedge \end{aligned} \tag{5.4}$$

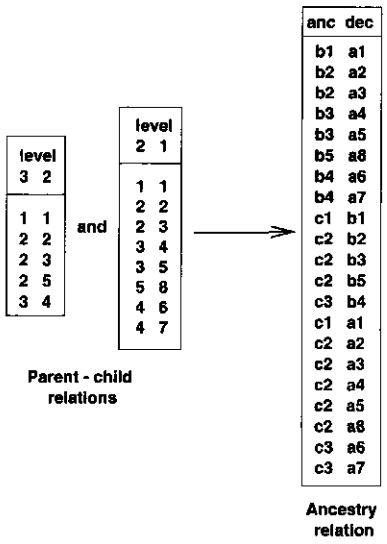

Figure 5.13: Generation of ancestry relation. Segment identifiers are a combination of a level (a, b, ...) and a segment number (1, 2, ...)

$$\begin{aligned} &\forall_{t \in \mathrm{anc}(s)} : \neg \mathrm{pure}(t) \ \wedge \\ &\forall_{t \in \mathrm{dec}(s)} : \neg \mathrm{pure}(t) \ \wedge \\ &\forall_{t \in \mathrm{anc}(s)} \exists_{u \in \mathrm{dec}(t)} : \mathrm{pure}(u) \bullet s \} \end{aligned} \tag{5.5}$$

These expressions define a rather simple segment selection scheme. Important is, however, the flexibility of the mechanism, which allows to refine the selections, according to application requirements.

For example:

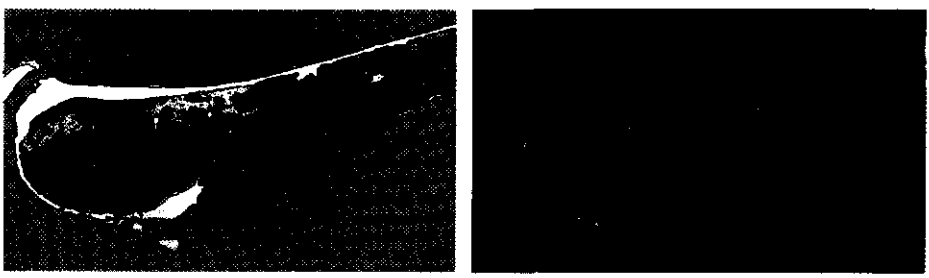

Figure 5.14: Selected pure and mixed segments

- At 30 m image resolution, a single *residential* pixel within a *forest* area of 30 pixels can be considered noise. Although there may really be a house, class assignment to an isolated pixel is unreliable, due to the mixed-pixel effect. A threshold value of, say, $t = 0.95$ adds the pixel to the forest. However, if the prior probabilities for a segment of 1000 pixels are 0.96 for *forest* and 0.04 for *residential*, a settlement covering 40 pixels (3.6 ha at 30 m resolution) may not be detected.
  Improvement is achieved by making the threshold a function of the segment size. In the example of the large forest, the selection will go deeper into the tree. Eventually, if a settlement is present, a pure *residential* segment will be found, or at least a mixed segment with a significant *residential* coverage. Otherwise, if the *residential* pixels are scattered, the segment will be split into several smaller segments that will become pure *forest*, due to a smaller threshold function value. In the latter case, a further refinement could still identify the original segment as pure *forest* (see below).
- Some land use types are characterized by a mixture of land cover classes in a typical (textural) configuration. For example, at a sufficient resolution a *residential* area, with houses, streets, gardens and trees, may be a mixture of *grass*, *bare soil* and *forest*. The selection scheme could discover that in a subtree of the pyramid the relative areas of these cover types do not change over several levels, and then decide to identify the segment at the top of the subtree as a single *residential* area.
- With the simple thresholding selection scheme, adjacent mixed segments are merged, unless their nearest common ancestor also has pure descendents. This may be undesirable when the mixtures in the segments are different, *i.e.*, involve different classes, or the same classes in different proportions.
  On the other hand, segmentation pyramid classification does not always merge adjacent pure segments of the same class, because their common ancestor may also contain segments of other classes and, therefore, be mixed.
  When judging these two situations, application requirements play a role. However, in both cases we observe that the merging decision is influenced by the presence of other (pure) segments in the neighborhood, which is unfavorable. The first case can be improved upon with a refined selection scheme, the second requires post-processing in addition.

5.3.5 Final classification

With the set of selected segments and the associated class statistics, *object* and *pixel* classifications can be made.

An object classification is a labeled map according to the predominant class in each segment.

A pixel classification is obtained by selecting the maximum a-posteriori probability

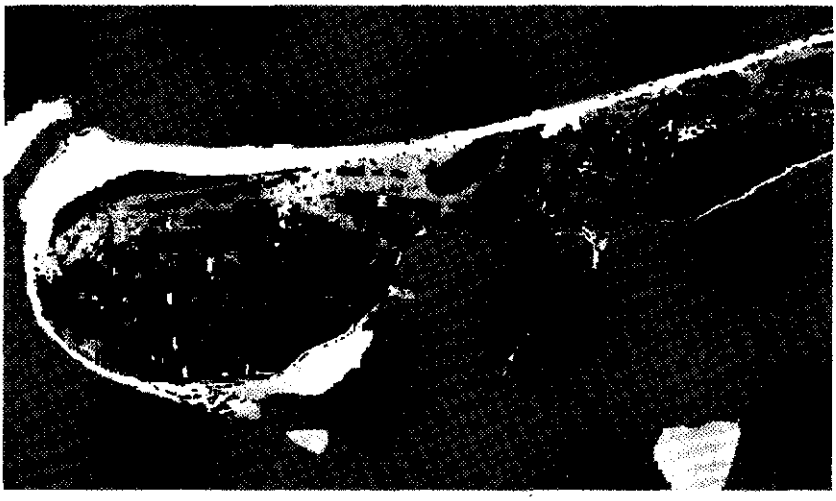

Figure 5.15: Object classification of selected segments

class, after applying Bayes' formula once more for each pixel, using local class prior probabilities and local class probability densities within the selected segments. (Fig. 5.16 and Table 5.4).

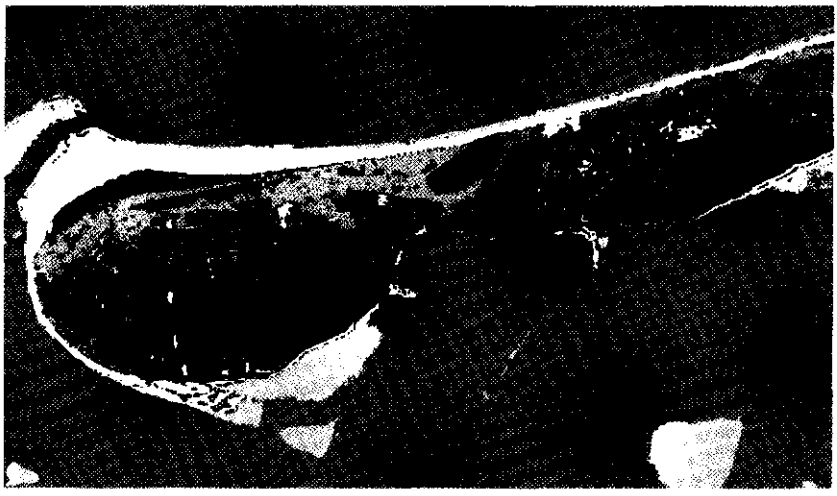

Figure 5.16: Pixel classification using local class prior probabilities and local class probability densities within the selected segments

Both error matrices (Table 5.3 and Table 5.4) show improvement in comparison to

	grass	for	water	beach	built	dune	marsh	bare	UNCL	ACC
grass	466	0	0	0	0	0	0	1	3	0.99
forest	0	191	0	0	0	0	6	0	20	0.88
water	0	0	484	0	0	0	0	0	0	1.00
beach	0	0	0	193	0	0	0	0	0	1.00
built-up	0	0	0	0	83	0	0	0	16	0.84
dune	0	0	0	0	12	389	14	0	6	0.92
marshl.	0	37	14	0	19	0	165	0	26	0.63
bare	0	0	0	0	0	0	0	61	0	1.00
REL	1.00	0.84	0.97	1.00	0.73	1.00	0.89	0.98		

average accuracy = 90.83 %
average reliability = 92.67 %

overall accuracy = 92.11 %
overall reliability = 95.18 %

Table 5.3: Error matrix of object classification of selected segments (see Fig. 5.15)

	grass	for	water	beach	built	dune	marsh	bare	UNCL	ACC
grass	427	0	0	0	23	1	0	16	3	0.91
forest	0	191	0	0	0	0	6	0	20	0.88
water	0	0	484	0	0	0	0	0	0	1.00
beach	0	0	0	193	0	0	0	0	0	1.00
built-up	0	0	0	0	74	1	0	8	16	0.75
dune	0	0	0	0	12	389	14	0	6	0.92
marshl.	0	40	8	0	16	0	171	0	26	0.66
bare	0	0	0	0	0	0	0	61	0	1.00
REL	1.00	0.83	0.98	1.00	0.59	0.99	0.90	0.72		

average accuracy = 88.94 %
average reliability = 87.63 %

overall accuracy = 90.21 %
overall reliability = 93.21 %

Table 5.4: Error matrix of pixel classification with local statistics (see Fig. 5.16).

conventional maximum likelihood classification (Table 5.1). Visual inspection gives the impression that pixel classification provides more detailed information than object classification. Apparently, according to the error matrices, this extra detail does not contribute to classification accuracy and reliability. This is partly caused by errors due to noise and mixed pixels in the pixel based classification. Obviously segmentation succeeds to avoid such errors. However, the error matrices depend on criteria concerning noise and mixed pixels that were used while creating the evaluation set. For example, when a forest area contains a few isolated pixels that look different from the surrounding, three actions can be taken:

1. They can be considered noise and added to the forest, which is favorable for the evaluation of object classifications.
2. They can be considered an object of a different class. The class itself is diffi-

cult to determine from the image and additional ground truth is required. If such objects are relevant for the application, pixel classification is preferable, although errors are likely to occur.

3. They can be excluded form the evaluation set. However, any classification looks better if "difficult" pixels are not evaluated.

The choice between the first and the second strategy depends on application requirements and on the relation between resolution, object size and desired map scale. The third alternative creates data that should not be used as an evaluation set.

5.4 Conclusions

By integrating segmentation and classification, this Chapter describes image analysis involving spatial and spectral image characteristics.

Comparison of the error matrices shows that one of the purposes of segmentation, to lead to classification improvement [Schoenmakers, 1995], was reached.

The goal of segmentation pyramid classification, however, is not just to make a classification, but to delineate and identify GIS data base objects. It distinguishes between pure (crisp) and mixed (fuzzy) objects and provides more statistical information than can be expressed in a classification map.

Chapter 6

Decision analysis

Increasingly, remotely sensed data are used for taking decisions in geographical information systems. Decision making can in principle be based on a classification of such remotely sensed data into nominal information classes. Such a classification, however, typically includes errors and uncertainty. Moreover, when processing spatial data for decision making, not only uncertainties inherent in these data but also objectives and preferences of the decision maker have to be taken into account. This Chapter proposes to exploit concepts from mathematical decision analysis for integrating uncertainties and preferences. It aims to solve complex decision problems on the basis of remotely sensed data. The feasibility of the decision-analytic approach to the interpretation of spatial data is demonstrated by means of a case study.

6.1 Introduction

Classification generally introduces uncertainty in the information classes assigned to the spectral objects. This uncertainty propagates through the subsequent stages of the decision making process [Lunetta et al., 1991]. The uncertainty can be reduced by using ancillary data and information, usually derived from sources such as domain experts, maps, field work, aerial photographs, or thematic maps from former classifications. Such evidence can be exploited before, during, and after classification and hence contribute to its accuracy in various different ways [Strahler, 1980]. Despite all efforts to reduce uncertainty introduced by classification, imperfections may still seriously affect the adequacy of using classification results for taking environmental decisions. For example, the commonly used *maximum a posteriori probability classification* discards useful information that may serve to yield insight into the uncertainties. In this approach to classification, the posterior probabilities that are computed for each spatial object within an information class distinguished during sampling, are used *only* to select the most likely class. The entire probability

distribution for the object, however, reflects highly valuable information about the *extent* and *distribution* of uncertainty which could be further utilized in a GIS.

If decisions are to be made on the basis of remotely sensed data, uncertainty tells only part of the story: the *objectives* to be pursued with interpretation of the data become crucial. In the presence of uncertainty, the best decisions are those that, in view of the objectives, carefully weigh the *benefits* of *correct* interpretation of the data on the one hand and the *losses* due to *incorrect* interpretation on the other hand. This idea is illustrated by an example dealing with fraud with subsidies assigned to agricultural crops by the European Union. In this example, the main objective is to detect illegal declarations of subsidized crops by taking remotely sensed images from crops on parcels, to avoid waste of public resources. From this objective alone, the number of detected illegal declarations should be maximized. However, unjust implication of fraud is highly unfavorable as it results in extra costs for verification and in loss of face. Therefore, the number of unjust implications should be kept to a minimum. In pursuing both objectives simultaneously, overlooking fraud is considered worse than over-estimating. It now depends on the probabilities computed for the various possible crops for a parcel under consideration whether or not fraud should be implied. Interpretation of remotely sensed data for decision making therefore involves both the extent and distribution of uncertainty introduced by classification and the preferences of the decision maker. These preferences concern the objectives that are being pursued with the interpretation and therefore differ from knowledge about the *subject* of the interpretation as referred to by [Strahler, 1980]. Both types of knowledge equally contribute to the interpretation, yet at different levels.

Further elaborating on the idea that remotely sensed data can serve as a basis for decision making, the question arises whether or not it is necessary to derive a complete classification before considering viable decisions. In principle, decisions can be taken on the basis of a classification. However, classification contains errors and uncertainties of which the extent and distribution are unknown. By making

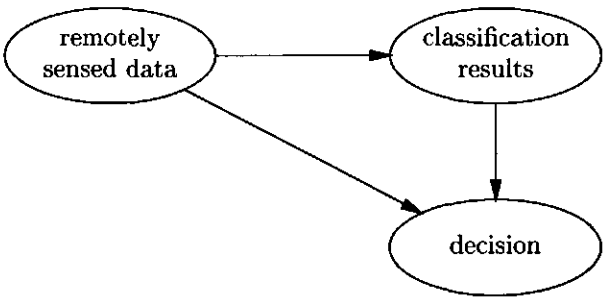

Figure 6.1: Founding decision making on data

decision directly on the data, full knowledge about the uncertainties involved can be included, thereby allowing for making better decisions. As decision making does not so much involve classification results as the extent and distribution of uncertainty introduced by classification, deriving a complete classification is no longer required and, in fact, has become obsolete (Figure 6.1). However, an accurate classification nevertheless serves various purposes beyond decision-making.

This Chapter addresses the interpretation of remotely sensed data in view of the objectives that are to be pursued when exploiting the data for decision making. To this end, various concepts from decision analysis are introduced which allow integration of uncertainties and a decision-maker's preferences. Section 2 expresses the interpretation of remotely sensed data as a decision problem and introduces the mathematics for solving this problem. Section 3 describes the assessment of the various parameters involved in quantification of uncertainties and preferences. A case study will be presented in Section 4, demonstrating the feasibility of the decision-analytic approach.

6.2 Interpretation of data: a decision problem

Interpretation of remotely sensed data is in essence a *decision problem*: the problem is to decide upon which decision to take for each spatial object on the basis of available data. The solution to this problem is for each object the decision that is expected to best meet the objectives that are being pursued with the interpretation. The field of *Decision analysis* provides the mathematical framework for solving complex decision problems such as the data-interpretation problem [Raiffa, 1968, von Winterfeldt & Edwards, 1986, Smith, 1988]. It offers means for structuring decision problems and for computing solutions. In this section, we express the interpretation problem and its solution in decision-analytic terms.

A decision problem involves two types of variable:

- a *decision variable* is a variable that represents viable decisions or actions that can be taken in the context of the problem at hand;
- a *chance variable* is a variable that represents the true 'state of the world'; the value of such a variable cannot be selected by the decision maker.

In the data-interpretation problem, there is only one variable of each type: a chance variable C that represents the true information class of a spatial object O and a decision variable D that represents the possible decisions that can be taken with regard to this object.

A variable in a decision problem can take its value from among a pre-defined set of values. We assume that $C_1, \ldots, C_N$, $N \geq 1$, are possible information classes of O. These classes therefore are values for the chance variable C. We further assume

that the decision variable D takes its value from among the decisions $D_1, \ldots, D_m$, $m \geq 1$.

In a decision problem, there typically is uncertainty regarding the true values of the chance variables involved. In data interpretation, there is uncertainty concerning the true value of the chance variable C since the true information class of O is unknown at the time of interpretation. This uncertainty is expressed as a probability distribution $\mathrm{P}(C)$ for the variable C, specifying for every possible information class C_i the probability $\mathrm{P}(C = C_i)$ that C_i is the true class of the object. Note that this probability distribution will not be influenced by the various decisions that can be taken.

In addition to uncertainties, a decision problem involves preferences. The desirability of a decision and its consequences, with each other called a *scenario*, is quantified by means of its *utility*. In our data-interpretation problem, each combination of a decision $D = D_i$ and a true information class $C = C_j$ has associated a utility $u(D = D_i \wedge C = C_j)$. The utility expresses the desirability of the scenario where the decision D_i is taken with regard to a spatial object, while it has C_j as its true information class. Actual utilities associated with the various scenarios depend upon the objectives that are being pursued with the interpretation.

Structuring all aspects of a decision problem can be done with a *decision tree*. A decision tree is a pictorial, tree-like representation of the problem. The various variables and values of the problem are organized in a (rooted) tree. Each node in the tree models a variable; the edges emerging from a node represent the values of its associated variable. The topological structure of the tree is an explicit representation of all scenarios that can possibly arise from a decision. The root node of the tree represents the initial situation before any decision is taken and each path from the root node to the tip of a terminal edge corresponds with a scenario. Figure 6.2 shows a tree organizing the variables of our object-interpretation problem. To distinguish between the decision and chance variable, the former is depicted as a square box and the latter is shown as a circle. In the tree, the uncertainties concerning the chance variable's values are depicted with the appropriate edges; the utilities are depicted at the tips of the terminal edges of the tree.

Once a decision problem has been structured in a decision tree, the best decision for the problem is easily computed. For this purpose, the tree is evaluated by *fold-back analysis*. Fold-back analysis starts at the tips of the terminal edges, works its way through all intermediate nodes and edges, and ends at the root of the tree. In fold-back analysis, for each viable decision the desirability of taking this decision is computed. The desirability of a decision depends on the values of the chance variables modeling its consequences. However, these values are not known before the decision is taken. The desirability of a decision therefore is computed by *weighting* the utilities of the various possible scenarios that can arise from taking this decision with the probabilities that these scenarios actually do occur. For each

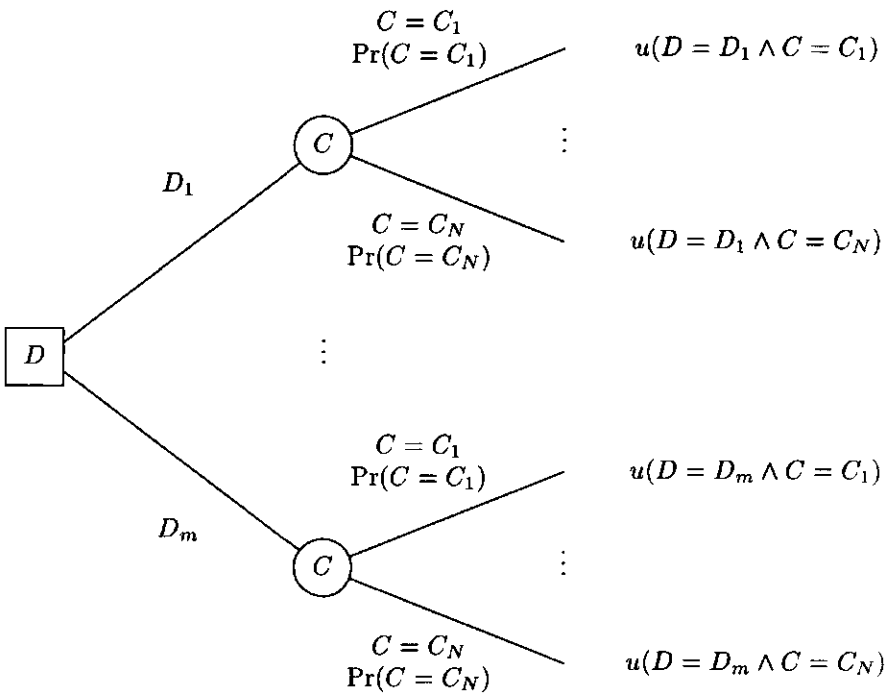

Figure 6.2: A decision tree for the data-interpretation problem.

chance variable, the *expected utility* over its values is computed, which expresses the expected utility of taking the decision corresponding with the incoming edge of the node modeling the chance variable. For each decision variable, the *maximum expected utility* over its values is computed. In a fold-back analysis of the decision tree for the data-interpretation problem, the expected utility $\hat{u}(D = D_i)$ for each decision $D = D_i$ is computed as

$$\hat{u}(D = D_i) = \sum_{j=1}^{n} u(D = D_i \wedge C = C_j) \cdot \mathrm{P}(C = C_j).$$

The best decision is the decision D_k with the highest expected utility. Computing the best decision with regard to a spatial object as outlined before will be coined *decision-analytic data interpretation*.

The statistical description of decision analysis provides a general and flexible framework for data interpretation. In fact, the framework also provides for conventional classification by taking for the decision the various possible information classes; the utilities then express the severity of different types of misclassification. As an

example, we express the common *maximum a posteriori probability classification*. The only objective pursued in maximum a posteriori probability classification is to maximize the probability of correct classification: every misclassification is considered equally undesirable. This objective can be expressed in terms of utilities by taking $u(D = D_i \wedge C = C_i) = 1$, for all $i = 1, \ldots, N$, and $u(D = D_i \wedge C = C_j) = 0$, for all $i, j = 1, \ldots, n, i \neq j$, where D_i is the decision to assign class C_i to a spatial object.

6.3 Assessing parameters

For decision-analytic data interpretation, a decision tree to model the interpretation problem is evaluated. This decision tree includes the various uncertainties and preferences involved. The accuracy of the assessment of these quantities directly determines the quality of the decision computed for the problem. This section briefly addresses the assessment of the quantities required for the decision-analytic approach.

6.3.1 Probability assessment

The uncertainties involved in data-interpretation are expressed as probability distributions over the various information classes distinguished for a spatial object under consideration. The probabilities in these distributions are computed from remotely sensed data as posterior probabilities given the spectral attributes of these data. As before, given a vector $\mathbf{x}$ of spectral attributes, for each information class C_i, $i = 1, \ldots, N$, the posterior probability $\mathrm{P}(C = C_i \mid \mathbf{x})$ is computed using Bayes' formula:

$$\mathrm{P}(C = C_i \mid \mathbf{x}) = \frac{\mathrm{P}(\mathbf{x} \mid C = C_i) \cdot \mathrm{P}(C = C_i)}{\mathrm{P}(\mathbf{x})}$$

where $\mathrm{P}(\mathbf{x} \mid C = C_i)$ is the probability that the vector of spectral attributes $\mathbf{x}$ occurs in the data given that the true class of the object is C_i. $\mathrm{P}(C = C_i)$ is the prior probability that the object has class C_i for its true class and $\mathrm{P}(\mathbf{x})$ denotes the probability of the vector $\mathbf{x}$ occurring in the data. $\mathrm{P}(\mathbf{x})$ is the same for every information class and does not have to be computed independently: $\mathrm{P}(\mathbf{x})$ is obtained by normalizing the enumerators of the right-hand side of the formula over all information classes. The probabilities $\mathrm{P}(\mathbf{x} \mid C = C_i)$ and $\mathrm{P}(C = C_i)$, however, have to be assessed explicitly for each class C_i.

Since the method works on *a posteriori* class probability values, rather than on labeled pixels, the accuracy of those values is very important. Therefore, depending on the availability of ancillary data, the algorithms, described in the previous Chapters can be fully exploited here:

- Non-parametric estimation of (local) class probability densities

- Iterative estimation of class prior probabilities
- Region-merging multi-spectral image segmentation
- Integration of the above (segment pyramid classification).

6.3.2 Utility assessment

The utilities of a decision problem are derived from the objective which is pursued and express the desirability of the various scenarios that can arise from a viable decision. In most decision problems several different objectives are pursued simultaneously. Therefore, a utility can be a complex combination of quite different commodities, such as monetary gain, status, and time. Decision analysis offers various, more or less formal, methods for performing this task [von Winterfeldt & Edwards, 1986].

The simplest, and least formal, method for utility assessment is to *visualize* all possible scenarios of a decision problem on a linear scale. The least desirable and the most desirable scenarios are identified and assigned to the ends of the scale. Every other scenario is now positioned on the scale, where the distance between two scenarios is indicative of the difference in desirability between these scenarios. Once all scenarios have been positioned, for each scenario a utility is yielded by projecting its position onto a matching numerical scale. Figure 6.3 illustrates the basic idea for two scenarios s_i and s_j.

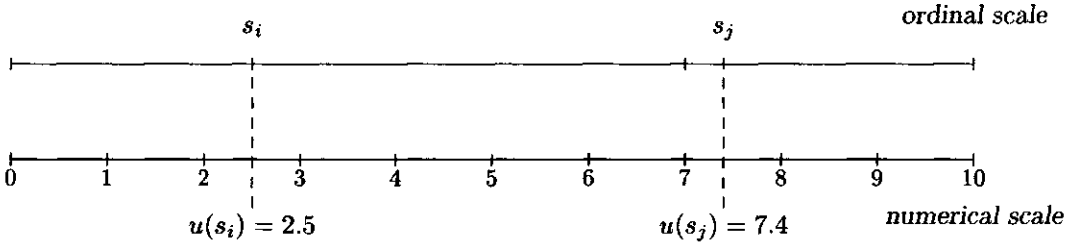

Figure 6.3: The visualization method for utility assessment.

Instead of first visualizing the differences in desirability among scenarios, these differences can be quantified directly, by using a *standard reference gamble*. A standard reference gamble serves for comparing three scenarios with regard to their desirability. Let s_i, s_j, and s_k be scenarios such that s_i is less desirable than s_j, and s_j in turn is less desirable than s_k. In assessing utilities for these three scenarios, a probability p is found such that scenario s_j is as desirable as a gamble that yields scenario s_k with probability p and scenario s_i with probability $1 - p$. Through this probability p, the utilities $u(s_i)$, $u(s_j)$, and $u(s_k)$ have now been assessed to satisfy

$$u(s_j) = p \cdot u(s_k) + (1 - p) \cdot u(s_i)$$

By using the standard reference gamble for appropriate three-tuples of scenarios, a system of equations is obtained from which a set of utilities is computed. The use of a standard reference gamble tends to yield better calibrated utilities than the visual method; the method, however, is more time-consuming.

If the utilities of a decision problem are composed of various commodities that are hard to compare, utility assessment can be especially cumbersome. The assessment then often is simplified by decomposing the utilities into their separate commodities. In terms of these separate commodities, *marginal utilities* are assessed, for example using one of the techniques outlined above. These marginal utilities subsequently are combined to yield overall utilities [von Winterfeldt & Edwards, 1986].

6.4 A case study

The decision-analytic approach to data interpretation has been applied to a case study. Although the situation described in the study in itself is hypothetical, it emerges from a real-life issue. The study concerns fraud with subsidies provided by the European Union to support the cultivation of certain agricultural crops. These subsidies are paid on the basis of declarations submitted by farmers. A fraud detection mechanism can make use of remotely sensed data. For each parcel, the viable decisions to consider on the basis of the data concern approval of the declaration on one hand, and an implication of fraud followed by further investigation on the other hand.

The study area is located around the village of Biddinghuizen in the province of Flevoland, the Netherlands. A Landsat Thematic Mapper image of the area is available (we used spectral bands 3, 4 and 5) from June 1987, as well as crop maps from 1986 and 1987. Seven different land-cover classes are distinguished: grass, wheat, potatoes, sugar beets, peas, beans, and onions. The crop maps, originating from an initial survey that included interviews with farmers, likely contain errors and uncertainties. In our study, we have used the 1986 map to calculate local prior probabilities. In the calculation, crop rotation cycles have been taken into consideration; so, the land-cover classes in successive years are not independent. Part of the 1987 crop map has been used for training sample selection, in combination with a color composite of the image. From the 1987 map we have subsequently extracted the fields with peas or beans, and considered them as farmers' declarations for subsidy on those two crops.

To investigate viable decisions, various utilities have been assessed. The decision to imply fraud and suggest further investigation is very advantageous if the farmer's declaration specifies peas or beans and there is a different agricultural crop reality: this scenario uncovers an illegal declaration. The scenario is assigned a utility of 10. The decision to not inspect such as field is extremely bad. This scenario is assigned

a utility of 0. If a declaration turns out to be legal after further investigation, we have put ourselves (or the farmer) through unnecessary trouble. However, an investigation that turns out superfluous is not so bad as overlooking a false declaration. This scenario therefore is assigned a utility of 3. Avoiding superfluous investigations is more advantageous anyway: we assign a utility of 8. These utilities are summarized in Table 6.1. Based on these utilities, we have applied our decision-analytic

	inspection	
crop	yes	no
grass	10	0
wheat	10	0
potato	10	0
sugar beet	10	0
pea	3	8
bean	3	8
onion	10	0

Table 6.1: Utilities for the detection of illegal farmer declarations.

method to the decision for each *pixel*. The result is a binary raster map, indicating the decision per pixel. Subsequently, a majority criterion has been applied to identify the *fields* that have been indicated for further investigation. These results are shown in Figure 6.4. Of 81 fields with a declaration of peas or beans, 22 will be inspected.

Now consider a slightly different (perhaps less realistic) situation in which the subsidies paid are rather small and the fraud detection agency is under-staffed. In this situation, farmers generally will be given the benefit of the doubt and only very suspicious looking declarations will be inspected. The utility assigned to the scenarios for this situation are shown in Table 6.2. After applying our decision-analytic method to the same data with these new utilities, the number of fields to be investigated has decreased from 22 to 16 as expected.

	inspection	
crop	yes	no
grass	10	6
wheat	10	6
potato	10	6
sugar beet	10	6
pea	0	20
bean	0	20
onion	10	6

Table 6.2: Modified utilities for the detection of obvious illegal declarations

6.5 Conclusion

Remotely sensed data are exploited to an increasing extent for decision making. For processing spatial data for this purpose, the objectives and preferences of the decision maker have to be taken into account. In principle, decisions may be taken on the basis of a complete classification of the data at hand. However, as taking the best decision involves the full extent and distribution of the uncertainty in the data, decision making is better founded directly on the data themselves. Decision-analytic interpretation, provides such an approach by integrating preferences and uncertainties in a mathematically well-founded way. The aim of the method is to assist a decision maker in taking the *best* decision and not so much to reconstruct reality, thereby contrasting conventional classification.

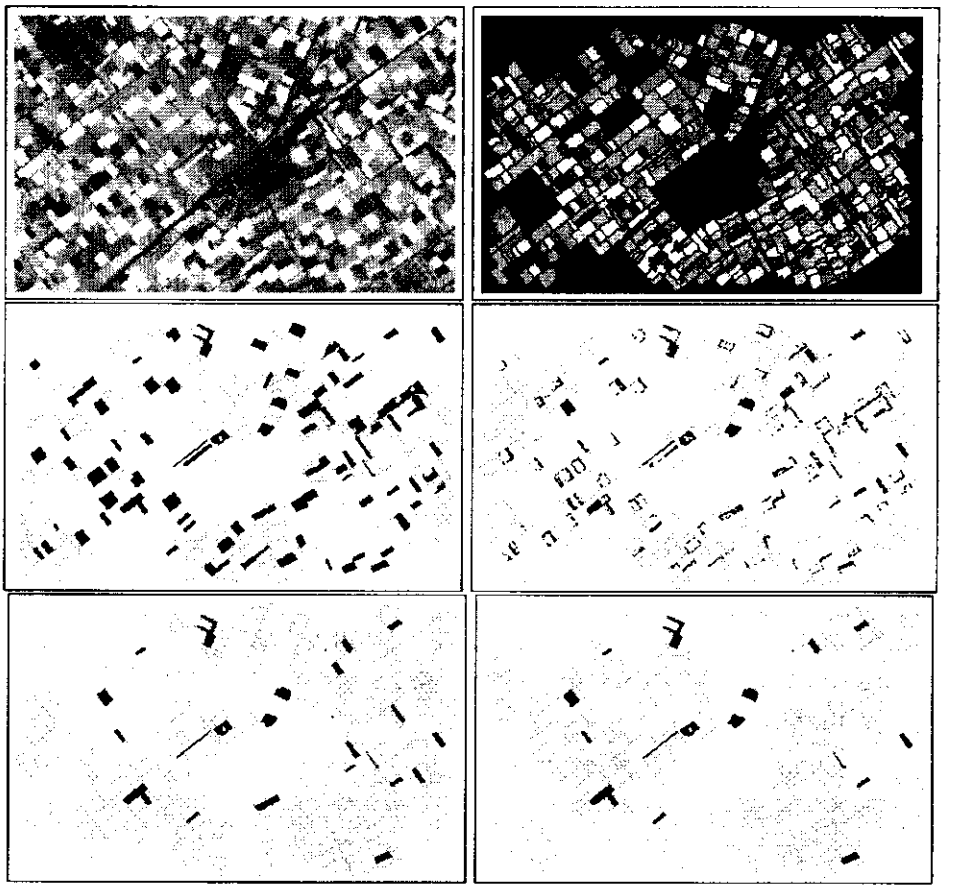

Figure 6.4: Experimental results. Upper Left: Landsat TM Image (band 4). Upper Right: Fields under consideration. Center Left: Fields with declaration for 'peas' or 'beans'. Center Right: Pixels with positive 'inspection' decision. Lower Left: Fields to be inspected. Lower Right: Fields to be inspected after modification of utilities.

Chapter 7

Conclusions and recommendations

Research for this thesis has resulted in a number of improvements in (semi-) automatic satellite image interpretation.

1. Extended k-Nearest Neighbor class probability density estimation
2. Local class probability density estimation after image regionalization
3. Iterative (local) prior probability estimation
4. Quadtree-based *region merging* segmentation
5. Segment pyramid classification
6. Decision-analytic interpretation of *fuzzy* classifications

Improvement of land-use/land-cover classifications is achieved in a variety of circumstances, depending on the availability of ancillary data and information. It appears from this study that the following topics are relevant:

Regionalization and expert knowledge: Data from Geografic Information Systems may provide a regionalization of the image area, such that different class mixing proportions occur in each region. Regionalization may be based on, for example, soil data, digital elevation models, previous land-use/land-cover data, and on certain kinds of administrative units. If the expected class mixing proportions in each region can be estimated, typically requiring *expert knowledge*, classifications can be improved by translating this knowledge into spatially distributed (local) *a priori* probabilities. Proper application of Bayes formula, however, requires that local *a priori* probabilities in a region are combined with class probability densities that are also local, *i.e.* pertain to the same region. Of course, global class probability densities, which are independent of the position in the image, can be used as estimates for local ones, but the study has shown that in the circumstance of many small and/or homogeneous regions these estimates are very poor. To estimate densities using different training sets for each region (classical stratification) is not feasible in such circumstances, because very large amounts of training

samples would be required. In such cases, local class probability density estimation from a single (global) set of training data, is preferable (Chapter 3).

Regionalization only: Often, like in the previous case, a regionalization is available, based on ancillary data, but the relation with class mixing proportions is unknown. Also now local class probability density estimation is appropriate, but more important is the method to iteratively estimate class mixing proportions per region on the basis of class probability densities. The estimated mixing proportions can be used as area estimates, when multiplied by the area of each region, and as prior probabilities in a Bayesian classification. Chapter 3 provides a mathematical foundation and demonstrates a significant classification improvement.

Without ancillary data: Extended k-Nearest Neighbor provides an estimate of unconditional feature densities, in addition to the conditional class probability densities. This opens the possibility to estimate probabilities for the *unknown* class, which, for example, is crucial when estimating areas of the "known" classes that are part of the training data.

When no ancillary data are available to supply a regionalization of the image area, it can be obtained by image segmentation. Any segmentation algorithm could be used, but usually the results suffer from *order dependency* and *region fragmentation/merging* problems. Therefore, Chapter 4 describes a new segmentation method, which avoids order dependency by making a few iterations in which the *merging criteria*, involving spectral distance and covariance, are gradually relaxed. Moreover, the method creates a segmentation *pyramid* by outputting a segmentation after each iteration. Whereas region fragmentation and merging are inherent to data-driven segmentation, these problems can be solved by segment pyramid classification, *i.e.* class mixing proportion estimation in all segments within the pyramid. This leads to integration of segmentation and classification (Chapter 5). The image is analyzed on the basis of both spectral and spatial characteristics, such that delineation and characterization of terrain objects is provided. It is important that the entire process is (almost) threshold-free. The necessity to provide spectral-distance thresholds for segmentation is greatly relaxed by integration with classification, where a suitable spectral-distance value is established automatically for each object, following a *class purity* criterion.

In each case, classification accuracies and reliabilities significantly improve, compared to Maximum Likelihood classifications.

Moreover, the algorithms yield much more information than only a classification. Regionalized class mixing proportions, as well as posterior probabilities are estimated with high precision, and could be stored in a GIS that supports objects with fuzzy class membership values, allowing for queries such as: "Show all areas which have a probability of larger than 0.8 to have changed from agricultural into indus-

trial over the past ten years". Updating the GIS data base can be implemented as updating posterior probabilities, using the current ones to establish priors and new data to derive class densities.

Chapter 6 shows how probabilistic data may be exploited to support and optimize decision making. In principle, decisions may be taken on the basis of a classification. However, taking the best decision involves the full extent and distribution of the uncertainty in the data. Decision-analytic interpretation provides an approach that integrates decision-maker's preferences and class membership uncertainties in a mathematically well-founded way. The aim of the method is to assist a decision maker in taking the best decision and not so much to reconstruct reality.

The decision-analytic approach to the interpretation of spatial data has been illustrated by means of a simple case study, which does not demonstrate the potential power of the approach. However, it illustrates the issue of *customization*: from a single set of spatial data, various results can be obtained, tailored to a decision-maker's objectives, by interpreting the data with different sets of utility assessments.

The approach is based on a well-known and long-established mathematical framework from decision analysis for solving complex decision problems. The field of decision analysis provides a wealth of methods, for example for assessing probabilities and utilities, that can be applied to the problem of interpreting image data. Thanks to its flexibility and mathematical well-foundedness, the framework has the potential to become an integral part of geographical information systems.

All methods, developed in this study, yielded algorithms that were carefully implemented and can be readily applied. Although the programs, for example concerning their user-interfaces, are still experimental, considerable effort was spend to optimize notably the segmentation and k-Nearest Neighbor implementations. Without these optimizations the practical usefulness of the programs would be very limited. Integration of image analysis with Geographic Information Systems technology, which was a design goal during the entire project, is well reflected in the software. Segmentation (Chapter 4) is executed completely in the quadtree domain, as is segment pyramid classification (Chapter 5), which shows numerous examples of integration of geometric with attribute data.

Recommendations

The algorithms require representative sampling of the class populations during the training stage. The measures to be taken when this requirement cannot be satisfied can be a future research topic.

Another interesting issue that remains to be addressed is the performance of the decision-analytic approach to data interpretation at a level beyond pixels. The approach is suitable for decision making for spatial objects instead of for individual

pixels, as the concepts involved remain the same; however, an image segmentation pre-processing step is required. Applying the approach to spatial objects is expected to benefit from (probabilities of) geometrical and topological properties of objects for decision making.

In context of decision analysis, but also with respect to the above-mentioned probabilistic GIS-model, it should be noticed that in this thesis a *posteriori* class probabilities are only estimated per pixel. Decision analysis at segment or object level, as well as probabilistic GIS models, would require these probabilities per object. Also this needs further investigation. For example, during probabilistic (as opposed to spectral) segmentation, where class densities govern the merging criteria, it is not entirely clear how the probabilities that two adjacent segments belong to a certain class can be combined into a single probability that the merged segment belongs to that class, or to a common superclass. It might be interesting to look at those issues from an aggregation/generalization perspective, taking into account that a spatial compound of fuzzy land-covers might form a distinct land-use at a higher aggregation level. A starting point for this kind of investigations could be a further exploration of the object selection rules in segment pyramid classification.

Bibliography

[Abkar, 1994] A.A. Abkar (1994). *Knowledge-based classification method for crop inventory using high resolution satellite data.* M.Sc. thesis, ITC, Enschede, 1994.

[Acton, 1996] S.T. Acton (1996). On unsupervised segmentation of remotely sensed imagery using nonlinear regression. *IJRS* 17 (7), pp. 1407-1415.

[Argialas and Harlow, 1990] , D. Argialas and C. Harlow (1990). Computational image interpretation models: an overview and perspective. *Photogrammetric Engineering & Remote Sensing.* Vol. 56, No. 6, pp. 871 – 886.

[Ballard and Brown, 1982] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, N.J., 1982.

[Burrough, 1986] *Principles of geographical information systems for land resources assessment.*, Clarendon Press, Oxford, 1986.

[Chang and Li, 1995] Y.-L. Chang and X. Li, "Fast image region growing", *Image and Vision Computing*, Vol. 13, pp.559-571, 1995.

[Chiarello *et al*, 1996] . E. Chiarello, J.-M. Jolion and C. Amoros (1996). Regions growing with the stochastic pyramid: application in landscape ecology. *Pattern Recognition* Volume 29 Number 1, pp. 61 – 75.

[Cochran, 1977] W.G. Cochran, *Sampling Techniques*, John Wiley & Sons, 1977.

[Conese and Maselli, 1992] C. Conese and F. Maselli (1992). Use of error matrices to improve area estimates with maximum likelihood classification procedures. *Remote Sensing of Environment*, no. 40, pp. 113 – 124.

[Cross *et al*, 1988] A.M. Cross, D.C. Mason and S.J. Dury (1988). Segmentation of remotely-sensed images by a split-and-merge process. *IJRS* 9(8), pp. 1329-1345.

[Dasarathy, 1979] B.V. Dasarathy, CODE: Cognition of Deliberate Entrants to a Scene, *Proceedings of the SPIE Symposium on Digital Processing of Aerial Images*, Vol. 186, pp.208-215, 1979.

[Dasarathy, 1980] B.V. Dasarathy, Noising around the neighbourhood: a new system structure and classification rule for recognition in partially exposed enviroment, *IEEE Trans. PAMI*, Vol. 2, 1980.

[Dasarathy, 1991] B.V. Dasarathy (1991). *Nearest neighbor (NN) norms - NN pattern classification techniques.* IEEE Computer Society Press, Los Alamitos, CA.

[Date, 1981] C.J. Date. *Introduction to database management systems*, 3rd ed. Addison-Wesley, Reading, 1981, 574 pp.

[Dubuisson and Masson, 1993] B. Dubuisson and M. Masson, A statistical decision rule with incomplete knowledge about classes, *Pattern Recognition*, Vol. 26, pp. 155-165, 1993

[Duda and Hart, 1973] R.O. Duda and P.E. Hart, *Pattern classification and scene analysis.* John Wiley and Sons, New York, 465 pp. 1973.

[the ERDAS Field Guide, 1993] *ERDAS field guide*, third edition, ERDAS Inc., Atlanta, 1993.

[Fisher and Pathirana, 1990] P.F. Fisher and S. Pathirana, The evaluation of fuzzy membership of land cover classes in the suburban zone, *Remote Sensing of Environment*, Vol 34, 1990, pp. 121 – 132, 1990.

[Franklin and Wilson, 1991] S.E. Franklin and B.A. Wilson. Spatial and spectral classification fo remote-sensing imagery. *Computers and Geosciences*, Vol. 17 No. 8, 1991, pp. 1151-1172.

[Fu and Mui, 1981] . A survey on image segmentation, *Pattern Recognition*, 13:3-16, 1981.

[Fukunaga & Hummels, 1987] K. Fukunaga, D.M. Hummels (1987). Bayes error estimation using Parzen and kNN procedures, *IEEE Trans. PAMI*, Vol. PAMI-9 No. 5, pp. 634 – 643.

[Gallego, 1995] F.J. Gallego. *Agriculture estimates with area frame sampling.* JRC publication, 1995, Ispra.

[Goodchild *et al.*, 1994] M.F. Goodchild, L. Chih-Chang and Y. Leang. Visualizing fuzzy maps. In H.M. Hearnshaw & D.J.Unwin (eds.): *Visualisation in geographical information systems.* Wiley & Sons, Chchester, 1994, pp.158-167.

[Gorte *et al.*, 1988] B.G.H. Gorte, R.Liem and J.P. Wind. The ILWIS software kernel system". *ITC Journal* 1988-1, Enschede.

[Gorte, 1995a] B.G.H. Gorte (1995) *Tools for Advanced Image Processing and GIS using ILWIS.* I.T.C. Publication No. 24, 55 p. ISBN 90 6164 101 2

[Gorte, 1995b] B.G.H. Gorte, *Experimental quadtree software*, Technical report, ITC, Enschede, 1995.

[Grosky and Jain, 1986] W.I. Grosky and R. Jain (1986). A pyramid-based approach to segmentation applied to region matching. *IEEE Trans. PAMI*, Vol. PAMI-8 No. 5, pp. 639 – 650.

[Haralick, 1979] R.M. Haralick, Statistical and structural appraoches to texture, *Proceedings of the IEEE* 67(5), 1979.

[Haralick and Shapiro, 1985] R.M. Haralick and L.G. Shapiro. Survey: Image segmentation techniques. *Computer, Vision, Graphics and Image Processing*, 29, 1985, pp 100-132.

[Hellman, 1970] M.E. Hellman, The nearest neighbor classification rule with a reject option, *IEEE Trans. Syst. Man Cybern.*, Vol. 3, pp. 179-185. 1970.

[Horovwitz and Pavlidis, 1976] S.L. Horovwitz and T. Pavlidis "Picture segmentation by a tree traversal algorithm", *J.ACM*, Vol.23, pp.368-388, 1976.

[Iron and Petersen, 1981] J.R. Iron and G.W. Petersen, Texture transforms of remote sensing data, *Remote Sensing of Environment* 11, 1981, pp.359-370.

[Janssen, 1994] L.L.F. Janssen (1994), *Methodology for updating terrain object data from remote sensing data*, Ph.D. Thesis, Wageningen, 1994, 173 pp.

[Janssen and van der Wel, 1994] L.L.F. Janssen and F.J.M. van der Wel (1994). Accuracy Assessment of satellite derived land-cover data: a review. *Photogrammetric Engineering & Remote Sensing.* Vol. 60, No. 4, pp. 419 – 426.

[Johnsson, 1994] K. Johnsson (1994). Segment-based land-use classification from SPOT satellite data. *Photogrammetric Engineering & Remote Sensing.* Vol. 60, No. 1, pp. 47 – 53.

[Kullback, 1954] S. Kullback. *Information and Statistics.* Wiley & Sons, New York, 1954.

[Lat, 1996] G.A. Lat. *Supervised segmentation of remotely sensed imagery.* M.Sc. thesis, ITC, Enschede, 1996.

[Le Moigne and Tilton, 1992] J. Le Moigne and J.C. Tilton. Refining image segmentation by integration of edge and region data. In *Proceedings of the International geoscience and remote sensing symposium (IGARSS92)*, Houston, 1992, pp. 1406-1408.

[Lunetta *et al.*, 1991] R.S. Lunetta, R.G. Congalton, L.K. Fenstermaker, J.R. Jensen, K.C. McGwire, and L.R. Tinney (1991). Remote sensing and geographic information system data integration: error sources and research issues. *Photogrammetric Engineering & Remote Sensing*, vol. 57, no. 6, pp. 677 – 687.

[Mather, 1987] P.M.Mather, *Computer processing of remotely-sensed images, an introduction*, Wiley, 1987.

[Meyer-Roux, 1987] J.M. Meyer-Roux. The ten year research and development plan for the application of remote sensing in agricultural statistics. *Technical Report SP.1.87.39/EN*, Commission of the European Communities, JRC, Ispra, 1987.

[Middelkoop and Janssen, 1991] H. Middelkoop and L.L.F. Janssen (1991). Implementation of temporal relationships in knowledge based classification or satellite images. *Photogrammetric Engineering & Remote Sensing.* Vol. 57, No. 7, pp. 937 – 945.

[Molenaar, 1998] M. Molenaar. *An introduction to the theory of spatial object modelling for GIS.* 1998. (in press, preliminary version 233 pp.)

[Morris *et al.*, 1986] O.J. Morris, M. deJ. Lee, A.G. Constantinides, "Graph theory for image analysis: an approach based on the shortest spanning tree", *IEE Proc.-Part F.* Vol. 133, pp.146-152, 1986.

[Morrison, 1995] J.L. Morrison (1995). Spatial data quality. In: S.C. Guptill & J.L. Morrison (ed.). *Elements of Spatial Data Quality*. Pergamon, Oxford, pp. 1 – 12.

[Mulder, 1976] N.J. Mulder (1976). *Physics of remote sensing*. Lecture notes, I.T.C. Enschede.

[Mulder and Middelkoop, 1990] N.J. Mulder and H. Middelkoop (1990). Parametric versus non-parametric maximum likelihood classification. *Proc. ISPRS Comm. III*, Wuhan.

[Pavlidis, 1986] T. Pavlidis. A critical survey of image analysis methods, *IAPR-8*, 1986, pp502-511.

[Pavlidis and Liow, 1990] T. Pavlidis and Y-T. Liow. Integrating region growing and edge detection, *IEEE Trans. PAMI*, Vol. 12(3), 1990, pp. 225-233.

[Pratt, 1978] W.K. Pratt, *Digital Image Processing*, John Wiley and Sons, New York, 1978.

[Press *et al*, 1992] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery: Numerical recipes in C - the art of scientific computing, Second edition, Cambridge University Press, 1992.

[Raiffa, 1968] H.A. Raiffa (1968). *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Addison-Wesley, Reading, Massachusetts.

[Richards, 1993] J.A. Richards, *Remote Sensing Digital Image Processing, an Introduction*, Springer-Verlag, 1993.

[Ripley, 1996] B.D. Ripley, *Pattern recognition and neural networks*, Cambridge University Press, 1996.

[Rosenfeld, 1975] A. Rosenfeld, *Visual texture analysis, and overview*, TR-406. Univ. of Maryland, COmputer Science Center, College Park, MD, 1975.

[Samet, 1990] H. Samet, *The design and analysis of spatial data structures*, Addison-Wesley, 1990.

[Schoenmakers, 1995] *Integrated Methodology for segmentation of large optical satellite images in land applications of remote sensing*, JRC, EC, Ispra, 1995, 171pp.

[Shannon, 1948] *The mathematical theory of communication*. Bell Systems Technical Journal, Vol.27, 1948, pp. 379-423

[Sharifi *et al.*, 1996] M.A. Sharifi, M.C. Bronsveld and M.B.W. Claveaux. *Development of crop inventory and forecasting system for the major agricultural commodities in Hamadan province, Islamic republic of Iran*. Project report ITC, Enschede, 1995, 48 pp.

[Shresta, 1998] A. Shresta, Incorporating DEM data in image classifications of mountainous regions, M.Sc. thesis, ITC, Enschede (in preparation).

[Smith, 1988] J.Q. Smith (1988). *Decision Analysis. A Bayesian Approach*. Chapman and Hall Ltd., London.

[Strahler, 1980] A.H. Strahler (1980). The use of prior probabilities in maximum likelihood classification of remotely sensed data. *Remote Sensing of Environment*, no. 10, pp. 135 – 163.

[Swain and Davis, 1978] P.H. Swain and S.M. Davis (Eds.), *Remote Sensing: The Quantitative Approach*, McGraw-Hill. 1978.

[Talukdar, 1997] K.K. Talukdar, *Recognition and extraction of spatial objects from satellite data using GIS and image processing techniques for urban monitoring*, M.Sc. thesis, ITC, Enschede, 1994, 100pp.

[Therrien, 1989] C.W. Therrien, *Decision, estimation and classification* John Wiley & Sons, 1989.

[Tilton, 1989] J.C. Tilton. Image segmentation by iterative parallel region growing and splitting. *IGARSS89*, Vancouver, 1989, pp2235-2238.

[Tomlin, 1990] C.D. Tomlin, *Geographic information systems and cartographic modeling*, Prentice-Hall, Englewood Cliffs.

[Tsichritzis and Lochovsky, 1982] D.C. Tsichritzis and F.H. Lochovsky, *Data models*. Prentice-Hall, Englewood Cliffs, 1982, 381 pp.

[van der Wel and Gorte, 1995] F.J.M. van der Wel and B.G.H. Gorte. *CAMOTIUS Hoofdfase, 1 jan 1993 - 31 dec 1994* BCRS rapport 95-06, 89 pp. ISBN 90 5411 155 0.

[von Winterfeldt & Edwards, 1986] D. von Winterfeldt and W. Edwards (1986). *Decision Analysis and Behavioral Research*. Cambridge University Press, New York.

Publications related to this thesis

[Gorte, 1995b] B.G.H. Gorte (1995). Improving spectral image classifications by incorporating context data using likelihood vectors. *Procs. IPA 1995*, IEE Conf. Publ. no. 410, pp. 251 – 255.

[Gorte, 1995c] B.G.H. Gorte (1995). Classification and class area estimation of satellite images in a GIS environment. *GISDATA specialists meeting on remote sensing and urban change*, Strassbourg, July 1995.

[Gorte and Kroupnova, 1995] B.G.H. Gorte and N.H. Kroupnova (1995). Non-parametric classification algorithm with an unknown class. *Proc. ISCV '95, IEEE International symposium on Computer Vision* pp. 443 – 448.

[Gorte et al, 1996] B.G.H. Gorte, L.C. van der Gaag and F.J.M. van der Wel (1996). Decision-analytic interpretation of remotely sensed data. *SDH '96*, International symposium on spatial data handling.

[Gorte, 1996] B.G.H. Gorte (1996). Multi-spectral quadtree based image segmentation. *IAPRS XXXI Comm. III*, 18th ISPRS Congress, Vienna, 1996.

[Gorte and Stein, 1998] B.G.H. Gorte and A. Stein, Bayesian Classification and Class Area Estimation of Satellite Images using Stratification, *IEEE - TGRS*, Vol. 36, No. 3, May 1998, pp. 803 – 812.

[Kroupnova and Gorte, 1996] N.H. Kroupnova and B.G.H. Gorte (1996). Method for multi-spectral image segmentation in case of partially available spectral characteristics of objects. *Proc. IS&T/SPIE's symposium on electronic imaging: science and technology*, San Jose, CA.

[Mesev and Gorte, 1998] V. Mesev and B.G.H. Gorte (1998). Enhancements to per-pixel image classification techniques and their application to urban remote sensing. In *Remote sensing and urban analysis* (ed. M. Barnsley and J-P. Donnay), Taylor and Francis. *In preparation.*

[van der Wel *et al*, 1998] F.J.M. van der Wel, L.C. van der Gaag, B.G.H. Gorte, Visual exploration of uncertainty in remote sensing classification, accepted for publication in *Computers and Geosciences.*

Samenvatting

Dit proefschrift behandelt het gebruik van kanstheorie bij het inwinnen van geografische informatie uit satelliet-opnames van het aardoppervlak. Dergelijke opnames bestaan uit metingen van de intensiteit van electromagnetische straling, doorgaans zonlicht, die door het aardoppervlak wordt gereflecteerd. Het gebied dat door een satellietopname wordt bedekt is denkbeeldig in rijen en kolommen verdeeld waardoor terreinelementen ontstaan. De afstand tussen aangrenzende rijen (resp. kolommen) bepaalt de resolutie van het beeld. De gemeten reflectie van de terreinelementen wordt digitaal in beeldelementen opgeslagen. Wanneer de beeldelementen op de rijen en kolommen van een computerscherm worden afgebeeld met intensiteiten die met de reflectiemetingen overeenkomen, wordt het terrein zichtbaar. In dit proefschrift worden hoofdzakelijk multispectrale beelden gebruikt, waarvan elk beeldelement uit reflecties in verscheidene spectrale banden bestaat, bijvoorbeeld een aantal zichtbare kleuren en infrarood. Om informatie uit beelden te verkrijgen moeten de meer-dimensionale continue reflectiemetingen worden omgezet in discrete objecten, die van elkaar onderscheiden worden volgens een discrete klassificatie. Helaas is het verband niet eenduidig. Binnen verschillende objecten van dezelfde klasse en zelfs binnen een enkel object kunnen verschillende reflecties voorkomen. Omgekeerd zijn verschillende thematische klassen in een satellietbeeld soms moeilijk te onderscheiden zijn omdat ze bijna dezelfde reflectie vertonen. In dergelijke gevallen zijn deterministische methodes niet afdoende. Een probabilistische aanpak daarentegen kan de spectrale variatie binnen een klasse beschrijven en bovendien de kans op foutieve klasse-toewijzingen zo klein mogelijk maken.

Klassificatie

Klassificatie kiest voor elk beeldelement een thematische klasse uit een verzameling die vooraf door de gebruiker bepaald is. De keuze wordt gemaakt op grond van de reflectie-metingen die in het beeldelement zijn opgeslagen. Klassificatie kampt met een aantal problemen, waardoor de resultaten vaak tegenvallen. De gemeten reflecties hangen niet alleen af van de thematische klasse, maar bijvoorbeeld ook van atmosferische omstandigheden, bodemvochtigheid en lichtinval, waarbij de laatste beïnvloed wordt door de zonnestand en de helling van het terrein. Ook het toeval,

in de vorm van ruis, speelt een (kleine) rol. Bovendien bestaan sommige thematische klassen uit samenstellingen van bodembedekkingen met verschillende spectrale kenmerken. De klasse *stedelijke bebouwing* bijvoorbeeld omvat huizen (daken), wegen, tuinen, plantsoenen enzovoort, die tot verschillende reflecties aanleiding geven. Anderzijds komt het voor dat terreinelementen van verschillende klassen dezelfde reflectie-metingen opleveren. De betreffende beeldelementen kunnen dan niet met zekerheid geklassificeerd worden.

Een ander probleem treedt op in elementen die objecten van verschillende klassen bevatten. Dit kan veroorzaakt worden doordat objecten kleiner of smaller zijn dan een terreinelement, zoals bij huizen of wegen in de gangbare satellietbeeld-resoluties het geval is. Daarnaast worden sommige terreinelementen doorsneden door de grens tussen objecten die op zichzelf groot genoeg zijn, iets dat bij elke resolutie optreedt. Bovendien zijn sommige objecten ook in het terrein niet duidelijk afgebakend. Zo is de grens tussen bos en hei, of tussen stedelijke bebouwing en het omringend landbouwgebied, niet altijd duidelijk te trekken. Er is dan een overgangszône die verscheidene terreinelementen breed kan zijn, afhankelijk van de resolutie. De overeenkomstige beeldelementen zijn moeilijk te klassificeren.

Een ander probleem waarop in dit proefschrift wordt ingegaan, doet zich voor indien het terrein objecten bevat waarvan de klasse niet tot de verzameling behoort die door de gebruiker is gekozen. Vooral indien de klassificatie gemaakt wordt om het areaal van de diverse klassen te bepalen is het noodzakelijk om met de zogenaamde restklasse rekening te houden.

Vaak wordt statistiek gebruikt om deze problemen het hoofd te bieden. Men kan klassificatiefouten daarmee niet voorkomen, maar de kans erop zo klein mogelijk maken. Bovendien kan men deze kans op fouten bepalen, zo nodig voor elk beeldelement afzonderlijk. Bayesiaanse klassificatie berekent in elk beeldelement de a *posteriori* kans voor iedere klasse op grond van schattingen van kansdichtheid en a *priori* kans, en kiezen vervolgens de klasse met de grootste kans. Bij het schatten van kansdichtheden neemt men vaak aan dat de reflecties binnen elke klasse uit een Gaussische verdeling afkomstig zijn. De parameters voor de verdelingen worden berekend aan de hand van trainingsgegevens, voorbeelden van beeldelementen die de gebruiker voor elke klasse in het onderhavige beeld aangewezen heeft. Vervolgens kan de kansdichtheid voor elke willekeurige reflectie-vector en voor elke klasse eenvoudig bepaald worden. De gebruiker kan soms de a *priori* kans specificeren als het verwachte aandeel van elke klasse in het totale oppervlak. In bepaalde gevallen is het beter om gelijke a *priori* kansen voor alle klassen te veronderstellen – dit komt weliswaar de klassificatie-nauwkeurigheid niet ten goede, maar de resultaten worden minder bevooroordeeld.

Lokale kansen

Statistische klassificatie-methodes proberen niet altijd de diverse kansen zo nauwkeurig mogelijk te schatten. Men hoopt wellicht dat de grootste kans ook met vrij ruwe schattingen bij de juiste klasse optreedt. Dit proefschrift toont verfijnde schattingen van de diverse kansen door ze *lokaal* te maken, dat wil zeggen betrekking te laten hebben op deelgebieden in het beeld, in plaats van op het hele beeld. De benodigde onderverdeling van het beeld kan gemaakt worden met additionele (kaart-)gegevens, die bijvoorbeeld in een geografisch informatiesysteem aanwezig zijn. Het uitgangspunt is dat bij het nemen van een klassificatie-beslissing voor een bepaald beeldelement, statistische gegevens over bijvoorbeeld de specifieke grondsoort van het element relevanter zijn dan statistische gegevens over het hele gebied. Tot dusverre was het bezwaar tegen deze werkwijze dat dergelijke specieke gegevens doorgaans niet voorhanden zijn. Dit proefschrift toont aan dat zulke gegevens uit reflectieverdelingen geschat kunnen worden.

Wanneer de *a priori* kansen in Bayesiaanse kansrekening betrekking hebben op een deelgebied, moeten de kansdichtheden over hetzelfde deelgebied gaan. Wil men kansdichtheden gebruiken die voor het hele beeld gelden, dan moet men veronderstellen dat deze inderdaad onafhankelijk zijn van de positie in het beeld, bijvoorbeeld dat de reflectie van een gewas niet afhangt van de grondsoort. Naarmate de deelgebieden kleiner worden, wat gunstig is voor de verfijning van de schattingen, wordt deze aanname steeds twijfelachtiger. Daarom presenteert dit proefschrift een tweetal manieren om kansdichtheden lokaal te schatten op grond van een enkele verzameling trainingsgegevens. Dit laatste is cruciaal als er veel deelgebieden zijn – afzonderlijke trainingsverzamelingen per deelgebied zijn dan niet verkrijgbaar. Naast een parametrische methode, die Gaussische verdelingen veronderstelt, wordt een niet-parametrische methode beschreven, die willekeurige verdelingen kan schatten, indien voldoende trainingsgegevens voorhanden zijn.

Een gebruiker die op deze wijze additionele gegevens inbrengt krijgt een hogere klassificatie-nauwkeurigheid. De grootste kans valt nu af en toe op een andere klasse dan voorheen en dit levert gemiddeld een betere keuze op. De conclusie is dat de lokale schattingen van kansdichtheden en van *a priori* kansen tot *a posteriori* kansen leiden die relevanter zijn. Zeer veel soorten additionele gegevens zijn geschikt, zoals bodemkaarten, geologische kaarten, hoogtegegevens en historische landgebruikskaarten. De enige eis is dat ze het terrein in stukken verdelen, waarin verschillende mengverhoudingen van klassen verwacht mogen worden. Deze mengverhoudingen hoeven niet vooraf bekend te zijn.

Segmentatie

Binnen de automatische beeldanalyse neemt segmentatie een belangrijke plaats in. Segmentatie is probeert aangrenzende beeldelementen samen te voegen tot segmen-

ten die overeenkomen met objecten in het terrein. De gevormde segmenten zijn homogeen ten aanzien van een bepaald kenmerk, in het eenvoudigste geval reflectie.

In tegenstelling tot klassificatie is segmentatie van satellietbeelden niet erg gangbaar. Segmentatie is erg rekenintensief, wat bij een satellietbeeld van bijvoorbeeld 6000 × 6000 elementen zwaarder telt dan bij een video-opname van 512 × 512 elementen. Bovendien is de resolutie van satellietbeelden een beperkende factor bij de meeste toepassingen. Men zoekt vaak naar objecten die maar een paar beeldelementen groot zijn, zodat er weinig te groeperen valt. De populariteit van segmentatie stijgt waarschijnlijk naarmate computers sneller en beeldresoluties hoger worden.

Er zijn verschillende methodes. Sommige zoeken eerst naar reflectie-overgangen tussen aangrenzende beeldelementen, om zodoende objectgrenzen op te sporen. Vervolgens worden de segmenten bepaald die door deze grenzen omsloten worden, waarbij het een probleem is dat er gaten en andere topologische inconsistenties in de grenzen kunnen zitten.

Andere methodes proberen rechtstreeks segmenten te vormen, bijvoorbeeld door op willekeurige plaatsen een nieuw segmentje te starten en daaraan één voor één aangrenzende elementen toe te voegen, zolang deze nog voldoende op elkaar lijken (region growing). Een andere mogelijkheid is het hele beeld eerst in gelijke vierkantjes te verdelen en vervolgens te proberen deze recursief samen te voegen zolang het resultaat homogeen genoeg is, of ze anders recursief in kleinere vierkantjes op te delen totdat die allemaal homogeen genoeg zijn (split-and-merge). Vervolgens worden aangrenzende vierkantjes, die nu verschillende groottes hebben, samengevoegd, waarbij wederom de homogeniteit van het resultaat betracht moet worden. Deze methodes hebben als bezwaar dat lang niet alle mogelijkheden om elementen of deelsegmentjes samen te voegen telkens helemaal doorgerekend kunnen worden. Er worden arbitraire keuzes gemaakt, die afhangen van de volgorde waarin de beeldelementen tijdens de berekeningen in beschouwing genomen worden.

Binnen sommige terrein-objecten komen verschillende reflecties voor, terwijl elders in het beeld aangrenzende objecten vrijwel dezelfde reflectie kunnen hebben. In het ene geval levert een object soms verschillende segmenten op en in het andere kunnen verschillende objecten in één segment terechtkomen. Beide soorten fouten komen voor in elke segmentatie, ongeacht de segmentatie-methode. Het probleem wordt over het algemeen bij de gebruiker gelegd. Deze moet over de gewenste 'fijnheid' van de segmentatie beslissen en met een drempelwaarde de toegestane heterogeniteit kiezen, dan wel bepalen wanneer een reflectie-overgang een grens is.

Het proefschrift beschrijft een nieuwe segmentatie-methode, uitgaande van een algorithme om uniforme segmenten te nummeren in een *quadtree*. De methode behandelt multispectrale beelden en biedt de mogelijkheid om met homogoniteits-criteria te experimenteren. Het algorithme vormt segmenten door recursief beeldelementen en deelsegmenten samen te voegen, waarbij de deelsegmenten tijdens alle fasen van

de recursie willekeurige vormen mogen hebben. Afhankelijkheid van de volgorde van de beeldelementen in het *quadtree*-bestand wordt vermeden door een beeld een aantal keren te doorlopen, waarbij het homogeniteits-criterium langzaam verruimd wordt.

Wanneer men het resultaat van iedere doorloop bewaart, ontstaat een serie segmentaties met verschillende fijnheden. Omdat in iedere doorloop grotere segmenten ontstaan door eerder gevormde segmenten samen te voegen, ontstaat een hiërarchie van segmentaties, die segmentatie-pyramide wordt genoemd.

Integratie

Klassificatie en segmentatie proberen beide terreininformatie uit beelden te winnen, waarbij klassificatie gebruikt maakt van spectrale kenmerken in combinatie met trainingsgegevens, terwijl segmentatie ruimtelijke kenmerken van objecten beschouwt. Aan beide methodieken kleven bezwaren, maar ze zijn complementair, zodat ze samen betere resultaten kunnen geven dan ieder afzonderlijk. Tijdens het onderzoek dat in dit proefschrift wordt beschreven, is gezocht naar een geschikte integratiemethode.

Indien additionele kaartgegevens ontbreken, kan segmentatie een onderverdeling van het beeld geven, waarbinnen vervolgens een klassificatie met lokale kansen uitgevoerd kan worden. Dit levert doorgaans inderdaad een lichte verbetering van de klassificatienauwkeurigheid op, maar voor de gebruiker is volstrekt onduidelijk welke segmentatie-fijnheid het meest geschikt is, en derhalve welke drempelwaarde hij moet kiezen.

Een betere methode is gebaseerd op segmentatie-pyramides. Het uitgangspunt is dat er voor ieder terreinobject ergens in de pyramide een segment bestaat. Een object met een zeer homogene reflectie komt als segment voor in een 'fijne' segmentatie, waar een strikt homogeniteits-criterium is toegepast. Minder homogene objecten worden gerepresenteerd door segmenten in segmentaties met ruimere homogeniteitscriteria. Om voor ieder object het bijbehorende segment op te sporen, wordt de klassificatie met lokale kansen op elke segmentatie in de pyramide toegepast. Zodoende worden de mengverhoudingen van klassen bepaald in alle segmenten van elke segmentatie van de pyramide, waarna het mogelijk wordt om pure segmenten, waarin één klasse domineert, van gemengde segmenten te onderscheiden. Om segmenten te selecteren, beschouwen we vervolgens een van de segmentaties. De pure segmenten in deze segmentatie komen voor selectie in aanmerking, tenzij ze op een hoger niveau in de pyramide (met grotere segmenten) ook puur zijn - in dat geval wordt dat hogere niveau gekozen. Bij gemengde segmenten gaat de voorkeur in principe uit naar kleinere segmenten, lager in de pyramide, in de verwachting dat ze dan puur genoeg worden om voor selectie in aanmerking te komen. Indien dit echter niet het geval is, wordt een gemengd segment geselecteerd, wederom op een zo hoog

mogelijk niveau. Blijkbaar bevat dit deel van het terrein een mengsel van klassen die niet onderscheiden kunnen worden, ten gevolge van resolutie-beperkingen of spectrale overlap. De uiteindelijke verzameling segmenten bedekt het hele gebied. Wanneer in elk segment de dominerende klasse geselecteerd wordt, ontstaat een object-klassificatie. Een elements-gewijze klassificatie wordt verkregen wanneer de mengverhoudingen in de geselecteerde segmenten als *a priori kansen* genomen worden en de kansdichtheden voor elk element afzonderlijk geschat worden. In beide gevallen is de nauwkeurigheid hoger dan die van een conventionele klassificatie.

De geïntegreerde methode wordt probabilistische segmentatie genoemd. Merk op dat geen additionele gegevens vereist zijn. Bovendien is de methode niet kritisch ten aanzien van drempelwaarden. Het resultaat bestaat, naast de thematische klassificatie, uit statistische gegevens die betrekking hebben op segmenten en op afzonderlijke beeldelementen.

Besluitvorming

Vaak wil men informatie uit satellietbeelden halen om beslissingen te nemen, bijvoorbeeld voor planologische doeleinden. Gedurerende het besluitvormingsproces wordt veelal een klassificatie vervaardigd, omdat de beslissingen afhangen van de ruimtelijke verdeling van thematische klassen. Een nadeel van deze werkwijze is dat fouten in de klassificatie, die nooit helemaal te voorkomen zijn, de besluitvorming beïnvloeden.

Nadat dit proefschrift klassificatie-onzekerheid kwantificeert in de vorm van kansen en mengverhoudingen, wordt in het laatste hoofstuk een methode gepresenteerd die de onzekerheid meeneemt in de besluitvorming. Daartoe wordt een besliskundig model gehanteerd dat gebaseerd is op een utiliteits-begrip en waarin mogelijke beslissingen en thematische klasses aan elkaar gerelateerd worden. De gebruiker specificeert wat bepaalde beslissingen op zouden leveren als thematische klassen met zekerheid bepaald waren. Het algorithme berekent vervolgens de verwachte opbrengst van elke beslissing, rekening houdend met de onzekerheid in de klassificatie.

Experimenten

De theorie in dit proefschrift wordt geïllustreerd met een aantal experimenten. In een *Thematic Mapper* beeld met een resolutie van 30 m, van een gebied rond Biddinghuizen in de Flevopolder worden de oppervlakte-percentages voor zeven gewassen geschat, uitgesplitst naar de gebieden waarin dezelfde gewassen het jaar ervoor verbouwd werden. Dit levert een verbeterde klassificatie op, alsmede een gewasrotatiematrix, die zeer nauwkeurig blijkt te zijn. Hetzelfde beeld wordt gebruikt om de 'restklasse' in kaart te brengen: de gebieden die niet bij een van de zeven klassen horen. Bovendien wordt een besluitvormings-experiment beschreven, met als doel

frauduleuze aanvragen voor landbouwsubsidies op te sporen, waarbij objectieve en subjectieve gevolgen van verkeerde beslissingen een rol kunnen spelen.

Een verrassend voorbeeld van klassificatieverbetering met lokale kansen betreft een *Thematic Mapper* beeld van Twente, onderverdeeld in de (vier-cijferige) postcodegebieden. Hoewel de sommige landgebruiksklassen in dit voorbeeld spectraal sterk op elkaar lijken, terwijl andere zeer heterogeen zijn, wordt een alleszins acceptabele klassificatie verkregen, wat met gangbare methodes niet het geval is.

Probabilistische segmentatie wordt geïllustreerd aan de hand van multispectraal *SPOT* beeld van Ameland, met een resolutie van 20 m. De object-klassificatie en de elementsgewijze klassificatie die aldus verkregen kunnen worden zijn beide nauwkeuriger dan gangbare klassificaties.

Curriculum Vitae

Ben Gorte was born in Losser, the Netherlands on July 21, 1953. He completed secondary school (Gymnasium β) at the Twents Carmellyceum in Oldenzaal in 1971, after which he studied applied mathematics at the Twente University in Enschede. He received a B.Sc. degree in 1977. In 1980, he started to work at the International Institute for Aerial Survey and Earth Sciences (ITC) in Enschede as a systems programmer, and meanwhile completed his education at Twente University, obtaining an M.Sc. degree in 1983. In 1986 he became a lecturer for image processing at ITC and since 1995 he is an assistent professor for knowledge based information extraction from image data in the division for geoinformatics and spatial data acquisition. Before starting the research for this thesis, he was leading the software development for the ILWIS system for image processing and GIS, between 1986 and 1993.