Geographic Object-based Image Analysis

By the Faculty of Geosciences, Geo-Engineering and Mining

of the Technische Universität Bergakademie Freiberg

approved

THESIS

to attain the academic degree of

Doktor Ingenieur

Dr.-Ing

submitted

by **MSc Prashanth Reddy Marpu**

born on the 12 March, 1982 in Kisannagar, India.

Assessors: Dr. Irmgard Niemeyer, Freiberg, Germany
Jun. Prof. Dr. Richard Gloaguen, Freiberg, Germany
Prof. Dr. Thomas Blaschke, Salzburg, Austria

Date of the award: 17 April, 2009

Dedicated to my late grandparents and my family

Abstract

The field of earth observation (EO) has seen tremendous development over recent time owing to the increasing quality of the sensor technology and the increasing number of operational satellites launched by several space organizations and companies around the world. Traditionally, the satellite data is analyzed by only considering the spectral characteristics measured at a *pixel*. The spatial relations and context were often ignored. With the advent of very high resolution satellite sensors providing a spatial resolution of $\leq$ 5m, the shortfalls of traditional pixel-based image processing techniques became evident. The need to identify new methods then led to focusing on the so called *object-based image analysis* (OBIA) methodologies. Unlike the pixel-based methods, the object-based methods which are based on segmenting the image into homogeneous regions use the shape, texture and context associated with the patterns thus providing an improved basis for image analysis. The remote sensing data normally has to be processed in a different way to that of the other types of images. In the geographic sense OBIA is referred to as *Geographic Object-Based Image Analysis* (GEOBIA), where the GEO pseudo prefix emphasizes the geographic components. This thesis will provide an overview of the principles of GEOBIA, describe some fundamentally new contributions to OBIA in the geographical context and, finally, summarize the current status with ideas for future developments.

Zusammenfassung

Das Arbeits- und Forschungsfeld der Erdbeobachtung (Earth Observation, EO) hat in jüngerer Zeit dank einer kontinuierlichen Qualitätszunahme in der Sensortechnologie und dank der zahlenmäßigen Zunahme von Satelliten verschiedener Raumfahrtorganisationen und -unternehmen eine enorme Entwicklung erfahren. Traditionell werden Satellitendaten so analysiert, dass nur die spektralen Eigenschaften von Pixeln Beachtung finden während ihre räumliche Beziehung oftmals aussen vorgelassen wird. Mit der Nutzung sehr hoch auflösender Satellitensensoren, die eine räumliche Auflösung $\leq$ 5m erreichen, wurden die Nachteile traditioneller pixelbasierter Bildverarbeitungstechniken offensichtlich. Die Suche nach neuen Ansätzen führte dann zur Fokussierung auf Methoden der so genannten objektbasierten Bildanalyse (OBIA). Anders als die pixelbasierten Methoden beruhen die objektbasierten Methoden auf der Segmentierung des Bildes in homogene Regionen, den sogenannten Bildobjekten, und der Bildanalyse auf der Grundlage der vielfältigen spezifischen Objektmerkmale, wie Farbe, Form, Textur und Kontext. Fernerkundungsdaten erfordern im Vergleich zu anderen Bilddaten eine besondere methodische Herangehensweise. OBIA, wird im geographischen Sinne auch als geographische objektbasierte Bildanalyse (GEOBIA) bezeichnet, wobei der Pseudopräfix ‚GEO' die geographischen Komponenten hervorhebt. Diese Arbeit gibt einen Überblick über die Prinzipien der GEOBIA, stellt einige fundamental neue Beiträge zur OBIA im geographischen Kontext vor und fasst schliesslich den gegenwärtigen Wissensstand und Ideen für zukünftige Entwicklungen zusammen.

Preface

This thesis is based on the work done by the author in two projects of the European "Global Monitoring for Environment and Security" (GMES) initiative (http://www.gmes.info) funded by the Sixth Framework Programme (FP6): The Network of Excellence "Global Monitoring for Security and Stability" (GMOSS, http://gmoss.jrc.it) and the Integrated Project "Land/Sea Integrated Monitoring for European Security" (LIMES, http://www.fp6-limes.eu). The author was employed as a third-party funded researcher within the Geomonitoring Group, Institute for Mine-Surveying and Geodesy, TU Bergakadamie Freiberg.

Acknowledgements

There can't be a better chance than now to thank everyone who helped me reach this level. It has been a big journey from my life at school in a small village in India to the universities in Europe. As the first thing, I would like to thank all the teachers of my school, St.E.A.S High School, Kisannagar, India. It is there, where my journey to be a researcher started. Those ten years, as I look back are the most cherishable moments of my life. Then there is this wonderful contribution from two maths teachers Prof. A. Padmanabham and Prof. B. Rameshwar Rao during my intermediate which I am indebted to for all my life. Going to Denmark to study for my masters took everything to a next level. The amazing atmosphere at the Technical University of Denmark (DTU) has fuelled me with all the zeal I need to be a researcher for the rest of my life. I have to specially thank Prof. Henning Skriver and Prof. Joergen Dall at DTU for all their support during my time in Denmark.

My life then in Freiberg was more than an academic achievement. It was a life experience. There are no pages enough to write all the praise for my supervisor Dr. Irmgard Niemeyer, head of the Gemonitoring Group, TU Freiberg for all her amazing support and composure. She was always there to help me. I have to acknowledge the wonderful support of Florian Bachmann, Bjarnheidur Kristinsdottir and Christian Daneke from the Geomonitoring Group for some of the best discussions I had. Special thanks to Florian Bachmann for providing me with some of the programs used in this thesis.

Prof. Richard Gloaguen, head of the Remote Sensing Group, TU Freiberg deserves a special mention for his tremendous support in more than one way, academically and personally. The first on the list would be to thank him for allowing me to use the facilities of his working group. The room in Humboldt-Bau has become my second home during my stay in Freiberg. Many thanks to Arief Wijaya, Faisal Shahzad, Syed Amer Mahmood, Moncef Bouaziz and Yash Gandhi for helping me with the proof reading of the thesis.

I owe a lot to Dr. Morton Canty, Research Center Juelich, Germany for a lot of things I could accomplish during my short research career. He is always a great source of inspiration. I also thank him for helping me with the proof reading of my thesis. Special thanks Prof. Thomas Blaschke, University of Salzburg, Austria for reviewing my thesis and also for his words of encouragement.

My late grandparents are always in my heart for all their love and care. My parents

always supported me all my life. Every decision I made was encouraged and every level I could scale was celebrated. And then there is my wonderful brother Srikanth who always took so much care of me all my life. My aunt, Ms. Shamantha has a special place in my life for supporting me in every step of mine.

I have to specially mention Dr. Vikas Baranwal who has helped me during my stay in Freiberg. He is such a wonderful and caring person whom I consider in high regard. Last but not the least many many thanks to all my friends who were there for me at all the stages of my life.

Contents

List of Figures

Chapter 1

Introduction

1.1 General introduction

The field of earth observation (EO) has seen tremendous development over recent time owing to the increasing quality of the sensor technology and the increasing number of operational satellites launched by several space organizations and companies around the world. The growing number of applications of the available remote sensing data is in turn feeding the appetite for new and improved technologies. *While remote sensing (RS) made enormous progress over the last years in terms of improved resolution, data availability and public awareness, a vast majority of applications rely on basic image processing concepts developed in the 70s per-pixel classification of data in a multi-dimensional feature space* (Blaschke and Lang, 2006). With the advent of very high resolution satellite sensors providing a spatial resolution of $\leq$ 5m, the shortfalls of traditional pixel-based image processing techniques became evident. The need to identify new methods then led to focusing on the so called object-based image analysis (OBIA) methodologies.

Unlike the pixel-based methods, the object-based methods which are based on segmenting the image into homogeneous regions use the shape, texture and context associated with the patterns thus providing an improved basis for image analysis. Meanwhile, *eCognition* (Baatz and Schaepe, 1999), the first commercial software for OBIA was already being developed and the first studies (De Kok et al., 1999, Niemeyer et al., 1999, Buck et al., 1999, Blaschke et al., 2000) were promising. The full release of *eCognition* in 2000 made generally available the tools for OBIA which had been restricted to the research community till then. Several studies have since then confirmed the efficiency of the object-based methods over the pixel-based methods (Baatz et al., 2008). These developments have recently led to the emergence of the new paradigm referred to as Geographic Object-Based Image Analysis (GEOBIA), where the GEO pseudo prefix emphasizes the geographic components (Hay and Castilla, 2008). This thesis will provide an overview of the principles of GEOBIA, describe some fundamentally new contributions to OBIA in the geographical context and, finally, summarize the current status with ideas for future developments. The aim is to provide general methodologies in the context of GEOBIA

and thus there is no focus on any particular application of RS.

1.2 Overview of the thesis

The remainder of this chapter gives a brief introduction to RS, types of RS data and the methods for interpreting them by means of classification. Several types of sensors now provide data in a wide range of the electromagnetic (EM) spectrum from visible to microwave frequencies. The knowledge of sources and characteristics of RS data is thus important to understand the information that they provide. The different types of resolutions which are a result of the tradeoff between different indicators of the sensor performance have to be understood to analyze the patterns in the image. A brief summary will also be given of the different distortions induced in the RS data. The basics of image classification are then introduced. Finally, a critique of the pixel-based approaches is provided to understand why object-based approaches can be better.

In chapter 2, the principles of GEOBIA are introduced. It is a general claim that OBIA tries to replicate human interpretation of images (Definiens User Guide, 2007). A reference to the existing theories of vision perception only confirms that this is not completely the case (Gordon, 2004). Firstly, there exists no single theory which can explain human perception correctly and secondly, it is already established that human perception depends a lot on prior knowledge. However, OBIA is still based on a lot of general concepts of visual perception provided the primitive regions (or image objects) are identified correctly. GEOBIA is OBIA applied on geographic images. Then, the following questions will be specifically answered,

1. What makes GEOBIA different from OBIA?
2. Is it object-based or object-oriented?
3. What is the difference between geo-object and image-object?

Finally, a complete workflow of GEOBIA will be presented which includes pre-processing of the data, segmentation, feature identification, classification and post-classification data analysis. Image segmentation and classification are treated independently in this thesis. The pre-requisite of a good classification result is a good segmentation result. However, it also requires a good classification algorithm which can handle a lot of complexity.

Chapter 3 deals with the pre-processing methods to ensure a good segmentation. Images can be pre-processed using the spatial, spectral and morphological filters. Orthogonal transformations such as the principal components analysis (PCA), maximum autocorrelation factor (MAF) and minimum noise fraction (MNF) transformations help in reducing the dimensionality of the data. Moreover the spectral transformations such as the normalized difference vegetation index (NDVI) can add a lot of information to the image analysis. A new method has been developed to pre-process the images to increase the homogeneity in the image regions. The method adaptively filters the images by preserving the contrast at the edges.

Chapter 4 deals with the segmentation of the images. The first and important step of an object-based classification system is the segmentation of the image in to primitive objects. Several algorithms for image segmentation have been developed till this date not only for RS data but also for computer vision, biomedical imaging, etc. Almost all of these algorithms are based on segmenting the images using the spectral values and the resulting image segments mostly do not correspond to the image objects that can be perceived. A new segmentation algorithm based on constructing a minimum spanning tree over the image is developed. This algorithm also does not provide an accurate image segmentation but, it has certain advantages over a lot of existing algorithms. Also, there are no algorithms which deal with the shape of the image objects. For instance, imagine two roads crossing each other: One of the standard characteristics to classify a road object is the ratio of the length to the width of the object. At the junction of the roads, this feature cannot be used. So, it is important to segment both the roads as different objects. In this regard, an algorithm for shape segmentation has been developed.

In chapter 5, the features describing the objects will be explained. An object is characterized by a huge number of features such as spectral, spatial, shape, texture and context features. A big collection of the features will be presented here. There are several ways to calculate the length and width of the objects. The method used in Definiens software for instance uses the elliptical assumption of an object shape. The length of the major axis is the length of the object and the minor axis is the width of the object. This method does not estimate the correct length and width of the objects in all the cases. A new method to calculate the length and width of an object based on the skeleton will be presented.

Chapter 6 deals with the classification of image objects. This chapter details the transition from image-objects to geo-objects. Object-based classification can be done sequentially where classes are classified individually in a sequential way or in parallel where all the objects are simultaneously classified. Sequential classification provides the chance to use the relations of the objects to the classified objects. For instance, if shadows can be classified first based on the fact that shadows are dark objects, then the bright objects adjacent to the shadow objects are buildings. On the other hand, when the context of the other classes is not so important then all the objects can be simultaneously classified using any standard classification algorithm such as nearest neighbor, maximum likelihood, neural networks, support vector machines, etc. However, the biggest challenge is to identify features of interest which characterize the classes. Vegetation, for instance, can be characterized as having a high NDVI value, high values in the green and near infrared bands, low values in the red band, etc. Roads have a high length to width ratio. So, it is important to identify the features which are characteristic of the classes. An algorithm and software for feature identification will be presented. Also, a classification algorithm based on an ensemble of neural networks is developed to deal with the different types of features associated with the different classes. The transferability of classification schemes from one scene to other is very difficult as it is almost impossible to identify common characteristics of all the classes. However, it is possible to transfer the classification rules temporally over the same area of interest.

In chapter 7, it will be shown how post-classification analysis can be done. This

explains how GEOBIA acts as a bridge between RS and GIS. The framework required to implement a GEOBIA system will be explained. Also, the design of database management system for monitoring applications based on GEOBIA will be introduced. Such a system would be an essential component of the future of GEOBIA. Finally, in chapter 8, the current status of GEOBIA will be discussed and then conclusions are drawn by providing ideas for future research.

It has to be noted that the work has been carried out parallely in the areas of preprocessing, segmentation and classification. The solutions to specific problems are identified based on the experiences of the work in all the three areas. So, the chapters have to be dealt with independently. For example, the new segmentation algorithm presented here was developed last along with the neural networks architecture based on the ideas after doing some initial work with existing segmentation and classification methods in the Definiens Developer software environment.

1.3 Introduction to remote sensing

Remote sensing can be defined as studying an object without making any actual contact with the object. More precisely,

"...remote sensing in the most generally accepted meaning refers to instrument-based techniques employed in the acquisition and measurement of spatially organized (most commonly, geographically distributed) data/information on some property(ies) (spectral; spatial; physical) of an array of target points (pixels) within the sensed scene that correspond to features, objects, and materials, doing this by applying one or more recording devices not in physical, intimate contact with the item(s) under surveillance (thus at a finite distance from the observed target, in which the spatial arrangement is preserved); techniques involve amassing knowledge pertinent to the sensed scene (target) by utilizing electromagnetic radiation, force fields, or acoustic energy sensed by recording cameras, radiometers and scanners, lasers, radio frequency receivers, radar systems, sonar, thermal devices, sound detectors, seismographs, magnetometers, gravimeters, scintillometers, and other instruments" (Short, 2009).

This is an all-inclusive definition of remote sensing in general. However, in this thesis the main focus will be on remote sensing based on EM radiation, primarily the visible/infrared spectrum. Within this thesis, remote sensing refers to imaging the surface of the earth by the satellite or airborne sensors. The principles of GEOBIA can be applied to any kind of EO image data.

Remote sensing data consists of measured values of emitted or reflected energy from an object. An overview of the various wavelength ranges of the EM radiation is given in Fig. 1.1. Some sensors measure the spatial distribution of the reflected solar radiation in the ultraviolet, visible and near- to middle infrared range of wavelengths. Others, measure the spatial distribution of energy radiated by the earth itself which is dominant in the thermal infrared and microwave wavelength range and others still, like the radar sensors, measure the relative return from the earths surface of actual energy transmitted

from the sensor vehicle itself. Such systems where the energy source is provided by the sensor platform are categorized as active sensors. Those which depend on external energy source, such as the sun, are passive sensors (Richards and Jia, 1999).

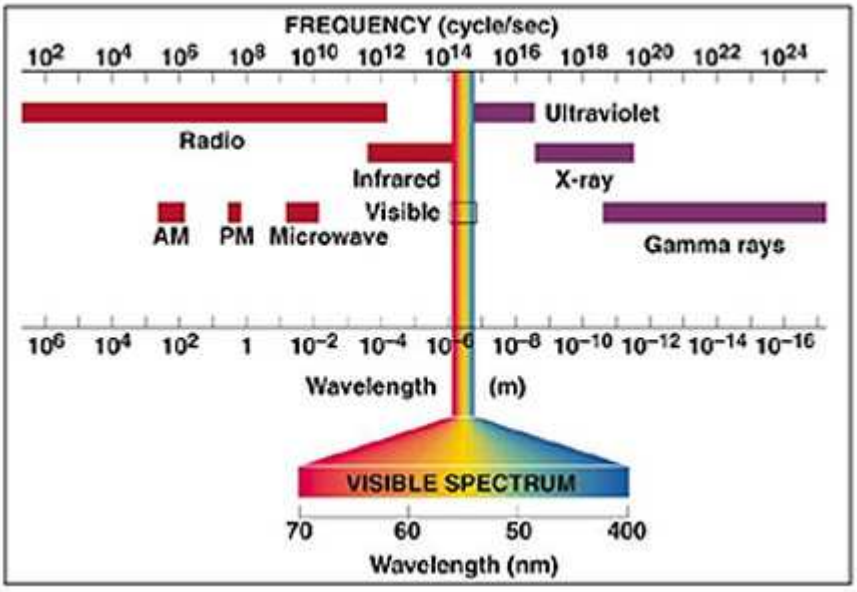


Figure 1.1: The different wavelength ranges in the EM spectrum (Short, 2009). Visible/infrared range (0.4 12μ m) and microwave range (30 -300mm) are the most common ranges used in EO.

The significance of the spectral ranges can be established by observing the spectral signatures of the different objects by the sensors. When the reflectance (fraction of incident electromagnetic power that is reflected at an interface) of the materials is plotted over a range of wavelengths, the connected curve represents the material's spectral signature (or spectral response curve). Properties such as the pigmentation, moisture content and cellular structure of vegetation, the mineral and moisture content of soils and the level of sedimentation in water determine the energy reflected to the sensor in the visible or infrared range. In the thermal range it is the heat capacity and other thermal properties such as the temperature, which determine the energy emitted by the objects. Whereas in the microwave range, using the active sensors, the magnitude of the reflected signal is determined by the surface roughness and the electrical properties of the object expressed in terms of complex permitivity which. in turn depends strongly on the moisture content (Richards and Jia, 1999). Fig. 1.2 shows the spectral signature of an example vegetation type where the factors affecting the spectrum in different wavelength ranges is indicated.

Vegetation reflects maximally in the near infrared (NIR) range between 0.7μm and 1.3μm where the plant cell structure dominates. The troughs in the shortwave infrared (SWIR) range are due to the water absorption bands at 1.4μm, 1.9μm and 2.7μm. In the visible range, it is the leaf pigments which determine the reflectance. For healthy

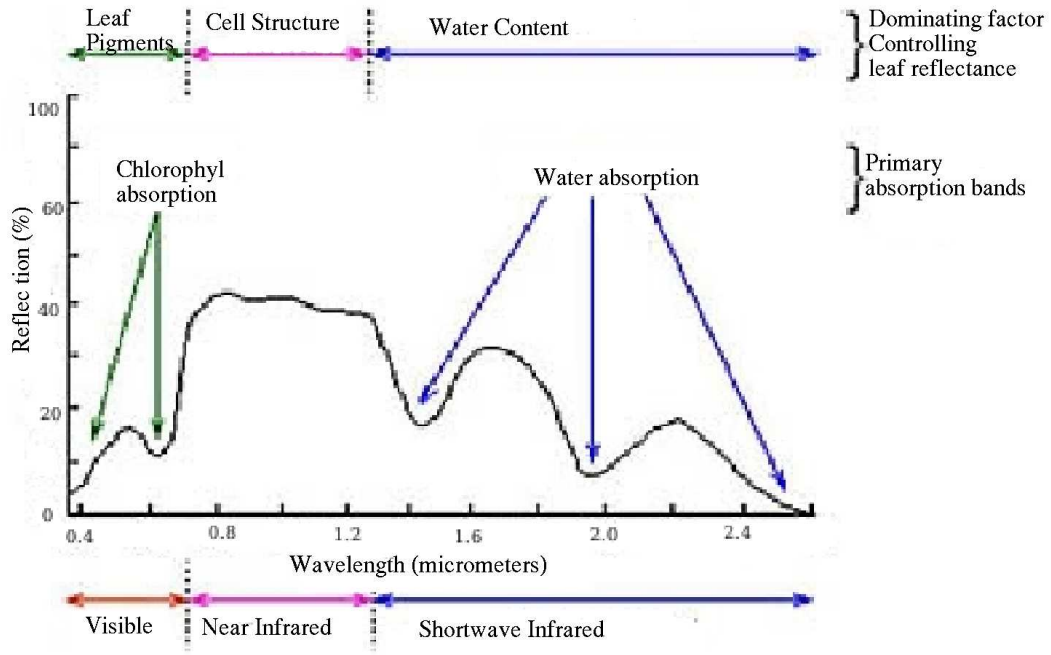


Figure 1.2: Spectral signature of vegetation (Short, 2009).

vegetation, like the example shown in Fig. 1.2, the chlorophyll absorbs the blue and the red bands leaving green reflection as dominant which is why the vegetation appears green for human eyes (Richards and Jia, 1999).

Different materials have their own spectral signature which helps to differentiate them into classes. For example, as shown in Fig. 1.3, the different types of materials show different relative reflectance values at different wavelengths. Sand reflects more energy than pinewoods and grasslands in the visible range and less energy in the infrared range, whereas water reflects almost no energy at all wavelengths.

The aspects of sensor characteristics, geometry and image formation will not be dealt in this thesis. A very good theoretical explanation of these topics is for instance given in Richards and Jia, 1999 and Schowengerdt, 2007. From now on bands in an image correspond to different wavelengths at which the data has been recorded. The pixels in the image are the smallest resolvable spatial units on the earths surface by the sensor. The concepts of resolution will be explained in Sec 1.5.

1.4 Sources of error

Several errors can be introduced while the sensors record the data. Radiometric errors (errors in the measured brightness of the pixels) can be a result of noise in the electronics and the interaction of the radiation with the atmosphere. Errors in geometry arise due

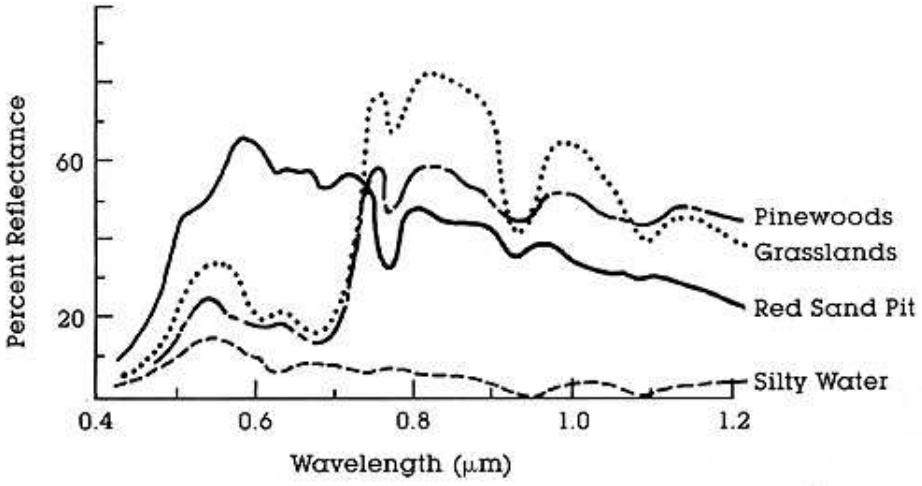


Figure 1.3: Spectral signatures of pinewoods, grasslands, sand and water (Short, 2009).

to the curvature of the earth, the motion of the sensor platform relative to the earth, the geometry of the earths surface, etc.

The radiometric distortion can be of two types (Richards and Jia, 1999):

1. The relative distribution of energy in a wavelength band is different to that in the ground scene;
2. the observed spectral signature is different to that of the spectral signature of the corresponding region on the ground.

Both the above types of errors can be a result of absorption and scattering in the atmosphere and of the instrumentation errors. The atmospheric distortions are wavelength dependent and their effect in RS data can be reduced by applying the atmospheric correction techniques. These techniques use radiative transfer models to correct for the different errors. This requires additional information such as the temperature, humidity, atmospheric pressure, visibility, etc. A detailed explanation of such models can be found in Schowengerdt, 2007. The instrumentation errors can be rectified to a great extent by making the assumption that the detail doesnt change much within a distance of one scan (Richards and Jia, 1999).

The geometrical errors can be introduced by many sources such as (Richards and Jia, 1999):

1. The earths rotation,
2. scan rate,
3. field of view (FOV),

4. earths curvature,
5. sensor non-idealities,
6. uncontrolled errors in the attitude, velocity and position of sensor platform and
7. panoramic effects of the imaging geometry.

The geometric errors can be corrected either by modeling the sources of distortion, if they can be characterized or by simply identifying mathematical relations between the pixels and the corresponding points on the ground. These relations can be used to correct for the image geometry irrespective of the knowledge of the distortion.

1.5 Concepts of resolution

The ability of the sensor to resolve the details is termed as the resolution of the sensor. The potential of the sensor to provide useful information about the earths surface is given by the resolution. There are four types of sensor resolutions: spatial, spectral, radiometric and temporal.

Spatial resolution corresponds to the size of the minimum resolvable spatial unit on the ground. A single pixel in the image is a representation of such a spatial unit. The digital image, which is a grid of pixels, is obtained as a result of the scanning the ground in cross-track direction (perpendicular to the direction of the motion of sensor platform) and in the along-track direction by the platform motion. A pixel is a sample of the continuous data stream of the scanning (in the case of pushbroom scanners, the individual charge coupled devices (CCD) sample the data in the cross track direction). The energy coming from different objects within this spatial unit is averaged, albeit weighted by the *Point Spread Function* (PSF). The PSF is the response of the imaging system to a point object. Normally, the energy from one pixel on ground also affects the corresponding neighboring pixels in the image plane resulting in blurring of the image. The extent of blurring is defined by the PSF. The PSF has contributions from the optical aspects such as diffraction, detectors and also the electronics.

The aggregated irradiance from a spatial unit (or pixel) is converted to an electrical signal at the sensor and quantized as an integer value, called the *Digital Number* (DN). The number of levels in to which the electrical signal is quantized represents the *radiometric resolution*. Like all digital systems, a finite number of bits are used to represent the numbers of quantization. The number of bits is used as a measure of the radiometric resolution. Higher the number of bits more are the levels of quantization and better is the contrast in the produced image.

The *spectral resolution* characterizes the ability of the sensor to distinguish between wavelength intervals. This applies only to the visible/infrared sensors. Prisms or diffraction gratings are used to split the reflected solar energy and spectral filters are used in different paths to aggregate the energy in a band of wavelengths. The better the splitting

of the reflected signal and greater the number of bands used and narrower their width, the higher the spectral resolution is. As seen in Sec. 1.3, the more are the number of sampling points on the spectral signature the better is the chance to differentiate between different materials. The number of bands in the image is a measure of the spectral resolution.

Regular imaging of the same area is vital for several applications of remote sensing involving monitoring. The *temporal resolution* is the revisit time of the sensor platform to image the same area on ground. The interval between the revisits entirely depends on the orbit of the sensors and their off-nadir capabilities.

1.6 Remote sensing satellites

The first Landsat Multispectral Scanner System (MSS) which was launched in 1972 marked the beginning of the era of satellite earth observation. The spatial resolution was 80m and 4 bands in the visible and near infrared region which were 100nm wide were used. Since then many more EO satellites have been launched and a great deal of data has been collected. The present day commercial remote sensing satellites sample nearly all the available parts of the EM spectrum with spatial resolutions ranging from 0.4m (GeoEye-1) to 1000m (MODIS). Also, there exist hyper-spectral systems with hundreds of spectral bands with a width of the order of 10nm. By the end of 2008, 49 optical remote sensing satellites from 22 different nations and 14 radar satellites from 8 nations are orbiting the earth (Stoney, 2008). There are as many as 12 optical satellites with a spatial resolution of $\leq$ 1m. Apart from this, several new satellites will be launched in the next few years. With the improvements in the resolution several applications of remote sensing images are coming in to light. A few important applications are (Schowengerdt, 2007):

- Environmental assessment and monitoring
- Global change detection and monitoring
- Agriculture
- Nonrenewable resource exploration
- Renewable natural resources
- Meteorology
- Mapping
- Military surveillance and reconnaissance
- News media

1.7 Interpretation of RS data

To interpret the images, they have to be classified into classes of interest. Classification is the process of systematic arrangement in groups or categories according to the established criteria. Classification can be viewed as creating thematic maps which show the spatial distribution of desired themes (or classes) in the image. A theme will have a sub-theme which in itself is a theme and hence a refinement of the entire image is attained. Either a general classification rule is defined to separate the different themes or, the clusters in the feature space are separated to form themes. The classifier can be either *supervised* or *unsupervised*. In supervised classification, prototype entity samples are already labeled using ground reference data, existing maps or photo interpretation. The classifier is trained based on the samples and the rest of the entities are classified based on the trained classifier. In unsupervised classification, prototypes are determined to distinguishing intrinsic data characteristics. The heterogeneous distribution of the entities is clustered into groups of entities or so called *clusters*. These clusters are assumed to represent the classes in the image. Supervised and unsupervised methods complement each other in the sense that the external knowledge on the analysis is used to constrain the classes in the former and in the later the inherent structure of the data is determined without any external knowledge (Schowengerdt, 2007). Here, the term *entity* is used instead of the traditionally used *pixel* so as to accommodate the concept of *object* (group of spatially connected pixels having similar characteristics). The concept of objects and a comparison to pixels will be presented in Sec. 1.8.

Historically, *photo interpretation* was used to analyze the images where a skilled and experienced image analyst identifies the features of interest. However, with the increasing information content in the images, the use of computers is now a standard practice. In recent years, the process of creating feature maps is being increasingly automated even though the visual interpretation cannot yet be completely supplanted by computer techniques. This view is called as *image-centered* (Schowengerdt, 2007).

In the second view, which is called as *data-centered*, the interpretation is based on the spectral values rather than the spatial features. As discussed in Sec. 1.3, the classification is done based on the spectral signatures of the classes. Based on the classification, spatial feature maps are then generated by grouping the spatially adjacent entities representing the same class. This view facilitates the automation of image interpretation.

The emerging object-based image analysis can be seen as a third view where it is both *image centered and data centered*. By identifying homogeneous regions in the image (called image objects), it is possible to use both the spatial as well as spectral relations. This characteristic of OBIA makes it a promising methodology in the direction of automation of image analysis tasks.

1.8 The need for OBIA

The size of a pixel is a very important factor when someone looks in to a satellite image. The objects in a natural scene never exist at the same scale, i.e., different objects of interests have different sizes. Moreover, it has to be noted that the pixel or the related instantaneous field of view (IFOV) on the ground consists of different objects and the representation of the pixel value is based on aggregation of energy components from the different objects that are a part of the IFOV. Moreover, the edge of IFOV used to construct a pixel on the image grid almost never overlaps with the edge of the object on ground. This leads to the so called *mixed pixels*. Added to this are the errors induced as discussed in Sec. 1.4.

Despite the problems with pixels, they were considered as basic units of image analysis mainly because of the assumption that different land cover classes behaved like distinct surface materials which can be analyzed using the spectral signature (Castilla and Hay, 2008). The pixel was considered as a sample on the desktop spectrometer. Several classification algorithms (e.g., nearest neighbor, maximum likelihood, artificial neural networks, etc) were developed based on grouping the pixels in a feature space with the axes representing the different wavelength bands (the term *feature* used in this context corresponds to the features of the pixel which are the measured values in different wavelength bands). The spatial structure of the objects could not be dealt with accurately at a lower resolution and was left out of the paradigm. However, with the increasing spatial resolution, the size of the pixels started becoming equal to or smaller than the size of the objects on ground thus, reducing the problem of mixed pixels. However, the internal variability and noise within the pixels of the same class increased with the increasing spatial resolution. Also, with the individual objects on ground clearly distinguishable, new classes began to emerge. For instance, it is very difficult to identify a building in 80m resolution image but, it is easily identified in a 1m resolution image. With the possibility of identification of new types of classes, the context and shape became very important. For example, trees exist in park and in forest as well. The context of the tree is different even if it has the same spectral signature and this means that it is necessary to have two class definitions for tree in this example namely, tree in forest and tree in park. As an example for shape complexity, the pond and river can only be distinguished by considering the shape. It is impossible for the pixel-based classification methods to handle this increasing complexity.

OBIA has emerged as an alternative to the pixel-based techniques. The basic units are no more pixels, but image objects. *Image object is a discrete region of a digital image that is internally coherent and different from its surroundings* (Castilla and Hay, 2008). The image objects are identified by a process called *segmentation*. These image objects are then analyzed based on their shape, texture, context and spectral properties to classify them under certain classes. This approach seems to be close to the way humans interpret images. However, the definition an image object can be quite misleading. The image objects perceived by humans are most of the time different from the image objects which are internally coherent in an image. Despite this, OBIA is still modeled on the basic concepts of visual interpretation. A detailed account of this is provided in chapter 2.

Chapter 2

OBIA and GEOBIA

2.1 A brief introduction to visual perception

The concepts of visual perception are hard to understand as the exact functioning of the human brain is not yet known. In the words of Gordon (Gordon, 2004), *"visual perception utilizes not only the eye which is a structure of formidable complexity but the brain, which in humans comprises ten thousand million cells interacting in ways as yet not understood. Underlying our experience of seeing is the most complicated system ever known."* Several researchers working in this field could come up with theories to explain many of aspects of visual perception. However, no theory can be seen as truly comprehensive. But, by understanding every new theory and analyzing the strengths and weaknesses the understanding of perception will grow.

The theories of perception are too big to be presented in detail within this thesis. So a brief overview of some of the relevant points is presented here. However, it has to be noted that some of the contributions in this thesis are a result of the authors own understanding of visual perception. The relevant points from the published theories of perception are quoted here to support the validity of the authors understanding of perception.

Gestalt theory (Gordon, 2004) is one of the first theories of visual perception and is still considered an effective one mostly because of its simplicity. Gestalt is a German word which means *form*. Some of the claims made by Gestalt school of thinkers were however rejected by subsequent experiments. But, the original laws of perceptual organization postulated by early Gestalt thinkers still have a lot of significance. Wilhelm Wundt in 1879 first proposed the model of perceptual organization called structuralism. It was later formalized and popularized by others. The first doctrine of structuralism is that perception is created by combining elements called sensations. But this single doctrine could not explain many perceptual phenomena. The second doctrine of Gestaltism which was then formulated says that whole is more than the sum of its parts. This basic idea led to development of theories of perceptual organization to define how small elements are grouped into larger objects. This is summarized by the set of laws of perceptual organization that specify how elements are grouped to wholes. These laws were first

mentioned by Wertheimer in 1923 (Wilson and Keil, 2001).

1. *Law of simplicity*: Every pattern is seen is such a way that the resulting structure is as simple as possible. The Olympic symbol in Fig. 2.1 is perceived as five circles because the circle is the simplest shape.

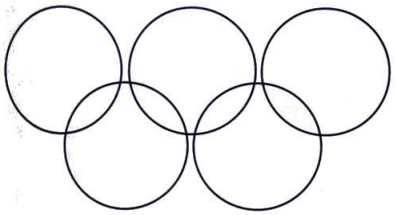


Figure 2.1: The Olympic symbol is perceived as five circles and not as 9 different shapes.

2. *Law of similarity*: Similar things appear to be grouped together. In Fig. 2.2 all the rows have a similar structure but it is visualized as columns of circles and squares.

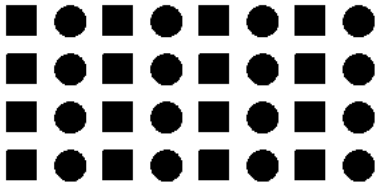


Figure 2.2: Grouping of similar objects.

3. *Law of good continuity*: Points when connected in straight or smoothly curving lines appear to belong together. In Fig. 2.3, the curves from A to B and C to D are connected smoothly.

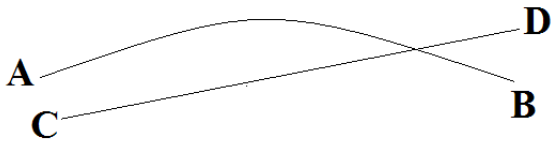


Figure 2.3: The curves go from A to B or C to D and not from A to D or C to B.

4. *Law of proximity*: Objects that are close together appear to be grouped. The law of proximity overrides the law of similarity. The objects in Fig. 2.2 can be visualized as rows if the distance between the columns is smaller than the rows. In Fig. 2.4, the circles are grouped based on the distance between them.

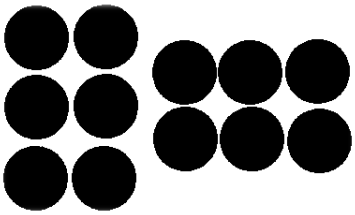


Figure 2.4: The circles in the left appear as columns and the circles in the right appear as columns.

5. *Law of common fate*: Things moving in the same direction appear to be grouped together. For example, the different line segments are grouped based on the orientation even if the length of the segments is not the same in Fig. 2.5

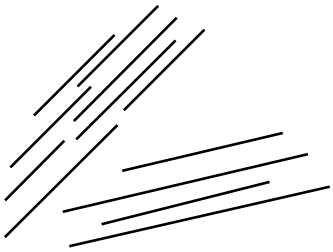


Figure 2.5: The lines are grouped based on the law of common fate.

6. *Law of familiarity*: Things are more likely to form groups if they appear familiar or meaningful. The groups of rocks and leaves resemble faces and can be recognized in Fig. 2.6

Figure 2.6: The Forest has Eyes by Bev Doolittle (1985). The patterns marked in red are perceived as faces based on the law of familiarity.

Another interesting theory from David Marr called Marrs computational approach is based on the assumption that human perception is similar to the way a computer

is programmed to understand things (Gordon, 2004). This assumption obviously is too general. One can agree with Marr on a philosophical level but then the question obviously is about how we are going to identify those algorithms running in the human brain. Marr tried to answer this by suggesting a model as given in Fig. 2.7. Marrs approach is often considered as the background to the artificial intelligence (AI) approach.

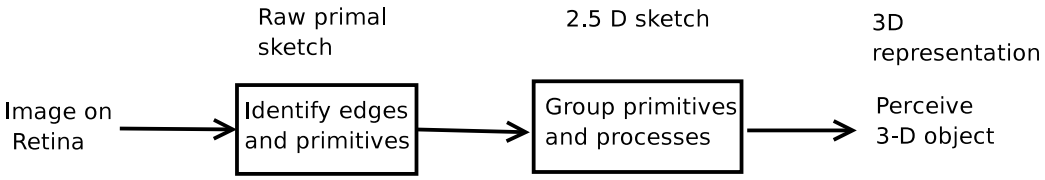


Figure 2.7: Marr's computational approach.

Marr considers that the edges are very important in defining shapes and hence in identifying the objects of interest. A raw primal sketch is first created based on the edges by identifying the collection of basic features such as circles, squares, rectangles, lines, smooth curves, etc. However, the edges are not only based on illumination but also on change in shape. The brain perceives the edges based on both these parameters. The lost parts of the edges due to the problems with illumination are reconstructed by constructing the basic features.

In the second stage, the information being extracted from the image is constructed only with reference to the viewer by understanding vaguely the effect of orientation and getting a sense of rough depth. This rough 2.5D sketch is then used to construct a complete 3D object. The building in Fig. 2.8, which is taken from a high resolution satellite image, is a very good example to illustrate the approach. The basic features in the raw primal sketch are the rectangular shape of the building and an L-shaped shadow. These features define the raw primal sketch. The association of shadow with the building helps in perceiving a depth at the edge of the building where the shadow appears. This is where we have a 2.5D sketch. Based on this, we now identify the building as cuboid structure which defines the 3D sketch. It can also be noted that the illumination differences are ignored on the roof of the building to facilitate extracting simple shapes as mentioned already. However, implementing this approach using a computer is not trivial. It requires making a lot of decisions in modeling the basic shapes while deciding the edges.

The feature integration theory (FIT), proposed by Anne Triesman divides perception into two stages namely, pre-attentive and focused attention stages (Triesman, 1980). In the pre-attentive stage, the individual components of an object are identified based on the boundaries which popup due to dissimilarities along the boundary between two regions. In the focused attention stage, the individual components are combined to perceive complicated shapes. These shapes are then compared to the objects in the memory and a final identification is made based on a match in the memory.

In his book on art and visual perception, Rudolph Arnheim, provides an interesting perspective of psychological forces (Arnheim, 1954). According to him, "what a person perceives is not only an arrangement of objects, colors, shapes, movements and sizes,

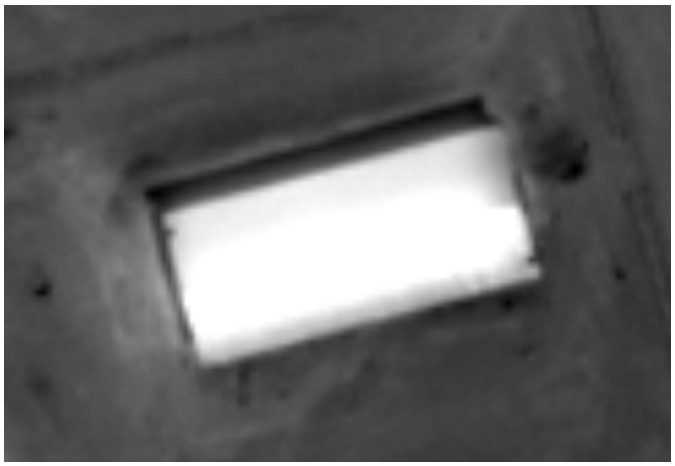


Figure 2.8: The building in this image is perceived based on its shape and the perception of its depth based on the shadow.

but, perhaps first of all, an interplay of directed tensions." The latter are inherent in any percept. Because they have magnitude and direction they are called psychological forces. Rudolph considers that the balance of these forces is very important in perceiving the object. Also, while dealing with shape, he emphasizes that skeletons of the object in combination with the edges are key to represent shapes.

Other significant point which can be distilled from other theories is the importance of context. The perception of an object strictly depends on the context in which it exists. A simple example is given in Fig. 2.9. The alphabets H and A have the same shape but are interpreted differently based on the context of their occurrence and previous knowledge of the words.

Figure 2.9: The alphabets 'H' and 'A' have the same shape but are understood differently based on the context

There are many more theories which explain a lot of interesting things about perception and it seems that this discussion about how we humans think never ends. A good explanation of some of the prominent theories of perception is given in Gordon, 2004. Now that some ideas of visual perception are presented, it is necessary to establish how OBIA is based on these concepts.

Any typical OBIA system starts with segmentation of the image in to primitive objects and these primitive objects are then grouped together into classes of interest depending on the color, shape and context. This processing is exactly similar to the above mentioned theories. But the fact that segmentation of the image into primitive objects is not a trivial problem is of much concern. There exists no segmentation algorithm today which can

reproduce the same primitive objects as the humans. The simplest reason for this is that almost all of the segmentation algorithms are based on color only, completely ignoring the shape and context. The very few algorithms which consider shape do not deal with it in the way humans perceive, as explained by Gestalts laws for instance. Moreover, there exists no comprehensive way to explain the human understanding of shapes in a variety of situations. However, the practitioners of OBIA try to overcome this problem by creating primitive objects which are as homogeneous as possible and try to group them depending on the class type to create objects which can come close to human perception. Segmentation is one step which is severely hindering the growth of this field.

2.2 A word about texture

Texture is one aspect of image processing which is easy to recognize but hard to describe. *Texture is a measure of the variation of the intensity of a surface, quantifying properties such as smoothness, coarseness and regularity* (Howe, 1995). In nature, texture is almost never built on a basic local pattern in any way. It is mostly random. A texture can be either directional like in plantations or non directional, like in a forest. It can be smooth like a calm lake or quite disturbed like in a stormy ocean. And within texture there can be many levels of abstraction. The texture is made up of elements which build a pattern. The segmentation of texture images in general is complimentary to segmentation of homogeneous regions as texture can be visualized as a group of homogeneous objects at a different level of abstraction, but having a specific ordering in terms of any of the characteristic defining the texture. In this work, the segmentation of the texture regions will not be dealt with. However, the features which can be used to describe texture are explained in chapter 5.

2.3 OBIA to GEOBIA

Till now, GEOBIA was defined as OBIA applied on geographic images. If that is so, then, what is the need of giving a new name and creating a new field? The field of OBIA and the corresponding development of techniques to analyze grey-level images already exist from a very long time in the fields of Computer Vision and Biomedical Image Analysis (Castilla and Hay, 2008). But, these techniques were not completely applicable to the RS images mainly because of the following reasons (Schiewe et al., 2001):

1. The RS sensors provide a variety of data not only in multiple wavelength bands but also at different resolutions. This increases the complexity and redundancy of the data.

2. Other types of data such as data based on GIS and very good elevation data are being made available with latest developments in technology. This allows extracting more information from the images.

3. The objects encountered in RS data have heterogeneous properties with respect to size, form, spectral behavior, etc.

4. It is very difficult to employ model-based interpretation because of the heterogeneity of the inherent object classes.

5. A very high accuracy is desired with respect to object identification using RS data.

There is a need to deal with RS data in a different way and dedicated methodologies have to be developed to interpret it. GEOBIA can be one of the right methodologies. The suffix GEO in GEOBIA is merely used to emphasize the geographic connection.

While image-objects are a result of the process of segmentation of the image, a geo-object is a group of connected image-objects which has a specific geographic meaning. It can be defined as a bounded geographic region that can be identified for a period of time as the referent of a geographic term (Castilla, 2003).

Since the GEOBIA methodology is based on image-objects and not real geographic-objects, it is not geographic-object based image analysis but instead geographic object-based image analysis. We can now finally define GEOBIA as follows (Hay and Castilla, 2008):

"GEOBIA is a sub-discipline of Geographic Information Science (GIScience) devoted to developing automated methods to partition remote sensing imagery into meaningful image-objects, and assessing their characteristics through spatial, spectral and temporal scales, so as to generate new geographic information in GIS- ready format."

The simple flowchart of GEOBIA is shown in Fig. 2.10. It starts with pre-processing the images so as to generate homogeneous regions in the images to aid the segmentation of the images in to primitive image-objects which are then grouped together to form geo-objects by means of classification. The geo-objects are used for further analysis to aid in the extraction of other geo-objects or analyze the group of geo-objects which make up a class.

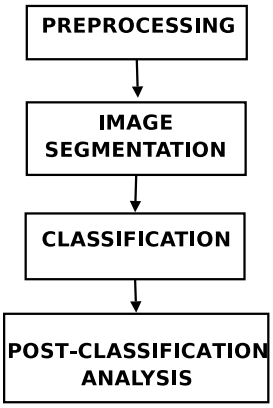


Figure 2.10: The GEOBIA workflow

Chapter 3

Pre-processing

The main objective of pre-processing the images in this thesis is to modify the data or create new image layers so that the data is suitable for segmentation algorithms. The extraction of the correct image objects is a key for successful OBIA. In most of the segmentation algorithms, the primary requirement of extracting the image objects is that the objects are homogeneous. So, the goal of the pre-processing step would be to increase the homogeneity in the image objects so that they can be extracted easily. Several filtering methods can be employed depending on the desired result or data transformations can be used. This chapter will describe briefly filtering techniques primarily in the spatial domain and orthogonal transformation schemes to transform the data into more suitable coordinate systems. Also, the image sharpening algorithms which are used to enhance the data quality by improving the spatial resolution will be explained. Other data transformations such as the Normalized Difference Vegetation Index (NDVI) which can be considered as indices for certain classes will also be detailed. Only a brief introduction is provided for the above methods so as to focus more on the contributions of this author rather than the existing methods.

3.1 Image filtering

The most common filtering techniques are in the spatial domain where the 2D image is convoluted with 2D filter matrices of size much lower than the dimensions of the image. A filter can be visualized as a mask window containing weights for every pixel represented by a window around the pixel of interest. These weights are used to calculate a weighted sum of the pixels in that window. The sum is the resulting filtered value of that pixel. The size of the filter is always an odd number to ensure that the pixel of interest stays in the center of the mask. For example, an image can be smoothed by averaging the pixels in the neighborhood of every pixel. Simple smoothing filters of size 3×3 is shown in Fig. 3.1. Smoothing increases the homogeneity of the objects in the image but it also reduces the contrast at the edges between objects. So, this might not be a desirable pre-processing step in the context of OBIA.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

Figure 3.1: 3×3 Smoothing filters

Line Detection

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal

-1	2	-1
-1	2	-1
-1	2	-1

Vertical

2	-1	-1
-1	2	-1
-1	-1	2

Diagonal

Gradients

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

North

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

South

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

West

Sharpening

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

Sobel edge filters

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

Horizontal

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

Vertical

Figure 3.2: Examples of 3×3 convolution filters

Several other types of filters can be formulated by modifying the weights in the filter windows for various applications such as line detection, edge detection, sharpening, etc. Some examples of convolution filters are shown in Fig. 3.2.

Filtering can also be done in the frequency domain where the image is converted from spatial domain to frequency domain using the discrete Fourier transform (DFT) and a mask is used to eliminate the unwanted frequencies. Smoothing is considered as removing the high frequency components and sharpening is removing the low frequency components. The fact that noise in the images is considered to be of high frequency allows to reduce the noise in the image by removing the high frequency components identified using the DFT.

Apart from convolution or frequency domain filtering, the statistical measures such as mean or median in a pixel neighborhood are also used. Morphological image pre-processing is a relatively new development in image pre-processing. Morphological filters use a so called structuring element which is a mask around a pixel of interest describing which pixels have to be considered for pre-processing. Typical structuring elements used are the masks in the shape of a square, diamond or a circle. In the case of binary (or

bi-level) images, the structuring element can also have pixel positions with the *don't care* values which means that those pixel positions can have any values.

The morphological operations are mainly defined for binary images. Erosion and dilation are the basic operations in morphological image pre-processing and other operations are built upon them. Erosion can be considered as shrinking the foreground with respect to the background and dilation as the reverse. This definition is valid for a bi-level image but for a gray scale image it is hard to distinguish between the foreground and the background.

Some of the morphological operations for binary images are defined below (Gonzalez and Woods, 2002):

1. *Dilation* expands the boundary of the foreground. The dilation of image A with respect to structuring element B is represented as $A \oplus B$. To manage the standard definition for both binary and gray level images, dilation can be defined as the maximum value of the pixels defined by the structuring element around the pixel of interest.

2. *Erosion* contracts the boundary of the foreground. It is represented as $A \ominus B$. It is the minimum value of the pixels defined by the structuring element around the pixel of interest.

3. *Opening* smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. It is erosion followed by dilation operation.

$$A \circ B = (A \ominus B) \oplus B \tag{3.1}$$

4. *Closing* smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. It is dilation followed by erosion operation.

$$A \bullet B = (A \oplus B) \ominus B \tag{3.2}$$

5. *Hit-or-miss transformation* identifies a set of pixels at which simultaneously, structuring element B_1 found a match ("hit") in A and structuring element B_2 found a match in the compliment of A

6. *Boundary extraction* identifies the boundary pixels of the foreground.

$$\beta(A) = A - (A \ominus B) \tag{3.3}$$

7. *Skeletonization* finds the skeleton $S(A)$ of a shape A and A can be reconstructed from skeleton subsets $S_k(A)$.

$$S(A) = \cup_{k=0}^{K} S_k(A)$$

$$S_k(A) = \cup_{k=0}^{K} \{(A \ominus kB) - [(A \ominus kB) \circ B]\} \tag{3.4}$$

Reconstruction of A:

$$A = \cup_{k=0}^{K} (S_k(A) \oplus kB) \tag{3.5}$$

K is the value of the iterative step after which the set A erodes to the empty set. The notation $(A \ominus kB)$ denotes the kth iteration of successive erosion of A by B.

The definitions of erosion, dilation, open and closing also apply for gray level images. Morphological smoothing is defined as performing opening and then closing operations. The morphological gradient is the difference of dilation and erosion operations.

3.2 A new adaptive filter

The aim of filtering in the context of OBIA as mentioned earlier is to enhance the data so as to help the segmentation. A better segmentation is achieved when the objects to be extracted are homogeneous. So, the main aim is to increase the homogeneity in the image while increasing the contrast at the edges. In this context, a filter for smoothing the image adaptively has been developed. The algorithm for adaptive smoothing is given below:

Algorithm for adaptive smoothing:
Parameters:
N (no. of iterations),
l (window size),
f_1, f_2 (fractions indicating allowable difference to the maximum and minimum values respectively).
Iterate steps 1 to 2, N times

1. Let mx $\leftarrow$ maximum pixel value in the window
   mn $\leftarrow$ minimum pixel value in the window
   px $\leftarrow$ pixel value of the pixel of interest and
   nr$\leftarrow$ pixel value in the window closest to px.

2. If (mx-px) $\leq f_1$*px, then result=(mx+px)/2
   Else
   If (px-mn) $\leq f_2$ * px then result=(mn+px)/2
   Else result=(px+nr)/2
   OR
   if px=mx then result=(px+nr)/2

The algorithm tries to differentiate the foreground and background at the pixel of interest and subsequently average it with the maximum value, minimum value or the pixel value closest to the pixel of interest. This process is iterated thereby increasing the homogeneity in the image while also improving the edge contrast. Fig. 3.3 shows a profile from an image processed using the above algorithm. It can be seen that the sinsuoidal edges are removed

and well defined crests and troughs which represent the homogeneous objects are created in the image.

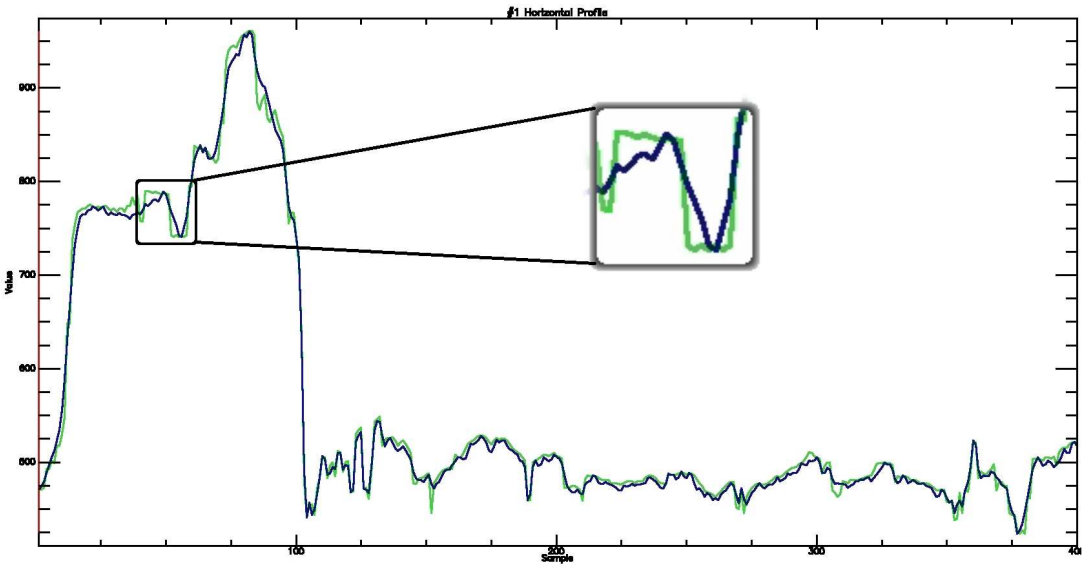


Figure 3.3: Comparison of a profile before and after adaptive smoothing. The profile in blue is before smoothing and the one in green is after smoothing with parameters f_1=0.02, f_2=0.02, window size=3, no. of iterations=3.

The algorithm can also be considered as a segmentation algorithm when appropriate values are selected for the parameters t1,t2 and the iterations are carried out till nothing changes in the image. This will produce groups of connected pixels with the same value which are in fact image objects. However, it would require a high computational cost as the iterations are carried out for all the pixels till the algorithm converges.

Another algorithm to only enhance the edge contrast without affecting the homogeneity much is also developed. A relative value depending on the edge strength is added or subtracted from the pixel value to increase the contrast at the edges.
The algorithm for edge enhancement is shown below:
Algorithm for edge enchancement:
Parameters:
C (multiplication factor)

1. Let mx ← maximum pixel value in the window
   mn ← minimum pixel value in the window and
   px ← pixel value of the pixel of interest
2. if (px-mn) (mx-px) then sign=-1 else sign=+1 or if px=mx or px=mn then sign=0
3. result=px+(C*(1-(mn/mx))*sign)

If the pixel value is closest to the maximum pixel value in the window the pixel value

is increased and decreased otherwise. The term (1- mn/mx) corresponds to the morphological gradient normalized by the maximum value. This fraction represents the strength of the edge in the image. The result therefore increases the contrast at the edges. Fig. 3.4 shows a profile from an image processed using the above algorithm. It can be seen that the edges are enhanced in the image. However, at the regions where very thin objects

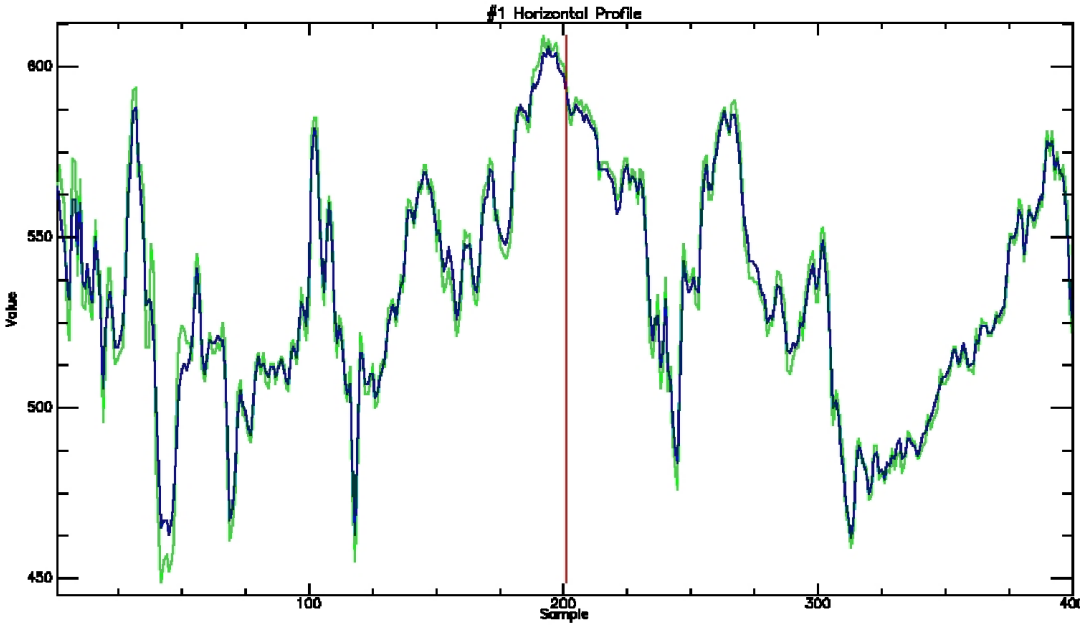


Figure 3.4: Comparison of a profile before (Blue) and after (Green) edge enhancement with c=100

exist, the edge intensity and direction as calculated in this algorithm do not give correct values but result in changing the pixel values in an undesirable way. The combination of the two algorithms may produce better results.

The effect of adaptive smoothing on the segmentation can be observed in the images of Fig. 3.5 . Fig. 3.5a shows the segmentation without pre-processing and the Fig. 3.5b shows segmentation after pre-processing. The image is a small part of a scene acquired by *Quickbird* satellite over Esfahan, Iran in July, 2003. It can be observed that the under-segmentation effect is reduced by a great extent by adaptive smoothing and edge enhancement. The multi-resolution segmentation algorithm of *Definiens Professional* software is used here (See chapter 4).

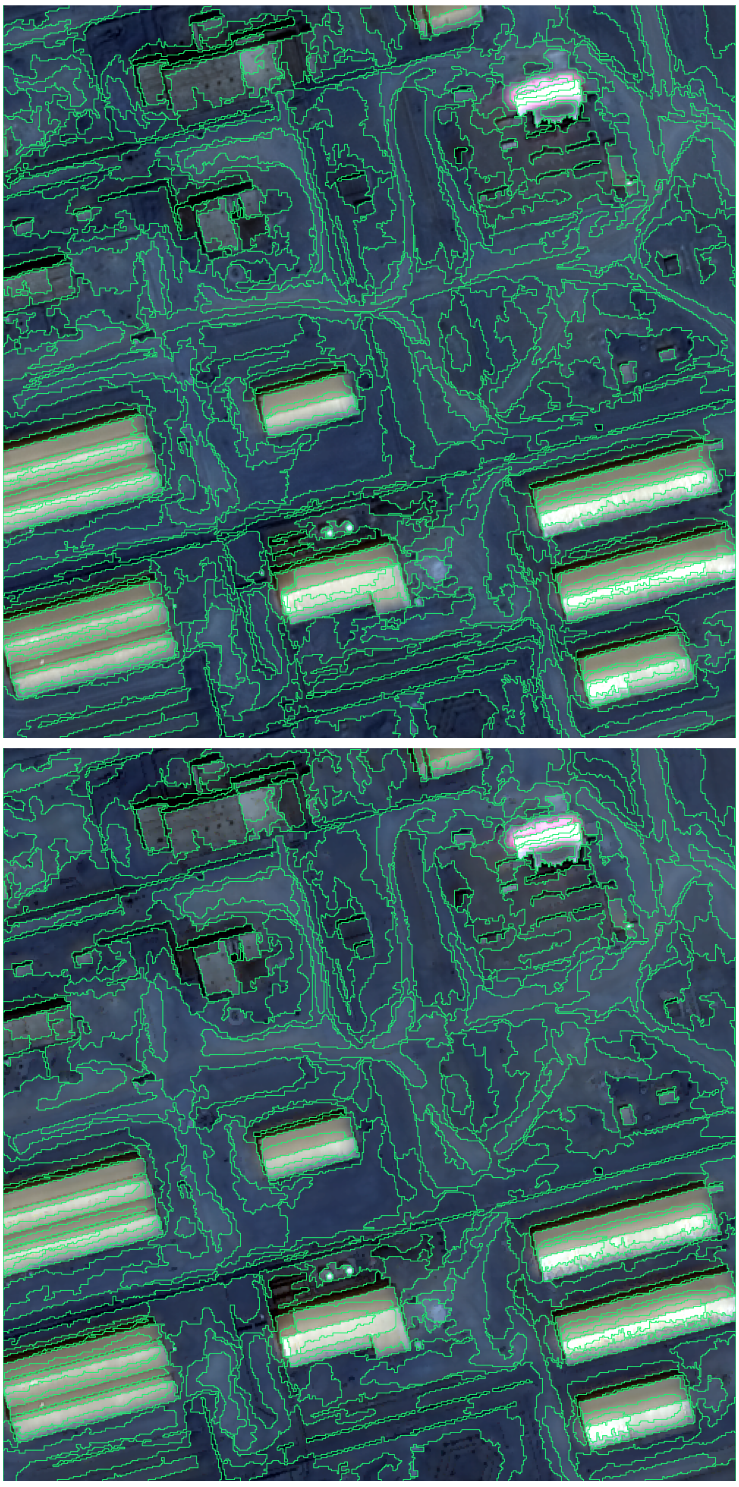


Figure 3.5: Segmentation of Quickbird image with scale parameter=75, shape factor=0.1, compactness=0.5. (a) before smoothing and (b) after smoothing.

3.3 Orthogonal Transformations

It is often convenient to transform the data into a different coordinate system for a variety of purposes which include reducing the dimensionality and increasing the contrast. Orthogonal transformations do not modify the structure of the original data but only use the rotation, translation and scaling operations to represent the data in a new set of axes of the same dimension as the original data. So, it is possible to retrieve the original structure of the data by a inverse transformation. The most commonly used orthogonal transformations are the principal components (PC), maximum noise fraction (MNF) and minimum/maximum autocorrelation factor (MAF) transformations. The transformations can be used to decrease the redundancy of the data which arises because of the covariance between the variables. An explanation of several other orthogonal transformations can be found in (Anderson, 1984).

3.3.1 Principal Components (PCs)

This method was developed by Hotelling in 1933 based on the technique described by Pearson in 1901 (Hotelling, 1933). The principal components of a stochastic multivariate data are calculated based on a linear transformation which produces uncorrelated bands of decreasing variance. An estimate of the dispersion matrix is required to calculate the PCs. The covariance matrix of the different bands is normally used. The PCs are calculated so as to maximize the variance in every component. A higher order PC is the linear combination of the original bands with maximum variance subject to the constraint that it is uncorrelated with all the other components.

Consider an image of K bands represented as a random vector G with zero mean. The constraint of zero mean allows to calculate the mean covariance matrix as

$$\Sigma = \left\langle GG^T \right\rangle . \tag{3.6}$$

The aim is to find a linear combination $Y = a^T G$ which maximizes the variance $a^T \Sigma a$ under the constraint $a^T a = 1$. Maximizing the unconstrained Lagrangian function given in Eq 3.7 leads to the eigenvalue problem given by Eq. 3.8

$$L = a^T \Sigma a - \lambda(a^T a - 1), \tag{3.7}$$

$$\Sigma a = \lambda a. \tag{3.8}$$

The eigenvectors are the new principal axes and the corresponding linear combinations $a_i^T G$ which are the projections along the principal axes are called as principal components where, a_i is the eigenvector corresponding to the highest eigenvalue. The eigenvector with the highest eigenvalue corresponds to the component with highest variance. So, the components are ordered in the descending order of the eigenvalues. The fraction of the

total variance accounted for by the first i principal components is given by

$$\frac{\sum_{k=0}^{i} \lambda_i}{\sum_{k=0}^{K} \lambda_i}. \tag{3.9}$$

Normally only the first few principal components account for the most of the variance. This fact is used to reduce the dimensionality of the data by considering only the first few principal components which account for the most of the variance.

3.3.2 Maximum Noise Fraction (MNF)

The PCs may not always be the components of decreasing image quality. The maximization of variance is not the best approach if we are working with spatial data. The *Maximum Noise Fraction* (MNF) transformation (Green et al., 1988) maximizes the *Signal-to-Noise Ratio* (SNR) rather than the variance. So, if the aim is to reduce the noise this MNF transformation is preferred over PC transformation.

Let the image, G be represented as a sum of signal and noise contributions,

$$G = S + N. \tag{3.10}$$

If Σ_S and Σ_N are the covariance matrices of signal and noise respectively, the covariance matrix of the image G is given as

$$\Sigma = \left\langle GG^T \right\rangle = \left\langle (S+N)(S+N)^T \right\rangle = \left\langle SS^T \right\rangle + \left\langle NN^T \right\rangle, \tag{3.11}$$

under the assumption that the signal and noise are not correlated.
Therefore,

$$\Sigma = \Sigma_S + \Sigma_N. \tag{3.12}$$

The SNR in the ith band is often expressed as:

$$SNR_i = \frac{var(S_i)}{var(N_i)}. \tag{3.13}$$

Now, we have to find the linear combination $Y = a^T G$ for which the SNR is maximum. The SNR for this linear combination is

$$SNR = \frac{var(a^T S)}{var(a^T N)} = \frac{a^T \Sigma_S a}{a^T \Sigma_N a} = \frac{a^T \Sigma a}{a^T \Sigma_N a} - 1. \tag{3.14}$$

Maximizing this leads to solving the *generalized eigenvalue problem* for a as,

$$\Sigma_N a = \lambda \Sigma a. \tag{3.15}$$

It can be derived that

$$SNR_i = \frac{1}{\lambda_i} - 1. \tag{3.16}$$

So, the projection $Y_i = a_i^T G$ corresponding to the smallest eigenvalue λ_i will have the largest SNR. For the MNF transformation, we have to estimate the noise covariance matrix. This can be estimated by using the fact that the *autocorrelation* of the image is close to 1 i.e., the pixel intensities of the neighboring pixels are very similar. This explained in connection with the MAF transformation.

3.3.3 Maximum Autocorrelation Factor (MAF)

While no spatial information is used in the PC and MNF transformations, the MAF transformation uses the spatial correlation between the neighboring pixels. The aim is to find a transformation which maximizes the autocorrelation of the image.

The spatial covariance $\Gamma(x, \Delta)$ is defined as

$$\Gamma(x, \Delta) = \left\langle G(x)G(x + \Delta)^T \right\rangle, \tag{3.17}$$

x stands for the position of the pixel. $G(x+\Delta)$ is the image $G(x)$ shifted by $\Delta = (\Delta_1, \Delta_2)^T$

Using the *second order stationarity assumption*,

$$\Gamma(x, \Delta) = \Gamma(\Delta) \Leftrightarrow \text{independent of } x,$$

$$\Gamma(0) = \left\langle GG^T \right\rangle = \Sigma. \tag{3.18}$$

If we define $Y_i = a^T G$ as projections in the new coordinate system. The *spatial autocorrelation of the projections* can be deduced as

$$corr(a^T G(x), a^T G(x + \Delta)) = 1 - \frac{1}{2}(\frac{a^T \Sigma_\Delta a}{a^T \Sigma a}). \tag{3.19}$$

So, the MAF transformation determines that vector a which maximizes the above correlation. This again leads to a generalized eigenvalue problem as in the case of MNF transformation as below

$$\Sigma_\Delta a = \lambda \Sigma a. \tag{3.20}$$

Now, this leads to the conclusion that under fairly general considerations Σ_Δ and Σ_N are proportional to each other. It can be deduced that Σ_N is approximately equal to $\frac{1}{2}\Sigma_\Delta$ (Canty,2007).

3.4 Other Transformations

Apart from the orthogonal transformations, we can also transform the data using other empirical or theoretical transformations related to spectral properties of the classes of interest. For instance, the *Normalized Difference Vegetation Index* (NDVI) based on the data in the Near Infrared (NIR) and Red channels is defined as follows

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{3.21}$$

NDVI scaled to the range of the other layers can be used as an additional image layer for segmentation as will be seen in chapter 6. This aids in segmenting images with vegetation as NDVI clearly distinguishes between vegetation and other classes. Several such indices exist in literature for different types of classes.

3.4.1 Ratios

Sometimes band ratios similar to the NDVI can be useful in separating certain classes. Although they can not be scientifically justified like NDVI, the band ratios, which are in fact non-linear transformations of input bands, can help in extracting certain classes. A program has been developed to generate coefficients for combining bands in all the possible ways at a reasonably high speed. A decimal number to binary number converter is used to first produce all the possibilities of the positive coefficients and then the possibilities having negative coefficients are included by just changing the sign and adding back to the possibilities with positive coefficients. This method allows avoiding loops which are quite time consuming to iterate to get all the possibilities. This method can be used to generate coefficients for uptill 63 bands as most of the programming languages only have support for 64 bit integers. For example, for the data set with three bands, the coefficients will be as follows:
(1, 0, 0), (0, 1, 0), (1, 1, 0), (0, 0, 1), (1, 0, 1), (0, 1, 1), (1, 1, 1), (1,-1, 0), (1, 0,-1), (0, 1,-1), (1, 1,-1), (1,-1, 1), (-1,1, 1)
A two level loop over these coefficients for the numerator and denominator respectively then produces all the possible band ratio combinations. Based on the samples for the classes a separability measure can be used to identify band ratios that can best separate the combinations of the classes (see Sec. 6.2).

These band ratios however are only valid for the image on which they are calculated. They may or may not be transferable for other datasets. Some examples were tried to check the effectiveness of the identified ratios. The band ratio $(b2+b3-b4)/(b2-b3+b4)$, where $b2, b3, b4$ are the Green, Red and NIR channels of a Landsat scene from Kalimantan forest in Indonesia seems to extract the road network through the forest region. In another example, the band ratio $(b1-b3-b4)/(b1+b3+b4)$, where $b1, b3, b4$ are the Blue, Red and NIR channels of a Landsat scene over the coast of Israel and Gaza strip seems to clearly identify the coastline irrespective of the suspended sediments in the coastal seawaters whereas the ratio $(b2 - b4)/(b2 + b4)$ separates the regions with and without suspended sediments along the coastline in the Mediterranean sea. This way of identifying features depends on the selected samples and the identified ratios can change for a different set of samples. The above examples were only random trials and the above way of identifying band ratios is not used in any of the examples in this thesis. However, this method can be used to generate artificial layers to extract some classes which show good separability for certain ratios identified using samples of the class.

3.5 Image Sharpening

Often satellite data in different frequency bands is acquired at different resolutions, or a panchromatic image layer which has the highest resolution of all the acquired image layers is most of the times provided. It is possible to improve the data quality of the lower resolution images by fusing the geometry of the higher resolution image layer with the spectral values of the low resolution image layers. This fusion is often referred to as image sharpening. Since, most of the times panchromatic image layer and multispectral data are involved the term *pan-sharpening* is used in general. Several methods exist for such fusion. Based on the observations by Canty, 2007 and Nussbaum and Menz, 2008 using the Wang-Bovik quality index (Wang and Bovik, 2002) and after visual inspection, three algorithms are selected to be used in this work. As reported in (Nussbaum and Menz, 2008), the performance of the algorithms is not consistent for different regions with different classes. So, it is not possible to decide about the quality of the results of various algorithms. The three algorithms which are found to be more consistent are:

1. Discrete wavelet sharpening
2. Á trous cubic spline filter sharpening
3. Gram-Schmidt spectral sharpening

The following explanation of the sharpening methods will use the panchromatic data as high resolution data and the multispectral bands as the low resolution data. Only a brief introduction of the methods is provided here. A detailed explanation can be for instance found in (Canty, 2007) and (Nussbaum, 2008).

3.5.1 Discrete wavelet sharpening

The Discrete Wavelet Transformation (DWT) filter bank can be used to sharpen the low resolution multispectral bands using the high resolution panchromatic band (Ranchin and Wald, 2000). The image is filtered using the high pass and low pass filters first and the resulting images are downsampled by a factor of two along the columns and again filtered using the high pass and low pass filters. This process yields four components of the image which are again downsampled along the rows. These four components are represented as four quadrants in a single image with the low pass-low pass component in the top left part. This component also corresponds to the degraded version of the original image. The low pass-high pass component is in the top right part with the high pass-low pass component in the bottom left and high pass-high pass component in the bottom right quadrant. The same process can be iteratively repeated in the top-left quadrant to generate coefficients at multiple levels.

For the purpose of image sharpening, the DWT filter bank is repeatedly applied over the panchromatic band till the size of the top left quadrant (the degraded image) is equal to that of the multispectral band. The degraded image is then replaced by the

multispectral band and the filter bank is inverted to produce the sharpened version of the multispectral band. Before doing this, the radiometric normalization coefficients are estimated by once applying the DWT filter bank to the multispectral band and the degraded image simultaneously and matching the statistics (mean and standard deviation) of the components.

3.5.2 Á trous cubic spline filter

The DWT is not *shift invariant* which means that the small spatial displacements in the panchromatic band can cause major variations in the wavelet coefficients at their various scales. So, when a multispectral band is injected for inverse filtering, this can lead to spatial artifacts (Yocky, 2006). The *À trous Wavelet Transform* (ATWT) is proposed as an alternative for DWT (Aiazzi et al., 2002).

Unlike the DWT, instead of downsampling the panchromatic band, the multispectral band is upsampled here. In every iteration the low pass filter is modified by simply inserting zeros between elements (hence the name à trous = holes; the zeros correspond to holes). A B-spline filter is often chosen as the low pass filter as it is symmetric (Núnez et al., 1999). The high pass filter is simply a difference of all pass filter and low pass filter i.e., the result of low pass filtering is subtracted from the original image.

For sharpening, the multispectral band is first upsampled and the à trous transformation applied for the upsampled multispectral band and panchromatic band simultaneously. The radiometric normalization coefficients are calculated by matching the statistics of the high frequency components and the component of the panchromatic band is normalized with those coefficients. The low frequency component of the panchromatic band is replaced by the low frequency component of the multispectral band and the transformation is inverted to result in a pansharpened multispectral band.

3.5.3 Gram-Schmidt spectral sharpening

The Gram-Schmidt fusion (Laben et al., 2000) first simulates a panchromatic band from the lower spatial resolution spectral bands by averaging the multispectral bands. A Gram-Schmidt transformation is performed for upsampled versions of the simulated panchromatic band and the multispectral bands with the simulated panchromatic band employed as the first band. The first Gram-Schmidt band is then replaced by the high spatial resolution panchromatic band. And finally, an inverse Gram-Schmidt transformation creates the pansharpened multispectral bands.

3.6 Relative radiometric normalization

The relative radiometric normalization of the images of the same area but of different times can be done by using the change detection methods (Canty and Nielsen, 2008, Canty et al., 2004). A change detection is first done over the images of two times and then the pixels that did not change in between the two times are only considered to find the coefficients to normalize the data in every band independently based on the assumption that the relation between the at-sensor radiance recorded at two different times can be approximated by linear functions. There exist several methods for unsupervised change detection. A review of different methods can be for instance found in (Coppin et al., 2004). The *Iteratively Re-weighted Multivariate Alteration Detection* (IRMAD) method (Nielsen 2007) has been extensively used in this thesis to perform a relative radiometric normalization of images.

3.6.1 Iteratively Re-weighted Multivariate Alteration Detection (IR-MAD)

Consider two K band multispectral images F, G of the same scene but acquired at two different times. Assume that both the images have zero mean (This is generally achieved by subtracting the images by their respective means). We can represent the observations in different bands of the multispectral images as the random vector $F = (F_1, F_2, F_3, ...F_K)^T$ and $G = (G_1, G_2, G_3, ...G_K)^T$. Consider the random variables U, V generated by any linear combinations of the intensities of the spectral bands as

$$U = a^T F = a_1 F_1 + a_2 F_2 + ... + a_K F_K, \tag{3.22}$$
$$V = b^T F = b_1 G_1 + b_2 G_2 + ... + b_K G_K. \tag{3.23}$$

The random variable created by the difference $U - V$ represents the combined change information in a single image. The task is to find suitable vectors a and b. This can be done by maximizing the correlation between U and V given by (Nielsen, 2007),

$$\rho = \frac{cov(U, V)}{\sqrt{var(U)var(V)}}. \tag{3.24}$$

A constraint is chosen to avoid the case of arbitrary multiples of U and V resulting in components with same correlation.

$$var(U) = var(V) = 1. \tag{3.25}$$

This leads to solving two *generalized eigenvalue problems* which are coupled via the parameter ρ as follows:

$$\Sigma_{fg}\Sigma_{gg}^{-1}\Sigma_{fg}^T a = \rho^2 \Sigma_{ff} a, \tag{3.26}$$
$$\Sigma_{fg}^T\Sigma_{ff}^{-1}\Sigma_{fg} b = \rho^2 \Sigma_{gg} b. \tag{3.27}$$

where Σ_{ff} is the covariance matrix of the first image,Σ_{gg} is the covariance matrix of the second image and Σ_{fg} is the cross-covariance matrix between the two images.

The desired projections $U = a^T F$ and $V = b^T G$ are given by the eigenvectors $a_1...a_K$ and $b_1...b_K$ respectively of the above equations sorted in the order of increasing eigenvalues for convenience.

The following K differences are called as *MAD variates*, M:.

$$M_i = U_i - V_i = a_i^T F - b_i^T G,\ i = 1....K, \tag{3.28}$$

The corresponding ρ_i are called as *canonical correlations* and the quantities U_i, V_i are called as *canonical variates*. The canonical variates and the MAD variates are mutually uncorrelated. This defines the MAD transformation. It can be easily derived that the MAD transformation is invariant to any linear affine transformations.

Now, let the random variable Z represent the sum of the squares of standardized MAD variates:

$$Z = \sum_{i=1}^{K} \left(\frac{M_i}{\sigma_{M_i}} \right)^2. \tag{3.29}$$

Based on the fact that the no-change observations are normally distributed and uncorrelated, the realizations z of the random variable Z should be chi-square distributed with K degrees of freedom. This gives us a chance to define the change/no-change probabilities as (Nielsen, 2007),

$$Pr(\text{no change}) = 1 - P_{\chi^2;K}(z), \tag{3.30}$$

where χ^2 represents chi-square distribution. $Pr(\text{no change})$ is the probability that a sample z drawn from a chi-square distribution could be that large or larger. The smaller the value of z the higher is the probability. The above probabilities of no-change can be used as the weights of the observations and the entire process is iterated until a stopping criterion is met. The criteria can either be a fixed number of iterations or when there is no significant change in the canonical correlations. This defines the iterated version of the MAD or the IR-MAD.

Chapter 4

Image Segmentation

Image segmentation is the first step in OBIA. The aim of the image segmentation is to divide the image into meaningful segments (or *image objects*). Every image object is characterized by a huge set of features which define the spectral properties, shape, texture, context, etc (see chapter 5). These image objects may or may not correspond to the real world objects. However, they are further analyzed to define real-world objects in the image. This process is called *image classification* (see chapter 6). Image segmentation is the most important step of OBIA, because of the fact that the basis for extracting the real world objects is the proper segmentation of image in to the object components i.e., image objects. There exists hundreds of image segmentation algorithms but till now, there is no proper review of the segmentation algorithms available. This is probably due to the fact that most of the algorithms are developed for specific tasks and are not consistent for all the types of images. Moreover, most of the algorithms are not suited for remote sensing data which is different from other images in a lot of ways e.g., several layers of data from a lot of spectral channels, the radiometric errors induced in the images, etc. For remote sensing images, contributions such as (Carleer et al., 2005, Neubert et al., 2008) tried to evaluate the segmentations based on benchmark datasets by manually defining segments and then finding the goodness of the algorithms based on some definitions of quality measures. A simple method for segmentation evaluation is also presented in this thesis to evaluate the segmentation results(see Sec. 4.5). However, it is not easy to quantify the goodness of the algorithms based on a few examples. The algorithms should stand the test of segmenting images from a variety of sensors and in a variety of environments which requires a very large benchmarking dataset. Image segmentation in itself is an enigma and as discussed in chapter 2 there are no specific rules of how homogeneity in the object can be defined consistently over the entire image.

4.1 Definition of segmentation

Image segmentation is defined as grouping the pixels in to a set of n segments

$$S = S_1, ..., S_n. \tag{4.1}$$

where, S refers to the set of image segments S_i such that,

$$S_i \subseteq I, \tag{4.2}$$
$$\forall j \neq i,\ S_i \cap S_j = \phi, \tag{4.3}$$
$$\bigcup S_i = I, \tag{4.4}$$

where, I is the set of all the pixels i.e., the image.
Every segment is a set of pixels such that,

$$S_i = x_{rc} \mid \exists x_{kl} \in S_i,\ r-1 \leq k \leq r+1,\ c-1 \leq l \leq c+1, \quad if |S_i| > 1 \tag{4.5}$$

r, c are the row and column indices of the pixel in the image.
i.e., there exists at least one connected pixel for every pixel belonging to the same segment if there are more than one pixels in the segment.

4.2 Types of segmentation algorithms

The different types of segmentation algorithms can be broadly grouped in to the following cases:

1. *Clustering* algorithms first identify a predefined number of clusters in the feature space with the individual bands of the image representing the axes by using the unsupervised clustering algorithms such as K-means, Kohonen self organizing map, expectation maximization algorithm, etc. See (Canty, 2007, Richards and Jia, 1999, Schowengerdt, 2007) for a description of the unsupervised clustering algorithms. After clustering, the connected components of pixels belonging to the same cluster are the image segments. This method needs to deal with all the pixels for clustering and obviously is computationally expensive in terms of time and memory management.

2. *Edge-based* algorithms first try to identify the edges and then try to link the edges to create close regions or segments. However, the problems arise in edge detection due to noise effects, breaks in the edges due to non-uniform illumination and other effects. Also, edge identification in images with a very high number of channels is a complicated procedure. The edge linking procedures are also complicated and often slow (Gonzalez and Woods, 2002). So, this algorithms are mostly ruled out.

3. *Region growing* methods are based on growing the random seeds spread over the entire image based on pre-defined criterion. These methods are classified as bottom-up segmentation algorithms where you start with pixels and end up with segments

which are groups of connected pixels. The seeds mark the objects to be segmented. The criteria for growing the region defines the growth of the region. For instance, the difference between the pixel intensity and the mean of the region is used to decide if the pixel can be allocated to that region or not. The pixel is allocated to the region with which it has the shortest distance. Using an another criteria, the region around the seed is thresholded based on a condition that the difference of the pixel intensities to that of the seed pixel should fall in the pre-defined range. The connected region consisting of the seed is the grown segment. Several algorithms are based on the region growing method where, different types of growing criterion are employed. These methods are in a way highly dependent on the input seeds. The region-growing algorithms result in the so called *hierarchical segmentation* where the image segments can represented at different levels as shown in Fig. 4.1 starting with the pixels and ending up with the objects at the highest level. Different levels indicate different stages of the growth of the objects.

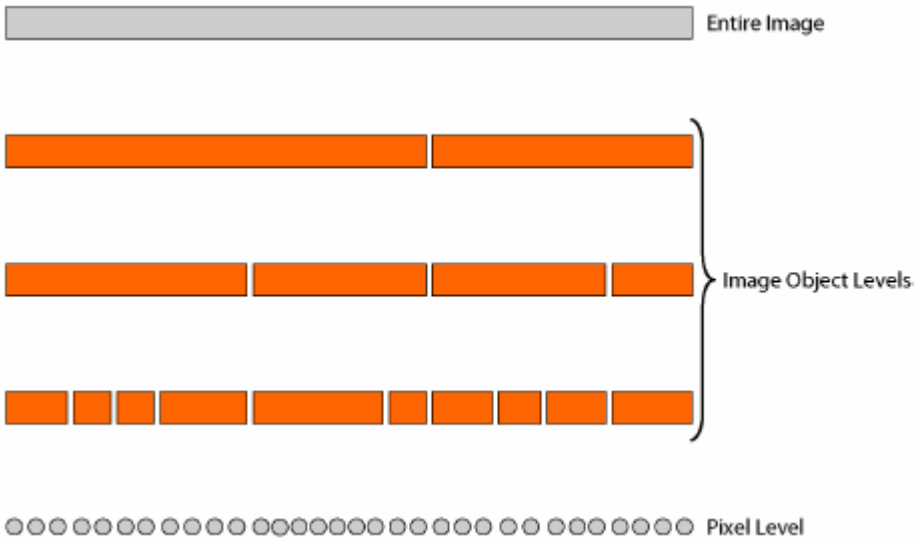


Figure 4.1: Region growing algorithms start with the pixels and create an hierarchy of image object levels based on the level of homogeneity in the image objects (Definiens User Guide, 2007).

4. *Region splitting* algorithms are classified as the top-down segmentation algorithms. They start with the entire image and iteratively split the individual regions into regions of pre-defined shape (mostly square or rectangle). These regions are checked for the homogeneity condition and if the condition is not satisfied, the regions are split further into smaller segments. The final segmentation is achieved when all the segments satisfy the homogeneity condition. This induces wrong segmentation at the edges of the split regions. To avoid this a merging procedure is used to iteratively merge the neighboring objects till the homogeneity condition is valid.

5. *Graph-based* algorithms employ the graph theoretic approaches for segmentation.

The image is represented as a graph where each pixel is a node in the graph and an edge is formed in between the pairs of pixels with a weight defined by the *measure of similarity*. The image is partitioned in to sub-graphs by removing the the edges connecting the pixels which are not similar. Several algorithms have been developed based on this concept (Shi and Malik, 1997, Felzenszwalb and Huttenlocher, 2004). A new algorithm is presented in this thesis which uses the principles of graph theory and region growing algorithms to segment the images.

Apart from the above mentioned types of algorithms there also exist several other types of algorithms such as the ones based on Markov chains (Tu and Zhu, 2002), simulated annealing (Cook et al., 1996), watershed segmentation (Beucher and Meyer, 1993), etc. In this work only multi-resolution segmentation algorithm which is a region growing algorithm available in the Definiens software is used. There are several other segmentation algorithms for a variety of purposes available in the Definiens software environment (Definiens, 2007). Apart from it, a new segmentation algorithm has been developed as a solution to a few general problems of segmentation of remote sensing images (see Sec. 4.4). It was not possible to use the proposed algorithm during the thesis as the software suitable for images larger than 400 $\times$ 400 pixels has been implemented only towards the end of the thesis.

4.3 Multi-resolution Segmentation (MRS)

The segmented image objects are homogeneous regions in the image which can be assigned to be a part of a particular class. The size of these objects depends on the scale the image is viewed at.

4.3.1 Criterion for segmentation

The criterion for segmenting the image into objects is based on defining *heterogeneity* of objects. The average heterogeneity of pixels should be minimized inside an object. As the result, the initial segmentation generates image object primitives. Heterogeneity is the opposite of homogeneity and both the terms are used to describe the object depending on the situation.

Three criterion for calculating the heterogeneity are defined.

1. *Spectral heterogeneity*

$$h_c = \sum_c w_c \sigma_c, \tag{4.6}$$

where σ_c is the standard deviation of the object in the layer c and w_c are the weight assigned to the layer.

2. *Spatial heterogeneity 1*

$$h_{s1} = \frac{l}{\sqrt{n}}, \tag{4.7}$$

n is the number of pixels in the object and l the defacto border length of the object. This criterion represents the smoothness of the object.

3. *Spatial heterogeneity 2*

$$h_{s2} = \frac{l}{b}, \tag{4.8}$$

b is the shortest possible length of the bounding box. This criterion represents the compactness of the object.

Starting with unity scale objects i.e., pixels, the objects are iteratively merged till a termination condition is reached. A fusion value is calculated everytime a decision has to be made to combine two objects. This fusion value is compared to the threshold, which is the square of the scale parameter. *The scale parameter is an abstract term which determines the maximum allowed heterogeneity for the resulting image objects* (Definiens, 2007). By modifying the scale parameter we can vary the size of image objects. If the threshold is reached the algorithm is terminated. As the objects are merged,the information of merging is stored hence forming a *hierarchical network of objects*. The overall fusion value,f when merging two objects is:

$$f = w_{color} h_{color} + (1 - w_{color}) h_{shape}, \tag{4.9}$$

w is the weight assigned to color heterogeneity. The terms involved in Eq. 4.9 are defined as

$$h_{color} = \sum_c w_c (n_{merge} \sigma_c^{merge} - (n_{obj1} \sigma_c^{obj1} + n_{obj2} \sigma_c^{obj2})), \tag{4.10}$$

$$h_{shape} = w_{cmpt} h_{cmpt} + (1 - w_{cmpt}) h_{smooth}, \tag{4.11}$$

where,

$$h_{smooth} = n_{merge} \frac{l_{merge}}{b_{merge}} - (n_{obj1} \frac{l_{obj1}}{b_{obj1}} + n_{obj2} \frac{l_{obj2}}{b_{obj2}}), \tag{4.12}$$

$$h_{cmpt} = n_{merge} \frac{l_{merge}}{\sqrt{n_{merge}}} - (n_{obj1} \frac{l_{obj1}}{\sqrt{n_{obj1}}} + n_{obj2} \frac{l_{obj2}}{\sqrt{n_{obj2}}}). \tag{4.13}$$

An example result of multi-resolution segmentation algorithm is shown in Fig. 4.6.

The problem of which object pairs to be merged first is solved in a lot of ways. Starting from any two neighboring objects, A and B , the following decision heuristics can be distinguished (Baatz, 2000, Nussbaum and Menz, 2008):

1. *Fitting*: Fuse A with any neighboring object B that fulls the homogeneity criteria;

2. *Best Fitting*: Fuse A with the neighboring object B that best fulls the homogeneity criteria, i.e. that minimizes the change of homogeneity

3. *Local Mutual Best Fitting*: Find a neighboring object B for A that best fulls the homogeneity criteria (best tting). Now nd the neighboring object C for B for which B best fulls the homogeneity criteria. If C is A, fuse the objects, otherwise repeat the procedure with respect to B for A and C for B ;

4. *Global Mutual Best Fitting*: Fuse the pair of neighboring objects that best fulls the homogeneity criteria in the entire scene.

In multi-resolution segmentation local mutual best fitting is used. However, it is easily understood that the global mutual best fitting is the best possible solution. But, to keep a track of the relations of all the objects in the entire image turns out to be a computationally expensive procedure in terms of memory and speed and hence the choice of local mutual best fit as a compromise. A new method based on the optimum graph is formulated in this thesis to support the global mutual best fitting scheme without any huge burden on the computational ability (see Sec. 4.4)

4.4 A new segmentation algorithm

There already exists a lot of segmentation algorithms which are based on the graph theoretic approaches (Wu and leahy, 1993; Shi and Malik, 1997; Felzenszwalb and Huttenlocher, 2004). Almost all of them try to create disconnected sub-graphs (or sub-trees) in different ways and these sub-graphs (or sub-trees) represent the object. A new segmentation algorithm is developed in this thesis based on the ideas from graph theory. This algorithm is in fact a fusion of graph theoretic approach and region growing approach. The graph is used to guide the merging process. Two versions of the same algorithm will be presented here. The second version is developed to overcome some of the shortcomings of the first version. The algorithm is based on building a graph over the image connecting all the objects (Normally, a minimum spanning tree is used. The graph here is built in the similar manner as the minimum spanning tree but the constraint that there should not be any closed loops is removed). The objects are then merged such that the best object pair based on the weights of the edges in the connected graph is merged first. This solves the global mutual best fitting problem. The *Standard deviation to Mean Ratio* (SMR) is used as the homogeneity criterion while merging the objects. Higher value of this ratio will yield bigger objects and vice-versa.

The segmentation algorithm is inspired from the clustering algorithm explained in (Duda and Hart, 2001) based on minimum spanning tree where natural clusters are yielded as shown in Fig. 4.2. A minimal spanning tree first connects all the nodes. Then the inconsistent edges which have the length more than that of the average length incident upon a node are removed, thus producing natural clusters. This example also illustrates how edge distances of different lengths co-exist. This is also the case in remote sensing data where the standard deviation in an object of interest often depends on the pixel intensity.

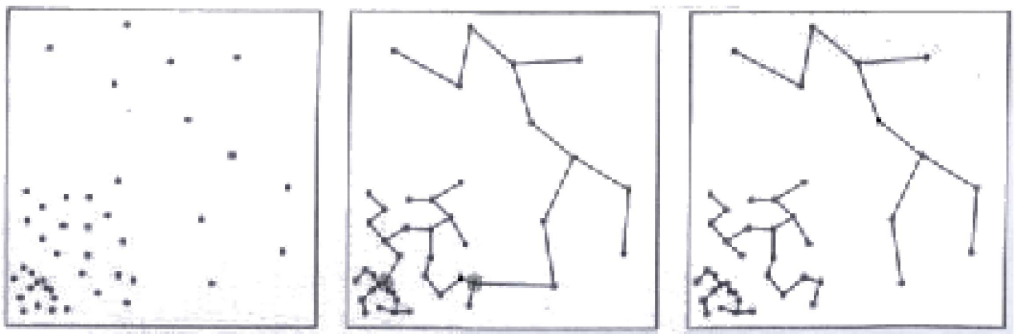


Figure 4.2: Natural clusters are yielded when the inconsistent edges are discarded in a minimal spanning tree. Image taken from (Duda and Hart, 2001)

For example, buildings with bright roofs have a larger standard deviation compared to a lake or some other dark region in the image. However, to build a minimum spanning tree over an image is an expensive procedure in terms of memory and processing time. Moreover, the same clustering algorithm when applied on an image will only yield very small objects because of the influence of the noise and the relatively low spatial correlation of the pixel intensities due to the factors mentioned in chapter 1. But the algorithm described in Fig. 4.2 very easily leads to the idea that the connectivity of the pixels and objects in the subsequent levels using suitable graphs will aid in the segmentation of the image.

The advantage of using graph theoretic approaches in image segmentation is that the complete graph structure is already known. It is just the pixels as nodes connected in a well-structured rectangular grid. If the diagonally connected pixels are not considered as neighbors, every node, i.e., pixel is connected to 4 neighboring nodes or else 8 neighboring nodes in the case of 8-neighborhood. A 4-neighborhood is considered in this thesis. So, the number of comparisons to be made to build any sub-graph is always 4 for every pixel.

4.4.1 The first algorithm

This algorithm first groups the pixels in to objects based on disconnecting the edges in the graph and then iteratively merges the objects based on the homogeneity criterion. The pixels are treated as objects of unit size. The segmentation algorithm tries to build a graph structure over the image connecting all the objects in two stages. The terms *node* and *object* are used to represent the image object. The term *node* is used while referring to the graph and *object* while referring to the segmentation level. In the first stage, the aim is to identify those regions which are as homogeneous as possible. The goal is to achieve something over the image which is analogous to the result in Fig. 4.2 but without actually creating the complete sub-graph over the entire image. This is achieved as follows:

1. Starting with the pixel at the top left of the image, a graph is constructed by

connecting every node to the neighboring node having the smallest distance. The distance is then the weight of the edge between the nodes. The distance between the objects is calculated using the simple Euclidean distance in the feature space of the image described as follows:

$$d_{ij} = \sum_{k=1}^{K} (\mu_i^k - \mu_j^k)^2 \tag{4.14}$$

where, K is the number of image layers used for segmentation. μ_i^k and μ_j^k are the means of the objects i and j respectively in k^{th} image layer.

2. If the edge connecting the current node the closest node is already in the graph, continue with the next closest node and so on.

3. When no more connections are possible for a node, check which of the edges incident on the node are inconsistent and disconnect them. This is done by taking an average weight of all the edges and checking which edges have weights greater than α times the average weight. The multiplication factor α is considered to account for the case where there are only two edges incident on the node. The edges which are inconsistent are disconnected and all the connecting nodes are grouped together to form a new object. The mean and the standard deviation are the representative values of the objects in the algorithm. So, every time the objects are grouped, the mean and standard deviation have to be updated as shown in Eq. 4.15 and Eq. 4.16. The expressions for the update of the mean and standard deviation can be easily derived using the basic statistics definitions of mean and standard derivation for all the p image layers used for segmentation.

$$\mu_{12}^i = \frac{n_1\mu_1^i + n_2\mu_2^i}{n_1 + n_2} \qquad i = 1, 2, ..p \tag{4.15}$$

$$\sigma_{12}^i = \sqrt{\frac{1}{n_1 + n_2 - 1}\left[(n_1 - 1)(\sigma_1^i)^2 + (n_2 - 1)(\sigma_2^i)^2 + \frac{n_1 n_2}{n_1 + n_2}(\mu_1^i - \mu_2^i)^2\right]} \tag{4.16}$$

where n_1, n_2 are the number of pixels in the objects to be merged, μ_1^i, μ_2^i are the means and σ_1^i, σ_2^i are the standard deviations of the objects in ith image layer. μ_{12}^i and σ_{12}^i are the mean and standard deviation of the merged object.

Consider the graph connecting the nodes with values as shown in Fig. 4.3. In Fig. 4.3a, it is not possible to discconnect any of the edges as there exists no inconsistent edges. However, as in Fig. 4.3b, when the nodes A and B are grouped, the new edge weight is calculated from AB to C as 7.5. If α, the multiplication factor to check for inconsistency is 1.5, the edge is consistent with respect to the node C and we merge the node C in to AB. In Fig. 4.3c, the edge to D has a value of 10. Clearly, this is inconsistent with the edge weight of 5 between D and E and is disconnected hence forming to groups {A,B,C} and {D,E}. This is the reason for creating and merging objects in between without creating the complete graph.

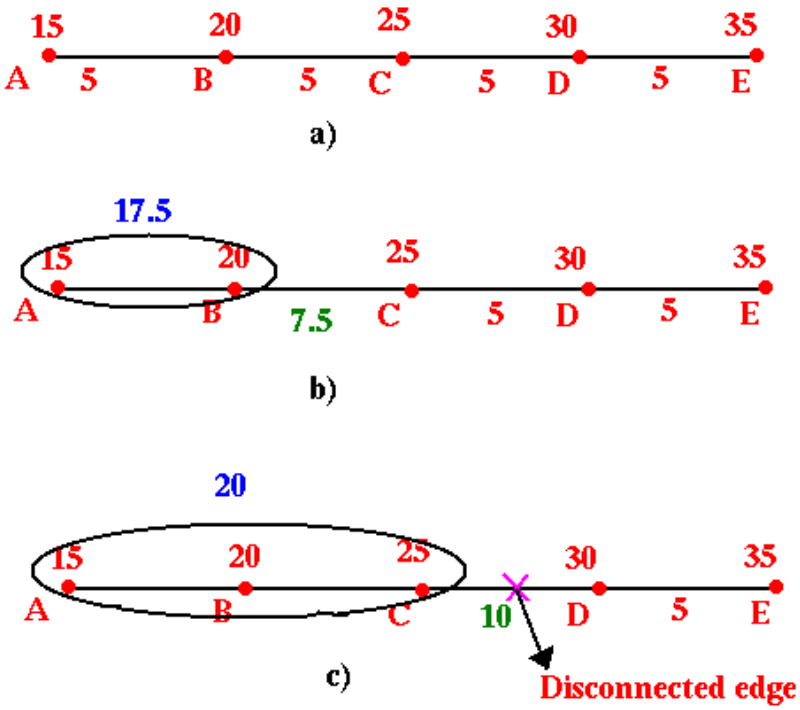


Figure 4.3: (a) A graph representation where the edges can not be disconnected based on the condition of inconsistency of the edges. (b) By merging the nodes the subsequent edge weight can be altered (c) We can now find a point where the inconsistent edge occurs.

4. The first iteration normally generates very small objects. So, the process is to be iterated a few times till sufficiently large image objects are formed. Typically, one to two iterations are used to generate sufficiently large segments to be used in the next stage. More iterations will lead to the problem where the initial segments do not satisfy the homogeneity criterion anymore.

The second stage merges the initial segments formed in the first stage in to final image objects which satisfy the homogeneity condition in an iterative process. Unlike the first stage, the complete graph has to be built first connecting all the nodes and then, merging decisions are made based on the homogeneity condition. That is the reason why iterations are performed in the first stage to have initial segments of sufficiently large size which decreases the number of nodes to be dealt with in the second stage. The merging of initial segments is done as follows:

1. Consider the initial segments from the first stage as the nodes to construct a graph.
2. By selecting the nodes in any fashion (randomly/ pre-defined order) connect it to the closest node with an edge of minimum weight. The weight is calculated as the distance between the nodes as defined in Eq. 4.14.

3. If the edge is already a part of the graph, connect with the next closest node and so on. The creation of the graph is illustrated in Fig. 4.4 where the points in a two dimensional space are connected using the edges and a graph is created. The minimum spanning tree of the complete graph is a subset of the graph shown here. This graph is created in the same lines as the creation of the minimum spanning tree. However, the condition of closed loops is ignored. So, the graph consists of closed loops whereas a minimum spanning tree will not.

4. When all the nodes are connected, sort the edges in the ascending order of the weight.

5. Starting with the smallest edge, merge the objects if they satisfy the homogeneity condition of the standard deviation to mean ratio (SMR),

$$SMR^i = \frac{\sigma^i_{merge}}{\mu^i_{merge}} \leq T, \qquad i = 1, 2, ..p, \tag{4.17}$$

   where $\sigma^i_{merge}, \mu^i_{merge}$ are the mean and standard deviation of the object formed by merging any two objects in the ith image layer. T is the threshold which specifies the homogeneity of the objects. Note that the condition should be satisfied in all the p image layers.

6. When two objects are merged all the edges connecting the corresponding node are updated with the weights calculated with respect to the merged object. Correspondingly, the sorted order of the edges might change because of the updated edges.

7. Check for the closest node to the newly formed object among the neighbors. If there is no edge connecting the closest node create a new edge and insert it in the sorted list.

8. This process continues until the edges do not change anymore. At that stage, the final segmentation is achieved. An example of the segmentation using this algorithm is shown in Fig. 4.6.

4.4.2 The modified algorithm

There are several drawbacks of the first version of the algorithm. Though the formation of initial segments using the concept of breaking the edges in the first stage is a very effective approach, it is computationally expensive. Moreover, the use of the Euclidean distance in the feature space as shown in Eq. 4.14 is not well suited in the case of having different types of data layers for segmentation. For instance, NDVI has a range in between -1 and 1 and the values of NDVI for vegetation has a very high range relative to the other classes. A normalization of the range of NDVI to suit the other layers may not be the correct idea sometimes as it only increases the variance in the vegetation objects. This problems are accounted for in the modified version of the algorithm.

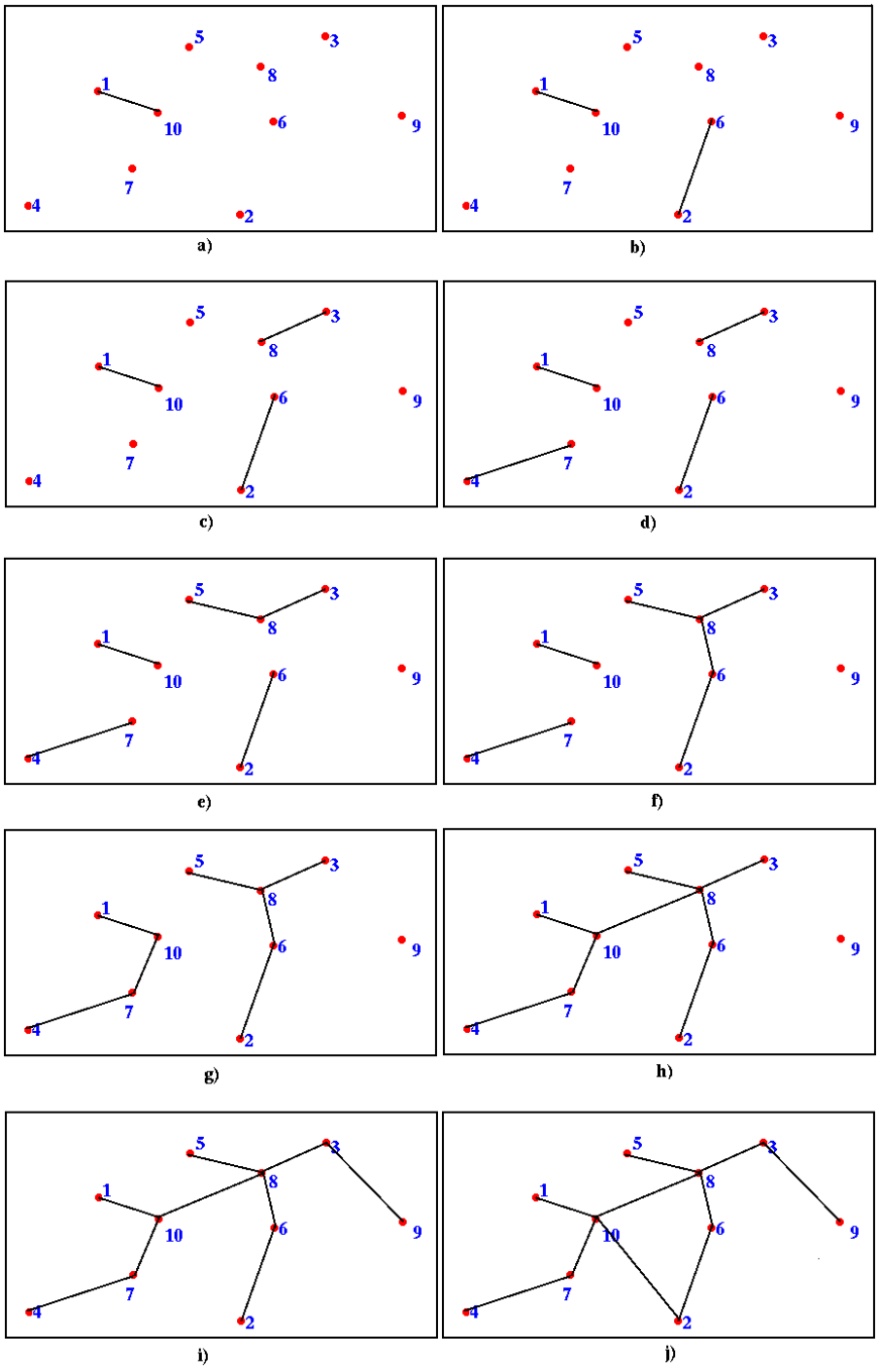


Figure 4.4: Creating a graph representation where all the points are for sure connected to the closest nodes. The numbers indicate the order in which the nodes are connected to their nearest neighbors. If the edge to the nearest neighbor already exists, the next closest node is connected with a new edge.

The modified algorithm also works in two stages. In the first stage initial segments are created in a different way to that of the first version. These initial segments are then merged in a similar way by creating the graph. However, the weights of the edges are not based on Eq. 4.14 but on the standard deviation to mean ratio. The initial segments are created in the first stage by growing the seeds. The seeds are not randomly selected like in the case of multi-resolution segmentation but are selected based on the information of edges (or edge intensity).

It is more convenient not to have seeds at the edges as they can lead to wrong growth of the segment. This the reason why the edge information is considered to avoid having the seeds close to the edges. Again, pixels inside brighter objects have bigger differences compared to that of the pixels in the darker objects. For this reason, the data is first normalized using a sigmoid function as given in Eq. 4.18 as:

$$f(I) = \frac{1}{1 + e^{(-\alpha_1 I)}} \tag{4.18}$$

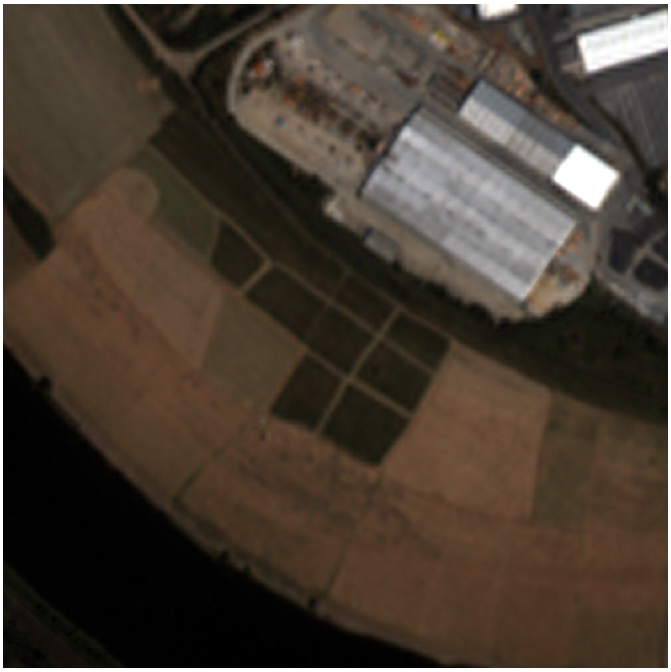
where I is the image and α_1 is the multiplication factor to change the shape of the sigmoid function depending on the range of the image data. Typically, α_1 is chosen such that the transformed range of the image is between 0 and 5 because the value of the sigmoid function is close to 1 around the input value of 5. This transformation decreases the relative standard deviation of the brighter regions in comparison with the darker regions.

A morphological gradient using a 3×3 square structuring element is then calculated to describe the edges as shown in Fig. 4.5. This process defines the edges very well compared to directly calculating the gradient. The maximum value of the gradient in all the bands at a pixel then represents the edge intensity at that pixel. After finding the edge intensity, the inital segments are formed as follows:

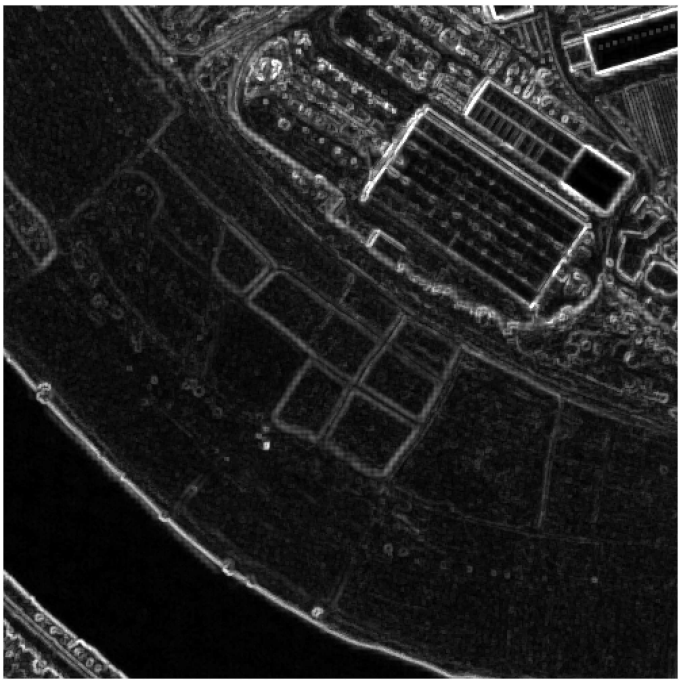
1. We start with the pixel having the lowest edge intensity value. The lowest value here represents that a pixel is in the most homogeneous neighborhood i.e., in the interior of an object. The seed is grown in a small neighborhood (for instance 51×51 pixels around the pixel) such that all the connecting pixels should be in a range of $p_i * (1 \pm s_t)$, where p_i is the value of the seed and s_t is the first parameter of the algorithm which defines how much difference is allowed with respect to the value of the pixel. So, the first step is to grow the seed by connecting all those pixels satisfying the condition in all the image layers.

2. This is repeated till all the pixels belong to any of the grown segments. The pixel with the smallest edge intensity in the group of pixels that do not belong to the already grown regions is identified and used as the next seed.

3. This creates the initial segments.



(a)



(b)

Figure 4.5: (a) A subset of an IKONOS scene of Saechsische Schweiz, Germany. (b) The edge intensity based on morphological gradient after the transformation using a sigmoid function. The pixels with lower edge intensity are first grown.

Once we have the initial segments then we can build a graph over these segments as it is done in the first version of the algorithm. A final segmentation is achieved in the second stage as follows:

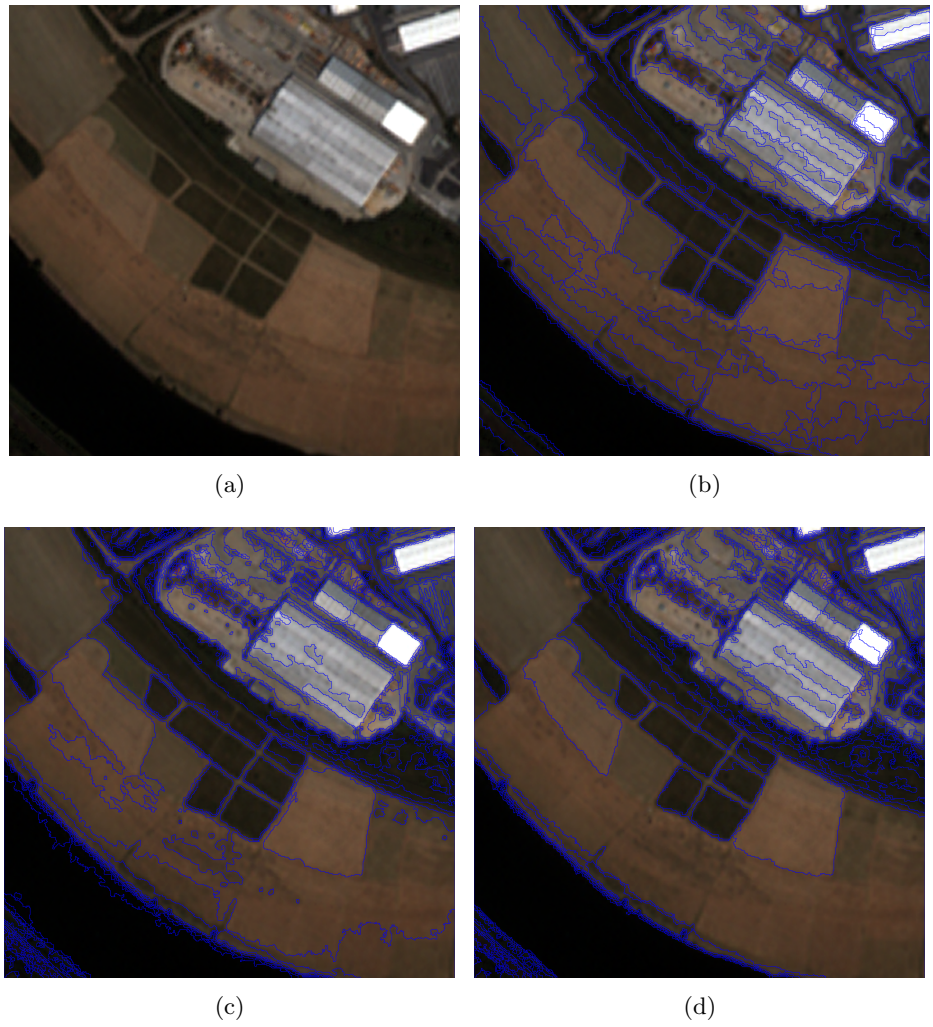
1. Consider the initial segments from the first stage as the nodes to construct a graph.
2. Unlike the earlier version of the algorithm, the edge weights are now calculated based on the standard deviation to mean ratio and not the distance measure. This adaptation allows us to use different image layers without normalizing the original data. The edge weight between two nodes is defined as

$$w_{mn} = max(\frac{\sigma^i_{mn}}{\mu^i_{mn}}) \qquad i = 1, 2, ..p, \tag{4.19}$$

   where μ^i_{mn} and σ^i_{mn} are the mean and standard deviation of the object formed by objects defined by nodes m, n in ith image layer. Note that the maximum of the mean to standard deviation ratio in all the p image layers is considered.
3. By selecting the nodes in any fashion (randomly/pre-defined order) connect it to the closest node with an edge of minimum weight as described in Fig. 4.4.
4. If the edge is already a part of the graph, connect with the next closest node and so on.
5. When all the nodes are connected, sort the edges in the ascending order of the weight.
6. Starting with the smallest edge, merge the objects if they satisfy the homogeneity condition.

$$w_{mn} \leq T, \tag{4.20}$$

   where, T is the threshold which specifies the homogeneity of the objects.
7. When two objects are merged all the edges connecting the corresponding node are updated with the weights calculated with respect to the merged object. Correspondingly, the sorted order of the edges might change because of the updated edges.
8. Check for the closest node to the newly formed object among the neighbors. If there is no edge connecting the closest node create a new edge and insert it in the sorted list.
9. This process continues until the edges do not change anymore. At that stage, the final segmentation is achieved. An example of the segmentation using this algorithm is shown in Fig. 4.6.



(a) (b)

(c) (d)

Figure 4.6: (a) The subset of IKONOS image of Saechsische Schweiz, Germany (b)The segmentation result of multi-resolution segmentation (scale parameter = 0.75, shape parameter = 0.1, compactness= 0.5) (c) Segmentation with he first version of the proposed algorithm ($\alpha = 1.2$, $T = 0.09$) (d) Segmentation with the modified version of the algorithm ($s_t = 0.035$, $T = 0.09$)

4.5 Evaluation of segmentations

Several segmentation algorithms are developed for a variety of purposes. It is therefore necessary to have an evaluation measure to decide which algorithm can be better for a particular task. For instance, the algorithms proposed in Sec. 4.4 are most suited for homogeneous regions and not textured regions. Like segmentation itself, there is no standard way of evaluating the segmentation results. There are always two cases of segmentation

1. *Over-segmentation*, where a real world object is segmented in to sub-objects.
2. *Under-segmentation*, where a real world object is a sub-object of the segmented object.

There is a chance that the real world objects can be reconstructed if there is the problem of over-segmentation but when there is under-segmentation, the real world object can not be recovered. So, a segmentation algorithm which produces over-segmentation rather than under-segmentation is desired. But then the sub-objects should be of reasonable size relative to the object of interest.

Two broad classifications of the methods are the *unsupervised* and *supervised* methods (Zhang et al., 2005). Unsupervised methods try to evaluate the quality of all the segments based on empirical criterion for quantifying homogeneity such as *entropy* (Zhang et al., 2004). Supervised methods try to evaluate the segmentations based on reference segments by comparing the shape and homogeneity of the objects. A summary of the existing evaluation methods is presented in (Neubert et al., 2008). In (Neubert et al., 2008), they used supervised evaluation combined with visual quality assessment to evaluate a variety of the existing segmentation algorithms for remote sensing data. The following criterion are used to evaluate the segmentations:

1. Average difference of the area [%]: The percentage of the difference of area between the reference object and the corresponding objects inside the reference object averaged over all the reference objects.
2. Average difference of perimeter [%]: The percentage of the difference of the perimeter between the reference object and the combination of segmented objects inside the reference object averaged over all the reference objects.
3. Average difference of shape index [%]: The difference of the shape index of the reference object and the combination of the segmented objects inside the reference object averaged over all the reference objects. The shape index is defined as

$$Shape\ Index = \frac{P}{4\sqrt{A}} \tag{4.21}$$

where P is the perimeter and A, area of the object.

4. Average number of partial segments: The number of partial segments inside a reference object averaged over all the reference objects. This value indicates the amount of over-segmentation.

5. Finally, an index visually rated between 0 and 2 (0 corresponds to poor and 2 corresponds to perfect segmentation) gives an idea of the quality of the overall segmentation.

A comparison of the quality of different segmentation algorithms is presented based on the above measures in (Neubert et al., 2008). However, the fact that only 20 reference areas were selected to quantify does not justify the results. Moreover, the average value is not a good statistical measure when a relatively smaller number of reference segments are considered.

In this thesis an attempt is made to evaluate the results of segmentation in the same lines as the method in (Neubert et al., 2008) but at a more abstract level to neatly account for all the artifacts of segmentation. The quantile values are presented as evaluation parameters instead of the average. Consider a reference image object as shown in Fig. 4.7. After segmentation, the reference object can consist of sub-objects. The sub-object which consists of maximum number of the pixels (in this case, A) gives us the information regarding the over-segmentation. Some sub-objects gather the pixels from outside the reference area. The sub-object with such mixed pixels can still be attributed to the reference object if it consists of majority of the pixels from the reference object (objects C and F). The pixels added here are termed as *extra pixels*. In the same way if the sub-objects consists of majority of the pixels from outside the reference object then they cannot be attributed to the reference object and the pixels will be lost. These pixels are termed as *lost pixels* (as in objects G,H). There should be a clear definition of what percentage of pixels are considered as majority. In this example 60% is chosen as the threshold to decide if the sub-object can be considered or not. For a strict evaluation, this value should be higher. The assumption is that even if there are 40% external pixels, the representative value of the sub-object would not change much. This cannot be true always. In that case the threshold should be higher. The final shape of the object that can be effectively reconstructed using the sub-objects is shown in Fig. 4.7b. This is a very good example and in certain cases it can get worse. The percentage of lost pixels and extra pixels gives the information regarding the under-segmentation.

The segmentation result is evaluated using the reference segments. Instead of providing an average, the quartiles (or sometimes deciles) are provided. The following measures are considered:

1. Size of the biggest sub-object [%]

2. Area of the lost pixels [%]

3. Area of the gained pixels [%]

In (Costa et al., 2008), they also use the concept of lost pixels and extra pixels and define a fitness index, F as in Eq. 4.22. It is just the average of total redundancy at the

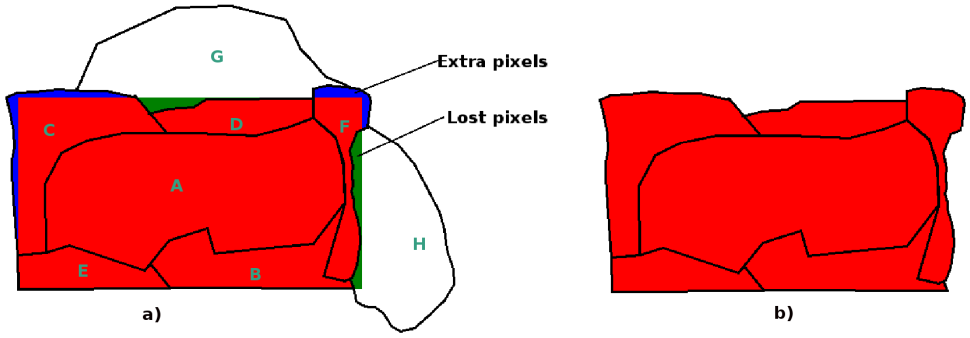


Figure 4.7: (a) An example of possible scenarios of segmentation. The blue regions indicate the extra pixels added to the object and green pixels are lost. (b) The effective shape of the reference object that can be reconstructed after segmentation.

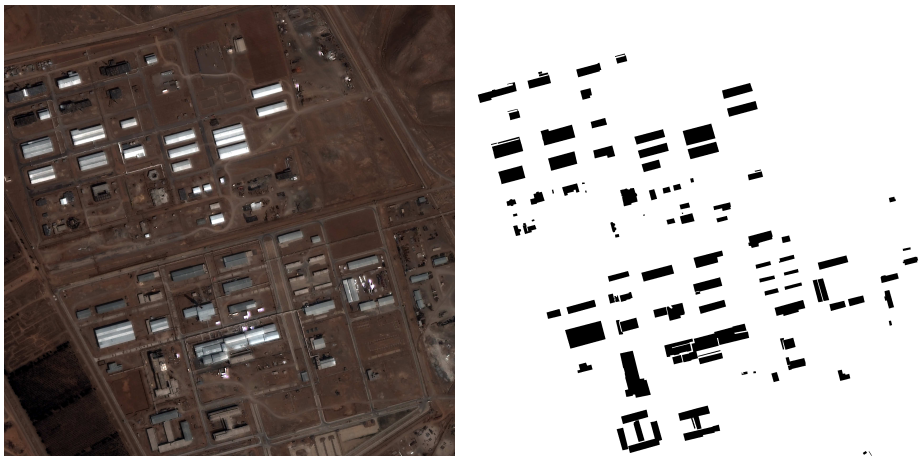
reference object. Again, the average value may not be the best criteria to evaluate the segmentation.

$$F = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{Area of lost pixels} + \text{Area of extra pixels}}{\text{Area of the reference object}}, \tag{4.22}$$

where, n is the number of reference segments.

As a demonstration, a subset of Quickbird scene over Esfahan, Iran acquired in 2004 is chosen to evaluate the multi-resolution segmentation of Definiens and the algorithm proposed in Sec. 4.4. The class of buildings is chosen as a reference class to be evaluated. In total 104 reference objects of various sizes were drawn and a mask is created as shown in Fig. 4.8.

The image is segmented using the multi-resolution segmentation and the proposed segmentation algorithm using graphs. Appropriate parameters that are identified as best parameters by visual inspection are used (For MRS, scale parameter=100, shape parameter= 0.3, compactness=0.5. For graph based algorithm, $\alpha = 1.2$ and $T = 0.1$) . The results of the evaluation are given in table: 4.5 where the quartile values of the parameters are provided. It can be seen that the graph based algorithm performs better as a lot of pixels are lost while segmenting with the multi-resolution segmentation algorithm.



(a) (b)

Figure 4.8: a)A subset of an Quickbird scene of Esfahan, Iran. b) The mask showing reference objects for the buildings class

	Q1(25%)	Q2(50%)	Q3(75%)	Q4(100%)
Area [%]				
Graph-based	34	52	77	100
MRS	30	47	87	100
Lost Pixels [%]				
Graph-based	1	3	**7**	100
MRS	2	7	**30**	100
Extra Pixels [%]				
Graph-based	2	4	9	53
MRS	1	4	11	61

Table 4.1: Evaluation of segmentation results. Graph-based algorithm performs better than the multi-resolution segmentation (MRS) algorithm.

4.6 Shape segmentation

Imagine a two roads showing similar spectral characteristics in the image crossing each other. A general characteristic to identify a single road is based on its shape that it is long and has a specific width or the sub-objects of the road have high assymetry. However, all these descriptions of the road fail at the crossing of two roads. To avoid such problems, we could turn to shape segmentation algorithms. No literature review will be presented here as the author working independently on a different problem related to counting of fission tracks in microscopic images developed this methodology. The same methodology can be used in the case of the image objects to extract individual direction components of the objects. It is based on identifying the skeleton of the object using morphological operations. Both the edge and the skeleton are segmented first to identify the individual directional components of the edge and skeleton. The directional components of the edge and the skeleton are then grouped to extract the directional components of the entire object. This topic is described here only as a direction for future research and has not been used during the thesis.This algorithm is a direct consequence of the understanding of visual perception theories of Marr and Rudolph (Sec. 2.1)

Consider the object shown in Fig. 4.9. It consists of 6 components and the aim is to extract the components. The following sequence of steps is used:

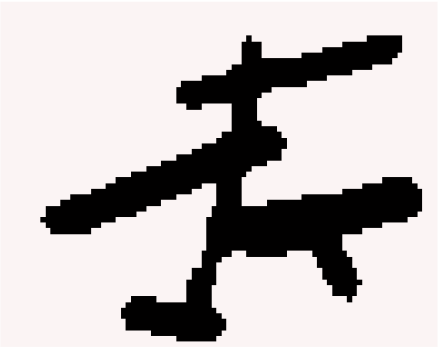


Figure 4.9: Object consisting of six directional components

1. Edge segmentation
2. Skeletonization
3. Skeleton segmentation
4. Grouping

4.6.1 Edge segmentation

The edge is first identified and refined such that the edge does not contain any pixels from the interior of the object which are exposed to the background through the diagonal pixel.

1. By using the templates shown in Fig. 4.10 the edge is then segmented into two directions ,say horizontal and vertical. The templates identify the horizontal components and the remaining are labeled as vertical.

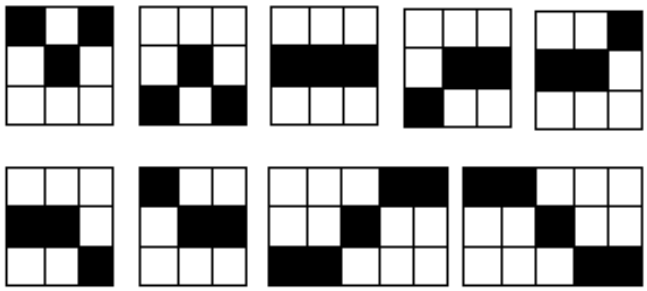


Figure 4.10: Templates to identify horizontal components

2. By following the edge, the horizontal and vertical components can be further segmented in to two more directions for each as shown in Fig. 4.11.

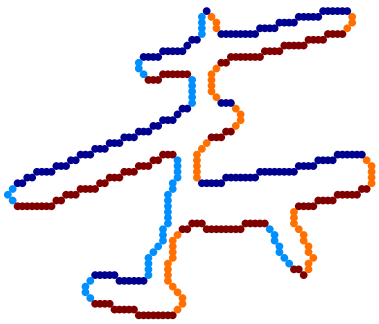


Figure 4.11: The segmentation of the edges in to four components

4.6.2 Skeletonization

Once the edge is segmented, the skeleton has to be drawn and segmented in the same lines as the edge segmentation. The skeleton is calculated as described in (Gonzalez

and Woods, 2002) in two stages. Consider the neighborhood arrangement as shown in Fig. 4.12

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Figure 4.12: Neighborhood arrangement for the skeletonization

1. A pixel p_1 is flagged for deletion if the following conditions are satisfied

$$\begin{aligned} \textbf{(a)}\ & 2 \leq N(p_1) \leq 6 \\ \textbf{(b)}\ & T(p_1) = 1 \\ \textbf{(c)}\ & p_2 \bullet p_4 \bullet p_6 = 0 \\ \textbf{(d)}\ & p_4 \bullet p_6 \bullet p_8 = 0 \end{aligned} \tag{4.23}$$

where $N(p_1)$ is the number of nonzero neighbors of p_1 and $T(p_1)$ is the number of 0-1 transitions in the ordered sequence $p_2, p_3, ...p_9, p_2$

2. In the second step only conditions **(c)** and **(d)** are changed as follows:

$$\begin{aligned} \textbf{(c)}\ & p_2 \bullet p_4 \bullet p_8 = 0 \\ \textbf{(d)}\ & p_2 \bullet p_6 \bullet p_8 = 0 \end{aligned} \tag{4.24}$$

Steps 1) and 2) are iteratively applied until no further points are deleted thus producing a skeleton of the object. Fig. 4.13 shows the skeleton of the object.

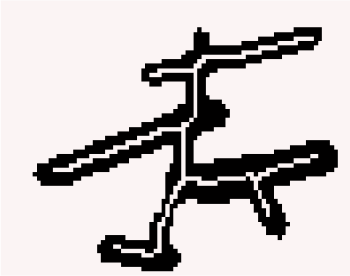


Figure 4.13: The skeleton of the object

4.6.3 Skeleton segmentation

After the skeleton is identified, it is segmented in the same way as the edges are segmented by using the templates for flagging the vertical and horizontal components. However.

since we have intersecting lines in the skeleton unlike the edge, some of the pixels at the intersection are flagged in both the directions. Such pixels are named as *joker pixels*. If there is line segment which consists of only joker pixels, it is ignored. The joker pixels are shown in Fig. 4.14 plotted in blue.

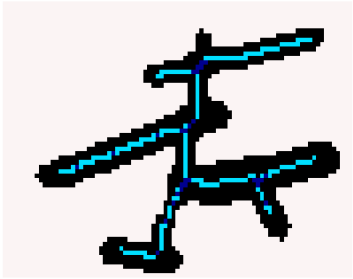
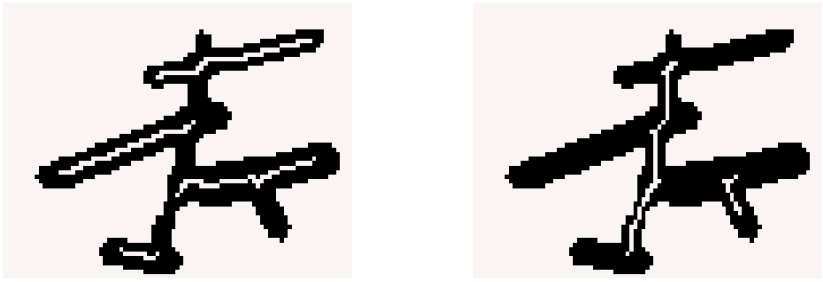


Figure 4.14: The pixels which belong to both the horizontal and vertical components are plotted in blue

We identify the horizontal and vertical components of the skeleton as shown in Fig. 4.15. It can be seen that the joker pixels that are not connected to the components are deleted to get the correct components of the skeletons. This components can now be used to group with the components of the edge to produce segmentation of the object based on the direction of the sub-objects.



(a) (b)

Figure 4.15: a) The horizontal components of the skeleton and b) vertical components.

4.6.4 Grouping

Once we have the components of skeletons and edges then the task is to group them in the best possible way. When there is a crossing, there obviously is a region shared by both the components. So for grouping purposes, this shared region is either segmented as a new region or allocated to the longest object. Fig. 4.16 shows the objects obtained

by independently grouping the skeletons and the edges in each of the directions. This components have the shared regions.

(a) (b)

Figure 4.16: a) The horizontal components of the object and b) vertical components.

4.7 Object-based change detection

There can be several other applications of segmentation apart from being used in object-based classification. One such application is the object-based change detection where the image objects are used as basic entities instead of pixels. The biggest advantage is that the salt and pepper effect of noise can be hugely reduced as the averaged values of the object are used instead of pixel values with higher variance. Moreover, we can also use a lot of features based on context and texture apart from the layer values of the image objects for change detection. An example is presented here to compare the results of MAD algorithm described in Sec. 3.6 when applied on pixels and objects. Fig. 4.18 shows the comparison of the MAD components using objects and pixels as basic entities respectively of the images shown in Fig. 4.17. The images are acquired using the QuickBird satellite over Esfahan, Iran in 2002 and 2003 respectively (see Sec. 6.5.1) for details of the images). It can be observed that the MAD transformation using objects is much better than transformation using pixels. Furthermore, using objects we can use more features to calculate the MAD components. As an example, the texture measure using gray-level co-occurence matrix (GLCM) mean (see Sec. 5.3.1) is used to calculate the MAD components of the two images. The MAD components based on texture measure are shown in Fig. 4.19.

(a) (b)

Figure 4.17: The subsets of QuickBird scenes from Esfahan, Iran from (a) 2003 and (b) 2004

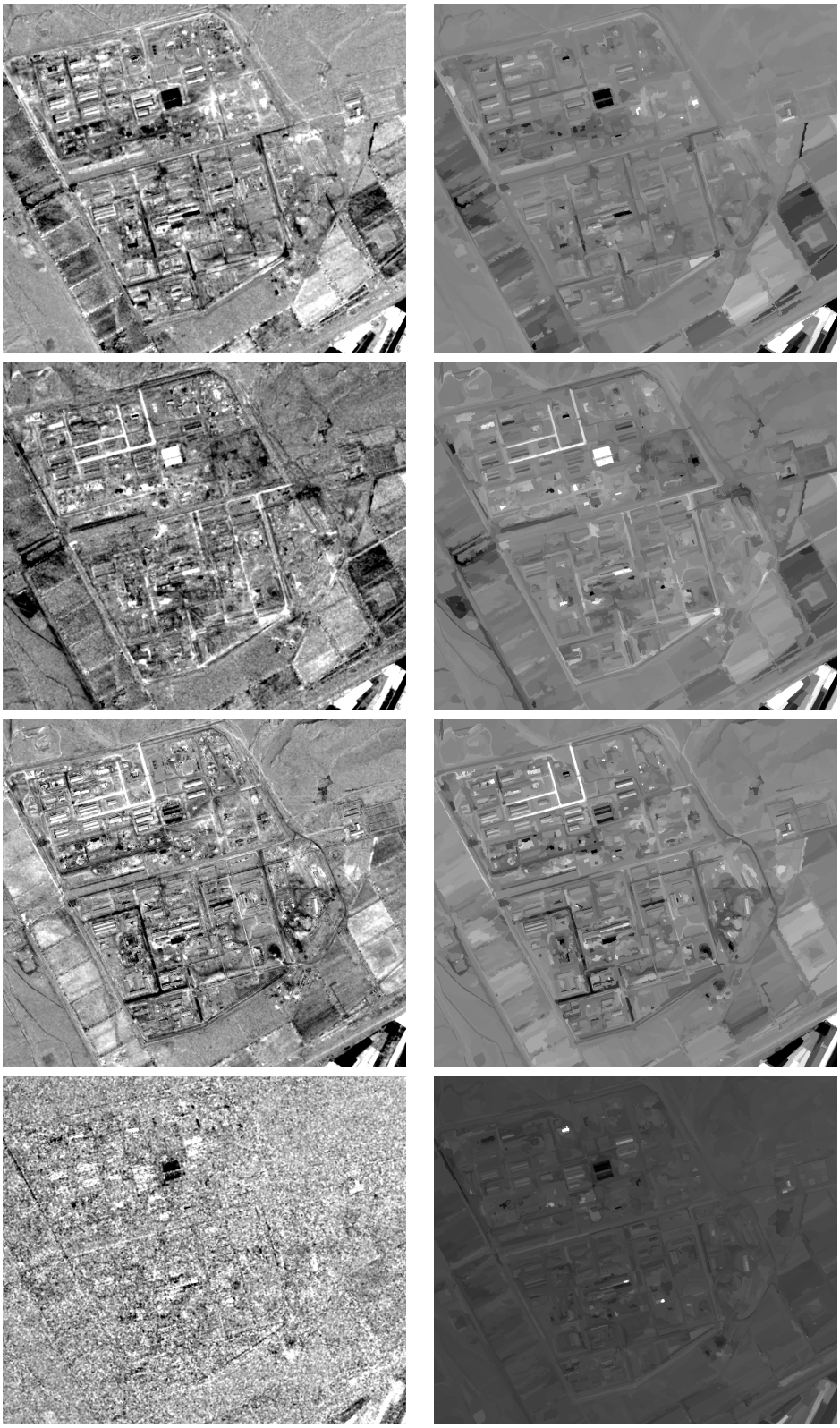
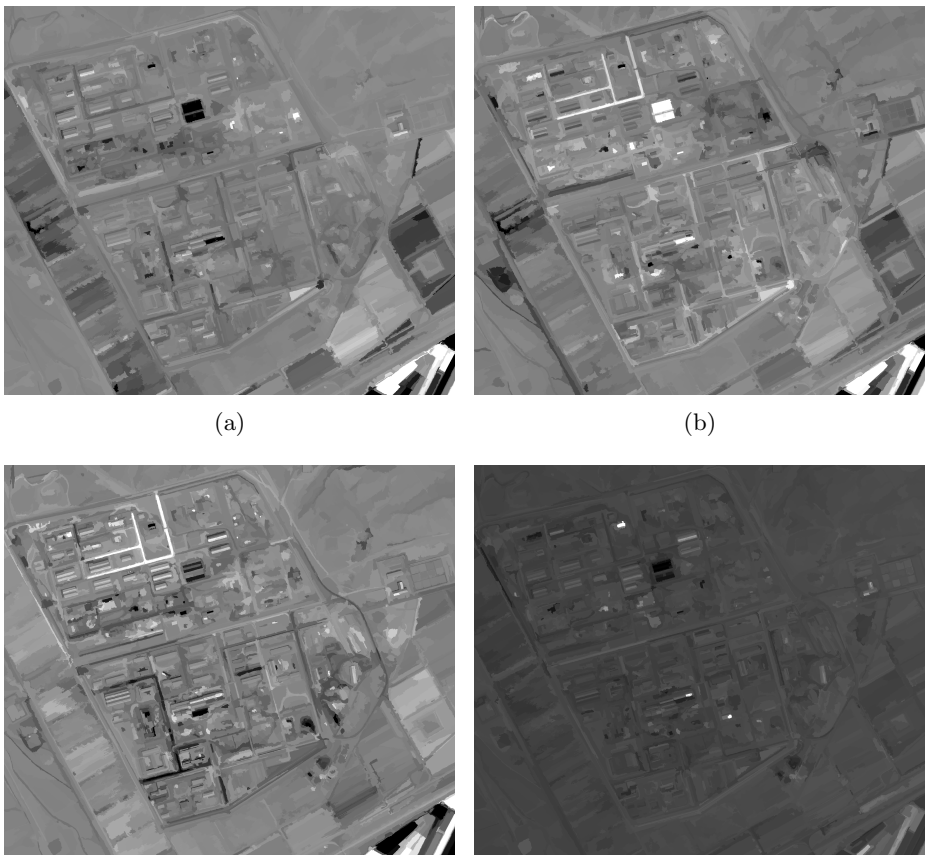


Figure 4.18: MAD components using pixels in the left side and objects in the right side



(a) (b)

(c) (d)

Figure 4.19: MAD components using GLCM Mean (a)MAD-4 (b)MAD-3 (c)MAD-2 (d)MAD-1

Chapter 5

Object features

The strength of OBIA is the fact that the image objects provide us with an improved basis for classification as we can use a lot of information apart from the spectral characteristics of the region. Every image object can be described by a variety of features. In this chapter an overview of some types of features that can be used to define the image objects is given. *Feature* is a term which is also often used to define a pattern in image processing. In this thesis it is strictly avoided to use the term *feature* for pattern. Here, a feature is an attribute representing the information of the image object. Several thousand types of features can be used to describe the objects. Only a brief overview of the object features is presented here. Features can be based on the following types of information:

1. Layer value
2. Shape
3. Texture
4. Context

5.1 Layer value features

The Layer value features of an object are calculated based on the layer values of the pixels inside the object. Some examples of the layer value features are:

1. *Mean values* of all the pixels inside the object in every level. The mean value is often considered as a representative layer value of the object.
2. *Standard deviation* of the values of the pixels inside the object.
3. *Minimum and Maximum* of the pixel values inside the object.

4. *Quantiles* are often used to represent the distribution of the data. For instance the *median* value can be a better representative layer value of the object when the image objects are affected by under-segmentation problem. The mean value is corrupted because of the layers values of pixels which do not originally belong to the object. In such cases using quantile values is a best option.

5. *Brightness* of an image object can be calculated in two ways either as the sum of the representative layer values under consideration or as the sum of the squares of the representative layer values.

6. *Maximum difference* between the layers under consideration.

7. *Ratios* are the rational functions of the combinations of ratios. Several ratios are possible by combining different layer values. the best combinations of layer values to calculate different types of ratios which can distinguish the classes can also be identified by means of samples of the classes as explained in Sec. 3.4.1, where the separability of the classes is calculated for all the possible combinations of ratios and the best ratios which have high separability are considered.

8. *Relations to the entire scene* based on the statistics of the entire layer such as the difference of the layer mean value of the object of interest to the mean value of all the pixels in the object. Similarly, the ratio between the object value and the scene value can also be calculated.

9. *Higher order statistical features* such as moments, skew, kurtosis can be calculated by considering the values of the pixels in the object as a distribution.

10. Moreover, several customized features can be calculated by user-defined arithmetic expressions based on the above features.

5.2 Shape features

The shape features are one of the most important features which make OBIA more advantageous compared to the pixel based measures. A lot of shape descriptors for objects are detailed in the literature (for e.g., Gonzalez and Woods, 2002). Some of the most commonly used shape features are given here.

1. *Area* is calculated based on the total number of pixels in the object.

2. *Length* and *width* of the objects can be calculated in a lot of ways. One simple way is based on the skeleton of the which calculated using the method explained in Sec. 4.6. The length is the sum of the length of the longest line of the skeleton of the object and the width of the object. The width of the object is twice the mean or median of the length of the pixels of the edge to the skeleton as shown in Fig. 5.1. The length and width can also be calculated based on the bounding box of the object where the longest edge represents the length. As in Definiens Developer

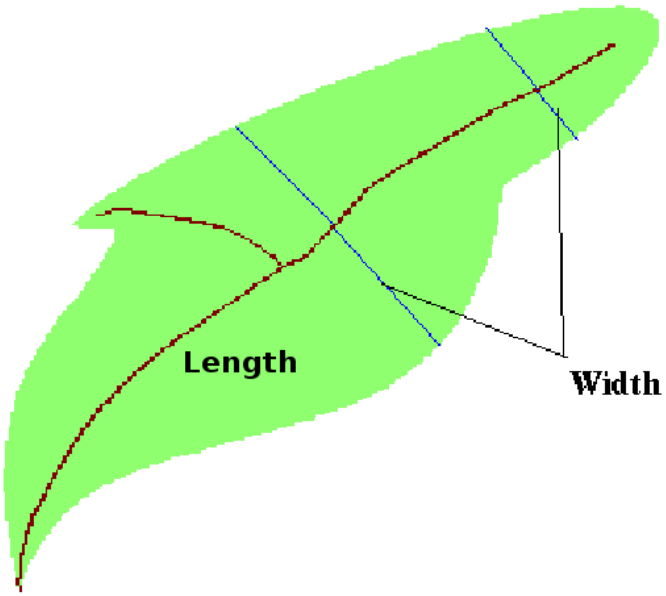


Figure 5.1: Length and width of the object based on skeleton

software (Definiens, 2007), the length and width can also be calculated based on the covariance matrix of the pixel coordinates. The ratio of the eigenvalues of the covariance matrix is equal to the length to width ratio.

$$\frac{length}{width} = \gamma = \frac{\lambda_1}{\lambda_2}, \tag{5.1}$$

$$length, l = \sqrt{N_p * \gamma}, \tag{5.2}$$

$$width, w = \frac{N_p}{\gamma}, \tag{5.3}$$

where, λ_1, λ_2 are the eigenvalues of the covariance matrix of the pixel coordinates, N_p is the number of pixels in the object. In Definiens Developer software, the length, width and γ are calculated in both ways using the bounding box and the covariance matrix and the smallest values are used. The length and width calculated based on the bounding box or covariance matrix is more convenient for compact objects. For curivilinear objects the skeleton gives more realistic values.

3. *Perimeter* is just the number of pixels on the edge.

4. *Asymmetry* which defines how symmetric the object is along the principal axis is calculated as (Definiens, 2007):

$$\frac{2\sqrt{(var(X) + var(Y))^2 + (var(XY)^2 - var(X) * var(y)}}{var(X) + var(Y)}, \tag{5.4}$$

where, X, Y are the set variables representing the coordinates of pixels.

5. *Border index* is the ratio of the perimeter of the object to the perimeter of the smallest enclosing rectangle (Definiens, 2007).

6. *Compactness* is calculated as the ratio of the product of length and width to the number of pixels in the object (Definiens, 2007).

7. *Density* of the object defines relation between the arrangement of the pixels in the object when compared to a square. The higher the density of the object the more it represents a square. It is calculated as follows (Definiens, 2007):

$$density = \frac{\sqrt{N_p}}{1 + \sqrt{(var(X) + var(Y))}}. \tag{5.5}$$

8. *Shape index* is defined by the following ratio:

$$shape\ index = \frac{perimeter}{4\sqrt{N_p}}. \tag{5.6}$$

9. *Convex hull* of the object can also be used to define all the shape features. A region R is convex if the straight-line segment between any two points of R lies completely inside of R. An example is shown in Fig. 5.2

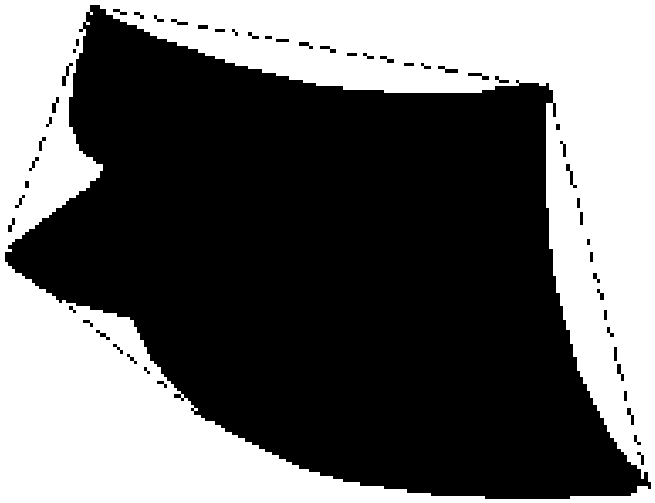


Figure 5.2: The convex hull of the object is shown using the dotted line

10. *Invariant moments* are the descriptors of the object shapes whose values are invariant to geometric transformations. M.K. Hu first used the geometric moments in 1962 to derive a set of invariant moments which are invariant to rotation, translation and scaling(Hu, 1962). Since then, several other complicated types of moments are invented for a variety of tasks. A detailed survey of invariant moments can be

found in (Kristinsdottir, 2008). The invariant moments defined by Hu are described here.

For a two dimensional continuous function $f(x, y)$, the $(p + q)^{th}$ order moment is defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy. \tag{5.7}$$

The *central moments* are then defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_0)^p (y - y_0)^q f(x, y) dx dy, \tag{5.8}$$

where, $x_0 = \frac{m_{10}}{m_{00}}$ and $y_0 = \frac{m_{01}}{m_{00}}$.
The *normalized central moments* are defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \tag{5.9}$$

where $\gamma = \frac{p+q}{2} - 1$ for $p + q = 2, 3, ...$
The *invariants moments* are then derived as

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02}, && (5.10)\\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, && (5.11)\\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, && (5.12)\\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, && (5.13)\\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&\quad -3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\quad [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], && (5.14)\\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad +4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}), && (5.15)\\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\
&\quad -3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \\
&\quad [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]. && (5.16)
\end{aligned}$$

The above set of seven invariant moments are invariant to rotation, translation and scale change. The seventh invariant moment ϕ_7 is skew invariant i.e., it can also identify mirror images. Fig. 5.3 shows an example object and the same object scaled by a factor of 2 and rotated. Both the objects have the same values of the Hu moments. The higher order moments are very sensitive to noise any small geometric distortion but the lower order moments show enough resilience to noise and slight geometric distortions. Some other examples of invariant moments are affine moments (Flusser and Suk, 1993), Zernike moments (Sim et al. 2004), Legendre moments (Teague, 1980) etc.

(a)

Object 1	0.34644	0.00063	0.00115	0.00194	-0.000003	0.000048	-0.0000007
Object 2	0.34633	0.00063	0.00115	0.00195	-0.000003	0.000049	-0.0000007

(b)

Figure 5.3: a)An example object and its scaled and rotated form. b) The values of the moments for the two objects

5.3 Texture features

Texture can be quantified using a lot of descriptors. Two of methods which are the descriptors using the *gray-level co-occurence matrix* (GLCM) and the fractal dimension are described here.

5.3.1 Gray-level Co-occurrence Matrix (GLCM)

Several texture measures can be defined using the GLCM (Haralick et al., 1973). For an angle θ and distance d we can dene a matrix $P = [p_{ij}]$, where p_{ij} counts how often a pixel with grey-level i occurs at d pixels distance (or offset) in the θ-direction from pixel j. In general the matrix P will have the dimensions $L \times L$ for a total of L brightness values possible ($L = 2048$ for 11 bit data). Dividing the matrix P by the total number of possible (i, j) pairs we get a GLCM denoted as $C = [c_{ij}]$.

The total number of GLCMs will depend on the chosen values of θ and d. The GLCMs calculated using various values of θ are used to check if the texture is orientation dependant and those calculated for various values of d are used to check scale of the texture variation. By averaging the GLCMs over all the directions, we assume that the texture is not varying significantly in different directions.

The following descriptors can be calculated from the GLCM:

1. *Mean*:

$$\mu = \sum_i \sum_j i c_{ij}. \tag{5.17}$$

2. *Variance*:
$$\sigma = \sum_i \sum_j (i-\mu)^2 \sqrt{c_{ij}}. \tag{5.18}$$

3. *Angular second moment*:
$$f_1 = \sum_i \sum_j c_{ij}^2. \tag{5.19}$$

4. *Contrast*:
$$f_2 = \sum_i \sum_j (i-j)^2 c_{ij}. \tag{5.20}$$

5. *Correlation*:
$$f_3 = \sum_i \sum_j \frac{(i-\mu)(j-\mu)c_{ij}}{\sigma^2}. \tag{5.21}$$

6. *Entropy*:
$$f_4 = \sum_i \sum_j c_{ij} log(c_{ij}). \tag{5.22}$$

7. *Homogeneity*:
$$f_5 = \sum_i \sum_j \frac{c_{ij}}{1+(i-j)^2}. \tag{5.23}$$

8. *Dissimilarity*:
$$f_6 = \sum_i \sum_j \mid i-j \mid c_{ij}. \tag{5.24}$$

5.3.2 Fractal dimension

Fractal dimension is often used as a texture measure. There are several methods of calculating the fractal dimension. The method based on the semi-variogram is explained here.
The semivariogram of any spatial distribution is given as (Carr, 1995):

$$\gamma(h_{ij}) = \frac{1}{2} E([Y(x_i) - Y(x_j)]^2), \tag{5.25}$$

Y is the spatial function, γ the semivariogram and h_{ij} is the lag.
It is calculated that

$$\gamma(h_{ij}) \propto 2|h_{ij}|^H \tag{5.26}$$

So, the slope of the line plotted between $\log \gamma$ and $\log h$ is H and is related to the fractal dimension, D of the surface as

$$D = 3 - \frac{H}{2} \tag{5.27}$$

5.4 Context features

The descriptors of context of the image object are very important features of the object as they define the relations to the other objects which inturn can be used for classification. For instance, the objects of the class shadows always have a bigger difference to the brighter neighboring objects; an island is described only by means of water objects and so on. There are several types of context features depending on what type of relations are required. The relations can be established with respect to the objects of the same segmentation level or to the parents and children at different segmentation levels. The representative layer values of the objects can be compared to the neighbors or even the shape features can be compared to establish geometric relations to the neighbors (e.g., relative border to the brighter/darker objects or comparison of both, area in comparison with the neighboring objects belonging to other class, etc). The position of the object with respect to the entire scene or any particular class of objects is another way of defining context.

Huge information can be extracted from the image objects. However, only a few types of features actually are characteristic of the classes to which the objects are assigned to. It remains a big challenge to extract the right type of description of a class by using the right features out of the thousands of available features. A way to find the best features characterizing the classes and methods to classify the objects based on the object features will be presented in chapter 6.

Chapter 6

Classification

Image classification using OBIA as mentioned earlier is more promising compared to pixel based classification as it allows us to use the shape, texture and context information along with the layer values. The classification can be done either "*parallely*," where all the classes are classified at once or "*sequentially*,"where objects belonging to one class are classified at a time. Sequential classification allows us to use the relations to the classifications as well. However, the task of identifying the characteristic features of the classes is an issue as we have to pick the right features from a huge set of available features. In some cases, the features are based on models or experience. For example, the shadow of the building can be identified using the *mean difference to the brighter neighbor objects* or when the buildings are already classified, the shadows are the darker neighbor objects of the buildings; the roads or rivers can be identified using the *width* of the objects in addition to layer values. In other cases, we can identify the features of the classes of interest based on the samples.

In this chapter, a method to identify the features characterizing the classes of interest is explained and examples are provided to explain how to use the features to classify the objects both in a parallel manner and sequential manner. Then, attention is given to check how a non-linear classifier such as neural network classifier can be used in OBIA. A new architecture using neural networks is designed specially for OBIA and is named as *class dependent neural networks*. Furthermore, it is shown how to use neural networks in sequential classification where only one class is classified at a time. But, before all that, a discussion will be provided about the validity of classification accuracies using objects as entities instead of pixels.

6.1 Accuracy assessment

It is a general requirement to provide an accuracy estimate of the classification result as an indicator of the quality. Accuracy assessment can be easily misunderstood while using the object-based methods. A lot of the authors report the accuracies by considering the image objects as basic entities. This author strictly opposes this practice. First of all,

there is no basis to adapt the image objects created using segmentation algorithms as the basic entities of image interpretation and secondly the object based methods are often compared to pixel-based classification methodologies where the basic entities are pixels. Moreover, the data values at the pixels are the actual measurements of the remote sensing sensors. So, it is more meaningful to have the accuracy assessment at the pixel level.

Another important consideration is that accuracy assessment is a statistical method which demands a sample size of reasonable size which may not always be possible with the image objects. After segmentation, we are left with a very small number of image objects compared to the number of pixels and hence a very small number of samples to be tested for accuracy. With a small sample set, the results of the accuracy assessment do not provide reliable measures.

Consider the classification as a random experiment and the possible outcomes are grouped in a set $\bar{A}$, A, where $\bar{A}$ is the set of misclassified samples and A is the set of correctly classified samples (Canty, 2007).
A real-valued function (i.e., a random variable), X is defined such that

$$X(\bar{A}) = 1,\ X(A) = 0. \tag{6.1}$$

The random variable X has a mass function as

$$Pr(X = 1) = \theta,\ Pr(X = 0) = 1 - \theta. \tag{6.2}$$

The mean and variance of X are then given as

$$mean,\ \langle X\rangle = 1.\theta + 0.(1-\theta) = \theta, \tag{6.3}$$

$$variance,\ var(X) = \langle X^2\rangle - \langle X\rangle^2 = \theta(1-\theta). \tag{6.4}$$

For n test data, the total number of misclassifications is the sample function

$$Y = X_1 + X_2 + ... + X_n. \tag{6.5}$$

So, the random variable describing the misclassification rate is Y/n. The mean of this is

$$\langle\frac{1}{n}Y\rangle = \frac{1}{n}(\langle X_1\rangle + ... + \langle X_n\rangle) = \theta \tag{6.6}$$

and the variance of the misclassification rate is

$$\sigma^2 = var\left(\frac{Y}{n}\right) = \frac{\theta(1-\theta)}{n}. \tag{6.7}$$

For y observed misclassifications, the estimate of θ is

$$\hat{\theta} = \frac{y}{n} \tag{6.8}$$

and the corresponding estimated variance is

$$\hat{\sigma}^2 = \frac{\hat{\theta}(1-\hat{\theta})}{n} = \frac{y(n-y)}{n^3}, \tag{6.9}$$

Y/n is a generally binomially distributed, but can be approximated by a normal distribution for a sufficiently large n. Eq. 6.9 has an interesting outcome. We can determine the size of the set of samples required to claim that two values resulted from the classifications using two different classifiers are significantly different.

For example, consider a misclassification rate of $\theta = 0.1$, i.e., 90% accuracy. If we want to claim that 95% and 90% accuracies are significantly different then the standard deviations should not be greater than 0.025 i.e., 2.5%. Using Eq. 6.9

$$0.1(1 - 0.1)/n \approx 0.025^2 \;\Rightarrow\; n \approx 144.$$

It has to be noted that we need 144 samples in the above case just to say that the accuracies are significantly different but then even more samples are required to actually establish that the difference of 5% is valid. As the differences decrease, even more samples would be required. This is a big ask when we consider image objects as entities for accuracy assessment. This is another good reason to use pixels as basic entities for classification accuracy assessment.

Normally a *contingency table* or *confusion matrix* is used to present the classification accuracies. For K classes the confusion matrix is defined as

$$C = [c_{ij}],\ i, j = 1, 2, ...K, \tag{6.10}$$

where c_{ij} is the number of samples belonging to class j which are classified as class i.

The estimated misclassification rate is

$$\hat{\theta} = \frac{y}{n} = \frac{n - \sum_{i=1}^{K} c_{ii}}{n}, \tag{6.11}$$

where only the diagonal elements of the confusion matrix are considered.
The *kappa coefficient* uses all the matrix elements to correct the classification rate when there is a possibility of chance correct classification. It is defined as

$$\kappa = \frac{Pr(\text{correct classification}) - Pr(\text{chance classification})}{1 - Pr(\text{chance classification})} = \sum_{i=1}^{K} \frac{c_{i.}c_{.i}}{n^2}, \tag{6.12}$$

where,

$$c_{i.} = \sum_{j=1}^{K} c_{ij} \text{ and } c_{.i} = \sum_{i=1}^{K} c_{ji}.$$

All the accuracies in this thesis are quoted as overall accuracies using pixels as the basic entities.

6.2 Separability and Threshold (SEATH)

The optimum features which can distinguish between classes can be calculated using the separability measure based on the sample distribution of the classes. We can also identify a threshold of separation using the Baye's rule (Richards and Jia, 1999; Nussbaum et al, 2005; Marpu et al, 2008).

6.2.1 Distance between random distributions

A popular measure of distance between two random distributions is the Bhattacharya distance measure, B (Bhattacharya, 1943). For two random distributions described by probability density functions $p_1(x)$, $p_2(x)$,

$$B = -ln\left(\int \sqrt{p_1(x)p_2(x)}dx\right). \tag{6.13}$$

For a discrete case with uni-modal distribution functions, we can approximate Eq. 6.13 as

$$B = -ln\left(\sum_i \sqrt{p_1(x_i)p_2(x_i)} \triangle x\right). \tag{6.14}$$

x_i refers to the discrete points and $\triangle x$ refers to the sampling interval or typically the width of the bins in the histogram. The discrete probability density function is obtained by normalizing the histogram with respect to the total area under the histogram. When both the distributions are assumed to follow normal distribution, then we can calculate B as

$$B = \frac{1}{8}(\mu_1 - \mu_2)^2 \frac{2}{\sigma_1^2 + \sigma_2^2} + \frac{1}{2}ln(\frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1\sigma_2}), \tag{6.15}$$

where μ_1, μ_2 and σ_1, σ_2 are the means and standard deviations of the two distributions respectively.
The range of B falls in half-closed interval $[0, \infty)$. This range is transformed into the closed interval $[0, 2]$ by using a simple transformation leading to so called Jeffries-Matusita distance measure, J,

$$J = 2(1 - e^{-B}) \tag{6.16}$$

$J = 0$ implies that the two distributions are completely correlated and hence inseparable. $J = 2$ implies that the distributions are completely uncorrelated and can be separated.

For every feature, we can calculate the separability between the samples of two classes using J. The features which have very high J value are the optimum features which characterize the classes.

6.2.2 Threshold of separation

We can distinguish between two random distributions by using a threshold of separation. For two classes C_1 and C_2 and an observation x, using the Bayes' rule we get,

$$p(C_1|x) = \frac{p(x|C_1)p(C_1)}{p(x)}, \tag{6.17}$$

and similarly for C_2. We then have

$$p(C_1|x) + p(C_2|x) = 1. \tag{6.18}$$

These are the only possibilities and the total probability should be 1. The best decision threshold then is given by the relation

$$p(C_1|x) = p(C_2|x). \tag{6.19}$$

On rearranging the terms using the above equations, we have

$$p(x|C_1)p(C_1) = p(x|C_2)p(C_2). \tag{6.20}$$

For discrete functions, we can find the threshold as

$$T = x_j, \text{ where } |p(x_j|C_1)p(C_1) - p(x_j|C_2)p(C_2))| \approx 0. \tag{6.21}$$

For the case where the class samples of classes C_1 and C_2 of size n_1, n_2 with means μ_1, μ_2 and standard deviations σ_1, σ_2 respectively are assumed to be normally distributed,

$$T = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 \pm \sigma_1\sigma_2\sqrt{((\mu_1 - \mu_2)^2 + 2A(\sigma_1^2 - \sigma_2^2)}}{(\sigma_1^2 - \sigma_2^2)},$$

$$A = log[\frac{\sigma_1}{\sigma_2} * \frac{n_2}{n_1}]. \tag{6.22}$$

The degree of misclassification using this threshold value depends on the separability of the classes.

SEATH was first used by Sven Nussbaum for OBIA but the principles behind it are rather basic results from Statistics theory (Nussbaum et al, 2005). One disadvantage of the method is that, it uses the assumption of normal distribution for the classes. However, as most of the class distributions can be modeled using the normal distribution this method works in most of the cases. If the classes are not normally distributed the value of the threshold will be significantly different but the values of the separabilities are still reasonably valid. A software to display the separabilities and thresholds in a graphical user interface (GUI) has been developed by this author. Fig. 6.1 shows a screenshot of the GUI. The tables in the column correspond to a class of interest compared with all the other classes.

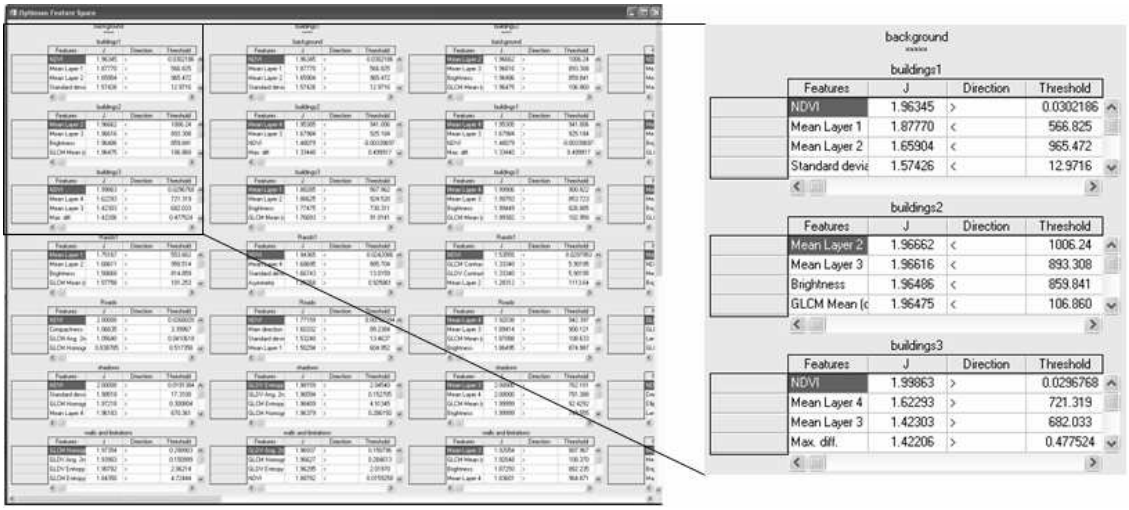


Figure 6.1: SEATH_GUI: A software for feature identification

6.3 Neural networks

Neural networks are from a long time, well established as one of the standard methods for classification of RS data (Bishop, 2006; Canty, 2007). There is often a comparison of the effectiveness of classification using neural networks and other important types of non-linear classifiers such as *support vector machines* (SVM). There are contradicting reports in the literature where some authors claim SVMs are superior (Huang et al, 2002; Foody and Mathur, 2004) and some others claim architectures based on neural networks perform better (Canty, 2009). In this chapter, the author does not like to take any position on which of the classifiers is superior. SVMs belong to the class of non-parametric methods where no assumptions of the probability distributions of the classes is made and simultaneously the posterior probabilities of the training data cannot be modeled. On the other hand, neural networks classifier belongs to the class of *semiparametric models* where no strong assumptions of the probability distributions of the classes are made but still the posterior class probabilities of the training data can be modeled. This is one of the reasons why neural networks are selected in this thesis to achieve some kind of probabilities of classes which can be used in a way suitable for OBIA to deal with a huge number of features effectively. First, a brief introduction to the neural networks and the training algorithms and then the details of a newly developed architecture based on the neural networks will be presented. This architecture allows us to effectively deal with the big feature space of the image objects. The architecture allows to characterize every class based on the feature space only defined by the features characterizing that class thus providing a better way to handle the complexity of the huge feature space available in OBIA.

6.3.1 Feed forward neural networks

The neural network classifiers belong to the category of semiparametric models for probability density estimation as mentioned earlier. Not only that they make no assumptions of the probability distribution but also can be adjusted flexibly to the complexity of the system to be modeled. This advantage sometimes is considered as a disadvantage as there is not way to identify the actual configuration of the architecture to suitably adjust for the complexity. There are no general theoretic rules concerning the choice of architecture and the decision is often made using a *trial and error* method. There are several architectures of neural networks (Bishop, 2006). Only two layer feed forward neural networks, which are more commonly used are presented here. The neural networks with single hidden layer can in principle, approximate any given decision boundary arbitrarily closely (Bishop, 1995). Using more hidden layers can define the training data very well but using more hidden layers can also lead to over-fitting of data which is not a desired result. The general architecture of the feed-forward neural networks is shown in Fig. 6.2 (Canty, 2007).

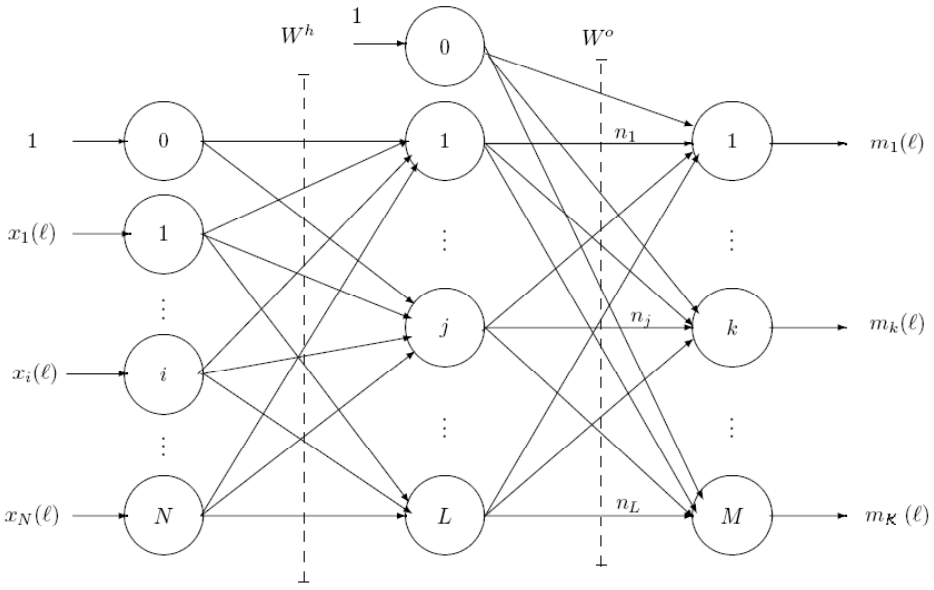


Figure 6.2: The architecture of feed-forward neural network

Consider a feature vector $x = (x_0, x_1, x_2, ...x_N)^T$ (x_0 =1 is added as the first element to add the bias) where N is the total number of features and the features are to be matched to an output vector $m = (m_1, m_2, ..m_K)^T$. The inputs are matched to the outputs through a *hidden layer* with $L + 1$ components. Every, element of the network is called a *node* (denoted by the circles in Fig. 6.2). The input to every node is a weighted combination of the nodes in the preceding layer as shown in Fig. 6.3. This representation of the node is termed *artificial neuron* in conjunction with the biological analogy and hence the name *neural networks*. In coherence with this, the inputs are termed as *signals*.

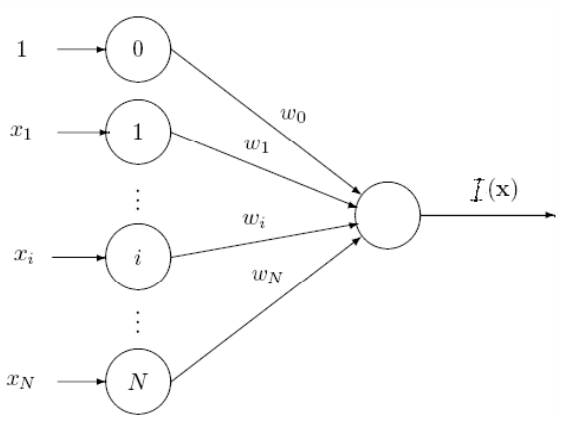


Figure 6.3: An artificial neuron

The input signals $x_0, x_1, ...x_N$ are multiplied with the *synaptic weights* $\omega_0, \omega_1, ...\omega_N$ and summed to produce an output signal $I(x)$.

$$I(x) = \omega_0 x_0 + \omega_1 x_1 + ...w_N x_N = \omega^T x + \omega_0. \tag{6.23}$$

The output signal $I(x)$ is then modified by a *sigmoid activation function* given in Eq. 6.24.

$$f(x) = \frac{1}{1 + e^{-I(x)}}. \tag{6.24}$$

For a training entity, ν, the input of the network would be

$$x(\nu) = (1, x_1(\nu)...x_N(\nu))^T. \tag{6.25}$$

This input signal propagates to all the L neurons in the hidden layer. An extra node with an input of 1 is also used in the hidden layer to account for the bias. So, the output of the hidden layer is a component vector of intermediate outputs of size $(L+1)$ as follows:

$$n(\nu) = (1, n_1(\nu)...n_L(\nu))^T, \tag{6.26}$$

$$n_j(\nu) = f(I_j^h(x(\nu))), \qquad j = 1...L. \tag{6.27}$$

The activation, I_j^h of the hidden neurons is:

$$I_j^h(x(\nu)) = {\omega_j^h}^T x(\nu), \tag{6.28}$$

where, $\omega_j^h = (\omega_{0j}^h, \omega_{1j}^h ... \omega_{Nj}^h)^T$ is the weight vector for the jth neuron of the hidden layer. The weight matrix of the hidden layer can then be written as

$$W^h = (\omega_1^h, \omega_2^h ... \omega_L^h). \tag{6.29}$$

Now the output of the hidden layer can be written as

$$n(\nu) = \left(1, \left(W^{h^T} x(\nu)\right)\right)^T. \tag{6.30}$$

The vector n is then fed to the *output layer* in the similar way. The weight matrix of the output layer is: $W^o = (\omega_1^o, \omega_2^o ... \omega_K^o)$ for K outputs. The output signal is then calculated as

$$m(\nu) = f\left(W^{oT} n(\nu)\right). \tag{6.31}$$

However, if a sigmoid function is considered for this activation at the output the sum of all the output is not unity. As mentioned earlier, the network outputs have to be interpreted as class membership probabilities ensuring that

$$0 \leq m_k(\nu) \leq 1, k = 1....K \tag{6.32}$$

and

$$\sum_{k=1}^{K} m_k(\nu) = 1. \tag{6.33}$$

This can be achieved by using a modified activation function called *softmax*. The softmax function is defined as follows:

$$m_k(\nu) = \frac{e^{I_k^o(n(\nu))}}{e^{I_1^o(n(\nu))} + e^{I_2^o(n(\nu))} + ... + e^{I_K^o(n(\nu))}}, \tag{6.34}$$

where,

$$I_k^o(n(\nu)) = \omega_k^{oT} n(\nu), \qquad k = 1...K. \tag{6.35}$$

6.3.2 Cost functions

The values of the synaptic weights has to be determined by training the neural network with sample data. Let the training data of size p be represented as a set of labeled pairs as follows:

$$\{x(\nu), l(\nu) \mid \nu = 1....p\}, \tag{6.36}$$

where the label, $l(\nu) = (0...0.1, 0...0)^T$) is a K dimensional column vector of zeros except at the kth position to indicate that the sample belongs to class k.

The *quadratic cost function* given in Eq. 6.37 can be minimized by means of adjusting the weights to train the network.

$$E(W^h, W^o) = \frac{1}{2} \sum_{\nu=1}^{n} \parallel l(\nu) - m(\nu) \parallel^2. \tag{6.37}$$

A more appropriate cost function for classification problems can be obtained by choosing to maximize the probability of observing the training data. The joint probability of observing the training data is

$$Pr(x(\nu), l(\nu)) = Pr(l(\nu) \mid x(\nu)) Pr(x(\nu)). \tag{6.38}$$

The neural network as explained earlier gives the approximate posterior class membership probability $Pr(l(\nu) \mid x(\nu))$. This can now be written using the terms of the neural network as:

$$\prod_{k=1}^{K} [m_k(x(\nu))]^{l-k(\nu)}. \tag{6.39}$$

By taking the logarithm and removing the independent terms, we end up with the *cross entropy cost function* given as:

$$E(W^h, W^o) = -\sum_{\nu=1}^{p} \sum_{k=1}^{K} l_k(\nu) log[m_k(x(\nu))]. \tag{6.40}$$

So, we get the synaptic weight parameters by minimizing the above cross entropy cost function. It can be derived that minimizing the cross entropy cost function is equivalent to that of minimizing the quadratic cost function (Canty, 2007).

6.3.3 Training algorithms

The cost function can be minimized in several ways. The easiest and most commonly used algorithm to train the feed forward neural network is the *backpropogation* algorithm, which is a gradient descent algorithm. These algorithms make use of the first derivatives of the cost function with respect to the synaptic weight parameters. One disadvantage of such an algorithm is when the cost function consists of local minima. The first derivative cannot differentiate between the local minimum and the global minimum of the cost function and hence results in undesired results. This can be solved by using the second derivatives of the cost functions or the *Hessian matrix* (Bishop, 1995) using the *scaled conjugate gradients* algorithm. But the calculation of the Hessian matrix is computationally expensive and the algorithm is very slow to converge. The *Kalman filter* training algorithm can however overcome this drawback. It can identify the global minimum of the cost function and moreover, it converges very fast. So, in this thesis the Kalman filter training is used. The back propagation algorithm is also used in addition to the Kalman filter training to further adjust the synaptic weights. Canty (Canty, 2007) suggests to use the scaled conjugate gradients method along with the Kalman filter for the refinement of synaptic weights. However, this author prefers to use the backpropagation algorithm because, when the kalman filter is employed, the result is already close to the global minimum and so, the backpropagation algorithm can also perform the same task of refining the synaptic weights to get more close to the global minimum. Using the backpropagation algorithm instead of the scaled conjugate gradients algorithm has the advantage that it is not computationally expensive like the later.

6.3.4 Backpropagation

Consider the local cost function for a training sample ν given as follows:

$$E(W^h, W^o, \nu) = -\sum_{k=1}^{K} l_k(\nu) log[m_k(x(\nu))] \qquad \nu = 1...n. \tag{6.41}$$

If this function is minimized for every training sample, we are also minimizing the overall cost function as given in Eq. 6.40. To make it more compact, we can replace $x(\nu)$ with ν to represent the νth training sample. Now the problem is to minimize the local cost function given in Eq. 6.41 with respect to the synaptic weights which are in turn elements of the two matrices W^h and W^o of sizes $(N+1)XL$ and $(L+1)XK$ respectively.

The general version of the backpropagation algorithm is described as follows:
Algorithm: Backpropagation

1. Initialize the synaptic weights with random numbers.
2. Choose a random training pair $(x(\nu), l(\nu))$ from the training data and determine the output response $m(\nu)$.
3. Replace all the elements, ω^o_{jk} of the matrix W^o with $\omega^o_{jk} - \eta\frac{\partial E(\nu)}{\partial \omega^o_{jk}}$
4. Replace all the elements, ω^h_{ij} of the matrix W^h with $\omega^h_{ij} - \eta\frac{\partial E(\nu)}{\partial \omega^h_{ij}}$
5. If $\sum_\nu E(\nu)$ ceases to change significantly, stop, or else choose another training pair randomly and go to step 2.

The algorithm scans through the training data, reducing the local cost function in every step. This is done by adjusting the synaptic weights by changing them by an amount proportional to the negative slope (the first derivative) of the local cost function with respect to that weight parameter. The constant of proportionality, η is referred to as the *learning rate* of the network.

To derive the expressions to calculate the partial derivatives of $E(\nu)$, consider the output neurons which generate the softmax output signals.

$$m_k(\nu) = \frac{e^{I^o_k(\nu)}}{e^{I^o_1(\nu)} + e^{I^o_2(\nu)} + ... + e^{I^o_K(\nu)}}, \tag{6.42}$$

where,

$$I^o_k(n(\nu)) = \omega^{oT}_k n(\nu), \qquad k = 1...K.$$

We wish to calculate

$$\frac{\partial E(\nu)}{\partial \omega^o_{jk}} \qquad j = 0...L, \quad k = 1...K.$$

By using the chain rule, we can rewrite it as

$$\frac{\partial E(\nu)}{\partial \omega^o_{jk}} = \frac{\partial E(\nu)}{\partial I^o_k(\nu)} \frac{\partial I^o_k(\nu)}{\partial \omega^o_{jk}} = -\delta^o_k(\nu) n(\nu), \qquad k = 1...K, \tag{6.43}$$

where,

$$\delta^o_k(\nu) = -\frac{\partial E(\nu)}{\partial I^o_k(\nu)}. \tag{6.44}$$

Using Eq. 6.41 and Eq. 6.42 and applying chain rule as follows:

$$-\delta^o_k(\nu) = \frac{\partial E(\nu)}{\partial I^o_k(\nu)} = \sum_{k'=1}^{K} \frac{\partial E(\nu)}{\partial m_{k'}(\nu)} \frac{\partial m_{k'}(\nu)}{\partial I^o_k(\nu)},$$

$$-\delta^o_k(\nu) = \sum_{K'=1}^{K} \frac{l_{k'}(\nu)}{m_{k'}(\nu)} \left(\frac{e^{I^o_k(\nu)} \delta_{kk'}}{\sum_{k''=1}^{K} e^{I^o_{k''}(\nu)}} - \frac{e^{I^o_{k'}(\nu)} e^{I^o_k(\nu)}}{\left(\sum_{k''=1}^{K} e^{I^o_{k''}(\nu)} \right)^2} \right), \tag{6.45}$$

where,

$$\delta_{kk'} = \left\{ \begin{array}{ll} 0 & \text{if } k \neq k' \\ 1 & \text{if } k = k' \end{array} \right\}. \tag{6.46}$$

This reduces to

$$-\delta^o_k(\nu) = \sum_{K'=1}^{K} \frac{l_{k'}(\nu)}{m_{k'}(\nu)} m_k(\nu)(\delta_{kk'} - m_{k'}(\nu))), \tag{6.47}$$

$$\begin{aligned} -\delta^o_k(\nu) &= \sum_{K'=1}^{K} \frac{l_{k'}(\nu)}{m_{k'}(\nu)} m_k(\nu)(\delta_{kk'} - m_{k'}(\nu))), \\ &= -l_k(\nu) + m_k(\nu), \quad k = 1...K. \end{aligned}$$

In vector form,

$$\delta^o(\nu) = l(\nu) - m(\nu). \tag{6.48}$$

Therefore, the update equation for the output layer becomes:

$$W^o(\nu + 1) = W^o(\nu) + \eta n(\nu) \delta^o(\nu)^T. \tag{6.49}$$

For the hidden weights, the same procedure is applied and chain rule is used repeatedly.

$$\begin{aligned} -\delta^h_j(\nu) &= \sum_{k=1}^{K} \frac{\partial E(\nu)}{\partial I^o_k(\nu)} \frac{\partial I^o_k(\nu)}{\partial I^h_j(\nu)}, \\ &= -\sum_{k=1}^{K} \delta^o_k(\nu) \frac{\partial I^o_k(\nu)}{\partial I^h_j(\nu)}, \\ &= -\sum_{k=1}^{K} \delta^o_k(\nu) \omega^{oT}_k \frac{\partial n(\nu)}{\partial I^h_j(\nu)}. \end{aligned} \tag{6.50}$$

Now, since $I_j^h = {w_j^h}^T x(\nu)$, the partial derivative inside the summation is only a function of jth hidden neuron. Therefore,

$$\delta_j^h(\nu) = \sum_{k=1}^{K} \delta_k^o(\nu) {\omega_{jk}^o}^T \frac{\partial n_j(\nu)}{\partial I_j^h(\nu)}. \tag{6.51}$$

The hidden neurons use the logistic activation function. So,

$$n_j(I_j^h) = f(I_j^h) = \frac{1}{1 + e^{-I_j^h}}. \tag{6.52}$$

The derivative of this function can be easily derived as:

$$\frac{\partial n_j(t)}{\partial t} = n_j(t)(1 - n_j(t)). \tag{6.53}$$

This leads to

$$\begin{pmatrix} 0 \\ \delta^h(\nu) \end{pmatrix} = n(\nu) * (1 - n(\nu)) * (W^o \delta^o(\nu)), \tag{6.54}$$

where $*$ represents *Hadamard* multiplication (element by element multiplication of the matrices).

We finally obtain the equation to update the synaptic weights of the hidden layer as:

$$W^h(\nu + 1) = W^h(\nu) + \eta x(\nu) \delta^h(\nu)^T. \tag{6.55}$$

The speed of the convergence of the algorithm depends on the η value. However, for high values, it might lead to oscillations. An additional parameter α termed as *momentum* is introduced to hold a portion of the previous weight changes in the current iteration. So, now the update equations will be

$$W^o(\nu + 1) = W^o(\nu) + \Delta^o(\nu) + \alpha \Delta^o(\nu - 1), \tag{6.56}$$

where $\Delta^o = \eta n(\nu) {\delta^o}^T(\nu)$ and a similar update for the hidden layer as well.

6.3.5 Kalman filter training

Kalman filter is a recursive estimator in the sense that only the current state and current measurement are required to predict a future state. It is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear ltering problem (Kalman 1960). Without giving much details about the history the application of the Kalman filter for neural network training will be presented here. the basic application of the Kalman filter is in recursive linear regression where the measurement data are sequentially presented and the best solution for the regression parameters is determined as the new data becomes available. In this process, no instances of the prior measurements are stored (Canty, 2007).

Recursive linear regression

Consider a statistical model,

$$Y_i = \omega^T x_i + R_i, i = 1...m, \tag{6.57}$$

which relates the n independent variables $x_i = (x_1, ...x_n)_i^T$ to a measured quantity Y_i using the parameters $\omega = (\omega_1...\omega_n)^T$. The random variables R_i represent the measurement uncertainties. The random variables R_i are assumed to be uncorrelated and normally distributed with zero mean and variance σ^2.

When ν measurements have been made, we can write the Eq. 6.57 in the form

$$Y_\nu = A_\nu \omega + R_\nu, \tag{6.58}$$

where, $(A_\nu)_{ij} = x_j(i)$, $Y_\nu = (Y(1), ...Y(\nu))^T$ and $R_\nu = (R(1), ...R(\nu))^T$.
The best least squares solution for the parameter vector ω is given by

$$\omega(\nu) = \left[(A_\nu^T A_\nu)^{-1} A_\nu^T\right] y_\nu = \Sigma_\nu A_\nu^T y_\nu, \tag{6.59}$$

where the expression in the square brackets is the pseudo inverse of A_ν and $\Sigma(\nu)$ is the covariance matrix of ω If there is a new observation,$(x(\nu+1), y(\nu+1)$ available, the equation for the least squares problem becomes

$$\begin{pmatrix} Y_\nu \\ Y(\nu+1) \end{pmatrix} = \begin{pmatrix} A_\nu \\ A(\nu+1) \end{pmatrix} \omega + R_{\nu+1}, \tag{6.60}$$

where $A(\nu+1) = x(\nu+1)^T$ is a row vector.
Using the Eq. 6.59 the solution of the new problem is

$$\omega(\nu+1) = \Sigma_{\nu+1} \begin{pmatrix} A_\nu \\ A(\nu+1) \end{pmatrix}^T \begin{pmatrix} Y_\nu \\ Y(\nu+1) \end{pmatrix}. \tag{6.61}$$

Solving this equation, we get a recursive formula for the new covariance matrix $\Sigma_{\nu+1}$ as:

$$\Sigma_{\nu+1}^{-1} = \begin{pmatrix} A_\nu \\ A(\nu+1) \end{pmatrix}^T \begin{pmatrix} A_\nu \\ A(\nu+1) \end{pmatrix}. \tag{6.62}$$

This can be reduced to

$$\Sigma_{\nu+1}^{-1} = \Sigma_\nu^{-1} + A(\nu+1)^T a(\nu+1). \tag{6.63}$$

Similarly, we can obtain a recursive relation to update the parameter set ω as

$$\omega(\nu+1) = \omega(\nu) + K : \nu+1[y(\nu+1) - A(\nu+1)\omega(\nu)], \tag{6.64}$$

where the *Kalman gain*, $K_{\nu+1}$ is given by

$$K_{\nu+1} = \Sigma_{\nu+1} A(\nu+1)^T. \tag{6.65}$$

Eqs. 6.63- 6.65 define the so-called *Kalman filter* for the recursive least squares regression problem.

The equations that define the Kalman filter are inconvenient as the Eq. 6.63 calculates the inverse of covariance matrix. The equations can however be reformed as follows in to a more convenient way:

$$\Sigma_{\nu+1} = [I - K_{\nu+1}A(\nu+1)]\Sigma_\nu$$
$$K_{\nu+1} = \Sigma_\nu A(\nu+1)^T[A(\nu+1)\Sigma_\nu A(\nu+1)^T + 1]^{-1}. \tag{6.66}$$

Training algorithm

The appropriate cost function is the local version of the quadratic cost function given as:

$$E(\nu) = \frac{1}{2}\|l(\nu) - m(\nu)\|^2. \tag{6.67}$$

Consider an isolate neuron as shown in Fig. 6.4.

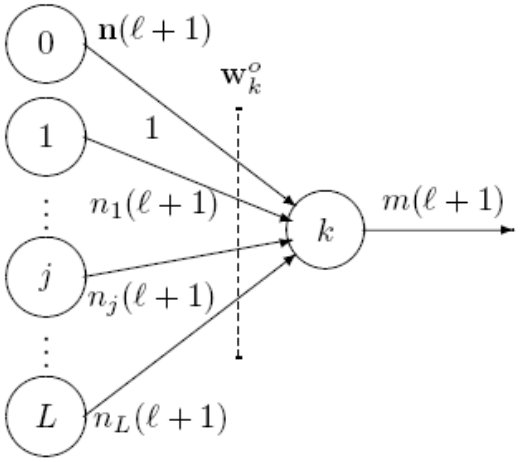


Figure 6.4: An isolate output neuron

The neuron receives its input from the hidden layer and generates the softmax output signal as:

$$m_k(\nu) = \frac{e^{\omega_k^{oT}(\nu)n(\nu)}}{\sum_{k'=1}^{K} e^{\omega_{k'}^{oT}(\nu)n(\nu)}}. \tag{6.68}$$

The derivatives of $m_k(\nu)$ with respect to ω_k^o and n yields,

$$\frac{\partial}{\partial \omega_k^o} m_k(\nu) = m_k(\nu)(1 - m_k(\nu))n(\nu)$$
$$\frac{\partial}{\partial n} m_k(\nu) = m_k(\nu)(1 - m_k(\nu))\omega_k^o(\nu). \tag{6.69}$$

For simplicity sake, the terms ω_k^o, m_k and l_k are represented as ω, m and l respectively.By just considering the isolated neuron, m is a function of ω only.

$$m(\nu) = m(\omega(\nu)^T n(\nu)). \tag{6.70}$$

The weight vector $\omega(\nu)$ is the updated vector after presenting ν training observations. Using the Taylor series expansion about a point $\omega(\nu)$ we can get the linear approximation to $m(\nu+1)$

$$m(\nu+1) \approx m(\omega(\nu)^T n(\nu+1)) + \left(\frac{\partial}{\partial\omega} m(\omega(\nu)^T n(\nu+1))\right)^T. \tag{6.71}$$

From Eq. 6.69 we then have,

$$m(\nu+1) \approx \hat{m}(\nu+1)(1-\hat{m}(\nu+1))n(\nu+1)^T(\omega - omega(\nu)), \tag{6.72}$$

where,

$$\hat{m}(\nu+1) = m(\omega(\nu)^T n(\nu+1)).$$

By defining a linearized input as

$$A(\nu) = \hat{m}(\nu)(1-\hat{m}(\nu))n(\nu)^T \tag{6.73}$$

we can write Eq. 6.72 as

$$m(\nu+1) \approx A(\nu+1)\omega + [\hat{m}(\nu+1) - A(\nu+1)\omega(\nu)] \tag{6.74}$$

The term in the square bracket results from the fact that the neuron's output signal is not simply proportional to ω. If we neglect it then we have

$$m(\nu+1) = A(\nu+1)\omega \tag{6.75}$$

which is in fact a linearized neuron output signal. This implies that we can use the Kalman filter to perform a recursive linear regression over this equation to update the synaptic weights ω. The least squares problem now looks like

$$\begin{pmatrix} l_\nu \\ l(\nu+1) \end{pmatrix} = \begin{pmatrix} A_\nu \\ A(\nu+1) \end{pmatrix}\omega + R \tag{6.76}$$

The Kalman filter equations for the recursive solution of this problem are again:

$$\begin{aligned} \Sigma_{\nu+1} &= [I - K_{\nu+1}A(\nu+1)]\Sigma_\nu \\ K_{\nu+1} &= \Sigma_\nu A(\nu+1)^T[A(\nu+1)\Sigma_\nu A(\nu+1)^T + 1]^{-1} \end{aligned} \tag{6.77}$$

and the recursive expression for the synaptic weights becomes

$$\omega(\nu+1) = \omega(\nu) + K_{\nu+1}[l(\nu+1) - A(\nu+1)\omega(\nu)] \tag{6.78}$$

This can be further improved by replacing the linear approximation $A(\nu+1)\omega(\nu)$ by the actual output of the $\nu+1$st training observation, i.e., $\hat{(m)}(\nu+1)$.

Based on this theoritical background, the algorithm for training the feed forward neural network using Kalman filter is explained here. This method was initially suggested by (Shah and Palmieri, 1990). A detailed explanation of this algorithm is given in (Canty, 2007; Canty 2009).
Algorithm: Kalman filter training

1. Set $\nu = 0$,$\Sigma_j^h(0) = 100.I^o$, $\Sigma_k^o(0) = 100.I^h$, $k = 1...K$ where I^h, I^o are the identity matrices of sizes $(N+1)\times(N+1)$ and$(L+1)\times(L+1)$ respectively. Also initialize the synaptic weight matrices $W^h(0)$ and $W^o(0)$ with random numbers.

2. Choose a random training pair $(x(\nu+1), l(\nu+1)$ and determine the hidden layer output
$$\hat{n}(\nu+1) = \begin{pmatrix} 1 \\ f\left(W^h(\nu)^T x(\nu+1)\right) \end{pmatrix}$$
and the quantities,
$$A_j^h(\nu+1) = \hat{n}_j(\nu+1)(1-\hat{n}_j(\nu+1))x(\nu+1)^T, \qquad j = 1...L,$$
$$\hat{m}_k(\nu+1) = m_k\left(\omega_k^{oT}(\nu)\hat{n}(\nu+1)\right),$$
$$A_k^o(\nu+1) = \hat{m}_k(\nu+1)(1-\hat{m}_k(\nu+1))\hat{n}(\nu+1)^T, \qquad k = 1...K,$$
and
$$\beta^o(\nu+1) = (l(\nu+1) - \hat{m}(\nu+1)) * \hat{m}(\nu+1) * (1-\hat{m}(\nu+1))$$

3. Calculate the Kalman gains of all the neurons according to
$$K_k^o(\nu+1) = \Sigma_k^o(\nu)A_k^o(\nu+1)^T[A_k^o(\nu+1)\Sigma_k^o(\nu)A_k^o(\nu+1)^T + 1]^{-1},$$
$$k = 1...K.$$
$$K_j^h(\nu+1) = \Sigma_j^h(\nu)A_j^h(\nu+1)^T[A_j^h(\nu+1)\Sigma_j^h(\nu)A_j^h(\nu+1)^T + 1]^{-1},$$
$$j = 1...L.$$

4. Update the synaptic weight matrices:
$$\omega_k^o(\nu+1) = \omega_k^o(\nu) + K_k^o(\nu+1)[l_k(\nu+1) - \hat{m}_k(\nu+1)], \quad k = 1...K$$
$$\omega_j^h(\nu+1) = \omega_j^h(\nu) + K_j^h(\nu+1)[W_j^o(\nu+1)\beta^o(\nu+1)], \quad j = 1...L$$

5. Determine the new covariance matrices:
$$\Sigma_k^o(\nu+1) = [I^o - K_k^o(\nu+1)A_k^o(\nu+1)]\Sigma_k^o(\nu), \quad k = 1...K.$$
$$\Sigma_j^h(\nu+1) = [I^h - K_j^h(\nu+1)A_j^h(\nu+1)]\Sigma_j^h(\nu), \quad j = 1...L.$$

6. If the overall quadratic cost function is sufficiently small the algorithms stops or else a new random training sample is chosen and the algorithm is repeated from step 2.

6.4 Class dependent neural networks

In this thesis, a new architecture is developed using the feed forward neural networks to especially facilitate the handling of the huge feature space related to the image objects. This architecture is named *class dependent neural network architecture* as the output of the individual networks used in the architecture only use the characteristic features of the classes independently and then make a final decision in the end. As discussed in the Sec. 6.2, every class has its own set of features which characterize the class and so it is necessary to deal with this feature set independently. Moreover, when the number of classes increase, the feature space which is a set of all the characteristic features of all the classes will again be huge and it is hard to deal with it.

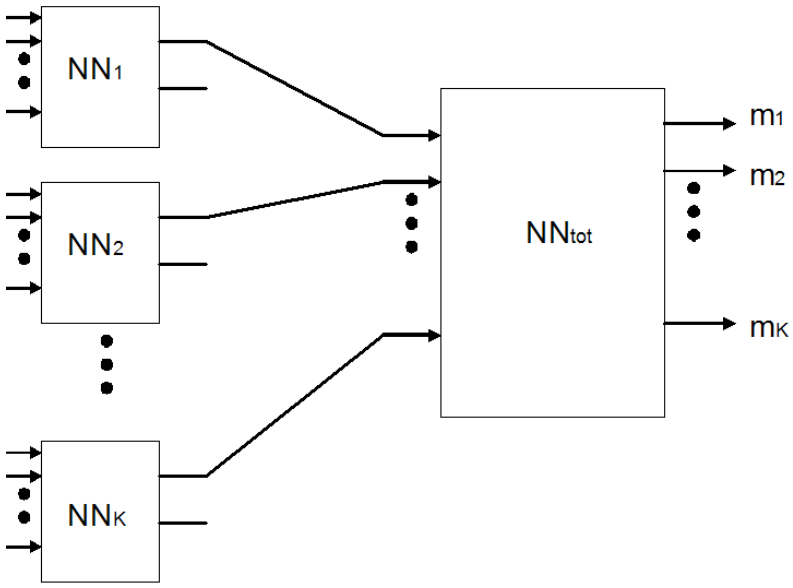


Figure 6.5: Class dependent neural network architecture

The proposed architecture is presented in Fig. 6.5. It consists of two layers of neural networks. This architecture can be seen as analogous to the fuzzy classification where a fuzzy value is first calculated as a membership value for every class and then these fuzzy values are defuzzified to get a final class probability. In the first layer there are exactly K networks NN_k representing K classes. The network representing the kth class is only fed with the characteristic features of that class. These characteristic features can either be identified using SEATH (Sec. 6.2) or based on prior experience. The networks in the first stage do not use the softmax ouput, but instead use the logistic activation function given by Eq. 6.24. This allows us to define a fuzzy value instead of a class probability. This is done because every network is fed with different types of features and hence are independent from each other. The output of every network only represents a fuzzy value of every class and has no relation to the class probability. This process is analogous to *fuzzification*. The outputs of the first layer of networks are then fed as inputs to the second

layer network. The second layer network, NN_tot uses the softmax output to give a class membership probability unlike the first layer networks. It combines allk, the fuzzy values of the first layer networks to produce a class membership probability for every class. This process is analogous to *defuzzification*.

So, exactly $K + 1$ neural networks are used to achieve a soft classification of the data by defining class membership probabilities. The networks of the input layer are first trained independently and when the first layer networks are completely trained, the outputs resulting from the first layer networks are used as training inputs to the second layer network. All the samples are used for training all the networks. While training a network of the first layer, a label of 1 is given to the samples of the class the network is representing and 0 to all the other samples. The second layer network is trained like the normal neural networks by giving a value of 1 to the class, the training sample is labeled to and 0 to all the other classes.

One disadvantage of this architecture or for that matter all the neural networks in general is that a relatively large number of training samples are required for modeling the probability distributions of the classes accurately. This can be a disadvantage in the case of OBIA. However, the results in this thesis suggest that the architecture can fare relatively well even with a few training samples. The biggest advantage of this architecture is that it allows us to model the class probabilities based on the characteristic features of the classes where the classes of interest are more distinct from the other classes hence reducing a lot of redundancy.

The fuzzy values represented by the networks in the first layer can also be used in the case of sequential classification. For a class of interest, the network gives two values as representative fuzzy values (not probabilities) to describe the chance that an entity belongs or does not belong to that class. This is illustrated in the examples presented here. The neural network architectures presented here including the class dependent network architecture developed by this author are implemented as plugins to the Definiens developer software by Mr. Florian Bachmann and are freely available at the website http://tu-freiberg.de/fakult3/mage/geomonitoring.

6.5 Classification examples

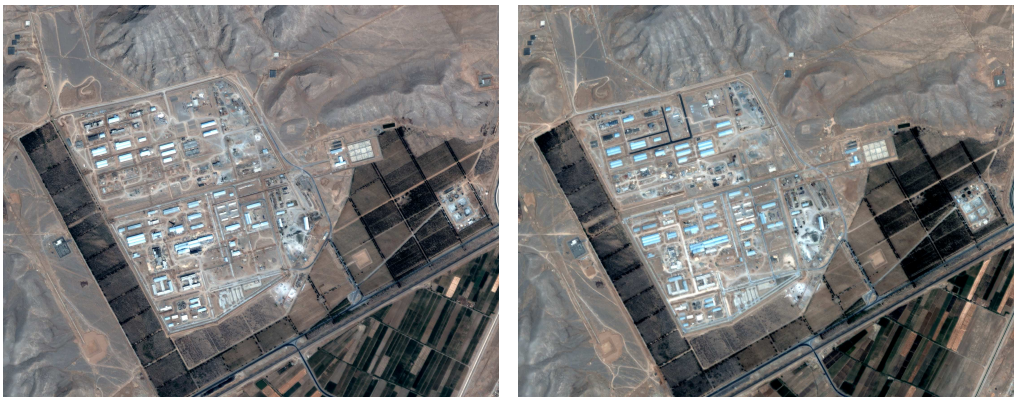
The features and thresholds calculated using SEATH based on the samples of the classes can be directly used to create rule bases for classification. Some examples will be presented here to illustrate the methodology. All the classifications are done using the Definiens Developer software. Also, in two of the examples, the results of classification using the proposed class-based neural network architecture are presented. The examples provided here will be a demonstration of the validity of the proposed methodologies and not a study of the applications of the results. The different examples are chosen from different types of settings to better validate the methods.

6.5.1 Example 1: Monitoring critical infrastructure

In this example, several direct applications of the OBIA will be presented. The aim is to develop a semi-automatic classification rule base for classifying a nuclear facility in Esfahan, Iran. These rule bases must be transferable to classify the images of the same sites at a future date. This work was done in cooperation with Mr. Christian Daneke within the framework of the GMOSS[1] (Global Monitoring for Security and Stability) Network of Excellence project (Daneke, 2008) . As the focus here is not on the application of the work but to demonstrate the effectiveness of OBIA and the proposed classification methodologies, no background information regarding the test cases is presented here.

The satellite imagery used here consists of a subset of bi-temporal set of images acquired by QuickBird satellite over the Esfahan nuclear facility in Iran on 24 July 2002 and 9th July 2003 respectively. QuickBird is a commercial EO satellite, owned by DigitalGlobe and launched in 2001. It acquires panchromatic imagery at 60-70 centimeters spatial resolution and multispectral imagery at 2.4- 2.8 meters spatial resolution. The multispectral images are pan-sharpened by using the *Á trous wavelet sharpening* (see Sec. 3.5) and registered. Furthermore, the radiometric normalization of the second image is done using the first image as the base using the method described in Sec. 3.6 in connection with using MAD transformation for change detection. The consequence of radiometric normalization to thebase image is that it allows the transferability of classification rule base from one time to another over the same area.

Fig. 6.6 shows the pre-processed images of the site at two times. The task is to develop



(a) (b)

Figure 6.6: Processed images of the subsets of an Quickbird scene of Esfahan, Iran in 2002 (a) and 2003 (b)

a rule base to classify the image in to six classes namely *buildings, paved streets, unpaved streets, shadows, vegetation and background*. SEATH is used to identify the features and thresholds. The following rule base is developed for this purpose:

[1] A network of excellence in the aeronautics and space priority of the Sixth Framework Programme funded by the European Commission's Directorate General Enterprise & Industry, *http : //gmoss.jrc.it*

1. The NDVI was calculated and modified into the range of 0-2000 to suit the range of QuickBird data. This modified layer is added as extra layer to aid the segmentation. Also, the NDVI layer as it is is used during the classification.

2. Different types of objects of different intensities are segmented properly at different levels. For this purpose, the image is segmented at two levels based on visual inspection. Multi-resolution segmentation algorithm is used for the segmentation of the image. The two levels are lvl30 (scale parameter=30, shape factor=0.3, compactness=0.5) and lvl60 (scale parameter=60, shape factor=0.3, compactness=0.5). The final classification will be done at lvl30. The samples of classes are visually identified over the image. 5-10 image objects are used as samples per class at each level to extract the features using SEATH. The 5 image layers used for classification are numbered as the pansharpened image bands (1-4) and NDVI (5).

3. Buildings are classified at lvl60 based on,
   Mean NDVI ≤ 0.025
   Mean Layer 2 ≥ 945

	lvl60
1.	unclassified at lvl60: building
2.	building Mean Layer 5 ≥ 0.025 at lvl60: unclassified
3.	building Mean Layer 2 $\leq$ at lvl60: unclassified
	lvl30
1.	unclassified with Existence of super objects building = 1 at lvl30: building
2.	[grow] building lvl30: ¡- unclassified Mean Layer 5 ≤ 0.01
3.	building with bright roof at lvl30: merge region
4.	building Ratio Layer 4 $\geq$ at lvl30: buildingvar
5.	buildingvar with GLCM Contrast (quick 8/11) Layer 4 (all dir.) ≤ 25 at lvl30: unclassifed
6.	buildingvar at lvl30: buildings

Table 6.1: Rulebase for buildings

4. The classification of buildings at lvl60 is transfered to lvl30. Two sub classes of buildings are used to extract the buildings based on the brightness of the roof. The objects already classified as buildings are grown using a region growing algorithm such that the shapes of the buildings are refined. The building objects in the NDVI layer are grown once to cover all the objects at the neighboring objects of the classified objects with a difference of less than 0.01. The Definiens rule base for finally classifying the buildings is shown in Table 6.1

5. The paved and unpaved streets are then classified using the features identified by SEATH as shown in Table 6.2. A class variable streetsvar is used to temporarily hold the classes.

	lvl60
1.	unclassifed at lvl60: streets(paved)
2.	streets (paved) with Mean Layer 5 $\geq$ 0.03 at lvl60: unclassified
3.	streets (paved) with Ratio Layer 1 $\leq$ 0.18 at lvl60: unclassified
4.	streets (paved) with Ratio Layer 2 $\leq$ 0.29 at lvl60: unclassified
5.	streets (paved) with Mean Layer 2 $\geq$ 890 at lvl60: streets (unpaved)
	lvl30
1.	unclassified with Existence of super objects streets (paved) (1) = 1 at lvl30: streets (paved)
2.	streets (paved) at lvl30: merge region
3.	unclassified with Border to streets (paved) > 1 Pxl at lvl30: streetsvar
4.	streetsvar with Ratio Layer 2 $\geq$ 0.3 at lvl30: streets (paved)
5.	with Existence of super objects streets (unpaved) (1) = 1 at lvl30: streets (unpaved)

Table 6.2: Rulebase for Streets

6. The vegetation is classified based on NDVI easily.

	lvl60
1.	unclassified with Mean Layer 5 $\geq$ 0.05 at lvl90: vegetation
	lvl30
1.	unclassified with Existence of super objects vegetation (1) = 1 at lvl30: vegetation

Table 6.3: Rulebase for vegetation

7. The shadows are finally classified based on fact that the shadow objects are always neighboring objects to buildings and have a very low brightness. All the remaining objects are then classified as background.

The results of the classification in the image of 2002 using the developed rule base is shown in Fig. 6.7. The same rule base is transfered to classify the image of 2003 after radiometric normalization using the MAD transformation. The results of the classification in the image of 2003 using the transfered rule base is shown in Fig. 6.8.

	lvl30
1.	unclassified with Distance to buildings < 10 Pxl at lvl30: shadowsvar
2.	shadowsvar with Brightness > 600 at lvl30: unclassifed
3.	shadowsvar at lvl30: shadows
4.	shadows at lvl30: merge region

Table 6.4: Rulebase for shadows

Figure 6.7: Classification of the subset of QuickBird image of Esfahan in 2002

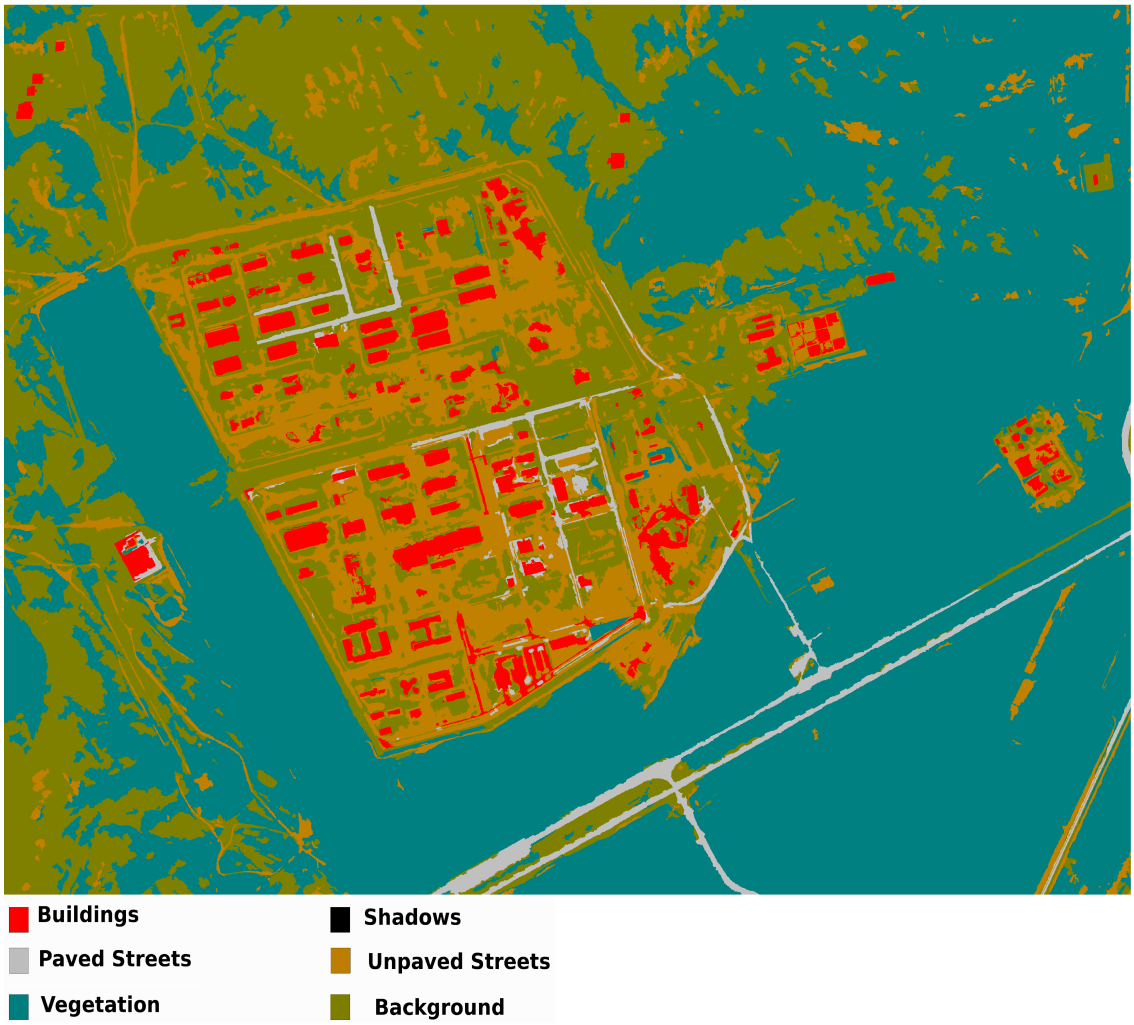


Figure 6.8: Classification of the subset of QuickBird image of Esfahan in 2003 using the rule base developed for the image of 2002

The classification results yielded a very high accuracy with an overall accuracy of 92% for the image of 2002 and and 90% for the image of 2003, excluding the shadows which consists of several misclassifications especially in relation with the buildings with dark roof as the shadows are under-segmented or the the building objects grow in to the shadows. It has proven to be a difficult task to separate the shadows of the buildings with dark roof. The accuracy assessment in this case is done by using the manually classified images as reference. The manual classifications of the different classes for the image of 2003 are shown in Fig. 6.9

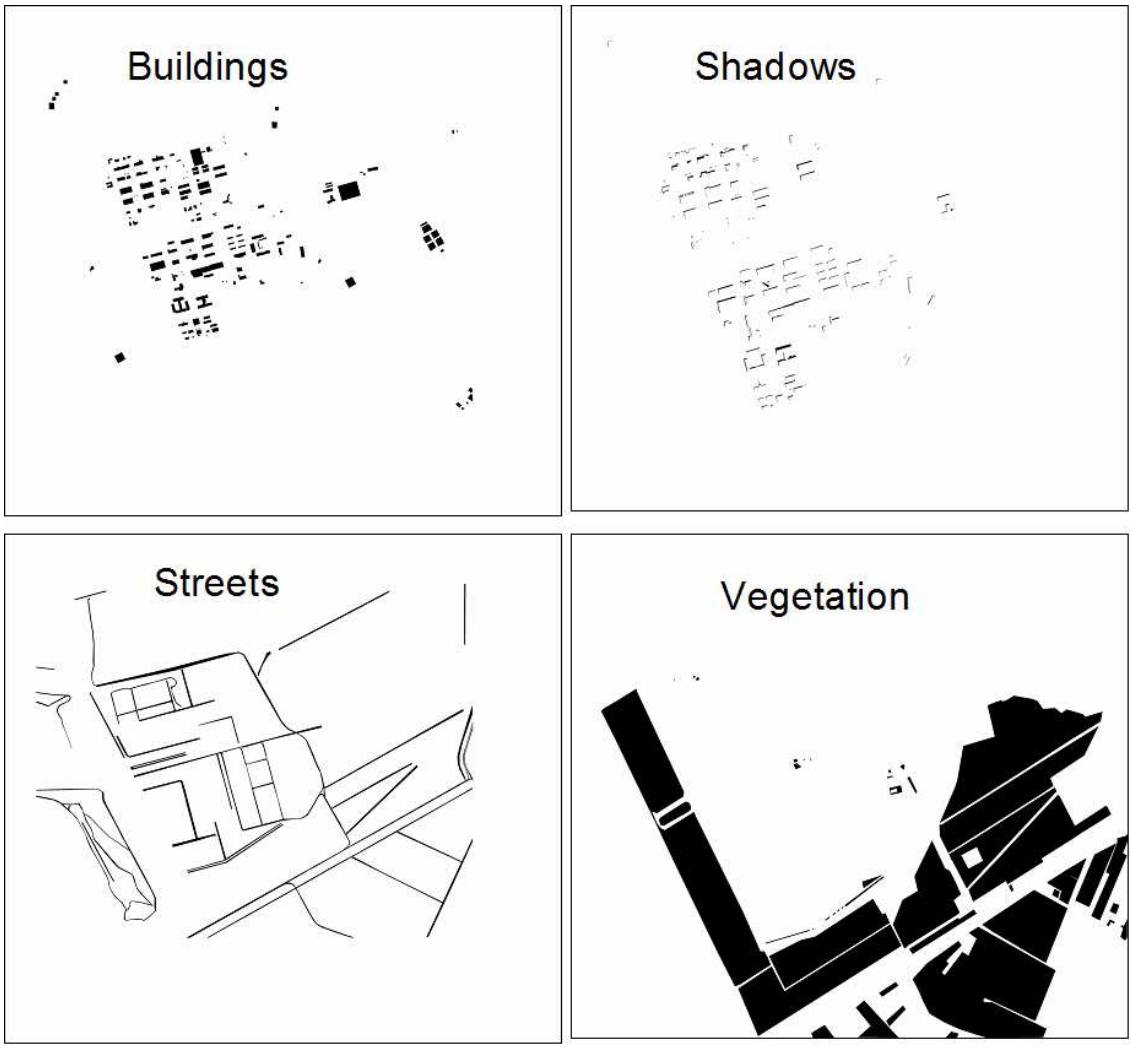


Figure 6.9: Manual classification of the subset of QuickBird image of Esfahan in 2003 for accuracy assessment

6.5.2 Example 2: Land cover classification of a rural environment

GEOBIA, as explained in the earlier chapters was mainly considered because of the advent of high resolution satellite data. However, it can also be used in the case of medium resolution satellite data such as the data from the ASTER and LANDSAT satellites. The satellite imagery used for the present study consisted of a 1000 × 1000 pixel spatial subset of an ASTER scene acquired over the town of Juelich, Germany on May 1, 20 07, processed to level 1b registered radiance at the sensor. This is exactly the same dataset used to compare the performance of several pixel based classifiers by (Canty, 2007; Canty, 2009). The six short wave infrared (SWIR) bands of the ASTER image were sharpened to the 15 m resolution of the three visual near infrared (VNIR) bands using the *Á trous wavelet sharpening*(see Sec. 3.5) so that the image consisted of nine spectral bands in all. The processed image, the training data and the class information has been kindly provided by Dr.Canty for this experiment. Ten classes of different land cover categories namely *Water, Canola fields, Beetroot fields, Settlement, Industries, Coniferous forest, Grain fields, Meadows, Deciduous forest and Open pit mine* are used. In all 7173 pixels from 30 different regions were used as the training samples. 5484 samples from 17 regions different from that of the training samples are used to test the classification results.

The task is to verify the performance of the class dependent neural networks over the segmented image in comparison with the advanced pixel-based classifiers. The additional advantage of using the segmented images is that the additional features such as texture, context and shape can be used. However, only texture is utilized in this work. The disadvantage is that the size of the training data set is remarkably smaller compared to the case of pixel-based methods. As reported in (Canty, 2009), the Adaboost classifier performs the best and results in a classification with an overall accuracy of 93.5%. Adaboost classifier is an ensemble of classifiers (in this case Kalman filter neural networks) where the misclassified training samples in the preceding network are given more priority in the next network and the result of all the classifers in the network is combined to produce a final result. A more detailed description of Adaboost algorithm for neural networks can be found in (Canty,2009). For comparison, the software developed by Mort Canty in the IDL/ENVI environment is used to classify the image using the Adaboost classifier (see Canty, 2009b).

For object based classification, the 9 ASTER bands are filtered using the first adaptive filter described in Sec. 3.2 with parameters of t1=t2=0.05. Principal component transformation was performed on the filtered image. The top 3 components having the highest eigenvalues are chosen for use in segmentation. NDVI was calculated and its value was modified into the range of 0-300 to suit the range of ASTER data. The three PCA bands and the modified NDVI layer were used for image segmentation using the multi-resolution segmentation which was done with scale factor 12, shape factor 0.2 and compactness 0.5. The pixels used as training samples are now constituted in 115 objects (Water: 2, Canola: 2, Beetroot: 5, Settlement: 22, Industries: 23, Coniferous forest: 5, Grain: 9, Meadows: 4 Deciduous forest: 8, Open pit: 35).

Fig. 6.10 shows a false color composite of the study area with bands 2,3 and 1 as red, green and blue layers respectively. Fig. 6.11 shows the false color composite of the first three principal components as red, green and blue layers respectively and Fig. 6.12 shows the NDVI modified to a range of 0-300 to suit the data range of the ASTER data.

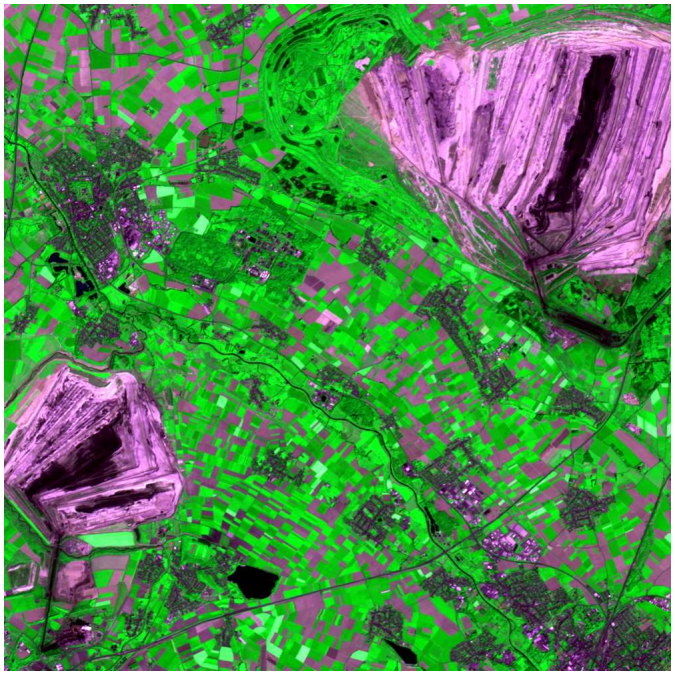


Figure 6.10: The false color composite of the Juelich study area.

The set of features distinguishing between the classes, identified based on the separability measure using SEATH are given in table: 6.5. Only the mean and median values of the objects in 9 layers of the filtered image, 9 principal components and the modified NDVI layer and texture values given by GLCM mean and homogeneity are used here. The layers are numbered in the order of 9 principal components (1-9), modified NDVI layer (10) and layers of the filtered image (11-19).

The classification is done in two ways:

1. Neural network (Kalman filter + Backpropagation) using the median values of the objects in the filtered image.
2. Class dependent neural networks using the features identified by SEATH

Neural network classification

The neural network is configured to have 11 nodes in the hidden layer and the network is trained for 50 epochs using the 115 samples of the classes. The result of the classification

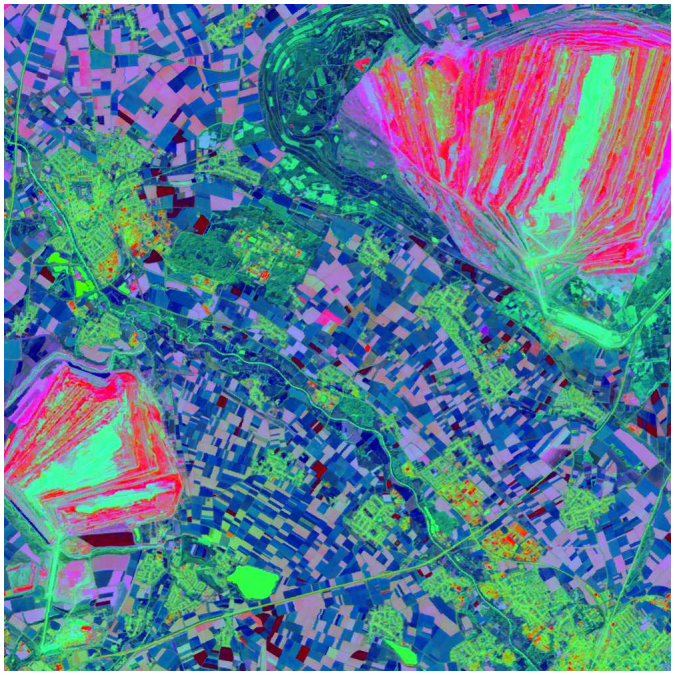


Figure 6.11: The false color composite of the first three principal components.

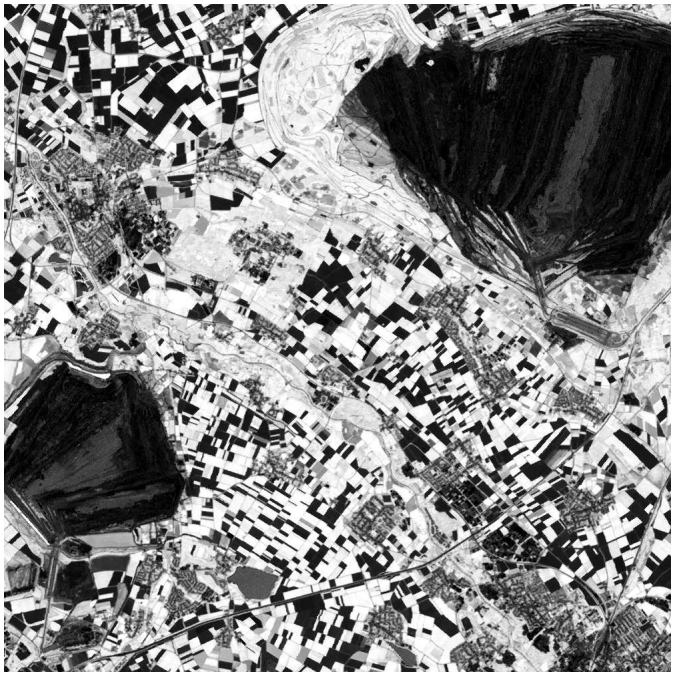


Figure 6.12: The value of the NDVI modified to a range of 0-300.

Classes	Features
Water	Median layer 2 Mean layer 10 Mean layer 12 Mean layer 13 Mean layer 15 Mean layer 18
Canola	Mean layer 12 Mean layer 13 Median layer 2 Median layer 10
Beetroot	Mean layer 12 Median layer 1 Median layer 2 Median layer 10 GLCM Mean (quick 8/11) Layer 2 GLCM Mean (quick 8/11) Layer 3
Settlement	Mean layer 13 Mean layer 14 Mean layer 17 Mean layer 18 Median layer 2 GLCM Mean (quick 8/11) Layer 1 GLCM Mean (quick 8/11) Layer 10 GLCM Homogeneity (quick 8/11) Layer 2
Industries	Mean layer 2 Mean layer 4 Mean layer 13 Mean layer 15 Mean layer 16 Mean layer 19 GLCM Mean (quick 8/11) Layer 1 GLCM Mean (quick 8/11) Layer 2 GLCM Mean (quick 8/11) Layer 10
Coniferous forest	Mean layer 8 Mean layer 14 Mean layer 16 Median layer 1 Median layer 2 Median layer 9 Median layer 10 GLCM Mean (quick 8/11) Layer 10
Grain	Mean layer 14 Median layer 1 Median layer 2 Median layer 9 Median layer 10 GLCM Mean (quick 8/11) Layer 2 GLCM Mean (quick 8/11) Layer 10
Meadows	Mean layer 4 Mean layer 14 Mean layer 16 Mean layer 17 Median layer 1 Median layer 2 Median layer 10 GLCM Mean (quick 8/11) Layer 10
Deciduous forest	Mean layer 4 Mean layer 10 Mean layer 11 Mean layer 14 Mean layer 16 Mean layer 17 Median layer 1 Median layer 2 Median layer 3 GLCM Mean (quick 8/11) Layer 10
Open pit	Mean layer 4 Mean layer 10 Mean layer 15 Mean layer 16 Mean layer 17 Mean layer 18 GLCM Mean (quick 8/11) Layer 2 GLCM Mean (quick 8/11) Layer 3 GLCM Mean (quick 8/11) Layer 10

Table 6.5: Features identified using SEATH

is shown in Fig. 6.13. An overall accuracy of 94.83% was obtained and the Meadows class accounted for the 45% of the misclassifications which is probably due to the fact

that only training samples are available. Around 15% of the test pixels were shown as misclassifications between the Settlements and Industries which can be understood.

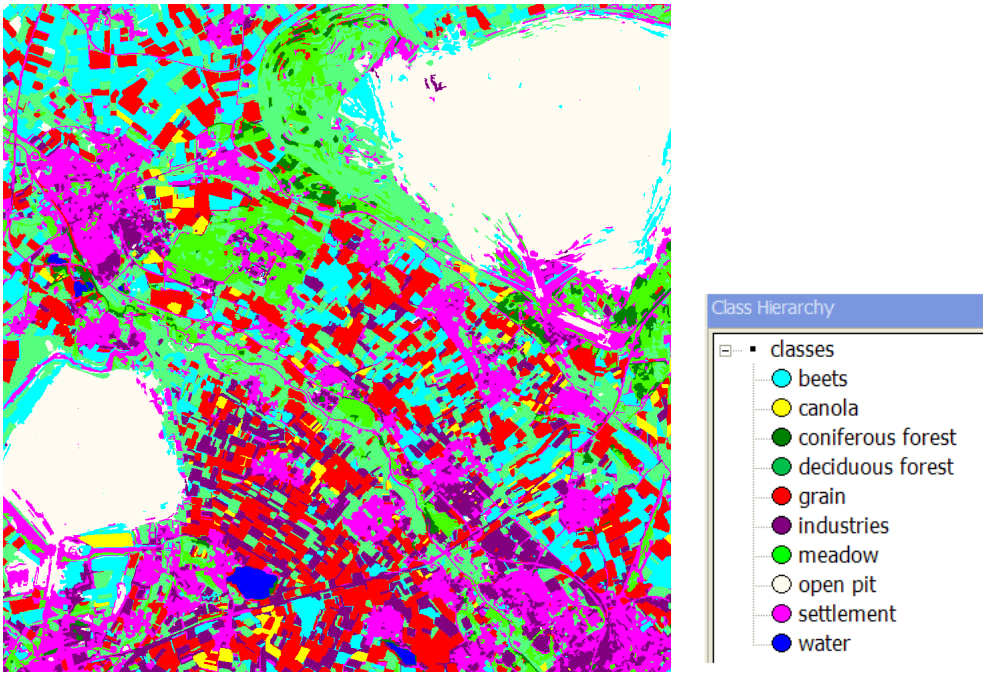


Figure 6.13: Image object based classification of the Juelich test site using the feed forward neural network

Class dependent neural networks classification

All the neural networks including the second layer neural network of the class dependent network architecture are configured to have 11 nodes in the hidden layer. A screenshot of the configuration of the class dependent neural networks is shown in Fig. 6.14 where it can be seen how we can use different features for different classes. The features shown in table: 6.5 are used for every class.

The result of the classification is shown in Fig. 6.15. An overall accuracy of 94.1% was obtained for this classification. Meadows accounted for 70% of the misclassifications and again the probable reason is the insufficient number of training samples. The misclassifications between Settlements and Industries account for 15%. So, it can be understood that the class dependent neural networks work very well when sufficient samples are provided. However, it is hard to decide how many samples are considered to be sufficient. It completely depends on the separability of the classes.

Also, as mentioned in the previous section, a single class dependent neural network from the first layer can be used to generate fuzzy values describing the cases of *belonging to the class* and *not belonging to the class*. These values can then be used in the sequential

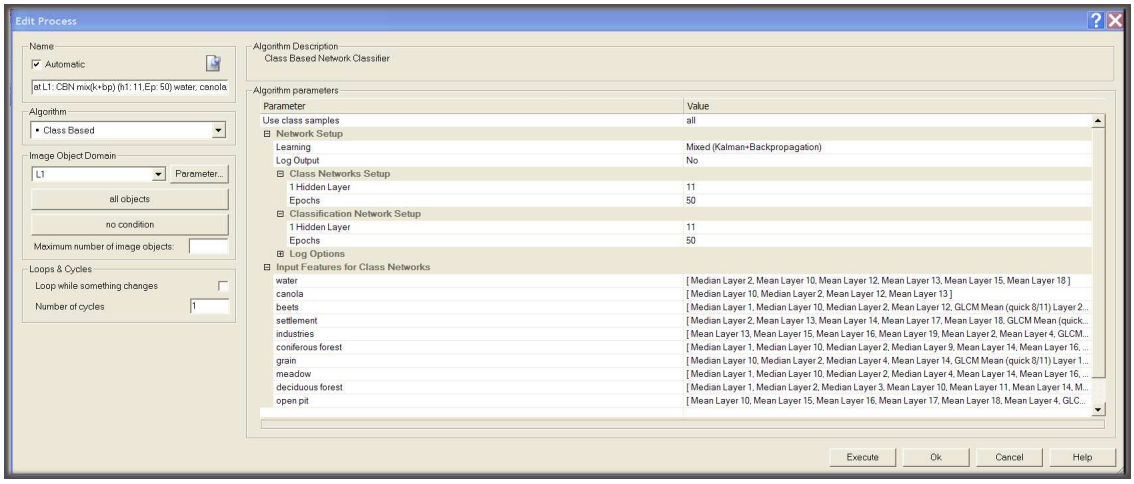


Figure 6.14: The configuration window for class dependent neural networks

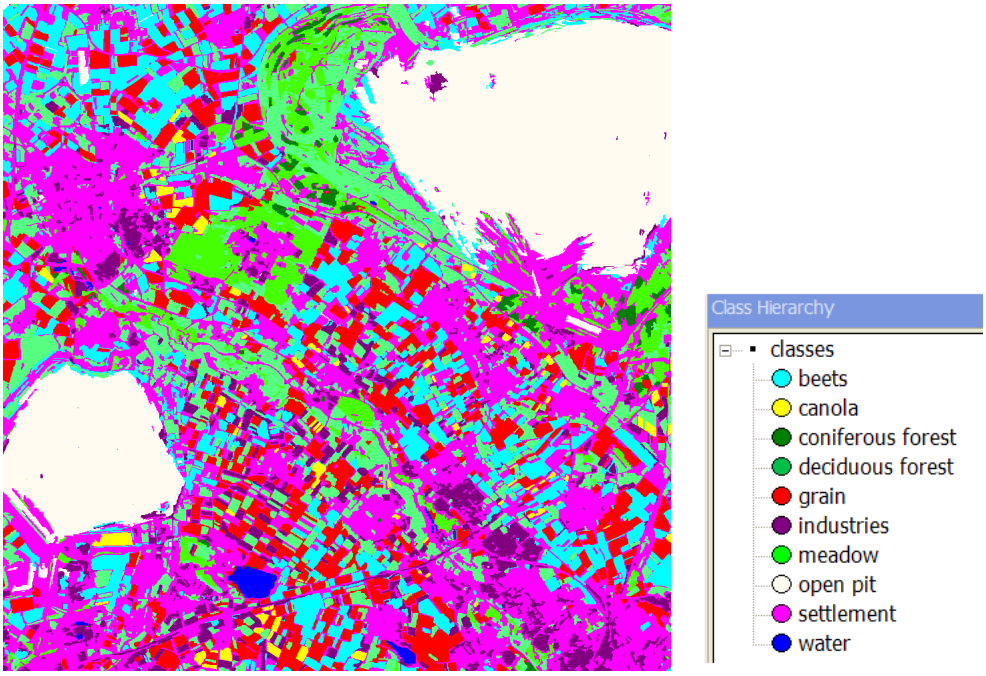
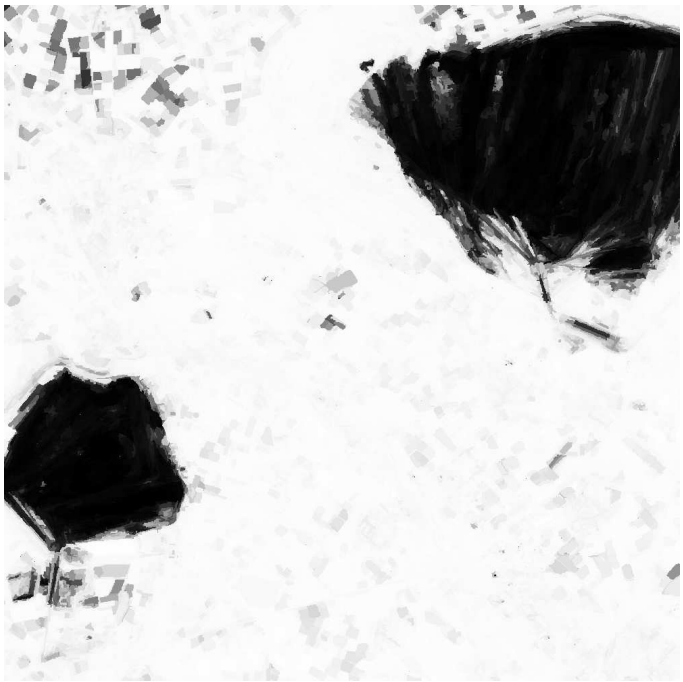


Figure 6.15: Classification of the Juelich test site using the class dependent neural networks based on features identified by SEATH

classification of the classes. As an example, Fig. 6.16 shows the images consisting of the two fuzzy values.

(a)



(b)

Figure 6.16: The fuzzy values of “is class” (top) and “ is not class” (bottom) to describe the open pit mine.

By using a threshold of 0.5 for the first image and 0.2 for the second image we get the classification of the Open pit mine as shown in Fig. 6.17.

Figure 6.17: The classification of the open pit mine by thresholding the fuzzy values.

6.5.3 Example 3: Land cover classification in a forest region

The second example further illustrates the application of OBIA and SEATH for classifying medium resolution data. The study area is located in Michoacan state, central west of Mexico, covering an area of approximately 58×60 km^2, within the longitude of 102° 00′ W and 102° 32′ W, and latitude of 19° 02′ N and 19° 36′ N. The predominant vegetation types include temperate forests, tropical dry forests, grasslands and tree plantations. The available data comprised of a Landsat ETM+ image obtained on 16 February 2003, containing 6 bands with a spatial resolution of 30m.13 land cover types are to be extracted: *orchards, dense temperate forests, sparse temperate forests, forests on the top of Tancitaro, sparse vegetation, fields with crops, dry fields without crops, wet fields without crops, lava flows, tropical dry forests, grass land, shadows, and human settlements.* Fig. 6.18 shows a color composite image using the bands 3,2 and 1 for red, green and blue layers respectively.

Principal Component Analysis (PCA) was carried out using the six Landsat bands. The first three components having the highest eigenvalues are chosen for use in segmentation. NDVI was calculated and its value was calibrated into the range of 0-300. The three PCA bands and the modified NDVI layer were used for image segmentation using the multi-resolution segmentation which was done with scale factor 20, color factor 0.7

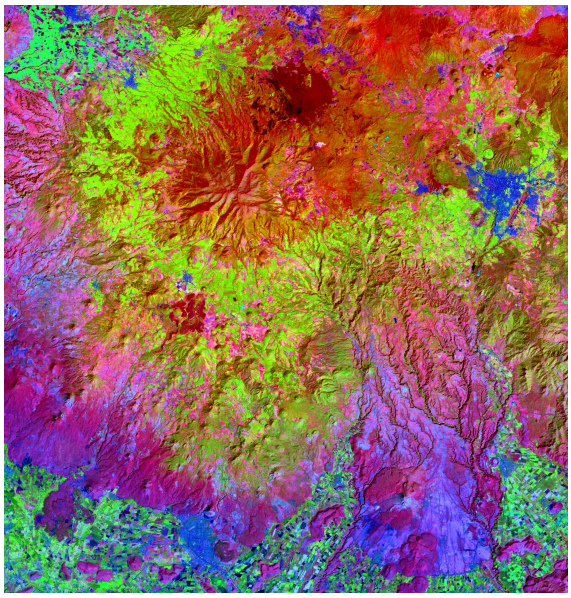


Figure 6.18: The false color composite of the Mexico study area

and compactness 0.5. Based on visual inspection, most of the expected objects are well-represented as individual segments. The rule base shown in table: 6.6 is created based on the features and thresholds identified using SEATH for 2476 sample pixels which amounted to 72 objects (roughly 5 objects per class). The image layers used for classification are numbered as the Landsat image bands (1-6), PCA image layers (7-9) and the modified NDVI (10).

The result of the classification using the thresholds identified by SEATH is shown in Fig. 6.19. An overall accuracy of $\approx 79\%$ is estimated. Other pixel based methods as reported in (Gao et al,2008) were resulting in accuracies of around 75%. Since, 1106 samples are used for testing, the improvement shown by the above classification is significant.

An attempt is made to classify the image using the class dependent neural networks even if there are only a few training samples. The result of the classification is shown in Fig. 6.20. The same features which are identified using SEATH are used as the characteristic features of the classes. The data is classified using neural networks with 10 nodes in the hidden layer and 65 epochs. An overall accuracy of 74% is achieved which is lower than the other other methods but it is a reasonable result owing to the fact that the network could not be trained properly. The most number of the misclassifications are in the class combinations (bareland, grassland) and (Orchards, Fields with Crops) where the training data is least separable with a maximum value of Jeffries Matusita distance being 1.71 for these class combinations.

The other application of the class dependent networks as discussed already, is to use a single network in the first layer to extract individual classes. An example of the settlements is shown in Fig. 6.21.

Classes	Rules
Orchards	Mean layer 1 $<$ 170 Mean layer 2 $\geq$ 199 Mean layer 5 $\leq$ 105
Sparse temperate forest	Max. diff. $<$ 1.7 Mean layer 2 $\leq$179 Mean layer 4 $\geq$ 143 Mean layer 5 $\geq$ 94.19 Mean layer 6 $\leq$ 191 Mean layer 6 $\geq$ 83
Dense temperate forest	Max. diff. $\geq$ 1.7 Mean layer 5 $\leq$72
Forest on the top of tancitaro	GLCM Mean (quick 8/11) layer 1 (all dir.) $\leq$ 187 GLCM Mean (quick 8/11) layer 1 (all dir.) $\geq$ 13.5 GLCM Mean (quick 8/11) layer 3 (all dir.) $\geq$ 165 GLCM Mean (quick 8/11) layer 4 (all dir.) $\leq$ 190 Mean layer 4 $\leq$ 187
Sparse vegetation	Mean layer 4 $\geq$ 60 Not forest type 3 Not dense forest Not sparse forest
Fields with crops	Mean layer 3 $\leq$ 105 Mean layer 4 $\geq$ 127 Mean layer 5 $\leq$ 233 Mean layer 5 $\geq$ 101
Wet fields without crops	GLCM Ang.2nd moment (quick 8/11) layer 3 (all dir.) $\geq$0.004 GLCM Mean (quick 8/11) layer 1 (all dir.) $\leq$ 197.5 Mean layer 1 $\leq$ 145 Mean layer 3 $\leq$ 77.5 Mean layer 4 $\leq$ 160 Mean layer 5 $\geq$ 130
Dry fields without crops	Brightness $\geq$ 206 GLCM Mean (quick 8/11) layer 1 (all dir.) $\geq$197 Lava flow Brightness $\leq$ 70 Mean layer 10 $\geq$ 43 Mean layer 2 $\leq$ 3.5 Ratio layer 1 $\leq$ 0.041
Lava flow	Brightness$\leq$ 70 Mean layer 10 $\geq$ 43 Mean layer 2 $\leq$3.5 Ratio layer 1 $\leq$0.041
Tropical dry forest	Max. diff $\leq$ 1.95 Mean layer 3 $\geq$ 91 Mean layer 5 $\geq$ 122 Not dry fields without crops Grassland Mean layer 10 $\geq$ 200 Mean layer 7 $\geq$ 219 Mean layer 9 $\geq$ 181 Not dry fields without crops Shadow Mean layer 10 $\leq$ 43 Mean layer 4 $\leq$ 49
Grassland	Mean layer 10 $\geq$ 200 Mean layer 7 $\geq$ 219 Mean layer 9 $\geq$ 181
Shadow	Mean layer 10 $\leq$ 43 Mean layer 4 $\leq$ 49
Human settlements	GLCM Mean (quick 8/11) (all dir.) $\leq$ 173.5 Mean layer 10 $\leq$ 233 Mean layer 10 $\geq$ 171 Mean layer 3 $\leq$ 100 Mean layer 9 $\leq$ 182

Table 6.6: Features and thresholds identified using SEATH

The class of interest appears bright if the network is properly trained with sufficient samples. In this example, even with only 6 samples of the class of interest and 66 samples combined from all the other classes, the network could effectively train to distinguish most of the settlement areas shown as bright regions in the image. This layer can be used to make a decision about the class. For instance, in this case a threshold of ≈ 0.35 classifies most of the settlements.

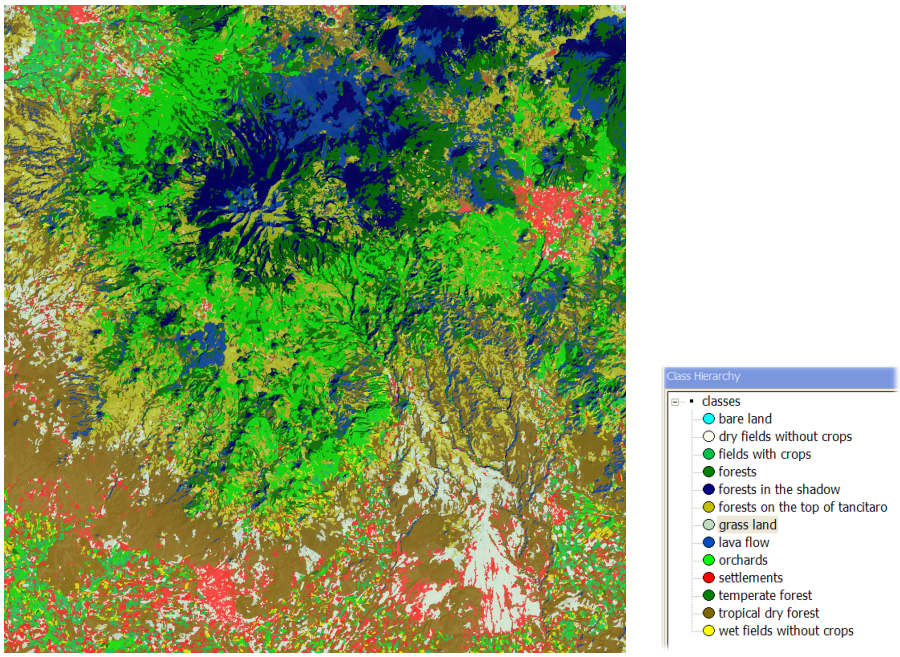


Figure 6.19: Classification of the Landsat image using SEATH

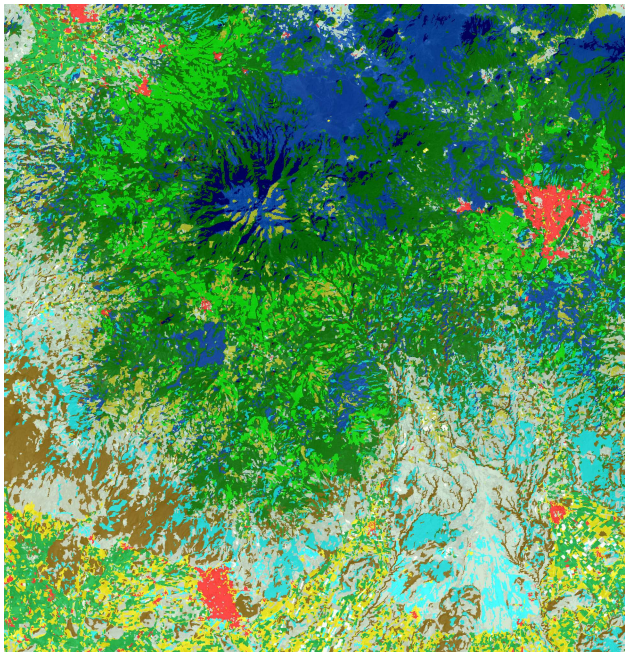


Figure 6.20: Classification of the Landsat image using the class dependent neural networks

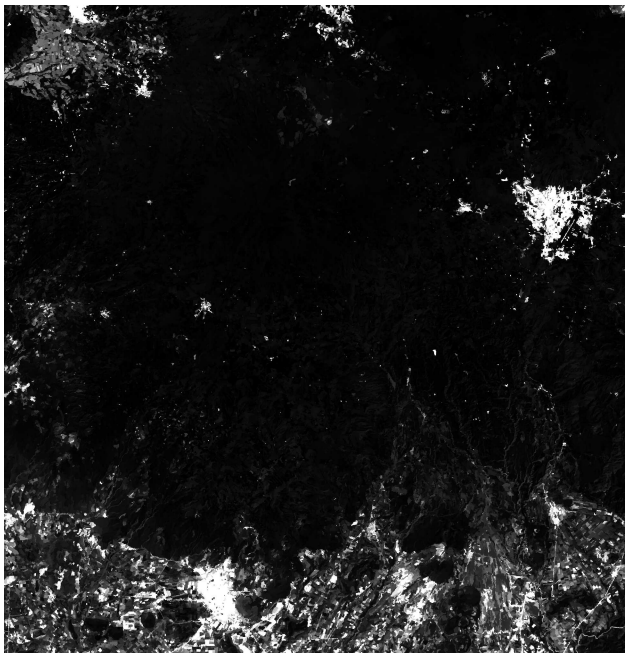


Figure 6.21: The fuzzy value of the class *settlements* given by the class dependent NN

6.6 Summary of classification examples

Though a lot of advanced classification rule bases can be developed using OBIA, the classification examples shown here are only provided to demonstrate the effectiveness of SEATH and class dependent neural networks. The example of classifying the Esfahan nuclear facility shows how we can use the multiple levels of the segmentations effectively for classification by extracting objects of different sizes at different levels. The classification of shadows with respect to the buildings is an example of using context information which is only possible in OBIA. In the Juelich and Mexico examples, the segmentation is done at only one level based on the choice of classes. The aim there was to demonstrate the use of class dependent neural networks. Even if the size of the training data is very small when image objects are used instead of pixels, the accuracies are comparable to that of the advanced pixel-based classifiers. This is due to the fact that we use the right features which distinguish between classes very well. As a summary, class dependent neural networks architecture is found to be an effective solution to create classification rule bases without having to calculate the thresholds as done using SEATH. The networks automatically map the characteristic features to the class labels. Furthermore, the usage of a single class dependent neural network to calculate the fuzzy values is proven to be very effective when sufficient training samples are available to be used in the sequential classification of the classes. The features identified by SEATH are shown to be effectively distinguishing between the classes. So, the combination of SEATH and class-dependent neural networks is a very promising step forward towards the automation of object-based classification.

Chapter 7

A GEOBIA system: An integrated system for remote sensing based monitoring tasks

The current trend is to use GEOBIA as a bridge between remote sensing (RS) and geographic information systems (GIS) (Lang and Blaschke, 2006). Segmentation of the image prior to classification means that we are already dealing with GIS ready entities in the form of objects. The bridge linking both the pixel domain of RS and vector domain of GIS is the creation of polygons from objects. So, essentially most of the work done using GEOBIA was just image classification. However, the concept of GEOBIA in itself is more than just classification. This author strongly supports the idea of a system centered around GEOBIA where RS and GIS are the components instead of GEOBIA being a part of the current system where it is just used as a link between RS and GIS. The concept as it sounds is not just a mere fusion of RS, GEOBIA and GIS, but instead a system which can handle more complexity than what can be handled by any present day systems. It will be a system which can interactively deal with any types of operations (including processing, querying and visualization) on the huge database that is designed to evolve with time. Such a system will be at the helm of any monitoring applications of RS data. Attempts have already been made to design such systems exclusively for certain types of monitoring tasks e.g. monitoring of nuclear facilities (Niemeyer et al, 2005). The advantage of focussing on GEOBIA system rather than the well developed GIS is that we can have more flexibility in handling the data processing while still acquiring all the functionality of the advances of the GIS systems. Effectively, we should be in a position both the pixel and vector data domains. So, there is a need for a paradigm shift in utilizing RS data at the application level. The new system should be able to provide us with a way to first store, process and visualize data and then allow us to intelligently use the spatial and temporal dimensions of the data in any further analysis thereby allowing the system to evolve by itself. This chapter will try to provide ideas for designing such a system. A partial implementation of a few ideas of the system has been done to get an idea of the data structures. This was again done in cooperation with Mr. Christian

Daneke (Daneke, 2008). The fact that all the components of this system are already well known makes it an even more simple system in practice. The only new thing about this is, how these individual components are grouped to make an intelligent architecture to deal with all sorts of applications related to RS applications. This should be the direction of the future of GEOBIA.

7.1 System design

The block diagram of the system is given in Fig. 7.1. The representation is more or less a blue print for the software implementation of the system. The reason for having an

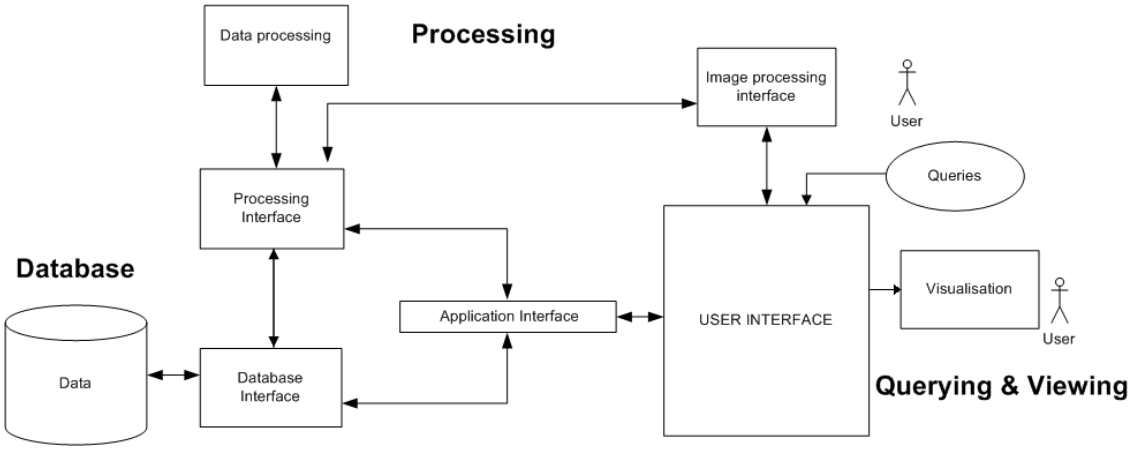


Figure 7.1: Block diagram representation of the GEOBIA system

interface for every component in the block diagram is to keep in mind that there are several types of solutions to realize the same components as will be explained here. The system has three basic components connected by interfaces.

1. *Database*, which stores all the data used by the system. Effective data structures have to be designed to handle the data in an efficient way. As the system evolves, the database gets bigger and hence the data organization at the basic level has a lot of consequences in the evolution of the system. Several implementations of the databases with spatial extensions to handle spatial data are available in the market commercially (e.g. ORACLE) and as open-source versions (e.g. PostGreSQL/PostGIS). The *database interface* component is introduced to account for this to switch between different types of databases.

2. *Processing*, which handles all the computational part of the system which involves image processing in the pixel and vector domain. Again several image processing softwares are already available in the market (e.g. Definiens, IDL/ENVI, etc). A typical image processing system comes with the three components *user interface, image processing interface* and *data processing*. In the block diagram they are

separated to fit the description of the system architecture. The *processing interface* has to be implemented to connect to the components of the commercial or open-source softwares (e.g. Definiens Developer software provides a software development kit (SDK) to interact with the components of the Definiens software)

3. *Querying & Visualization*, which handles all the user interactions with the system. This is a big part of the *user interface*. The *user interface* should handle all the complexity of the querying structure and visualization. The querying and visualization can be either using a standalone application to run on a computer or through a web browser in connection with the network components. When the network components are to be involved there are other interfaces required between the *application interface, user interface* and *database*

All the above components have a connection to the common block of *user interface* which is in fact the central component of the entire system. Imagine a system where we can perform any sort of queries on the existing data (e.g. query about the image, class in an image, class statistics in an image in comparison with the other image, object of a class, object features such as shape, texture, etc, context of the object). This functionality is already available in all of the GIS systems. Now imagine that we have some new data. If there is information (e.g. classification) available from the same region from an earlier date, we can either automatically classify the new data based on the classification rule bases already available for that scene (see Sec. 6.5.1) or alert the user to create a new classification rule base for the new data by allowing the user in to an image processing software environment. This is the duty of the *application interface* to switch between the processing mode and retrieve the data from the database. The data once classified will be stored again in the database for any future reference or querying. This functionality when attached to the existing GIS system possibly creates a GEOBIA system. The advantage of the GEOBIA system is that it can evolve into a wealth of information for a wide range of applications of RS.

7.2 Data structures

Fig. 7.2 shows an entity representation diagram (ERD) of the database implementation. This is a partial implementation mainly for demonstration purpose where a nuclear sites monitoring architecture is being developed (Daneke, 2008).

Five main data structures have to be independently represented in a relational database management system (RDBMS) which are inter-connected by means of common members in the structures.

1. *Imagery* is the structure used to represent the images. Every image is given a serial number (*image_id)* and a description (*classification_id*). The image corresponds to a site represnted by a serial number (*site_id*). The other information concerning the image are the date of acquisition (*acquisition_date*) and the type of the sensor (*sensor*).

Figure 7.2: The entity representation diagram of the implemented database

2. The *classes* are represented as a different structure (*distinct_classes*) with the name of the class (*class*) and a serial number given to that class (*class_id*).

3. The structure *site* holds the information regarding the sites with the serial number of the site (*gid*), the name of the site (*name*), a unique number given to the site (*number*), type of activity at the site (*activity*), any other description (*info*), the geographical location of the site (*xy_coordinates*), the number of images of the site in the database (*image_count*) and the classification rule base to classify any image of the future date (*ruletree*).

4. The structure *Classification shapes* is the representation of the image objects. The geometry of the image object (*the_geom*) is the key to retrieve the object from the raster images. The object is to be classified and hence has a class label (*class*) attached to it. The other members of the structure define the object features such as *area, length*, etc are used only for the demonstration. Owing to the large number of features available for every image object, the features of the objects under consideration should be calculated on demand.

5. In every monitoring system, the information of changes is the most important part. the structure *change_shps* accounts for this. It consists of the type of change classification (class) with a number(change_id) between different times (*time1, time2*) and all the change objects are ground in one structure (*table_name*).

This five structures allow all the types of queries in the image and also by storing the rule bases the right step is taken towards automation of the monitoring.

7.3 Summary of the GEOBIA system

Only ideas of the GEOBIA system are presented here to encourage some future work in the direction of a GEOBIA centric system instead of restricting GEOBIA to be a classification methodology. The fact that all of the components described in the system are in direct use already should be a motivating factor to concentrate on designing the standards of such a system. Moreover, with the new methodologies such as the class dependant neural networks (see Sec. 6.4), the updating of the rule bases and hence increasing the classification accuracy can become more practical. The methods such IRMAD (see Sec. 3.6) can be used to identify changes and the no change entities could be used as samples to modify the existing rule bases to create more adaptive rule base structure. For all the queries, the existing technology in the field of GIS is well advanced and this could easily be integrated in to the system. The future task would then only be to design much more robust and effective data structures to deal with any kind of data and application.

Chapter 8

Conclusion and future work

This thesis made an attempt to provide an overview of the author's contributions to GEOBIA while briefly explaining the underlying concepts of GEOBIA. Several general purpose methodologies have been presented with direct implications to all the applications of OBIA. The three stages of the OBIA are parallely studied so as to maintain the coherence to develop algorithms for the preceding stage with an overview of problems of the later stage. While, this style is an advantage in the expected sense but it has also restricted the regular order of application of the methods to all the classification examples presented in this thesis. For example, the new segmentation algorithm has been developed in the end to overcome several problems with the multi-resolution segmentation of Definiens and hence no examples used the algorithm. However based on the evaluation, it seems to be better than the multi-resolution segmentation algorithm. The work done to classify the nuclear facilities (see Sec. 6.5.1) was done prior to the development of the class dependant neural networks architecture (see Sec. 6.4). The effectiveness of the SEATH (see Sec. 6.2) in identifying the right features but not so accurate thresholds led to the idea of generating the rule bases automatically based on neural networks. The general observations and conclusions about all the contributions and an outlook for future research will be presented here first for every stage separately and then a general conclusion is drawn.

8.1 Pre-processing

Several pre-processing methods either to modify the data or to create new image layers to help in the segmentation process have been presented. Two new adaptive filters have been developed to increase the homogeneity of the objects. The algorithms are very effective but with the limitation of computational time. The improvement in the results of segmentation after filtering is a big motivation to develop more such filters. A method to identify suitable ratios (see Sec. 3.4.1) based on class samples has been presented but not used during the thesis. Such a transformation just based on the samples can be a better input for instance for the class dependant networks which can then model a fuzzy

probability of the classes. A focus on finding ways to effectively make a distinction between the class distributions based on samples is required apart from general transformations like principal components transformation. One good direction would be to work with kernel versions of the standard transformations based on samples. However, such methods require representative sample distributions of the classes with significant size.

8.2 Segmentation

A new segmentation algorithm has been developed to improve the results of segmentation based on graph theory (see Sec. 4.4. There are several advantages of the algorithm. It can effectively use different layers with different ranges. Also, it can be easily programmed to work with tiles of images and produce the same results even for images of larger size. The usage of the standard deviation to mean ratio as the homogeneity criteria is more close to human perception of homogeneous objects compared to the criterion used by several other algorithms which are either based on thresholding or standard deviation only, resulting in smaller objects in brighter regions and bigger objects in darker regions at a particular level of extraction. Though the algorithm is not tested for a variety of applications, the initial results suggest that it can perform well. The simple method to evaluate the segmentations where quartile values are considered instead of averaging is more comfortable to analyze the segmentation results based on reference segments (see Sec. 4.5. Finally, the algorithm to automatically segment the objects based on shape (see Sec. 4.6) is a particularly interesting application in a lot of cases as it gives the chance to refine the results of segmentation. The fact that it is less sensitive to shapes of the edges by means of using skeletons also makes it an effective algorithm. This algorithm is a direct consequence of the understanding of visual perception theories of David Marr and Rudolph Arnheim (2.1). OBIA, as it is presented now does not use the exact concepts of human perception. The future research should be more focused on bringing more concepts of the visual perception theories to build much better concepts of OBIA. Texture segmentation has not been considered in this thesis and this is one of the immediate problems to be solved in the future.

The fact that the quality of RS data is improving and more and more types of data are being available makes it necessity to develop more and more effective methodologies. The digital elevation model (DEM) or digital surface model (DEM) is one such information which has not been referred to in this thesis. This information can have several advantages. For example, it then becomes very easy to implement the Marr's computational approach for image understanding to group objects and prepare models similar to human visual perception (2.1).

8.3 Classification

Object-based classification has an improved basis compared to pixel-based methods as they use shape, texture and context information apart from the color information. How-

ever, this extra information has to be carefully handled and the right features have to chosen. The SEATH method (see Sec. 6.2) explained in this thesis seems to be an effective approach to identify the features distinguishing between classes. A new architecture based on neural networks named class dependant neural networks architecture (see Sec. 6.4) has been developed specially to deal with the object features. The results of the examples suggest that it is an effective architecture for object-based classification provided sufficient training samples are available (more than 5). Moreover, a single class dependant network can be used in the sequential classification and at multiple levels of segmentation. This can be a a very important method for most of the object-based classification applications. The algorithms are provided as general methodologies with an explanation of the theoretical background and hence a lot of examples were not provided. An example of how classification rule bases can be transfered from one time to another by radiometrically normalizing the images with each other is a promising way for the applications involving continuous monitoring (see Sec. 6.5.1). However the classification accuracies can reduce with time if the same rule base is used for longer time periods. One solution to solve this problem is again to use the IRMAD transformation to identify no change objects. This no-change objects can be used as samples to update the available rule bases. Using the no-change objects with the class dependant neural networks can be a very efficient way to update the rule bases and hence obtain a very good classification automatically.

8.4 Conclusion

The current status of GEOBIA is based on the basic concepts of human visual perception but still deviates a lot directly as early as in the segmentation step. A careful thought has to be given to focus more on building models close to human perception. The fact that more detailed height information is becoming available with improvements in technology will be a big advantage for object based image analysis as we can then have much better segmentation results based on height information. The opportunity of having 3D models will for sure help in getting close to generate classification results comparable to human perception. All the contributions presented in this thesis including the adaptive filters for pre-processing, segmentation using the graphs, shape segmentation, segmentation evaluation and class dependant neural networks are very promising. Finally, a reference is again made to the proposed GEOBIA system architecture (see chapter 7). GEOBIA has more to do than just classification. That would be the prime conclusion of this thesis. This architecture can be a very important step for the future of GEOBIA where all the steps related to application of RS imagery could be handled by one system. Since, object-based classification is already proven to be a better methodology compared to that of the pixel-based methods, the design of the entire system around this methodology is obviously a necessity.

Bibliography

[1] Aiazzi B., L. Alparone, S. Baronti and A. Garzelli, *Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution analysis.* IEEE transactions on Geoscience and Remote Sensing, 40(10), 2300-2312, 2002

[2] Anderson T.W., *An Introduction to Multivariate Statistical Analysis.* 2nd ed, John Wiley, New York, 1984.

[3] Arnheim R.,*Art and visual perception: A psychology of the creative eye.* University of California Press, Berkeley, 1954.

[4] Baatz, M. and A. Schaepe, *Object-Oriented and Multi-Scale Image Analysis in Semantic Networks.* In: Proc. of the 2nd International Symposium on Operationalization of Remote Sensing, Enschede, ITC, August 1620, 1999.

[5] Baatz, M. and A. Schpe,*Multiresolution segmentation -an optimization approach for high quality multi-scale image segmentation.* Angewandte Geographische Informationsverarbeitung XII, Beitrge zum AGIT-Symposium Salzburg 2000, pp. 12-23. Herbert Wichmann Verlag, Karlsruhe, 2000.

[6] Baatz M., C. Hoffmann and G. Willhauck, *Progressing from object-based to object-oriented image analysis.* In T. Blaschke, S. Lang and G.J. Hay (Eds) Object-based Image Analysis- Spatial Concepts for Knowledge-driven remote sensing applications. Springer-Verlag, Berlin, 2008.

[7] Beucher S. and F. Meyer, *The morphological approach to segmentation: the watershed transformation.* In Mathematical Morphology in Image Processing (Ed. E.R. Dougherty), pages 433-481, 1993.

[8] Bhattacharyya A., On a measure of divergence between two statistical populations defined by probability distributions, *Bull. Calcutta Math. Soc.* , 35, pp. 99-109, 1943.

[9] Bishop C.M., *Neural networks for pattern recognition.* Oxford University Press, Oxford, UK, 1995.

[10] Bishop C.M., *Pattern Recognition and Machine Learning.* Springer, Berlin, 2006.

[11] Blaschke T., S. Lang, E. Lorup, J. Strobl and P. Zeil, *Object-oriented image processing in an integrated GIS/remote sensing environment and perspectives for environmental applications.* In A.Cremers and K.Greve (Hrsg.): Umweltinformation fr Planung, Politik und ffentlichkeit / Environmental Information for Planning, Politics and the Public. Metropolis Verlag, Marburg, Vol 2: 555-570, 2000.

[12] Blaschke T. and S. Lang,*Object based image analysis for automated information extraction a synthesis.* In Proc. of Measuring the Earth II ASPRS Fall Conference, San Antonio, Texas, November 6-10 2006.

[13] Buck A., R. De Kok, T. Schneider and U. Ammer,*Improvement of a forest GIS by integration of remote sensing data for the observation and inventory of protective forests in the Bavarian Alps.* In Proc. IUFRO Conference on Remote Sensing and Forest Monitoring, Rogow, Poland, June 1-3, 1999.

[14] Canty M.J., A.A. Nielsen and M. Schmidt, *Automatic radiometric normalization of multispectral imagery.* Remote Sensing of Environment 91(3,4) 441-451, 2004.

[15] Canty M.J.,*Image Analysis, Classification and Change Detection in Remote Sensing, With Algorithms for ENVI/IDL.* Taylor & Francis, CRC Press, 2007.

[16] Canty M.J. and A.A. Nielsen, *Automatic radiometric normalization of multitemporal satellite imagery with the iteratively re-weighted MAD transformation.* Remote Sensing of Environment. 112(3), 1025-1036, 2008.

[17] Canty M.J.,*Boosting a fast neural network for supervised land cover classication.* Computers and Geosciences, doi:10.1016j.cageo.20 08.07.0 04, 2009.

[18] Canty M.J., *ENVI extensions for image analysis, classification and change detection in remote sensing.* 2009b. URL:$http://mcanty.homepage.t-online.de/$

[19] A.P. Carleer, O. Debeir and E. Wolff, *Assessment of very high spatial resolution satellite image segmentations.* Photogramm Eng Rem Sens 71(11):1285-1294

[20] Carr J.R.,*Numerical Analysis for the Geological Sciences*: Prentice-Hall, Inc;NJ, 1995.

[21] Castilla G., *Object-oriented analysis of remote sensing images for landcover mapping: conceptual foundations and a segmentation method to derive a baseline partition for classification.* PhD thesis, Polytechnic University of Madrid, URL: $http://oa.upm.es/133/01/07200302_castilla_castellano.pdf$

[22] Castilla G. and G.J. Hay, *Image objects and geographic objects.* In T. Blaschke, S. Lang and G.J. Hay (Eds) Object-based Image Analysis- Spatial Concepts for Knowledge-driven remote sensing applications. Springer-Verlag, Berlin, 2008.

[23] Cook R., I. McConnell, D. Stewart and C.J. Oliver, *Segmentation and simulated annealing.* In: G. Franceschetti, F.S. Rubertone, C.J. Oliver and S. Talbakhsh (Eds) Microwave sensing and synthetic apperture radar. Proc. SPIE2958:30-35, 1996.

[24] Coppin P., I. Jonckheere, K. Nackaerts, B. Muys and E. Lambin, *Digital change detection methods in ecosystem monitoring: a review.* Int. Journal of Rem. Sens., Vol. 25, No. 9, 1565-1596, 2004.

[25] Costa G.A.O.P., R.Q. Feitosa, T.B. Cazes and B. Feijo, *Genetic adaptation of segmentation parameters* In T. Blaschke, S. Lang and G.J. Hay (Eds) Object-based Image Analysis- Spatial Concepts for Knowledge-driven remote sensing applications. Springer-Verlag, Berlin, 2008.

[26] Daneke. C., *Automatisierte Analyse von Fernerkundungsdaten für umfangreiche Monitoring-Aufgaben im Rahmen nuklearer Safeguards* Diplomarbeit, Universität Wien, 2008.

[27] *Definiens Professional* Software reference guide, 2007. URL:*www.definiens.com*

[28] De Kok R., T. Schneider and U. Ammer, *Object based classification and applications in the Alpine forest environment.* In Proc. Joint ISPRS/EARSeL Workshop, Fusion of sensor data, knowledge sources and algorithms, Valladolid, Spain, June 3-4, 1999.

[29] Duda R.O., P.E. Hart, and D.G. Stork, *Pattern Classification, Second edition.* John Wiley & sons, pg 556-557, 2001.

[30] Felzenszwalb P. and D. Huttenlocherm, *Efficient graph-based image segmentation.* Int. Journal of Computer Vision 59, 2, 167–181, 2004.

[31] Flusser J. and T. Suk, *Pattern recognition by afne moment invariants.* Pattern Recognition, 26(1):167174, 1993.

[32] Foody G.M. and A. Mathur.*A relative evaluation of multiclass image classification by support vector machines.* IEEE Trans. on Geoscience and Remote sensing, Vol. 42, No. 6, June 2004.

[33] Gao Y., J.F. Mas, I. Niemeyer, P.R. Marpu and J.L. Palacio, *Objectbased image analysis for mapping landcover in a forest.* In Proceedings of the 5th ISPRS International Symposium on Spatial Data Quality, 2007.

[34] Gonzalez R.C. and R.E. Woods, *Digital image processing.* 2nd edition, Pearson education, Singapore, 2002.

[35] Gordon I.E., *Theories of Visual Perception.* Psychology Press, Hove, England, 2004.

[36] Green A.A., M. Berman, P. Switzer, and M.D. Craig,*A transformation for ordering multispectral data in terms of image quality with implications for noise removal.* IEEE Transactions on Geoscience and Remote Sensing, 26(1), 6574, 1988.

[37] Haralick R., K. Shanmugam and I. Dinstein,*Textural features for image classication.* IEE Transactions on Systems, Man and Cybernetics, 3(6):610621, 1973.

[38] Hay G.J. and G. Castilla, *Geographic Object-based Image Analysis (GEOBIA): Anew name for a new discipline.* In T. Blaschke, S. Lang and G.J. Hay (Eds) Object-based Image Analysis− Spatial Concepts for Knowledge-driven remote sensing applications. Springer-Verlag, Berlin, 2008.

[39] Hotelling H., *Analysis of a complex of statistical variables into principal components.* Journal of Educ. Psych., 24, 417441. 1933

[40] Howe D., *The free on-line dictionary of computing*, 1995. URL: $http://foldoc.org/$

[41] Hu M.K., *Visual pattern recognition by invariant moments.* IRE Transactions on Information Theory, 8(2): 179-187, 1962.

[42] Huang C., L.S. Davis and J.R.G. Townshend, *An assessment of support vector machines for land cover classification.* Intl. Journ. of Rem. Sens, vol.23, no. 4, pp.725-749, 2002.

[43] Kalman R.E., *A new approach to linear filtering and prediction problems.* Journal of Basic Engineering 82 (1): 3545, 1960.

[44] Kristinsdottir B., *Implications of Moment Invariants for Texture, Segmentation and Classication.* Master thesis, Institute for Mine-surveying and Geodesy, Freiberg University of Mining and Technology, 2008.

[45] Lang S. and T. Blaschke, *Bridging remote sensing and GIS - What are the main supporting pillars?.* International Archives of Photogrammetry, remote sensing and Spatial Information Sciences. Vol. XXXVI-4/C42, 2006.

[46] Laben C.A., V. Bernard, and W. Brower,*Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening.* US Patent 6,011,875, 2000.

[47] Lewinski S., *Applying fused multispectral and panchromatic data of Landsat ETM+ to object oriented classification.* In Proccedings of the 26th EARSeL Symposium, New Developments and Challenges in Remote Sensing, Warsaw, Poland, May 29-June 2, 2006.

[48] Neubert M., H. Herold and G. Meinel, *Evaluation of remote sensing image segmentation quality.* In T. Blaschke, S. Lang and G.J. Hay (Eds) Object-based Image Analysis-Spatial Concepts for Knowledge-driven remote sensing applications. Springer-Verlag, Berlin, 2008.

[49] Nielsen A.A., *The regularized iteratively reweighted MAD method for change detection in multi- and hyperspectral data.* IEEE Transactions on Image Processing. Vol. 16 No. 2 pp. 463-478, 2007

[50] Niemeyer I., M.J. Canty and M. Baatz, *Fractal-hierarchical Pattern Recognition for Safeguard Purposes.* In Proceedings of the 2 nd International Symposium on Operationalization of Remote Sensing, Enschede, ITC, August 1620, 1999.

[51] Niemeyer, I., S. Nussbaum and I. Lingenfelder,*Automated analysis of remote sensing data for extensive monitoring tasks in the context of nuclear safeguards*. Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 2005 (IGARSS 05). Volume 3, Issue 25-29 July 2005 Page(s): 2137-2140, 2005.

[52] Nussbaum, S., I. Niemeyer and M.J. Canty, Feature Recognition in the Context of automated Object-Oriented Analysis of Remote Sensing Data monitoring the Iranian Nuclear Sites, *Proceedings of Optics/Photonics in Security & Defence*, SPIE , 2005.

[53] Nussbaum S. and G.Menz, *Object-based image analysis and treaty verification: New Approaches in Remote Sensing-Applied to Nuclear Facilities in Iran*. Springer Verlag, 2008.

[54] Nùnez J., X. Otazu, O. Fors, A. Prades, V. Pala, R. Arbiol, *Multiresolution-based image fusion with additive wavelet decomposition.* Geoscience and Remote Sensing, IEEE Transactions on, Volume 37, Issue 3, Page(s):1204 - 1211, 1999.

[55] Richards J.A. and X. Jia, *Remote Sensing Digital Image Analysis*, 3rd ed., Springer-Verlag, Berlin, 1999.

[56] Ranchin T. and L.Wald, *Fusion of high spatial and spectral resolution images: the ARSIS concept and its implementation.* Photogrammetric Engineering and Remote Sensing, 66(1), 49-61, 2000.

[57] Schiewe J., L. Tufte and M. Ehlers. *Potential and problems of multi-scale segmentation methods in remote sensing.* In Proc. of GeoBIT/GIS 6: 34-39, 2001.

[58] Schowengerdt R.A., *Remote Sensing: Models and Methods for Image Processing*. 3rd ed., Elsevier Inc., 2007.

[59] Shah S. and F. Palmieri, *MekaA fast, local algorithm for training feed forward neural networks.* In: Proceedings of the International Joint Conference on Neural Networks, San Diego, USA, vol. I, no. 3, pp. 4146, 1990.

[60] Shi J. and J. Malik *Normalized Cuts and Image Segmentation*, IEEE Conference on Computer Vision and Pattern Recognition, pp 731-737, 1997.

[61] Short N.M., *NASA Remote sensing tutorial,* 2009 URL:$http://rst.gsfc.nasa.go$

[62] Sim D., H. Kim, and R. Park,*Invariant texture retrieval using modied zernike moments.* Image and Vision Computing, 22:331U342, 2004.

[63] Stoney W.E., *ASPRS Guide to Land Imaging Satellites* ,2008. URL: $http://www.asprs.org/news/satellites/satellites.html$

[64] Teague M.,*Image analysis via the general theory of moments.* Journal of the Optical Society of America, 70(8):920930, 1980

[65] Treisman A. and G. Gelade, *A feature integration theory of attention.* Cognitive Psychology, 12, 97-136, 1980.

[66] Tu Z. and S. Zhu, *Image segmentation by sata driven markov chain monte carlo.* IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.24, No.5, 2002

[67] Wang Z. and A.C. Bovik, *A universal image quality index.* IEEE Signal processing Letters, 9(3), 81-84, 2002

[68] Wilson R.A.and F.C. Keil (Eds), *The MIT Encyclopedia of the Cognitive Sciences (MITECS).* 2nd edition, The MIT Press, 2001.

[69] Wu Z., and R. Leahy, *An optimal graph theoretic approach to data clustering: theory andits application to image segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 15, Issue 11, 1101-1113, 1993.

[70] Yocky, D.A., *Multiresolution wavelet decomposition image merger of landsat thematic mapper and spot panchromatic data. Photogramm.* Eng. Remote Sens. v62 i9. 1067-1074, 1996.

[71] Zhang H., J.A. Frits and S.A. Goldman,, *An entropy-based objective segmentation evaluation method for image segmentation.* SPIE Electronic Imaging- Storage and Retieval Methods and Applications for Multimedia: 38-49, 2004.

[72] Zhang H., J.A. Frits and S.A. Goldman, *A co-evaluation framework for improving segmentation evaluation.* SPI Signal Processing and Target Recognition XIV: 420-430, 2005.

Appendix A

List of Abbreviations

ATWT	A- trous WaveletTransform
CCD	Charge Coupled Devices
DEM	Digital Elevation Model
DFT	Discrete Fourier Transform
DN	Digital Number
DWT	Discrete Wavelet Transformation
EM	Electromagnetic
EO	Earth Observation
ERD	Entity Representation Diagram
FIT	Feature Integration Theory
FOV	Field of View
GEOBIA	Geographic Object-Based Image Analysis
GIS	Geographic Information System
GLCM	Gray-Level Co-occurence Matrix
GMOSS	Global Monitoring for Security and Stability
GUI	Graphical User Interface
IFOV	Instantaneous Field of View
IR-MAD	Iteratively Reweighted Multivariate Alteration Detection
OBIA	Object-based Image Analysis
MAD	Multivariate Alteration Detection
MAF	Minimum/Maximum Autocorrelation Factor
MNF	Minimum Noise Fraction
MRS	Multi-Resolution Segmentation
MSS	Multispectral Scanner System
NDVI	Normalized Difference Vegetation Index
NIR	Near Infrared
PC	Principal Components
PCA	Principal Components Analysis

RS	Remote Sensing
PSF	Point Spread Function
RDBMS	Relational Database Management System
SMR	Standard Deviation to Mean Ratio
SNR	Signal to-Noise Ratio
SVM	Support Vector Machines
SWIR	Shortwave Infrared
VNIR	Visual Near Infrared