UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Automatic Road Network Extraction from High Resolution Satellite Imagery using Spectral Classification Methods

by

Andries Carl Hauptfleisch

Submitted in partial fulfilment of the requirements for the degree Magister Scientae in the Faculty of Engineering, Built Environment and Information Technology

University of Pretoria

February 2010

Abstract

Road networks play an important role in a number of geospatial applications, such as cartographic, infrastructure planning and traffic routing software. Automatic and semi-automatic road network extraction techniques have significantly increased the extraction rate of road networks. Automated processes still yield some erroneous and incomplete results and costly human intervention is still required to evaluate results and correct errors. With the aim of improving the accuracy of road extraction systems, three objectives are defined in this thesis: Firstly, the study seeks to develop a flexible semi-automated road extraction system, capable of extracting roads from QuickBird satellite imagery. The second objective is to integrate a variety of algorithms within the road network extraction system. The benefits of using each of these algorithms within the proposed road extraction system, is illustrated. Finally, a fully automated system is proposed by incorporating a number of the algorithms investigated throughout the thesis.

Keywords: automated road network extraction, remote sensing analysis, feature extraction, edge detection, segmentation, spectral classification.

Supervisor: Prof. A. P. Engelbrecht
Co-supervisor: Dr. F. van den Bergh
Department of Computer Science
Degree: Magister Scientiae

Acknowledgements

I would like to thank my family and friends for their continued support. They provided the foundation needed to ultimately pursue further studies.

I appreciate the guidance and direction offered by my supervisors Dr. Frans van den Bergh and Prof. Andries Engelbrecht. It was an exciting journey and I received the necessary guidance and direction to complete this thesis, while still having the space to pursue a number of avenues.

Finally, I would also like to thank, the Strategic Research Panel (SRP) for providing support for this research through the CSIR GenDySI (Generation and Harnessing of Dynamic Spatial Intelligence) project.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

"The important thing is not to stop questioning."

— Albert Einstein

Roads have played a central role in the lives of man since the beginning of time. Before the invention of sea, air and rail travel, roads were the solitary means of transporting goods and people from one location to another. Even in today's modern society, roads remain one of the mediums used most frequently for travel and transportation. In the European Union, roads are used to transport 44% of all goods and 85% of all people [132].

In light of the pivotal role roads play in our daily lives, information pertaining to the location of roads becomes essential. This information not only allows humans to make informed decisions regarding their environment in general, but also increases efficiency in the choice of routes for transporting goods and people.

At present, information regarding road locations and their characteristics is stored

digitally within geographic databases. This digital representation is flexible enough to enable numerous Geographic Information Systems (GIS) to use the road data. Road data enables GIS applications to facilitate a variety of services which include satellite navigation, route planning, transportation system modelling [97], health care accessibility planning [144], land cover classification [134] and even infrastructure management [6].

Two methods are typically used to obtain road data sets, namely ground surveying and delineating roads from Remotely Sensed (RS) imagery [174]. Ground surveying can be conducted by using devices such as receivers for the Total Station or the Global Positioning System (GPS). Delineating roads from remotely sensed imagery is known as *road extraction* and can be categorized as being a manual, semi-automated or fully automated process. Manual extraction entails a human operator delineating roads from remotely sensed imagery, while semi-automated extraction requires some human input to guide a set of automated processes. Finally, the automated process requires no human input.

Fully automated road extraction systems comprise a variety of algorithms, which can be roughly classified into three levels of processing [114]:

- the low-level operations that work with the raw image data,
- the medium-level processes that further refine the information gathered by the low-level algorithms, and
- the high-level algorithms that produce the final road networks.

The higher-level algorithms exhibit aspects of intelligence in their ability to reason on the structure and location of road networks in a fashion similar to humans.

The goal of this thesis is threefold: to develop a flexible road extraction system, to test various algorithms within this system and finally to propose a novel road extraction system. In addition to these goals, a comprehensive survey of the literature on road extraction is also conducted. In the remainder of the chapter the problem statement is formalized, motivation for this study is provided, concrete objectives are set, possible contributions are considered and a study outline is given.

1.1 Problem statement

Ground surveying is an extremely time consuming process since a surveyor has to obtain a series of physical ground measurements along the route being extracted. As it is considered less resource intensive, extracting roads from remotely sensed imagery could result in reduced costs and data acquisition time.

In general, humans are able to detect roads and paths in remotely sensed imagery with relative ease. Even partially occluded routes can, in most instances, be estimated with a high level of accuracy. Although road extraction is such a trivial task for humans, automating this process with feature extraction techniques remains a challenging problem.

In this thesis, a full automation is considered as system capable of completing an assignment without the need of an operator during any stage of processing. The system parameters should either be predefined constant values or determined dynamically by the system itself.

Digitally processing remotely sensed data with the intent of retrieving or enhancing man-made features dates back to the mid-1970s [81, 133]. In recent years, the field of feature extraction, and in particular Road Network Extraction (RNE), has become a very active research area, which is notable in a recent road extraction survey by Mena [114] where more than 250 articles are referenced. The current automated state-of-the-art systems [67, 76, 115, 177] manage an extraction quality of approximately 70%, with the quality of manual extraction remaining vastly superior. It is thus apparent that a clearer understanding of the manner in which Human Vision Systems (HVS) function is required.

1.2 Motivation

As established above, roads are a critical component in our daily lives. Knowledge regarding their location and characteristics allows utilization of this valuable resource efficiently and effectively.

Although humans posses the ability to extract roads from remotely sensed imagery with fairly high accuracy, the process remains slow and tedious. With the dawn of the information age, computers provided the ability to automate various functions at high speeds. Automating the road extraction process will also reduce road extraction times and thus reduce the costs of creating new maps. In a study by Shen *et al.* [143],

the extraction process of a semi-automated system was 30.03% faster than manual extraction.

In addition to creating new maps, automation of the road extraction process will also assist in updating existing road data sets. Brown [20] states that map products struggle to maintain currency given the rapid pace of development. This scenario is especially relevant in rapidly developing countries where high economic growth rates are prevalent. Economic growth can be translated to the expansion of infrastructure, which includes road networks. Existing road data sets become outdated rapidly and thus lose their relevance. It is therefore imperative that existing data sets are updated on a continuous basis.

Road extraction research deals not only with the issues of extracting roads, but also touches on a number of fundamental issues in Digital Image Processing (DIP) and HVS. Research in this area allows the scientific community to develop a clearer understanding of the manner in which humans are able to combine their cognitive strengths with their vision systems. These issues are highly relevant in our search to understand the way in which we function.

1.3 Objectives

The objective of this thesis is threefold. Firstly, a flexible automatic road extraction system is developed with the intent of extracting road networks with a high level of accuracy. This system comprises an image processing chain that includes a number of algorithms, ranging from the lowest to the highest level of processing.

The second objective is to develop a clearer understanding of the areas in which road extraction systems suffer the most. This is achieved by testing a variety of algorithms at different stages within the processing chain. The outcome serves as an indication of the areas on which future research could focus.

Finally, this study proposes a novel fully automated road extraction system with the intent of producing road data sets of high quality.

1.4 Contribution

The main contributions to the scientific community can be summarized as:

- A generic road extraction system is presented. The system is flexible and allows

various configurations to be tested.

- Various new approaches are tested within the extraction system. The results are presented and discussed.
- Through the development of this system and the testing of algorithms, a clearer understanding of road extraction is obtained.
- The generic system is fully automated, requiring no human intervention during processing.
- A thorough literature review of the current state-of-the-art is presented.

1.5 Thesis outline

Chapter 2 presents the background for this study. A brief introduction and history on road network extraction is provided. This is followed by a thorough literature review on road extraction techniques, quality metrics and popular data fusion methods. Finally, the different data types and colour spaces are discussed briefly. Chapter 3 describes the techniques used in this study in greater detail and presents a generic road extraction system. Chapter 4 investigates the possibility of using a long range edge detector over the well known Canny approach. Chapter 5 explores the prospect of improving road extraction results by using segmentation. Chapter 6 combines components from chapters 4 and 5 to develop a novel fully automated system. Finally, the thesis concludes in Chapter 7.

CHAPTER 2

Background and literature review

"I not only use all the brains that I have, but all that I can borrow."

— Thomas Woodrow Wilson

The process whereby roads are extracted from remotely sensed imagery is rather elaborate and involves a number of techniques from numerous fields within computer science. The aim of this chapter is twofold: firstly, it provides background information regarding the fields involved in the extraction process: secondly, it offers a literature review of the RNE techniques used in literature. This review is by no means exhaustive and serves only as an introduction to commonly used RNE techniques, with references to some of the more noteworthy works in the field.

Sections 2.1 and 2.2 comprise an introduction and brief history of road mapping. Section 2.3 offers a review of lower-level road extraction algorithms, while Section 2.5 focuses on some of the higher-level techniques. Data fusion and its relevance to RNE are discussed in Section 2.4, while Section 2.6 provides an overview of the various metrics

developed to determine the accuracy of extracted road networks. Section 2.8 describes briefly the various satellite data types available. The chapter concludes in Section 2.9.

2.1 Introduction

The digital processing of remotely sensed data, with the intent of retrieving or enhancing man-made features, dates back to the mid-1970s [81, 133]. The field of feature extraction, and in particular RNE, has since matured significantly. As mentioned in the opening chapter, a 2003 review conducted by Mena [114] cites more than 250 RNE studies, which serves as an indication of the vast amount of work already conducted in this field. Owing to the sheer volume of studies involved, Mena provides an overview of the various extraction methods rather than a detailed description. Mena organizes the studies into various categories, according to technique and the level of processing.

Although the survey by Mena is used as a basis for this chapter, a stronger emphasis is placed on the description of the RNE techniques, and the manner in which they are employed. An attempt is made to discuss recent studies, which serves as an update to the work by Mena. The categorization method employed in the review by Mena is followed to a large extent, but it has been restructured in this study.

2.2 Brief history

The technologies used to create road maps have evolved significantly from their humble origins. The exact dates of the first cartographic maps are unclear, but the early Babylonian maps date back to approximately the 14th-12th century BC [58]. Subsequent maps created by the ancient Greeks and Romans date back to the 6th century BC. These early maps were created by combining mathematical techniques with observations made by explorers.

Technological advances have changed the way in which modern cartographic maps are created. Two methods are currently used to obtain map information [174]. The first involves ground surveying where data is collected using traditional methods, such as total station, or through modern approaches, such as mobile mapping systems. Ground surveying requires a team of surveyors to go into the field physically and record the location of road structures. The process is time-consuming and costly. The second approach collects data on road positions by identifying them in RS imagery. The state-

of-the-art production road extraction systems are semi-automated, which implies that the system is driven by a human operator with elements of automation. Extracting roads from RS imagery holds some advantages over ground surveying. Even though extraction from RS imagery is still time-consuming, it is less intensive than ground surveying and has lower skills requirements. Extracting roads from RS imagery might require Very High Resolution (VHR) imagery, which could be quite costly.

The current operational cartographic RNE systems rely heavily on human input. Some systems provide a level of automation, but human operators still drive the process to a large extent [99]. Achieving automation in the field of RNE will greatly reduce the time and labour required. The knowledge obtained could also be made more general and applied to other RS and even HVS problems.

2.3 Road extraction techniques

In the introduction (refer to Section 2.1), it was mentioned that a great number of techniques for road extraction are available. Categorizing these methods will be helpful in adding structure to the literature review. The classification by Mena [114] forms the basis of the following categories used in the subsections:

- Automatic seeding (refer to Section 2.3.1)
- Classification (refer to Section 2.3.2)
- Edge detection (refer to Section 2.3.3)
- Hough transform (refer to Section 2.3.4)
- Mathematical morphology and filtrate (refer to Section 2.3.5)
- Multi-resolution analysis (refer to Section 2.3.6)
- Road tracking (refer to Section 2.3.7)
- Segmentation (refer to Section 2.3.8)
- Snakes (refer to Section 2.3.9)

As mentioned in Chapter 1, it is common practice in feature extraction literature to refer to the processing level of an algorithm as being low, medium or high. This indefinite

categorization typically illustrates three properties of an algorithm, namely the input data it receives, the type of computational operations it performs, and its position within an image processing chain. The hypothetical system shown in Figure 2.1 illustrates how various components can be arranged into a processing chain capable of detecting roads. As an example, the area opening operation conducted in the *Pre-Processing* step operates at a low level. The *Data Fusion* component can be considered as a medium-level operation, while the *Network Completion* step functions at a high processing level. The following subsections discuss road extraction approaches in each category and refer back to this hypothetical system to put the algorithms in context.

2.3.1 Automatic seeding

In the context of road extraction, seeding is the process whereby a marker is placed at certain points of interest within a road network. These points of interest can include markers along the centre of the road, points of high curvature, or intersections. The seeds are typically single points but can also be centreline segments or road regions. Seeding is not an extraction technique itself, but the markers are used as initialization points for extraction techniques, such as road trackers and snakes (refer to Sections 2.3.7 and 2.3.9). The seeds can also be used to generate road models (pattern classes), which can be used to train classifiers used to detect road objects in imagery. In addition, road network construction algorithms can also use seeds to connect the points using a high-level algorithm, such as presented in Section 2.5.2.

A number of semi-automated RNE approaches require a human operator to specify the seed points before a higher-level operation can continue with the extraction process. Automating the seeding process reduces the total extraction time of such a system significantly. According to Harvey [60], the performance of automatic road tracking algorithms depends to a large extent on the quality of the starting points. Apart from automating the process, the quality of the seed points is consequently also imperative.

Considering Figure 2.1, automatic seeding forms part of the *Geometrical Detector*, which receives its input from the *Edge Detector* component. In this instance, automatic seeding operates at a lower processing level. The following section explores established, as well as recent novel approaches to automatic seeding.

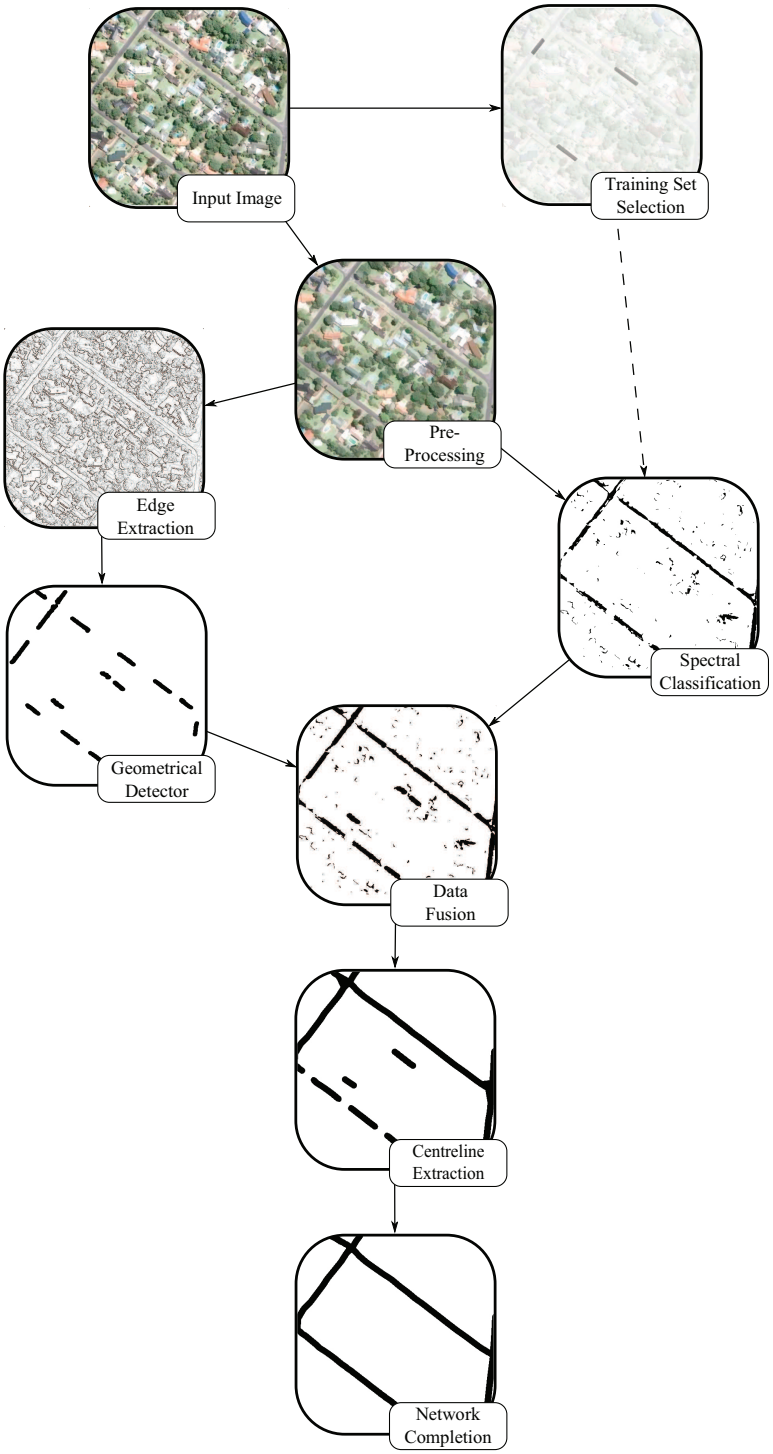


Figure 2.1: A hypothetical road extraction system

Automatic seeding techniques

A wide variety of techniques can be employed to detect road seeds automatically. Some of the techniques include parallel edge detection, geometric template matching, segmentation, Hough transformation and spectral classification. Automatic seeding algorithms can be categorized as low to medium-level processing techniques, as they typically receive raw image data or output from a low-level algorithm as input.

One of the most popular approaches to automatic seeding is the detection of parallel edges in medium to high resolution imagery. Of the twenty papers surveyed for their automatic seeding approaches, fifteen used a variation of parallel edge detection. A typical parallel edge detection algorithm begins by detecting the roadsides, which can be obtained by using an edge detector. The edge detection algorithm uses methods, such as the Laplacian, Roberts or Sobel filter, to assign a gradient value to each edge pixel. Edges with similar orientations within a certain local neighbourhood are then considered to be parallel. Seed points or lines are formed by calculating the midpoint between the parallel lines, whereas seed segments are typically created by connecting the end points of the parallel lines to form a rectangle.

Doucette *et al.* [38] present a more advanced approach for detecting seeds automatically through parallel edge detection. The Anti-parallel Centreline Extraction (ACE) algorithm follows more stringent criteria when selecting parallel edges. As roads are depicted as either brighter or darker than the non-road areas, only road edges with opposing gradients are considered. Minimum and maximum road width thresholds are used to reduce the detection of spurious road seeds. The road width is determined by calculating the perpendicular distance between the parallel edges using the edge gradient values. Provisional centreline road seeds are placed halfway between the edges, but a number of isolated erroneous centreline seeds remain evident. Considering that the centreline seeds should form strings of connected pixels, a connected component filter is used to remove many of the isolated seeds. The constant width thresholds within ACE are a limiting factor, leaving the algorithm unable to adapt and detect roads of different widths. Relaxing the width constraints introduces a great number of incorrectly detected road centre points. Zhou *et al.* [178] present a method which automatically detects road widths, which could be incorporated within ACE. Baumgartner *et al.* [11], Tesser and Pavlidis [149], Cai *et al.* [21] and Mei *et al.* [111] all present parallel edge-based seed detectors of their own.

Another frequently used approach is seed segment detection through geometric model

matching. In general, the method starts by segmenting an image and selecting candidate road segments based on certain criteria, such as shape or spectral properties. The candidates are then compared to predefined road models and, the correctly matched profiles are selected as seeds.

Koutaki and Uchimura [90] present an automated seeding method based on geometric model matching, where road intersections are used as initial seed points in aerial photographic imagery. Please refer to Section 2.8 for more information on aerial photography. First, the Iterative Self-Organizing Data Analysis Technique (ISODATA) spectral classifier is used to cluster and classify road segments with the same approach as used by Zhang *et al.* [172, 171]. The classified image is subsequently thresholded into a binary image. A number of intersection models are created and a novel template matching technique is used to identify intersections. A road tracker is used to follow the road between the detected intersections. A seed point is created at the location of every step the road tracker takes.

Dal-Poz *et al.* [32] also use geometric information to automatically extract road seeds from aerial imagery. Their method is based on four road models and the connections between the models. Each of the models is a representation of a straight road section. A model consists of a set of polylines that represent the edges of the road. As in the work of Doucette *et al.* [38], the Canny edge detector is used to determine the road edges. From the extracted edges, the road models are used to detect short straight road sections. A seed point is created where a match for a road model is found.

Hui *et al.* [72] extract initial seed points by adapting a method developed by Dell'Aqua *et al.* [34] for Synthetic Aperture Radar (SAR) imagery to work on QuickBird data. Please refer to Section 2.8 for a description of these sensors. The method is based on the premise that urban roads are usually parallel with two main edge directions. First, the method determines the two main edge directions within a given scene. A directional filter is used to identify features with comparable gradients, which results in the creation of two filtered images. A Hough transform is used to detect straight lines within these filtered images and the lines are fused together as preliminary seed points. The classification system by Mena and Malpica [115] is used to evaluate the preliminary seed points and in the process, all non-road seed points are discarded.

Summary

Automatic seed point detection is a key element in the development of a fully-automated road extraction system as it provides initialization points for a number of algorithms. Increasing the accuracy of initial road points enhances the higher level process of constructing accurate road networks. Automatic seeding is largely dependent on the underlying DIP techniques, such as spectral classifiers and edge detectors.

2.3.2 Classification

A classifier is a type of an inference model that seeks to identify patterns in data [74]. Classifiers are typically used to categorize data into different pattern classes through labelling or by assigning probability values to data samples. These probability values indicate the likelihood of an observation belonging to a pattern class. Classifiers can also be used to extract features from RS imagery and are often used to identify, amongst others, vegetation types, land usage, man-made objects, and natural features.

Classification can be formalized as the process of applying decision functions to a set of unknown patterns. Consider the set $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ with n patterns and W pattern classes $\omega_1, \omega_2, \ldots, \omega_W$. The objective of the classifier is to find W decision functions $d_1(\mathbf{x}), d_2(\mathbf{x}), \ldots, d_W(\mathbf{x})$ that, if a pattern $\mathbf{x}$ belongs to class ω_i, then

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \qquad j = 1, 2, \ldots, W; j \neq i. \tag{2.1}$$

Put differently, the unknown pattern $\mathbf{x}$ would be labelled as being part of pattern class i if $d_i(\mathbf{x})$ yields the largest value of all decision functions. Instances where ties occur would have to be resolved arbitrarily. From this definition, it is evident that the classification accuracy is directly correlated to the ability of the decision function to represent the pattern classes. Decision functions can model the pattern classes more accurately when the classes are separable.

Classifiers can be used to categorize any type of data, which implies that they can be used at various levels in the road extraction processing chain. When considering Figure 2.1, the *Spectral Classification* component serves as an example where the spectral signal of road models (pattern class) is used to classify a pre-processed image. Lower level classifiers are typically used to extract roads according to their spectral, textural and geometrical properties. Global context classifiers and automatic pattern class generators are typically more complex system involving algorithms at various levels.

When creating road models (manually or automatically) there tends to be a degree of overlap between the various pattern classes [7]. This overlap is evident when comparing, for instance, the spectral signals of a dirt road with bare soil, or an urban road and an open parking lot. Additional information has to be added to or derived from the data to improve the separability between pattern classes. Increasing this separability is one of the fundamental problems in feature extraction.

Classifiers can be used to categorize any type of data, but are often used in RNE system to perform any of the following:

- Spectral classification
- Textural classification
- Geometrical classification
- Contextual classification
- Automatic pattern class generation

Classifiers can also be used in the generation of the final topologically sound road network. The final construction of topologically sound road networks is discussed in Section 2.5. Each of the abovementioned application areas are discussed in greater detail in the following sections.

Spectral classification

Spectral classifiers are used to compare the spectral signature of a pattern class (road model) to that of an unknown sample. Figure 2.2 illustrates an example of the output created by a spectral classifier, where brighter values indicate a higher probability of a pixel belonging to the road class. In this instance, the Mahalanobis distance metric was used to determine the separation between the image pixels and a predefined road class.

A major issue with classifiers relates to the overlap that occurs between features in RS imagery. Considering that various features might have the same spectral signals, the classifier would be unable to make a clear distinction. This limitation can be reduced by considering additional information such as texture, structure and context. Spectral classifiers are, as a result, often used to generate an initial classification, which is further filtered and extended by higher level processes.

Figure 2.2: An example of spectral classification through classical statistics

A number of different classification methods are used in literature and include, amongst others, *classical statistics*, *Artificial Neural Networks* (ANN) and *fuzzy classifiers*. The following sections discuss the manner in which these and other methods are employed for spectral classification.

Classical statistics: Classical statistics refers to statistical techniques whereby information is gathered and quantified through the use of probability theory. These techniques are often used in feature extraction systems for their simplicity and computational efficiency. Methods such as minimum distance matching, matching through correlation, and Maximum Likelihood (ML) classifiers are often used. A discussion of the application of some of these methods to RNE follows.

Oddo *et al.* [122] use an ML classifier [59] to obtain an interim rough road classification. The classification is done for a single pattern class, which is assumed to conform to a Gaussian probability function. This assumption of road structures having uniform spectral values is made for most of the spectral classifiers. The pattern class is defined by the two multivariate parameters: mean and variance. All unknown patterns in a given image are labelled according to their position on the probability curve. The labelled image is thresholded to produce a binary image containing *road* and *non-road* (background) labelled pixels. This interim classification is then passed on to higher level processes where the road network is extracted.

The methods presented by Mena and Malpica [115] are based on the same principles as those of Oddo *et al.*, but the way in which statistics are gathered and used to assign likelihood values to unknown patterns are somewhat different. Firstly, the statistical properties for a single data point are calculated in three different ways: the statistics

are derived from single pixel's spectral values, sets of pixels' spectral values and texture feature. Secondly, likelihood values are assigned to unknown patterns through a distance function rather than calculating their positions on a probability curve as in [122]. The distance function calculates the separation between an unknown pattern and the pattern class. The resulting value is then transformed into a pseudo-probability value through a normalization function. The approach by Mena and Malpica is discussed in greater detail in Section 3.5.

A histogram Bayesian classification approach is followed by Chen *et al.* [26]. As with the previous approaches, a road pattern class is created by selecting a set of pixels representing a typical road surface. A non-road pattern is also created from a selection of points next to the roads. The spectral distributions of these pattern classes are determined and the final classification is made with the following function

$$\frac{p_{\mathrm{r}}(\mathbf{x})}{p_{\mathrm{b}}(\mathbf{x})} \leq \theta. \tag{2.2}$$

Here $p_{\mathrm{r}}(\mathbf{x})$ is the likelihood of the pattern $\mathbf{x}$ being a road pixel and $p_{\mathrm{b}}(\mathbf{x})$ being a non-road pixel. After applying the threshold (θ) a binary image is created, where every pixel above θ is marked as a road pixel. θ is an application specific variable and is determined through Receiver Operating Characteristics (ROC) techniques [26].

Artificial neural networks: An ANN is a non-linear statistical model, which is based on biological neural networks [52]. The ANN model consists of a multitude of interconnected non-linear components, which are known as neurons. In general, ANNs are dynamic systems able to 'learn' by adapting their internal structures according to information that flows through the network. ANNs are distribution free operators, which means that they have a distinct advantage over classical statistical models, where the influence of the sources has to be determined beforehand [29].

Mokhtarzade and Zoej [118] propose an ANN road extraction system that classifies Very High Resolution (VHR) imagery such as that obtained by the QuickBird and IKONOS satellites [45] (refer to Section 2.8) . The study also considers several different spectral ANN classifier configurations with the intent of finding the optimal road pixel classification solution. Each configuration is presented with a training set of pattern classes and is trained with the well-known steepest gradient descent algorithm [16]. The ANN is tested with different input vectors and varying hidden layer sizes. The pattern classes considered for input include a single pixel, single pixel with normalized

distance value, 3×3 window and a 3×3 window with normalized distance values. All pattern classes are spectral and created from the RGB bands. Each structure has one output indicating whether the considered pixel is a road pixel or not. The study found that an input vector consisting of a 3×3 window with normalized distance values yields the best results. As for the hidden layer, the authors conclude that no more than ten neurons are required. Overall classification accuracies of 95.15% and 95.22% are achieved for IKONOS and QuickBird scenes respectively. This illustrates that the spectral characteristics of neighbouring pixels, as well as their distance from the road mean point, can increase the classification accuracy. Spectral ANN classifiers have also been used by Bhattacharyya and Parui [15], Fiset *et al.* [42], and Li *et al.* [98].

Fuzzy classifiers: Fuzzy classifiers are considered to be classifiers based on fuzzy logic or fuzzy sets [92], which are based on set theory where the elements have a varying degree of membership. Fuzzy logic is expressed in terms of two truth values (true and false) while fuzzy sets allow multi-valued logic. As with classical statistical and ANN classifiers, fuzzy classifiers derive information for their membership functions from predetermined pattern classes and use the information to classify unknown patterns accordingly.

Mohammadzadeh *et al.* [116] identify five pattern classes and define a representative spectral membership function for each class. The membership functions are constructed by calculating the mean and standard deviation values in each band of every pattern class. An estimation of the class contribution in each band is determined, and a fuzzy classification is made by executing a MIN and MAX operation. The probabilistic membership values are thresholded, resulting in a binary image that represents the road and non-road classes.

Shackleford and Davis [140] also use spectral fuzzy classifiers, but they detected a significant overlap between the spectral values of the pattern classes (*tree* and *grass*, *road* and *building*). To discriminate between these classes, a hierarchical fuzzy classifier is constructed. The hierarchical structure allows additional information, specific to each class, to be used. Textural information is used to distinguish between the *tree* and *grass* classes, and geometrical information for the *road* and *building* classes. Tuncer [151] follows the same approach and obtains better results by adding geometrical information to the spectrally derived membership functions.

Other classifiers: A Support Vector Machine (SVM) is used by Song and Civco [145] to classify pixels into either a road or non-road class according to their spectral

characteristics. SVMs were created by Vapnik and Cortes [30, 156] to address the well known optimization problem of generalization against overfitting. They addressed this problem by balancing the relationship between the accuracy on training data against the accuracy on validation data. SVMs are linearly separable by design, which makes them ideal for two class problems such as detecting the road and non-road class. A kernel function is required to represent non-linear separable functions. Song and Civco use a Gaussian based Radial Basis Function (RBF) as the kernel function. As with the classifiers in the preceding sections, pattern classes are selected to create a representative road pattern class, which is in turn used to train the classifier. To evaluate the SVM's accuracy, the results are compared with those of a Gaussian ML classifier. The SVM shows an improvement over the classical statistics approach [145]. Zheng *et al.* [177] present another approach where an SVM is used as a binary spectral classifier.

Patterson [124] uses the Classification and Regression Trees (CART) algorithm to construct a spectral classifier in the form of a decision tree. Each split in the tree is created by evaluating the spectral signatures of ground-type pattern classes. These classes include *tar*, *asphalt*, *soil*, *pine needle*, *maple leaf*, *cement* and *wood board*. Each split or node within the tree is based on a threshold value for a given spectral band. An advantage of the CART algorithm is its built-in cross-validation procedure that ensures that an optimal tree is created. This cross-validation procedure also prevents the decision tree from overfitting. After running the CART algorithm it was found that IKONOS bands two and three (green and red) contained the most valuable information and are the only two bands used for splitting the tree.

Textural classification

Textural classifiers endeavour to classify image regions according to their textural characteristics. To classify these regions, texture has to be mathematically quantified. Even though no formal definition for texture exists, descriptors such as smoothness, coarseness and regularity are often used [52]. One of the most celebrated works on texture analysis was written by Haralick [57] in which 14 textural features are defined. A later study by Ohanian and Dubes [123] reveals that three of these features are sufficient for texture classification, namely *angular second-moment*, *contrast* and *entropy*. As with most of the aforementioned spectral classifiers, texture classification begins by firstly defining pattern classes, which are then used to classify unknown patterns. Texture classification is often used to either detect the global context of a scene or to detect the road surface

itself and can be considered as a low-level algorithm.

Mena and Malpica [115] present a method based on texture analysis. The approach, initially proposed by Mao and Jain [104], is used to calculate the similarity between pairs of pixels with the aid of a texture cube. A texture cube is created by considering the 3×3 neighbourhood around a candidate pixel (unknown pattern). This neighbourhood constitutes the width and height of the cube, while the depth consists of the RGB bands. Co-occurrence matrices are constructed to interweave the spectral and textural information contained within the cube. The following Haralick features are used as texture descriptors: *correlation*, *energy*, *entropy*, *maximum probability*, *contrast*, and *inverse difference moment*. A distribution from the six features is created for both the pattern class and the unknown pattern (candidate pixel). The distance between the two distributions is computed with the Bhattacharyya distance function [13]. The resulting distance is transformed to represent a pseudo-probability of the likelihood of a pixel being a road pixel. The method is discussed in greater detail in Section 3.5. This approach is also followed in the works of Hui *et al.* [72] and Xiong [166]. Haralick features are also used by Cai *et al.* [21] to train a C4.5 decision tree and an SVM classifier to classify road structures.

Dial *et al.* [36] create a panchromatic texture filter that calculates the variability of the area around a sample pixel through an Angular Texture Signature (ATS). An ATS is calculated by rotating a rectangular window 360° around the sample pixel and calculating the variability in each window. A histogram is constructed, where each bin represents a step through the 360°, and the bin value is the variability in that direction. Two pieces of information are recorded, namely the number of valleys and the angle at which the valleys occur within the histogram. Since road structures tend to have a low variability along the long axis, two valleys, roughly 180° apart should be detected for a straight road section. The information gathered by ATS is then passed to a higher level process for further classification. ATS is a popular approach and is used in studies by Haverkamp [61], Keaton and Brokish [80], Gibson [49] and Zhang *et al.* [175], amongst others.

Geometrical classification

Geometrical classifiers detect roads based on distinctive structural characteristics. Roads appear as long lines in low to medium resolution imagery and as elongated homogeneous regions in VHR imagery. During a typical classification process, structural information

is derived from a raw image through a lower level algorithm. Edge detectors, segmentation algorithms or the Hough transform are often used to characterize the structural properties of objects within an image. The final classification is based on the derived data, and geometrical classifiers can be considered as low to medium-level algorithms.

Zhang and Couloigner [175] expanded on the ATS method (described in Section 2.3.2) by also considering the shape characteristics of the results obtained from the ATS operation. Four shape features are extracted from the ATS values namely, *mean*, *compactness*, *eccentricity*, and *direction*. The mean value is calculated as the mean variation in each direction, which serves as a measure of the number of objects around the centre pixel. The mean value should be larger for a parking lot pixel than a road pixel. The compactness serves as a measure of how circular or elongated the object is within which the centre pixel is located. Again road structures should be elongated whilst parking lots will have a circular shape. The eccentricity defines the location of the centre pixel in relation to the centroid of its parent object. Pixels on the border of an object will have greater values than those located near the centre. The direction feature is a type of maximum directional connected component measure, which is a count of the neighbouring pixels with the same maximum directional value. A fuzzy logic classifier uses these four shape features to identify road structures.

Steger [146, 147] states that roads appear as ridges or ravines when an image is viewed as an intensity terrain model in VHR images. This view of roads is similar to the approach of parallel edge detection, but rather than using an edge detector, Steger constructs road models as polynomial functions, which describe the ridges and ravines in the intensity terrain model. Interpolation is used to match the pattern class to the intensity terrain model. This approach has two distinct advantages in that it is possible to obtain sub-pixel accuracies, as well as information indicating the direction of the road. Hu and Tao [70] also use a model matching technique, but use a binary template matching function to detect the initial centrelines rather than interpolating polynomials. The template matching function offers a speed advantage at the cost of a reduced accuracy. Post-processing is also required to clean and filter the results.

In the approach by Jin and Davis [76], the line extractor by Steger is employed to detect road structures from IKONOS imagery of densely vegetated suburban areas. The Normalized Difference Vegetation Index (NDVI) is a measure of the amount of vegetative cover, and it is calculated for every pixel in the image. Unless occluded, road structures have low NDVI values and appear as ravines through the vegetated areas. The results from the line extractor by Steger are fused with segments obtained by the *k-means*

clustering algorithm [103] and the result is handed to a higher level operation to connect the final road network.

Contextual classification

Contextual information is used by a number of RNE systems to guide the extraction process. Hinz *et al.* [68] identify two types of contextual information as being *global* or *local*. Global contextual information is used to define regions where roads have similar characteristics, such as *curvature*, *width* and *colour*. It is possible to optimize extraction techniques for specific regions. In identifying the types of roads that appear in a given region, the optimal road extraction approach can be followed. Local contextual information refers to the objects that are often found on or in close proximity to road structures, such as *cars*, *road markings*, *buildings* or *tree lanes*. Identifying these structures could help to reinforce the existence of a road, particularly in instances where the road is occluded. A few contextual classification approaches are summarized in the following paragraphs.

Yang and Wang [167] propose a method for global contextual classification based on edge densities within a given class. Four contextual classes are defined as *urban*, *rural*, *montane*, and *hybrids of suburban and rural regions*. An edge density histogram model is created for each class by first applying an edge detector to a typical class scene. The histogram model is calculated by binning edges with similar angles. These class models are then used to classify a new candidate scene. The scene is divided into overlapping tiles an edge density histogram for each tile is created. The resulting histograms are classified according to the four defined classes. If a tile is found to contain a hybrid of road scenes the tile is subdivided and the process repeated. This iterative process continues until all tiles are classified or until a threshold for the minimum tile size is reached.

Baumgartner *et al.* [10] and Hinz *et al.* [67] use both local and global contextual information to guide the classification process. Their approach first determines whether a specific region is forested, urban or rural. Baumgartner *et al.* identify five local contextual objects, namely *driveways up to buildings*, *parallel features*, *shadows*, *vehicles* and *rural driveways*, to aid the extraction of roads in all three global contexts. Hinz *et al.* focus solely on urban areas and find that *shadows*, *buildings*, *general occlusions*, *vehicles* and *convoys of vehicles* provide the best local contextual information. Baumgartner *et al* and Hinz *et al.* extract both local and global contextual information by considering

the Digital Surface Model (DSM) of the scene and a set of features extracted from the original image. The DSM is created by considering the edge frequency and surface roughness, while the set of features are edge frequency, edge straightness, the frequency of rectangular and parallel edges, and edge amplitude.

Automatic pattern class generation

According to Jain *et al.* [74], learning is an integral part of most pattern recognition systems. In the context of pattern recognition, learning entails the derivation of certain statistical properties of pattern classes with the goal of using these properties to classify future samples. Classification with respect to road extraction is no different, and samples of typical road structures are required for training. Selecting these samples through automation would form a key component within a fully automated RNE system. Considering Figure 2.1, the *Training Set Selection* component automatically generates pattern classes (road models) at a low level, which are used as input to the *Spectral classifier* component. A few methods, capable of generating road samples automatically, are discussed in following paragraphs.

Bacher and Mayer [8] present a method where pattern classes are generated automatically from IKONOS imagery. An edge detector is applied over a scene and all parallel edges (within a given threshold) are identified. The areas between the parallel edges are evaluated and added to a training set if they contain uniform grey values.

Doucette *et al.* [38] also identify parallel edges and extracted the centrelines that run between these edges. The Self-Organized Road Mapping (SORM) algorithm and a fuzzy grouping model are used to construct a preliminary road vector topology. A Self-Supervised Road Classification (SSRC) automatically selects training samples by considering a 3×3 window around nodes along the vector topology.

It is worth noting that not all pattern recognition methods require statistical properties derived from training samples. When using an unsupervised clustering method such as the k-means or ISODATA algorithm, the input parameters are not directly related to the statistical properties of the classes. In this case, the limitation of the algorithm is its inability to label the cluster produced, which means that a higher level process would be needed for labelling.

Summary

As is evident from the techniques discussed in this section, classification techniques can be applied to RS imagery in various ways. Spectral, textural and geometrical classifiers are typically applied at a lower level, while global context classifiers and automatic pattern class generators are more complex techniques involving algorithms at various levels. The techniques mentioned are by no means exhaustive and classifiers can be applied to numerous other areas in road extraction systems.

2.3.3 Edge detection

Edge detectors are algorithms that aim to identify discontinuities in the intensity within digital imagery. Put another way, edge detectors seek to find areas in images where the brightness changes rapidly over a short distance. The points where these changes occur are marked as edge points. Edges are also often referred to as ridges and are not to be confused with boundaries or borders. Boundaries form a closed loop and are often the section where two bordering segments meet. Edges are seen as *local* concepts while boundaries are seen as *global*.

A common edge detection approach begins by detecting areas within a local window where a considerable change in intensity occurs. The change in intensity at a given point might be gradual and span over a few pixels. A thinning algorithm can be used to reduce this area to a single point. The end result is a line running along the edge where the change in intensity is the greatest. This simplistic edge detection approach produces localized edges and is sensitive to noise or disruption along the edge line.

In general, road objects appear as linear structures at lower spatial resolutions and as homogeneous elongated segments in higher spatial resolution [63, 165]. Edge detectors can be applied to lower resolutions to extract linear features or to detect the roadsides of the elongated segments in higher resolutions. The versatility of these algorithms has made them very popular in feature extraction literature [124]. Out of the 217 RNE-based studies surveyed, 129 used edge information at some stage. Considering Figure 2.1, the lower-level *Edge Extraction* component provides input to the *Geometrical Detector*. The following section describes edge detection algorithms in more detail, while the subsequent section discusses their application to RNE problems.

Edge detectors

Various edge detectors have been developed, including Canny [23], Nevatia-Babu [121], Sobel [40], Laplace [18], Marr-Hildreth [105], Nalwa-Binford [119] and Sarkar-Boyer [137], amongst others. Edge detectors can be considered as being either search-based or zero-crossing based. Search-based methods typically use a first-order derivative to calculate the edge strength and direction using the gradient magnitude and orientation respectively. A search function considers the local edge strength and the directions in which edge locations are to be identified. Zero-crossing based approaches tend to calculate the second-order derivative in order to obtain a value such as the gradient rate of change. The algorithm will then search for points where the zero reference axes are crossed (zero-crossings). The local gradient maxima are extracted at the zero-crossing points. Edge detectors and particularly zero-based approaches, are sensitive to image noise. Gaussian filtering is commonly used to smooth the image before applying an edge detector. The processes followed by typical edge extractors are described in the following paragraphs.

One of the most popular edge detectors in DIP and RNE literature is the search-based Canny edge detector [23]. The algorithm is designed to fulfil the following objectives mathematically:

1. **Minimal error rate.** All true edges should be detected whilst keeping the number of spurious edges to a minimum.

2. **Localized edge points.** The edge points should be as close to the edges in the original image as possible.

3. **Single edge points.** Only a single point should be extracted for each true edge point.

The algorithm begins by smoothing the image with a Gaussian filter. The gradient direction and magnitude is calculated in the vertical and horizontal directions. The gradient direction is discretized in one of the eight directions, which is in the direction of a neighbouring pixel. The edge gradient is 'thinned' through non-maximum suppression, which is finally traced and thresholded with hysteresis. The thresholding process causes the pixels in the image to be marked either as edge or background pixels.

A comparative study by Heath *et al.* [62] compares four edge detectors, namely Canny, Nalwa-Binford, Sarkar-Boyer, and Sobel. The authors conclude that each of the

detectors, save Sobel, specialize in extracting edges from certain scenarios. The Canny algorithm did have the best overall accuracy, when optimized for the scene. Figure 2.3 presents an example of edges extracted by the Canny algorithm.

The Nevatia-Babu edge detector [121] is another popular approach which is often found in RNE literature [49, 76, 150]. The method seeks to fulfil the same objectives as Canny, but obtains the information at each step in a different way. The method begins by convolving an image through a set of edge masks to obtain the edge magnitude and direction. A thinning and thresholding operation extracts the edge pixels, which are finally linked according to their proximity and direction. The following section describes a number of applications of edge detection algorithms to RNE problems.

Road extraction from edge information

The information obtained through edge extraction can be used to extract road structures in numerous ways. Some of the techniques include *automatic road seed point selection*, *geometrical classification*, *segmentation* and *road tracking*. Edge extraction is often used to detect road seeds by considering parallel edges in medium to high resolution imagery (refer to Section 2.3.1). This technique is used in studies by Ruskone [136], Mei *et al.* [112], Chen *et al.* [27] and Dal-Poz *et al.* [32], amongst others. Baumgartner *et al.* [10] conclude that the precision of road extraction based on parallel edge detectors is poor and incorrect hypotheses are common. To improve the accuracy of RNE systems based on edge detection, Doucette *et al.* [38] add two components, namely a heuristic network building algorithm and a feedback loop that considered the spectral characteristics of the extracted road. These components removes unlikely road sections based on their

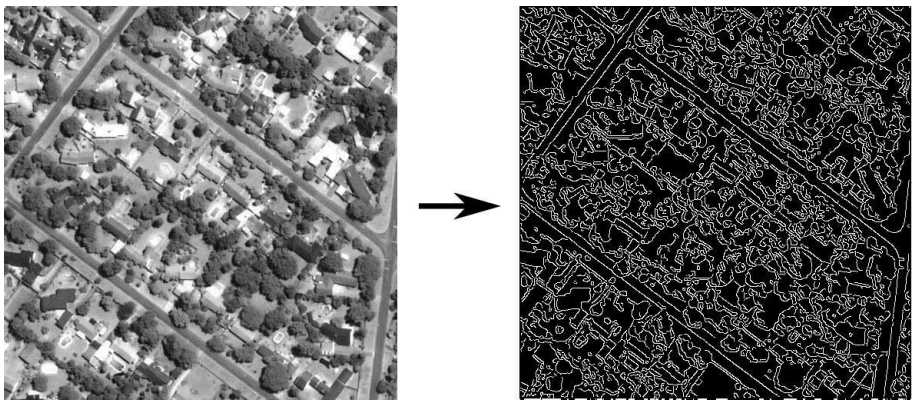


Figure 2.3: Canny edge detection

structural and spectral features.

Mei [111] propose another technique to increase the consistency of edge detectors. Considering that edge detectors are sensitive to noise and small occlusions, an edge could easily be broken by a tree or a shadow passing over the road. Mei extends the Canny edge detector to bias straight line segments by applying a Kalman filter [77] to every edge pixel. The Kalman filter searches for a neighbouring edge point based on its previous phase difference. If a neighbouring edge point is found, the phase difference is recalculated and the search for a new neighbouring edge point continues. This approach differs from most other approaches since the Kalman filter introduces an estimation component.

A different usage of edge detectors is presented by Youn and Bethel [169] where Canny edges are used to provide contextual information regarding the dominant road directions in an urban scene. Urban roads tend to be aligned in a grid, which implies they have two dominant directions perpendicular to each other. To determine these directions, a scene is divided into equal sized segments and the Canny algorithm is applied. A histogram is constructed by calculating the edge directions within a segment. Dominant edge directions should appear as peaks at 90° angles within the histogram. Wang and Zhang [161] also used this technique to provide input for a segmentation algorithm.

Hu *et al.* [69] present a system where Canny edges are used to conduct contextual classification (refer to Section 2.3.2). The system begins by automatically identifying a number of road seed points and extracts Canny edges for the scene. Road trackers (refer to Section 2.3.7) are initialized from the seed points and the kernel function determines a 'footprint' at each step. The footprint represents a wagon wheel, where the distance from the candidate pixel to the nearest edge in 360° is measured. A number of classifiers are then used to determine whether the shape is consistent with that of typical roads.

The Gradient Directional Profile Analysis (GDPA), proposed by Wang *et al.* [51, 160], is another approach that utilizes edge detection. This method calculates the predominant gradient direction in a 3×3 kernel over all pixels within a scene. The gradient is given by the maximum slope in intensity along the four immediate neighbours of the candidate pixel. Roads appear as ridges in the SPOT data (refer to Section 2.8) used by Wang *et al.* The data has a Ground Sampling Distance (GSD) of 10 m, which refers to the pixel size in ground units. To detect these ridges the concept of *lists* is introduced. A list comprises strings of connected pixels with similar gradient directions. A set of criteria is defined to describe the characteristics of roads in terms of the detected lists. The defined criteria include aspects such as the *maximum curvature of the list*, the *slope of*

the gradient and the *gradient rate of change*. These criteria are used to filter lists that are not probable road candidates.

Summary

Edge detectors are useful techniques for detecting roads and can be utilized in various ways with the image processing chain. The way in which these detectors are typically used is to detect roadsides in high resolution imagery, while they are also used to detect thin lines at lower resolutions. Edge detectors have, in addition, been used to provide information to higher level processes such as segmentation, contextual and geometrical classifiers. Edge detection is a very popular tool, but is not a definitive solution to road extraction since it is sensitive to noise and occlusions. A vast number of road extraction approaches based on edged detection can be found in literature and the techniques discussed here should be considered an introduction to the topic.

2.3.4 Hough transform

The Hough Transform (HT) is a technique used to find features of a particular shape in digital imagery. The classical HT is used to detect regular shapes, such as lines, circles and ellipsoids. The generalized HT is a parametric approach and can be used to detect arbitrary shapes. One of the advantages of using the HT is that it is relatively unaffected by noise and fragmentation.

The HT transforms features into single points in parameter space. Linear features in the original image can be detected by searching for peaks in the parameter space. Through thresholding, the number of detected features can be fine-tuned. The following section evaluates a number of road extraction methods using the HT.

Hough transform techniques

Geman and Jedynak [46] are among the first to use the HT within an RNE system. HT is used to detect areas of interest (where roads are present) which are then used to mask the non-road areas. The system begins by applying a set of convolution filters to the original RS image. These convolution filters act as edge detection masks and yield a fine scale result containing a considerable amount of noise. The noise is reduced by scaling the edge image from a 512×512 resolution to 16×16 pixels. The HT is then used to detect the predominant line directions within a window. These directions are then used

as indicators of the areas in the image where roads occur.

Jin and Davis [76] developed the Spatial Signature Weighted Hough Transform (SSWHT) to detect the grid-like shape of roads in dense urban scenes. The ATS method (described in Section 2.3.2) is used to calculate the spatial signature for all pixels in a given scene. The spatial signature produces a histogram containing the number of homogeneous pixels at discrete angles, 360° around the candidate pixel. The SSWHT uses the length of this spatial signature as a weighting for the HT rather than simply using a value of one as weighting. Through this weighting, the highest response in the (ρ, θ) plane corresponds to the lines in the scene with the longest spatial signatures. These polar plane points are projected back to the original image where they appear as lines. The result is a grid of lines that runs through the entire image. A clean-up operation is done to verify each line segment (road) within the grid.

Preliminary road segments are extracted with the HT in the work by Ye *et al.* [168]. The method begins by detecting roadsides from high resolution imagery through the application of the HT in a local neighbourhood. The detected lines are filtered by considering the gradients and spectral uniformity. These filtered road segments are then used as the initialization points for a road tracker.

Amberg *et al.* [3] use a Dynamic Programming (DP) method to extract road centrelines where the underlying structure has a low curvature. The result is a set of disconnected linear road segments. A method using HT is proposed to connect the end points of these centrelines. Considering that most connections have a high curvature, the target function is defined as the hyperbola

$$a(x-p)^2 + 2b(x-p)(y-q) + c(y-q)^2 = 1.$$

Two parameters (p and q) describe the centre of the curve and the remaining parameters (a, b and c) describe the shape. These parameters are calculated by a two step randomized HT. Firstly, the two lines under consideration are extended and their intersection selected as the centre point (p and q). Secondly, a and c are fixed and selected from the connecting lines while b is deduced from the underlying image data. The extracted structure is then connected to the two centrelines in question.

Summary

The HT is a popular approach used to mathematically describe the shape of the underlying objects within RS imagery. The HT can be used to detect regular geometrical shapes

such as lines, curves and circles. With regards to RNE, HT is often used to detect linear road sections, but it has also been extended to detect curved roads. One of the greatest benefits of using the HT is its aptitude in dealing with noise and fragmentation.

2.3.5 Mathematical morphology and filtrate

Mathematical morphology and filtrate was first introduced by Matheron [106] and Serra [138] for geometric analysis. Mathematical morphology is primarily used for binary image processing, but can also be extended to support greyscale images. It is expressed in the language of set theory. Set theory allows a number of mathematical operators such as *union*, *intersection* and *complement* to be applied between sets. By applying these operators in a certain fashion, useful representations of region shapes (such as edges or skeleton structures) can be obtained [52, 73]. In road extraction systems, mathematical morphology is often used as a filter during pre- or post-processing operations.

Mathematical morphology theory

Dilation and *erosion* are two fundamental mathematical morphology operations which can be used to define more complex operations. Dilation causes objects to grow in size whilst erosion has the converse effect.

Consider the illustration in Figure 2.4 of sets A and B in a two dimensional (2D) binary space, with set B the Structuring Element (SE), or kernel. The SE centre is indicated by the grey square and is applied to set A. The dilation of set A by B is defined as

$$A \oplus B = \{z|(B)_z \cap A \neq \varnothing\}. \tag{2.3}$$

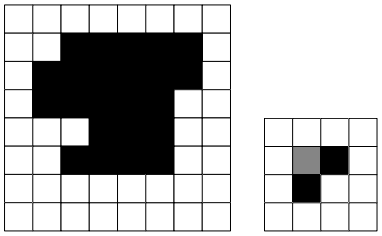
Here z is all the displacements of the origin of B (grey in Figure 2.4b) over A, where at least one element in B extends beyond the elements in A.

As shown in Figure 2.5a, the operation has the effect that the boundaries of the original object (A) are enlarged. The basic principle is that a union operation is performed between A and B for all z. The grey elements in Figure 2.5a indicate the newly added boundaries.

Erosion of set A by B is defined as

$$A \ominus B = \{z|(B)_z \subseteq A\}. \tag{2.4}$$

The erosion process is similar to dilation with the exception that the boundaries of A are



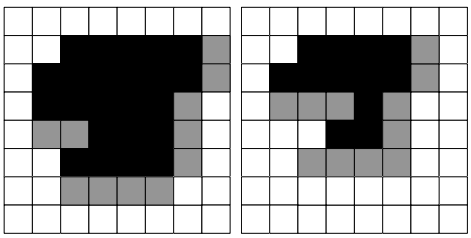
(a) Set A (b) Set B

Figure 2.4: Sets A and B in 2D space

reduced instead of enlarged (Figure 2.5b). The process performs a relative complement operation on A by B, for all z. The grey elements on Figure 2.5b indicate which elements are to be removed.

Other mathematical morphology operations found in RNE literature includes *opening*, *closing* and *thinning* [71, 115, 145]. The opening operation is defined as the erosion of A by B, followed by a dilation by B. The closing operation is defined as the dilation of A by B, followed by an erosion by B. Thinning is the process whereby a set of masks are iteratively applied to an object with the goal of reducing the object size to a single line of pixels. The opening transformation can be used to disconnect small objects from larger road objects, while preserving the shape of the road objects. Closing can be used to connect road objects and fill small gaps, while also preserving the shape of the objects. Thinning can be used to reduce road segments into a single line of pixels representing the road centreline.

When analysing greyscale imagery, the image surface can be considered to be three-dimensional where the intensity value serves as the height property. This surface model could contain features such as peaks, holes, ridges and valleys. The dilation operation would broaden peaks and ridges while filling holes and valleys, while erosion would have the converse effect, removing peaks and regions while broadening holes and valleys. The



(a) Dilation (b) Erosion

Figure 2.5: Dilation and erosion operations of A by B

following section considers some of the ways in which mathematical morphology has been applied within the field of RNE.

Road extraction through mathematical morphology

Mathematical morphology is a very popular set of tools in DIP and also in the RNE field. These tools can be applied to achieve numerous goals, but they are most often used in a pre-processing stage or to extract a final road skeleton. A few studies where mathematical morphology is used to great effect will now be discussed.

Chanussot and Lambert [25] designed a mathematical morphology filter based on three distinct road characteristics observed in SAR data. These features are defined as.

Rectilinear: Roads appear locally as elongated rectilinear structures with a certain minimum length.

Thin: The width of the extracted features should not be more than a given maximum threshold.

Contrast: Roads will appear darker in SAR data than the background.

Several filters are combined to detect the abovementioned features. These filters consist of opening and closing operations and variants thereof. The final filter comprised the following operations:

1. Non-flat peaks are removed through opening by reconstruction.
2. Non-linear valleys are removed with a direction closing operation, which is rotated at 40 different angles.
3. An opening operation removes the remaining peaks.
4. Wide valleys are removed by a closing top-hat operation.

The structuring elements used in these filters, as well as their size are set according to *a priori* information based on the road width and curvature. By thresholding the response obtained from the set of filters, a set of linear features, representing the underlying road structures, are extracted.

Katartzis *et al.* [79] build on the work of Chanussot and Lambert with the intent of detecting larger road sections with fewer incorrect points. The directional closing operator is sensitive to noise and was replaced by a soft morphological filter based on

weighted order statistics. The directional information for each pixel is recorded and used in subsequent processing.

Géraud and Mouret [47] applied a segmentation algorithm to greyscale satellite imagery. The images contained numerous local minima which resulted in over-segmentation. The mathematical morphology closing operation is ideally suited to suppress local minima. A disc is often used as a structuring element where no prior information is available. The disc shape has the drawback of creating artefacts when the filtering strength increases, for example, peaks tend to shift when a large number of minima are removed. The authors use an *area closing* operation [157] which does not have the same deficiencies as the classical closing operation.

The abovementioned filters are used to enhance or suppress features within an image. Mathematical morphology can also be used to extract road centrelines through a process known as thinning or skeletonization. Thinning is comparable to erosion in that the object size is sequentially reduced. Thinning preserves the object topology, which is not the case with erosion. Figure 2.6 illustrates the manner in which a classified image can be thinned to extract the centreline of the object. Due to the anomalies present in the final thinned image, further processing would be required to straighten lines, remove branches, breach gaps and remove the isolated incorrectly detected road centrelines.

Zhang *et al.* [172] use the thinning operation as part of the process of extracting road centrelines. The method begins by identifying road pixels based on their spectral values. Numerous spurious classifications are made and an opening operator is used to remove objects smaller than a given size. The classified result contains a few gaps between the road segments, and the closing operation is used in an attempt to connect these sections. Finally, the road centrelines are extracted through thinning.

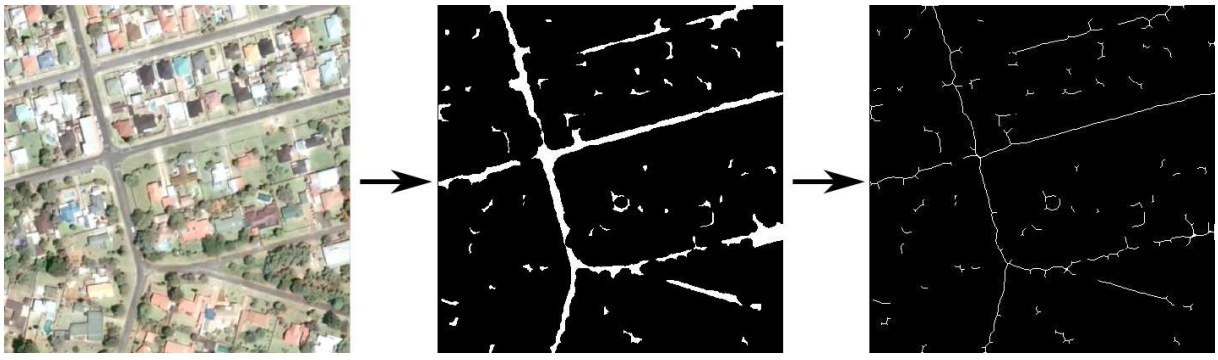


Figure 2.6: Mathematical morphological thinning

Summary

Mathematical morphology is a versatile technique which can be used in various ways during the process of road extraction. As illustrated in this section, morphology is often used in pre- or post-processing steps to enhance or reduce certain features. Mathematical morphology can be used as a feature detector capable of extracting roads and also to thin objects to extract road centrelines.

2.3.6 Multi-resolution analysis

Multi-Resolution Analysis (MRA), also known as Multi-Scale Analysis (MSA) refers to a theoretical analysis technique whereby imagery is represented in multiple resolutions. According to Gonzalez and Woods [52], MRA theory deals with the representation of images (or signals) in different resolutions (or frequencies). The usage of a certain name, be it MRA or MSA, merely places an emphasis on different aspects of the same concept [128]. As the resolution increases, the scale decreases. For notational convenience, MRA and MSA will be referred to only as MRA. Figure 2.7 presents an example of a Multi-resolution pyramid, consisting of a suburban scene at three scales.

MRA is useful when applied to road extraction since certain road characteristics become more apparent at different resolutions. Figure 2.8 illustrates what the Canny edges for the corresponding images in the image pyramid would look like. It is clear that certain features are more apparent at different levels. For example, roads appear as elongated regions at higher resolution, but as long lines at a coarse resolution. Other

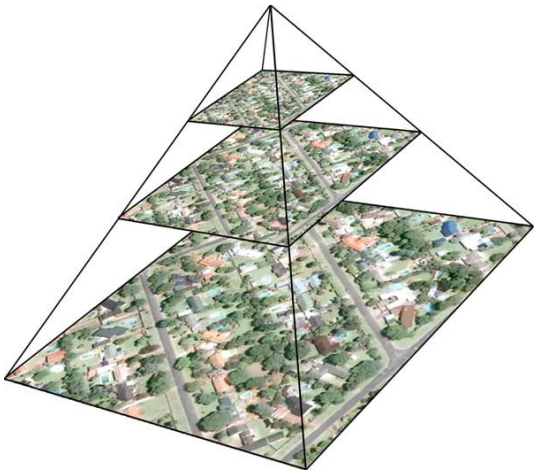


Figure 2.7: A multi-resolution pyramid

features, such as road markings and cars, are naturally only visible at a very high resolution. When considering the Canny output, the edge extractor is able to detect the roadsides at a high resolution, but these become less apparent as the resolution decreases. It would be feasible to apply different extraction techniques at different resolutions, with the aim of increasing the accuracy of road extraction systems.

MRA is also beneficial in reducing the processing time of road extraction systems. A common approach begins by detecting regions of interest at a coarse level and moving down to a finer level only in these areas. Considering that roads typically have a small surface area in relation to non-road areas, the amount of data to be processed at such a fine level can be greatly reduced. The following section takes a look at different types of MRA algorithms and various ways they can be applied in RNE systems.

MRA techniques

A great variety of methods are found in literature whereby remotely sensed imagery is translated into a number of given sub-levels with the aim of increasing the accuracy of RNE systems. A few interpolation methods, which can be used to obtain various sub-levels, as well as applications of MRA in RNE systems, are described in this section.

Various interpolation methods can be used to obtain the different levels within an image pyramid, with *nearest neighbour*, *bilinear* and *bicubic* being amongst the best

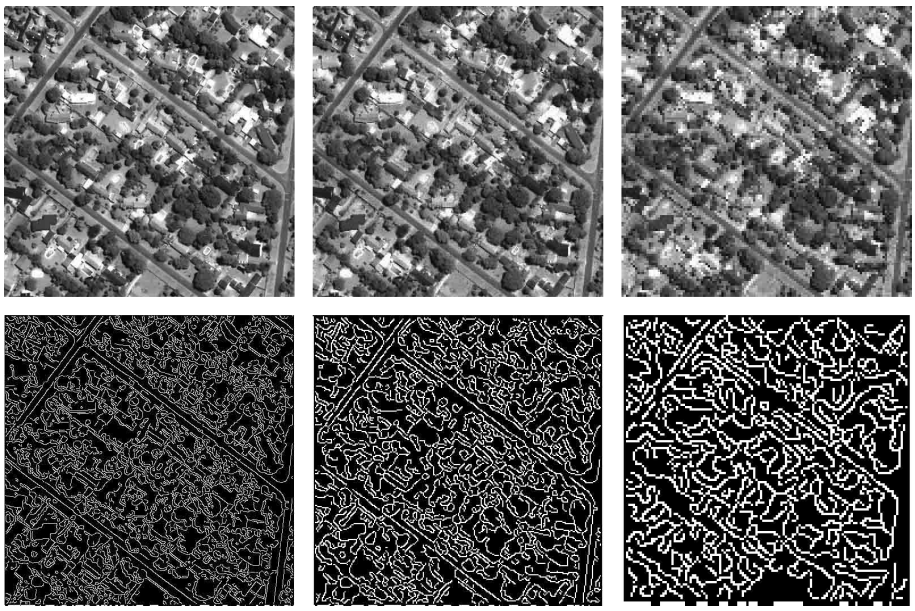


Figure 2.8: Canny edges at resolutions 512×512, 256×256 and 64×64

known. The pixel value at each level is determined by deriving information from the underlying level. The nearest neighbour approach selects the value of the nearest pixel in the higher resolution as the value to be used the lower resolution. The bilinear approach calculates a weighted average of the values in the underlying $n \times n$ pixel block. The size of the block (n) depends on the level of the lower resolution in the image pyramid. An $n = 2$ would result in an image half the size of the original image. Bicubic interpolation extends on the bilinear approach by calculating a weighted average over a pixel block of $n^2 \times n^2$. A higher weighting is given to pixels that are closer to the lower resolution pixel being calculated. The bicubic method provides a smoother result with fewer interpolation artefacts than nearest neighbour or bilinear interpolation. These artefacts are more apparent when interpolating from a low to a higher resolution [52]. The trilinear and tricubic methods can be used when interpolating across three image bands.

One of the earliest MRA road extraction approaches is presented by Steger *et al.* [148] where a different extraction method is used for each scale level. The two road extraction methods are applied to aerial panchromatic orthoimagery; one algorithm on a fine scale with a 25 cm ground resolution, and the other at lower resolution which was reduced by a factor of eight. Please see Section 2.8 for more information on the orthoimagery data source. The fine scale method extracted roads based on a structural model matching technique, while the lower scale method extracted lines based on the image intensity level. The outputs are combined by selecting roads that occur at both levels only.

A more common approach is to define a generic algorithm, capable of extracting roads at various levels. Such an example is provided by Lisini *et al.* [100] where roads are extracted at various scales in an image pyramid. Bilinear interpolation is used to create three sub-levels for $n = 4$, $n = 8$ and $n = 16$. The system applies a road extraction algorithm at each of the levels in the pyramid and combines the outputs through superimposition. More examples of the use of scale independent road extraction algorithms can be found in [33, 96, 152, 64].

Another popular MRA approach is based on the well known wavelet transform. Wavelet transforms can be applied to road extraction in two different ways [173]. The first is to derive multiple image resolutions through a wavelet transform. Such an example is presented in the work of Gruen and Li [54] where the wavelet acts as an impulse response of a band-pass filter. The wavelet transform applies the convolution with the band-pass filter which is dilated and translated. The wavelet acts as a scaling component where

features of different sizes are enhanced. The second approach is to perform analysis on the wavelet domain itself. Couloigner and Ranchin [31] present an algorithm where an edge detector is applied in the wavelet domain. The method begins by constructing an image pyramid using the wavelet transform. The wavelet coefficients for each level are determined by calculating the difference between level n_i and n_{i-1}. When an edge detector is applied to these wavelet coefficients, it is possible to detect the road centreline and sides.

A novel approach by Gibson [49] utilizes the Generalized Balanced Ternary (GBT) addressing system to represent image data and derived vector information at various levels. A grid of hexagon-shaped cells is created across the image forming a honeycomb structure. Each pixel is mapped to a cell and the image data and vector information is aggregated for each cell at level n_0. A cell and its six neighbours can then be combined to form an individual cell at level n_1. These cells contain additional information such as mean, variance, and first and second momentums of inertia. The process can be repeated for multiple levels of n. A series of road extraction methods is applied at various levels and finally a road topology is extracted from the cells at level n_4.

Summary

This section provided an overview of methods commonly used to construct multi-resolution pyramids, and studies where MRA is utilized within road extraction systems. It presented both level-specific and generic extraction methods. The level-specific approach identifies road structures in individual layers with a different extraction technique at each level, while the generic approach applies a scale-independent technique at all levels.

2.3.7 Road tracking

Road tracking can be considered as a line tracing method. Road trackers typically consist of two key components: a future position prediction algorithm and a matching or refinement algorithm. The future position prediction algorithm typically calculates where the next step should be taken by considering the current and preceding steps. The refinement algorithm is executed after a new step has been taken to determine whether the prediction has been accurate and also to determine the orientation of the road. The assumptions made by road trackers differ, but often include a number of the following:

- road surfaces are homogeneous,

- roads have low road curvatures,
- there are little or no occlusions occurring along the route, and
- all roads are interconnected to form a network.

A simplistic hypothetical algorithm is presented to illustrate the road tracking process. Road trackers require a starting *seed point*, which is typically provided by a human operator or an automated seeding algorithm (refer to Section 2.3.1). From this seed point, the tracker shifts along the road direction to a new candidate position, which can be refined by a matching algorithm. The road direction can be supplied by a human operator, by previous positions or by the information gathered from the surrounding pixel values. This process is repeated until the entire road segment has been extracted.

This simplistic approach will fail easily when it encounters real-world data and it will require additional features to improve its robustness. The following section presents a number of road tracking techniques which seek to provide a robust solution.

Road tracking techniques

In an early attempt at creating a robust road tracker, McKeown and Denlinger [108] propose a system where two independent low-level processes are used during tracking in high resolution aerial imagery. The first process uses *road surface texture correlation*, and the second uses *road edge following*. Each of the autonomous processes extracts the road centreline, gathering properties such as road width and surface intensity. An intermediate-level process is used to monitor the progress of the low-level operations. In addition to monitoring, more road properties, such as *changes in width*, *surface changes* and *occlusions* are gathered. The intermediate-level process evaluates the data gathered and might suspend a given tracker and restart it with model data gathered from other trackers. A description of the surface texture correlation tracker and the road edge following algorithms is given in the following two paragraphs.

The road surface texture correlation tracker begins with a human operator specifying a starting seed point, the road width and tracking direction. A cross section intensity profile is extracted perpendicular to the road direction and is used as an initial surface model. The tracker steps forward by fitting a parabola through the most recent road points. Considering that only a single road point has been selected, the next move will be in the direction specified by the operator. A cross section is extracted from the new position and the actual road position is determined with cross correlation of the surface

model. A score is assigned to each position based upon the accuracy of cross correlation. Only points with high scores are considered during the curve fitting process. In cases where correspondence is poor, the tracker will continue guessing the road position until a good correlation is found or until the tracker has gone too far. In the latter case, the tracker will assume that a new road surface has been encountered and a new surface model is extracted.

The road edge following tracker takes a similar approach, but it compares statistics derived from road edges rather than correlating surface models. The statistics are gathered by applying a Sobel edge detector [40] to the road cross section at the current position. The algorithm calculates a weighting score for each of the edge points and selects the highest scores as the road edge points. The score is taken as the sum of the following Sobel values: *edge strength*, *orientation*, *difference in magnitude from each neighbouring point*, and *difference in angle from each neighbouring point*. Where two edge points are identified, a mid point is marked as the road centre point. If a single point is extracted, it is assumed that the road width has remained constant and the centre point is marked at half the road width from the edge. Where no high weightings are found, the algorithm will continue guessing the following position as with the road surface texture tracker.

The aforementioned system also attempts to detect intersections at a higher level, but the low level road trackers are inherently designed to detect only curvilinear road segments. Shen *et al.* [143] propose a road tracker capable of detecting junctions by using the ATS method discussed in Section 2.3.2. As with the McKeown and Denlinger approach [108], the system is initialized by a human operator which starts a curve-fitting stepping function that predicts future road points. The road direction is determined by considering the valleys in the ATS histogram calculated at every point. Under ideal conditions, a single road section should have two valleys, one in each direction. Three or more valleys are indicative of a road junction, and the tracker will attempt to follow each branch until the entire network is extracted.

Vosselman and Knecht [158] propose substituting the curve-fitting prediction step by McKeown and Denlinger [108] with a stochastic modelling approach driven by least squares and a Kalman filter [77]. The Kalman filter is an efficient recursive process, capable of detecting the parameters in a linear dynamic system from a sequence of noisy measurements. The Kalman filter is used during tracking to make future predictions of the road position and shape. When disturbances along the road cause the matching process to fail, the Kalman filter uses one or more of the previous iterations to predict

future states. In a further improvement, least squares profile matching is chosen over cross correlation, because of its ability to estimate the precision of the profile shift. This precision is used to quantify the accuracy of the matching algorithm and provides input to the Kalman filter. Vosselman and Knecht found their approach to be robust enough to continue past obstructions such as highway overpasses and junctions.

Kim *et al.* [83] propose a more robust least squares matching algorithm which is used to track roads in IKONOS imagery. Rather than using a single cross section, a 25×25 window around the centre point is used to construct a road model. When moving to a new candidate point, the orientation and position of the candidate position is refined by locally shifting and rotating the template until the maximum correlation is obtained. To reduce the complexity of the search, the template is shifted only in a direction normal to the expected road direction, effectively reducing the search space by one dimension.

Summary

This section has presented a number of road trackers and discussed some of the aspects requiring consideration when using these techniques. A number of limitations have also been mentioned, as well as possible ways of addressing them.

2.3.8 Segmentation

Segmentation is the process of grouping data into subsets where the items in a subset have similar characteristics. In computer vision, these characteristics are generally defined by the discontinuities and similarities of intensity values, spectral radiation or textural patterns that occur within an image [52]. The result is a collection of pixel clusters, where the characteristics in each cluster differ significantly from those of its neighbours. The aim of segmentation is to add structure to the data which allows for faster and more accurate analysis [142].

Figure 2.9 provides an illustration of a hypothetical road extraction system using segment analysis. The system begins by segmenting the initial candidate scene into clusters based on the spectral characteristics within a segment. A classifier can then be used to analyse the segments and assign a road probability value to each cluster. This is only one example, although segmentation has been used in numerous ways to assist RNE [114]. The following list presents a number of examples that show the range of methods that exist, as well as a few studies where these segmentation methods have been applied [52, 142]:

- Clustering methods [1, 76, 101, 117, 161]
- Histogram-based methods [169]
- Boundary detection methods [130, 161]
- Region growing methods [129, 139, 145]
- Level set methods [21, 80]
- Graph partitioning methods [53, 129]
- Watershed transformation [47, 96, 111]
- Model based segmentation [67, 162]
- Multi-resolution segmentation [126]
- Neural networks segmentation [20, 118]

Considering the diversity of segmentation techniques, providing an in-depth discussion of each method is beyond the scope of this study. The following section highlights a few road extraction systems, which in turn illustrate the practical applicability of image segmentation.

Segmentation techniques

As mentioned in the previous section, various segmentation algorithms are used in studies pertaining to road extraction. Binary segmentation is a commonly used approach, where the two classes consist of a road and a non-road class [38, 115, 117, 145]. Unless explicitly

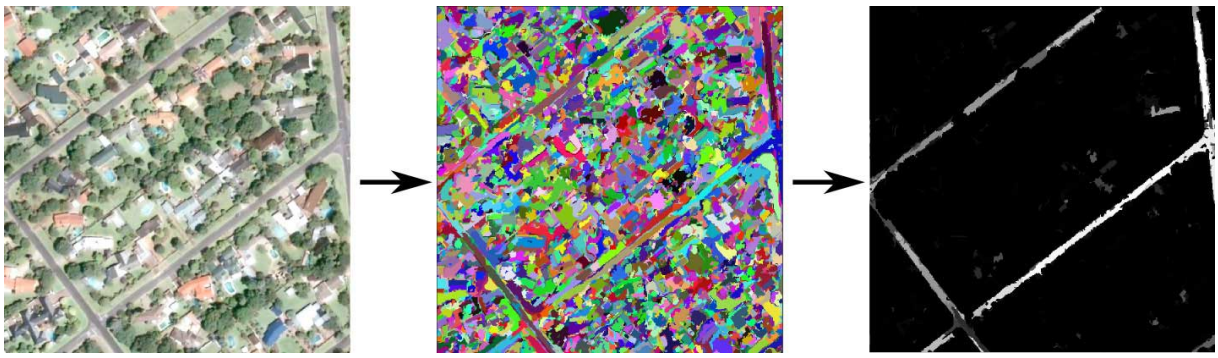


Figure 2.9: An example of segmentation and a subsequent spectral classification

modelled, the classifiers used in these approaches do not consider the spatial properties of the data points to create the binary segments. The remainder of this section considers spatial segmentation methods only. Please refer to Section 2.3.2 for more information pertaining to binary segmentation.

The well-known *k-means* clustering algorithm [103] is often used in image processing and road extraction literature with the intention of detecting objects or borders. The algorithm attempts to classify n samples into k clusters by considering the distance to the nearest mean. The pixel coordinates are commonly used to consider the spatial structure of the clusters [142]. A drawback of this algorithm is that the number of classes k has to be defined beforehand, but in spite of this drawback its simplicity and efficiency contributes to its popularity in the field of RNE [14, 37, 76, 102]. Zhang and Couloigner [175] clustered IKONONS and QuickBird multispectral images by considering all four bands (*red*, *green*, *blue* and *near-infrared*). A number of k values are considered, with six being the final choice. The road clusters are identified with a model-based spectral classification algorithm.

ISODATA [56] is another popular technique which is used in numerous RNE studies [37, 51, 90, 170, 172]. The appeal of the unsupervised ISODATA algorithm lies in its ability to change dynamically the number of clusters by splitting clusters with large standard deviation values and conversely merging clusters with low standard deviations. Zhang [170] used ISODATA to segment and classify aerial photographic stereo images. Zhang identifies five classes (*road regions*, *vegetation*, *shadow areas*, *dark roofs* and *red roofs*) for objects that appear in imagery. This approach is different from the two class classifications most methods follow. Rather than working on the raw Red, Green and Blue (RGB) bands, the image data is transformed in a novel manner. The first band of the transformed image is set to be the first component of the transformed principle component image. The second band is transformed into a *greenness* band by forming the ratio $(G - R)/(G + R)$. Finally, the third band is set to the saturation component of image data in Hue Saturation Intensity (HSI) colour space. Five models are created, one for each class, to which the ISODATA method is applied.

Hinz and Baumgartner [67] utilize segmentation to solve a number of problems within an intricate RNE system. The system firstly determines the global context (such as *forest*, *rural* or *urban areas*) from high resolution aerial photography (45 cm) by applying the texture segmentation by Laws [95] at a lower resolution (4 m). A series of methods, specific to the global context, are employed to identify the road structures themselves. A region of interest is obtained by evaluating bright lines with symmetric contrast [147]

and probable roadsides [93]. These regions are assembled to create lane segments by considering local context objects, such as *buildings*, *cars* and *road markings*. During subsequent processing, additional characteristics of the local objects, such as *shadows* and *orientation in relation to the road structures*, are used to expand the lane segments further. Finally, all the lane segments are fused into a single road network [162].

Summary

Considering that road structures typically appear as homogeneous regions in high-resolution RS imagery, segmentation is ideally suited to detect these areas. Segmentation techniques are not limited to detecting the road surfaces themselves, but can also be used at different levels in the processing chain. An example is given where segmentation is used to detect the global context [67], as well as local contextual information by considering the occurrences of buildings, cars and road markings [162]. When considering the diversity of segmentation algorithms and the numerous ways they can be applied to aid RNE, an exhaustive survey is beyond the scope of this study.

2.3.9 Snakes

Active contour models were first proposed by Kass *et al.* [78]. These are also known as "snakes", a reference that can be attributed to their snakelike morphological movements. Snakes are deformable lines that are adjusted to fit features of interest, which typically include edges, lines or boundaries.

The position of a snake is adjusted according to internal and image forces. Internal forces — elasticity and stiffness — stem from the snake itself and depend on the type of feature being modelled. Image forces originate from the image and indicate the presence of certain features of interest. These forces are combined into an energy function which can be visualized as a dynamic three-dimensional surface across the image face. The problem of finding the contour of the feature of interest can now be treated as an energy minimization problem.

The contour fitting process is localized, and the contour initialization position is of vital importance. Inaccurate initialization points could cause sections of the curve to deviate from the intended path into neighbouring local minima. Automated seeding algorithms (as discussed in Section 2.3.1) can be used to detect fitting initialization points.

The applicability of snakes to road extraction is apparent, since the forces can be implemented to fit the snake to the shape of a road structure. The method has also been applied to various image processing issues that include segmentation, motion tracking, and ridge and contour detection [50].

Two approaches are predominantly used in road extraction literature, the first being the original approach by Kass *et al.* and the second the ribbon snakes proposed by Fua and Leclerc [43]. Ribbon snakes build on the original approach by adding a width property, which allows the sides of the snake to be fitted to the boundaries of the feature. These two methods and their applicability to road extraction will now be discussed in greater detail.

Standard snakes

The original approach by Kass *et al.* mathematically and its use in road extraction literature is discussed in this section.

Considering Figure 2.10, a snake is defined as a parametric curve by a list in the image (x, y) plane by the list of n points,

$$\vec{v} = \{v_0, v_1, \ldots, v_n\}, \quad v_i = (x_i, y_i), \quad i = 0, \ldots, n. \tag{2.5}$$

The position of the snake corresponds to the local minimum of the sum of energies

$$E(\vec{v}) = E_{\text{img}}(\vec{v}) + E_{\text{int}}(\vec{v}), \tag{2.6}$$

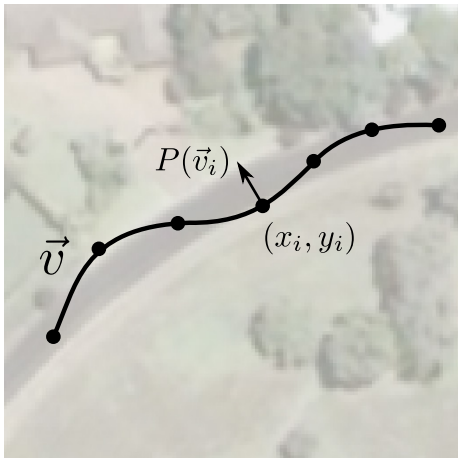


Figure 2.10: The force exerted by the potential function on a snake

where E_{img} and E_{int} represent the image energy and internal energy respectively [107]. The image energy is defined as

$$E_{\text{img}}(\vec{v}) = -\int_0^1 P(\vec{v})ds. \tag{2.7}$$

The potential field $P(\vec{v})$ will have higher values for image areas of interest. The image gradient magnitude could, for example, be used to represent the potential field as,

$$P(\vec{v}) = |\nabla I(\vec{v})|. \tag{2.8}$$

The internal energy determining the geometrical constraints can be defined as

$$E_{\text{int}}(\vec{v}) = \frac{1}{2}\int_0^1 \alpha(s)\left|\frac{\partial \vec{v}}{\partial s}\right|^2 + \beta(s)\left|\frac{\partial^2 \vec{v}}{\partial s^2}\right|^2 ds. \tag{2.9}$$

The $\alpha(s)$ parameter controls the rate at which the snake can change its length and is also known as the elasticity or tension parameter. The $\beta(s)$ beta determines the degree of curvature of the snake and is known as the rigidity or stiffness parameter.

To optimize the position of the snake the total energy has to be minimized. This can be achieved through variational calculus and more specifically the Euler-Lagrange differential equation of motion. This minimization method is characterized by a strong sensitivity to image noise and often converges to local minima. The DP method by Giraldi *et al.* [50] allows for convergence to the global minimum. The remainder of this section considers the practical uses of snakes within RNE literature.

Bentabet *et al.* [12] deliver an in-depth evaluation of the inner workings of snakes used to extract roads from SAR imagery. An existing road GIS database is used to initialize snake positions by dividing the contours into discrete sections using Hermite functions [5]. These polynomials have low degrees of freedom, causing a smooth appearance. Considering the fragmented nature of SAR data, this smoothness property allows the snakes to be less susceptible to noise. The elasticity parameter, $\alpha(s)$, in the internal energy is set to zero, since the intention of the study was only to refine the existing database without expanding it. By evaluating the roads in the existing database, the authors estimated a rigidity parameter of $\beta(s) = 95$ for a straight road, $\beta(s) = 54$ for a slightly curved road, and $\beta(s) = 27$ for a road with a strong curvature.

Unlike the aforementioned approach, snakes can also be represented as polylines rather than polynomials. Polylines are sets of connected linear lines with a node at each

connection point. In order to deform the snake, each node has to be updated by calling the energy functions. These function evaluations can be computationally intensive, which serves as motivation to optimize the number of nodes along the polyline. The goal is to obtain an accurate approximation of the road curvature by using as few nodes as possible.

Agouris *et al.* [2] follow an approach where redundant nodes are removed and new nodes inserted where a higher resolution is needed. When updating the node count on the polyline, the authors consider two sets of criteria, namely the radiometric quality and the curvature. The radiometric quality is defined as the image intensity gradient at a given point, which is indicative of the presence of a road edge. The curvature is measured by rate of change in the angle at a node. An iterative process is used to remove and insert nodes. The method starts by defining four thresholds, namely a high and low threshold for both radiometric quality (E_{RH} and E_{RL}) and curvature (E_{CH} and E_{CL}). The algorithm scans along the contour and calculates the radiometric quality and curvature (E_{Ri} and E_{Ci}). The authors consider nodes with a high E_R and low E_C as having a high likelihood of lying on a road segment with a low curvature and fewer nodes are required to accurately represent the road curvature. These nodes could be redundant and are removed if $E_{Ri} > E_{RH}$ and $E_{Ci} < E_{CL}$. Conversely, nodes with a low E_R and high E_C are considered to be points on a road segment with a high curvature and more nodes might be required to represent the road curvature. More nodes are added by inserting two new nodes on either side of a node when $E_{Ri} < E_{RL}$ and $E_{Ci} > E_{CH}$. After this process, the snake position will be optimized again, but only the manipulated points will be permitted to move. The process ends when no more updates are done or if the updates do not yield any reduction in the total energy of the snake.

Snakes and the way in which they can be optimized have been discussed in the preceding paragraphs. Now, a number of novel road extraction approaches involving snakes will be presented. Jeon *et al.* [75] use the unbiased curvilinear feature detector by Steger [147] during the calculation of the image energy function. The external energy is defined as a potential field over the entire candidate scene and is calculated by combining the presence of a ridge with the intensity of the SAR image. The potential field equation at a given position (x, y) is defined as

$$P(x, y) = \sum_{i=1}^{N} \frac{255 - G_i}{R_i + 1}, \tag{2.10}$$

where N is defined as the number of extracted ridge pixels, G_i is the grey level at ridge pixel i and R_i is the Euclidean distance from the centre pixel to the ridge pixel i. $R_i + 1$ is used to ensure a non-zero denominator and the maximum intensity value of 255 is negated by G_i. Roads appear as darker structures in SAR imagery, which will cause the numerator to increase. In turn, the closer the pixel is to a centreline, the smaller the denominator. The end result is a high potential value where the pixel value is low and close to a ridge.

Péteri and Ranchin [127] present a different, novel approach by combining two parallel snakes to detect the road edges. The energy function incorporates a multi-resolution approach with the Wavelet transform and both snakes are optimized simultaneously to ensure parallelism. The multi-resolution approach is used to reduce the computational time and the effects of local minima. Naturally, intersections would cause breakages along the parallel lines, and so these areas are extracted independently.

In the original snake method by Kass, the energy coefficients are presented as single integrals over the contour. Rochery *et al.* [135] propose the use of higher order snakes, incorporating multiple integrals in the energy coefficients. Multiple integrals allow complex long-range interactions between counter points to be modelled. It also allows intricate geometric information and sophisticated multi-point interaction between the snake and the image data to be used. The contour optimization is achieved through the level set method. The level set approach was extended to enable the use of the new non-local energy functions.

Youn and Bethal [169] found that the native snake approach with its globally defined coefficients is better suited for areas where roads have low curvatures. Road structures in urban areas are more complex, and the use of local energy coefficients is proposed for each individual snake. Different variants are defined, each depending on the local context, which is derived from the local dominant road directions.

This section defined snakes and discussed how they can be optimized when applied to road extraction. Finally a number of novel road extraction approaches using snakes were presented. The following section considers a snake variant, which is popular in road extraction literature.

Ribbon snakes

With the advent of VHR remotely sensed imagery, a variant of the original snake approach — called ribbon snakes — has grown in popularity. This section will define

ribbon snakes and present a few approaches where they have been applied to road network extraction.

According to Mayer *et al.* [107] the concept of ribbon snakes extends the original snake approach by incorporating a width parameter, which allows the sides of the snake to be fitted to a given feature. The original parametric curve (Equation 2.5) can be augmented to include the new width parameter w_i as

$$\vec{v} = \{v_0, v_1, \ldots, v_n\}, \quad v_i = (x_i, y_i, w_i), \quad i = 0, \ldots, n. \tag{2.11}$$

In this form, the ribbon snake has a centre point at (x_i, y_i) and a span of $2w_i$ for each slice i. As with standard snakes, the energy function consists of an image and internal coefficient (Equation 2.6). The representative ribbon snake in Equation 2.11 enables the use of the same internal energy function as defined in Equation 2.9. The w_i parameter will also be subject to the same elasticity and flexibility parameters.

The image energy information for ribbon snakes is not taken at the centre (x_i, y_i) of the curve, but at the left and right side. As illustrated in Figure 2.11, each slice v_i has a corresponding pair of vectors $\vec{v}_{Li}$ and $\vec{v}_{Ri}$, which are obtained by computing

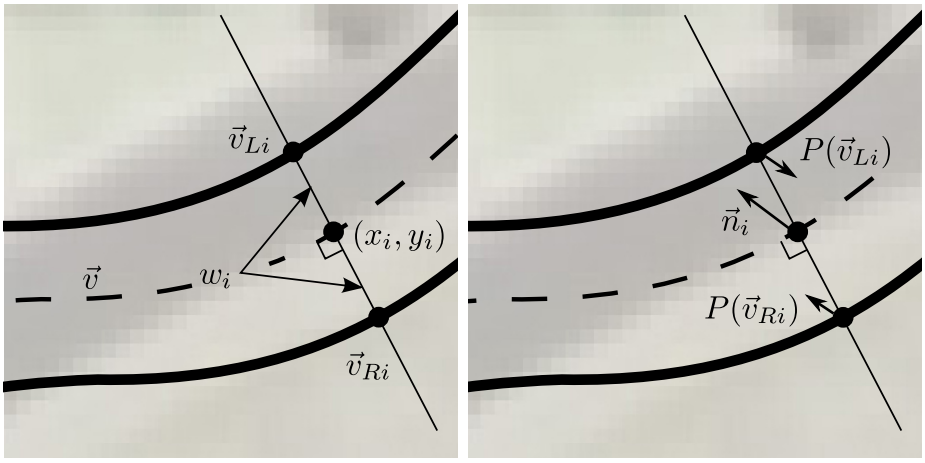
$$\vec{v}_{Li} = -w_i\vec{n}_i \tag{2.12}$$

$$\vec{v}_{Ri} = w_i\vec{n}_i. \tag{2.13}$$

Finally, the image energy function for a ribbon snake is defined as

$$E_{\text{img}}(\vec{v}) = -\int_0^1 \Big(P(\vec{v}_L) + P(\vec{v}_R)\Big)ds. \tag{2.14}$$

Now that ribbon snakes and their energy functions have been described, a number of studies using this method will be presented. Koutaki and Uchimura [90] use ribbon snakes to extract suburban road networks from aerial images with a ground resolution of 50 cm. Ribbon snakes are chosen, since roads appear as thin elongated surfaces in VHR imagery. The method begins by classifying the candidate scene with the ISODATA algorithm. Road intersections are then extracted with a model based matching method. A road tracking algorithm is used to initialize the snakes between the detected intersections. The image energy function, E_{img}, and internal energy function, E_{int}, are reduced by optimizing the snake positions and their shapes. E_{img} is computed by considering the classified ISODATA values within the snake in relation to the external classified values.



(a) Description (b) Forces

Figure 2.11: The components of ribbon snakes

Laptev *et al.* [94] use ribbon snakes to remove irrelevant structures extracted by a preliminary line detection algorithm at a coarse resolution. The method initializes a ribbon snake for each line detected and sets the width property to zero. The snake positions are optimized at a coarse scale to get a rough approximation of the road position. A second optimization process is used at a finer scale where the road position precision is increased and the width property expanded up to the structure boundary. Finally, road width thresholding is used to discard any irrelevant structures.

Summary

Snakes are inherently linear feature extraction techniques which makes them ideally suited to detect road structures in RS imagery. Snakes are often used to optimize the path between a set of road seed points, but can also be used to increase the road detection accuracy by discarding irrelevant structures. As found by Mayer *et al.* [107], snakes are particularly useful in bridging gaps caused by shadows or occlusions.

This concludes the review of the lower level road extraction methods. The next section introduces data fusion concepts.

2.4 Data fusion

Data fusion can broadly be defined as the process of combining a variety of data sources with the objective of obtaining more accurate measurements than a single source is able

to provide. The objective is to increase the accuracy of data sets through inference; conversely, data integration increases the data set size by merging multiple sources. Data fusion has been applied to various fields, including robotics, medicine, defence and remote sensing. The Society Data Fusion Technical Committee defines data fusion as [87]:

The process of combining spatially and temporally indexed data provided by different instruments and sources in order to improve the processing and interpretation of these data.

As with the road extraction approaches discussed in the preceding sections, data fusion can also be categorized according to the level at which processing takes place [87]. Low-level data fusion combines raw data, while mid-level fusion combines information derived from the raw data. High-level fusion operates on a cognitive level and involves processes such as the fusion of decisions. When considering Figure 2.1, the *Data Fusion* component is located at the mid-level and combines processed data from the *Geometrical Detector* and the *Spectral Classifier* components.

Klein [87] discuss a number of data fusion algorithms in detail, namely *classical inference*, *Bayesian fusion*, *Dempster-Shafer theory*, *Artificial Neural Networks*, *Voting Logic*, *Fuzzy Logic* and *Fuzzy Neural Networks*. These methods can be applied at various processing levels and can also be combined in numerous ways to produce novel fusion approaches. The most popular approaches, namely Bayesian and Dempster-Shafer fusion, will now be discussed in more detail. This discussion is followed by a few examples, which illustrate the practical applications of data fusion in RNE systems.

2.4.1 Bayesian fusion

Bayesian fusion combines multiple data sources through Bayes' theorem. The theorem is used to calculate the posterior probability of a single event given a number of observations. The probability of the events A and B occurring can be computed as

$$p(A, B) = p(A|B)p(B), \tag{2.15}$$

where $p(A)$ and $p(B)$ denotes the prior probability of the occurring events and $p(A|B)$ the conditional probability of A occurring, given that B already occurred [87].

Considering $p(A, B) = p(B, A)$, one can rewrite the equation as

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}. \tag{2.16}$$

Consider the example where two algorithms assign the probability values y_1 and y_2 to a pixel, which indicates its likelihood of belonging to the road pattern class x. The fusion of the output of the two algorithms can then be expressed as

$$p(x|y_1, y_2) = \frac{p(y_1|x)p(y_2|x)p(x)}{p(y_1, y_2)}. \tag{2.17}$$

By applying Bayes' rule again, y_i is swapped with x, and the following equation is obtained

$$p(x|y_1, y_2) = \frac{p(x|y_1)p(y_1)}{p(x)} \cdot \frac{p(x|y_2)p(y_2)}{p(x)} \cdot \frac{p(x)}{p(y_1, y_2)}. \tag{2.18}$$

The value $p(x|y_1, y_2)$ is the probability of the pixel belonging to x, given the output from the two algorithms y_1 and y_2.

The following section will discuss an alternative method, known as Dempster-Shafer fusion.

2.4.2 Dempster-Shafer fusion

Dempster-Shafer Theory (DST) is a theory of evidence, which is concerned with the combination of multiple pieces of empirical evidence with the objective of creating an accurate representation of reality. DST is based on the calculation of subjective probabilities or belief functions by Shafer [141] and the rule of combination by Dempster [35]. The method by Shafer is a generalization of Bayesian theory concerned with obtaining *degrees of belief*, and the method by Dempster integrates multiple beliefs into a single measure. Bayesian theory differs from DST in that it requires a probability for each question under consideration, whereas DST obtains the degree of belief through probabilities of a related question. The implementation of DST will now be discussed, starting with belief functions by Shafer and followed by the rule of combination by Dempster.

Shafer's belief functions

Consider a set Θ of n mutually exclusive propositions, also known as the *frame of discernment*, such that

$$\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}, \tag{2.19}$$

where θ_i represents each mutually exclusive proposition. The propositions within Θ must be exhaustive so that the closed world assumption holds.

The power set $\mathrm{P}(\Theta)$ includes all sub-set combinations of Θ. Consider the following example where $\Theta = \{\theta_1, \theta_2, \theta_3\}$. The power set is then defined as

$$\mathrm{P}(\Theta) = \{\emptyset, \theta_1, \theta_2, \theta_3, \theta_1 \cup \theta_2, \theta_1 \cup \theta_3, \theta_2 \cup \theta_3, \theta_1 \cup \theta_2 \cup \theta_3\}. \tag{2.20}$$

An empty or null set is represented by $\emptyset$.

A probabilistic function $m(A)$ is introduced, which assigns a mass value to any given member of $\mathrm{P}(\Theta)$. The mass value is indicative of the amount of evidence that an event belongs to A. This mass value is only applicable to A itself and makes no claims about any subset of A, which would have its own mass values.

According to Shafer, the mass value for a given element A_i allows two items of information to be deduced, namely an account of the amount of *belief* (also referred to as *support*) in the proposition A_i as well as the amount of belief in its negation $\overline{A_i}$. The degree of belief in a proposition and its degree of *plausibility* is an expression of these two items of information.

The basic belief assignment is defined as $m(\cdot) : \mathrm{P}(\Theta) \rightarrow [0, 1]$, which is subject to

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \in \mathrm{P}(\Theta)} m(A) = 1. \tag{2.21}$$

The belief function for a single proposition A_i is defined as

$$\mathrm{Bel}(A_i) = m(A_i). \tag{2.22}$$

When the proposition is either θ_1, θ_2, or θ_3 the belief function is defined as

$$\begin{aligned} \mathrm{Bel}(\theta_1 \cup \theta_2 \cup \theta_3) \quad = \quad & m(\theta_1) + m(\theta_2) + m(\theta_2) + m(\theta_1 \cup \theta_2) + \\ & m(\theta_1 \cup \theta_3) + m(\theta_2 \cup \theta_3) + m(\theta_1 \cup \theta_2 \cup \theta_3) \end{aligned} \tag{2.23}$$

Plausibility is defined as the belief in the negation of a proposition. Plausibility is defined as

$$\mathrm{Pl}(A_i) = 1 - \mathrm{Bel}(\overline{A_i}), \tag{2.24}$$

where $\mathrm{Bel}(\overline{A_i})$ represents the degree of belief in the negation of the proposition A_i.

Another method for calculating plausibility is through the summation of all masses of the subsets with a non-null intersection with A_i:

$$\mathrm{Pl}(A_i) = \sum_{A_i \in \mathrm{P}(\Theta), A_j \cap A_i \neq \emptyset} m(A_j). \tag{2.25}$$

Consider $\mathrm{P}(\Theta)$ as defined in Equation 2.20. The plausibility of θ_i can be calculated according to Equation 2.25 as

$$\mathrm{Pl}(\theta_i) = m(\theta_1) + m(\theta_1 \cup \theta_2) + m(\theta_1 \cup \theta_3) + m(\theta_1 \cup \theta_2 \cup \theta_3). \tag{2.26}$$

The *uncertainty* interval is consequently defined by the range $[\mathrm{Bl}(A_i), \mathrm{Pl}(A_i)]$, where $\mathrm{Bl}(A_i) \leq \mathrm{Pl}(A_i)$.

Dempster's rule of combination

The rule by Dempster allows the combination of Shafer's probability masses from multiple independent sources. The rule evaluates the intersection of propositions and selects the largest probability mass as the *joint mass*. Concurring propositions are therefore emphasized while any conflicting evidence is discarded through a normalization factor.

The joint mass of two sets of masses m_1 and m_2 can be calculated as

$$m(\cdot) = [m_1 \oplus m_2](\cdot) = \begin{cases} m(\emptyset) = 0 \\ m(A) = \frac{1}{1-K_{12}} \sum\limits_{\substack{X,Y \in \mathrm{P}(\Theta) \\ X \cap Y = A}} m_1(X) m_2(Y), \quad \forall (A \neq \emptyset) \in 2^{\Theta}. \end{cases} \tag{2.27}$$

The conflicting evidence between X and Y is calculated as

$$K_{12} \triangleq \sum_{\substack{X,Y \in \mathrm{P}(\Theta) \\ X \cap Y = \emptyset}} m_1(X) m_2(Y). \tag{2.28}$$

The normalizing factor in Equation 2.27 can result in a disproportionally large value when a high degree of conflict, K_{12}, exists between two masses. This behaviour can be

formalized as

$$\lim_{K \to 1} \frac{1}{1 - K_{12}} = \infty. \tag{2.29}$$

Considering that Dempster's rule is associative, it is possible to combine more than two mass values sequentially. The masses of $m_1(\cdot)$, $m_2(\cdot)$, and $m_3(\cdot)$ for a proposition A can be calculated so that $m(\cdot) = [(m_1 \oplus m_2) \oplus m_3](\cdot)$.

A discussion of the practical applications of data fusion in RNE systems is provided in the following section.

2.4.3 Fusion in road extraction literature

Hellwich *et al.* [65] present an approach where linear features are extracted from SAR imagery. The authors note a study by Hendry *et al.* [66] which illustrates the complementary nature between SAR intensity and interferometrics coherence data. This finding is enforced by Hellwich *et al.* who discovered a weak correlation between the two data sources, where linear features are often visible only in either data source. Bayesian data fusion is used to combine the two sources automatically. The prior probability for each data source is modelled by a Markov Random Field (MRF)name=MRF, description=Markov Random Field [84]. The MRF model is designed to give an indication of the probability of a line running through a pixel (line strength) and the local neighbourhood. The MRF is further designed to suppress line gaps caused by the sensor noise. The final result is an image where each pixel consists of the combined line strengths and directional information.

Data fusion is used at pixel level in a system by Mena and Malpica [115]. Three classifiers are proposed, each producing a per pixel pseudo-probability value in the range $[0, 1]$. The classifiers receive an RGB IKONOS image as well as a road model as input. Dempster-Shafer fusion is used to combine the results produced by the classifiers. The fusion algorithm requires an uncertainty value for each classifier and a mass value from each pixel. The uncertainty value is indicative of the level of confidence a classifier has in its own classifications. This uncertainty value is obtained by evaluating the ability of the classifier to classify the road model accurately. The pseudo-probability value assigned to each pixel is used as the mass value. The fusion process produces a plausibility value for each pixel which is indicative of the likelihood of a pixel belonging to the road model class.

An unconventional usage of fusion in RNE systems is presented by Gerke [48]. The author proposes a quality metric framework that allows objects in feature space to

be assessed automatically. An existing GIS road database is evaluated against roads extracted from recent aerial imagery through an automated process. The extracted roads are therefore used as a reference set as described in Section 2.6. The automated extraction system might extract more than one road object for a single database road object. Applying a quality metric between each extracted object and database object would lead to multiple measures which will then have to be combined to obtain a single measure. Both Bayesian and DST inference models are considered as inference models. Gerke concedes that a comparison between the two approaches is subjective, but notes that DST produced a more rational result.

Various other data fusion approaches can be found in literature. A number of studies used less sophisticated fusion techniques such as rule-based logic [21, 55], superimposition [76, 100], fuzzy logic [159] and mathematical morphology [25]. The selection of a fusion technique depends greatly on the amount and type of information under consideration.

2.4.4 Summary

Data fusion is a useful tool when extracting features from RS imagery. Features can be extracted by disparate data sources and extraction algorithms resulting in multiple representations of a single object. Data fusion provides the means for combining these representations into a single object. Two prominent data fusion techniques and their applicability to road extraction were presented in this section. Data fusion is not limited to these techniques and diverse methods can be found in literature. The selection of a fusion technique depends greatly on the problem at hand.

2.5 Road network construction

As mentioned in Section 2.1 road extraction systems consist of algorithms at various levels. Road Network Construction (RNC) techniques are usually higher level algorithms used to create a topologically sound network. The intermediate result obtained by the low and mid-level algorithms is usually not in a practically usable form and requires further processing to detect the road centrelines and to ensure network integrity. In a sense, RNC algorithms attempt to emulate the human ability to construct networks by using a priori information, DIP and Computational Intelligence (CI).

Considering Figure 2.1, the intermediate result produced by the *Data Fusion* component would require further processing before the road data set can be used in a

practical GIS. The image contains some background noise, incorrectly classified road sections and gaps where occlusions have occurred. The *Centreline Extraction* and *Network Completion* components can be considered as RNC techniques, which produce a topologically correct road network.

The following two sections consider firstly, a few approaches where raster data is converted into vector space and, secondly, a number of methods to construct and verify the topological structure of the road network.

2.5.1 Vectorization

Low and mid-level road extraction techniques, such as classifiers and segmentation algorithms, produce a raster-based output. As stated before, the purpose of RNC algorithms is to construct topologically sound road networks, which implies that they consider road objects in a feature space. Representing feature data in a raster format is not ideal and roads are often converted into feature space as sets of vectors. This conversion process is also known as *vectorization*. The results produced by certain low and mid-level algorithms, such as road trackers and snakes, are already in feature space, and no vectorization is therefore required.

Mena [113] proposes a process chain to convert a binary segmented image to a road vector data set. This processing chain consists of the following six stages:

1. **Cleaning stage:** A bit of noise is evident after a thresholding step and is removed before vectorizing.

2. **Pixel classification and edge smoothing:** Four classes are defined (*road*, *edge*, *frame* and *ground*) and each pixel is classified accordingly. The pixels in the edge set are smoothed.

3. **Skeleton extraction:** Through skeletonization the initial road centrelines are extracted from the pixels in the road class. The skeletonization process involves a thinning step, and pruning to remove short branches. After skeletonization, the system detects the centreline end nodes and junctions.

4. **Graph construction:** A graph is created by considering the edges between the nodes. If an edge is found to have a high curvature, more nodes are inserted to reduce the curvature between nodes. The slope between neighbouring edges is used to assign an “opened” or “sharp” class to each edge.

5. **Geometrical adjustment:** Edges classified as "opened" are defined as polynomials, while "sharp" edges are defined as polygons. Both the polynomials and polygons are iteratively adjusted at a sub-pixel level, to fit the underlying centreline.

6. **Topological adjustment:** Finally, the curvature of the road is evaluated at the nodes to ensure a topologically correct road network.

Christophe and Inglada [28] propose another novel approach, which begins with creating a set of vectors for each straight line segment. The initial vector positions are found by considering the weighted centre of mass of a number of pixels in a given direction. The centre of mass is obtained by calculating the mean value for each band. The directional information is obtained by calculating the original image gradient. The method uses sub-pixel accuracy, which reduces the number of vectors needed to represent a straight line segment. The gradient image tends to contain a lot of noise, which causes too many short vectors to be extracted. A cleanup process is used to merge sets of vectors with similar gradients. The result causes a number of sharp angles between connected vectors, which is reduced by splitting vectors where sharp angles are detected. The road network contains a number of gaps owing to the lower level extraction algorithms used. The vectorization algorithm attempts to breach these gaps by connecting the endpoints of vectors in close proximity with similar gradients. If a new connection creates sharp angles along the path, it will not be incorporated into the network. Finally, the network is evaluated and any remaining short paths are removed.

This section has presented two vectorization methods which convert the results obtained from lower level algorithms into feature space. The following section considers a number of RNC techniques which are capable of creating a topological sound road network.

2.5.2 Heuristic road network construction

Wiedemann and Hinz [165] propose a method of creating topologically correct networks by considering new possible road links (link hypotheses). The approach is based on the fact that road networks are normally constructed to provide the shortest path between points of interest (nodes). A concept called *detour factor* is introduced, which is defined as

$$\text{detour factor} = \frac{\text{network distance}}{\text{optimal distance}}.$$

The *network distance* is the summation of all the lengths of the road segments. The *optimal distance* is the shortest distance between two nodes. The detour factor will yield a high value when the route distance between two nodes is long and the Euclidean distance is small. A new link hypothesis is made between two nodes when the detour factor is above a certain threshold and if there is no other competing link hypothesis with higher detour factors. In addition, link hypotheses between nodes with a high Euclidean distance are discarded. The link hypothesis is then sent to the evaluation module for verification. A method by Wiedemann *et al.* [163] is used as the evaluation module. When a link is added the entire process is then repeated, since the topology of the network has changed. The process ends when no new link hypotheses are suggested. This RNC technique is also used in a number of other studies [32, 67, 162].

Yang and Wang [167] propose an RNC method for two types of structures called *blobs* and *lines*. A blob is defined as a squared segment with a vector running through the centre point. Lines are sets of vectors which have the same representation that one would obtain from a vectorization process. The blob vectors are linked by considering the cost function by Zhao *et al.* [176], given as

$$E_{ij} = |\alpha_{ij}| + |\beta_{ij}| + |\theta_{ij}|,$$

where α_{ij} is the distance between centre points of blobs i and j, β_{ij} is the cover rate and θ_{ij} the angle. Blob vectors are linked where the cost E_{ij} is lower than a specified threshold. The line vectors are connected by using a Genetic Algorithm (GA). Yang and Wang argue that most vector linking approaches determine whether a new link is credible by using constant thresholds, which lacks the necessary accuracy. A GA is used to validate new possible links between existing vectors. The method begins by selecting an initial vector $\vec{v}$ as well as n vectors ($\vec{w}$) in close proximity, which could be used to create new links to the initial vector. A chromosome is created for each $\vec{v}$ and consists of n binary numbers. Each binary number indicates whether the corresponding new link will be added or not. The fitness function incorporates four characteristics derived from the $\vec{v}$ and $\vec{w}_n$ namely, *proximity*, *parallelism*, *overlap* and *relative length* (the reader is referred to the original work for more detail). The GA is optimized by using selection, crossover and mutation. If the fitness of the GA remains unchanged for fifteen generations or evolves for 200 iterations, the optimization process is stopped. The optimized chromosome is used to create the links where the binary numbers are equal to one. If the road network structure changes, the process is restarted and repeated until no

more changes occur. Finally, the blob vectors and line vectors are merged into a single road data set.

Oddo *et al.* [122] present a novel RNC system which constructs a network from a raster image, rather than a set of existing vectors. The system combines spectral classification, *k-medians* clustering (a variation of the k-means algorithm), graph theory and geometrical model matching to delineate roads from high resolution RS imagery. The method begins by applying a ML spectral classifier to the scene, which produces a binary image with a *paved road* and *non-road* class. A grid of clusters is initialized over the image and the k-medians algorithm is used to find road segments. A Minimum Spanning Tree (MST) is created by connecting the road cluster centre points. A score is assigned to each link, which is based on the neighbouring cluster scatter measure (variance) and inter-cluster centre connection strength.

A fast, raster-based RNC system is proposed by Géraud and Mouret [47]. Their system uses watershed transformation and MRFs to evaluate the road segments, as well as the way in which they form a network. First, a "potential" image is obtained, which indicates the likelihood of a pixel being a centre road pixel. A filtering step is then applied by using the closing mathematical morphology operator and watershed transformation. A road network of connected line segments is obtained through this process. All non-road segments are removed using MRFs by expressing the network in terms of a global energy minimization problem. The energy of road segments is expressed according to its original "potential", local and global curvature and contrast with respect to nearby regions. Finally, a global road network is obtained by reducing the network energy through the simulated annealing algorithm.

2.5.3 Summary

RNC techniques are concerned with producing the final road network in a usable format and they form an integral part of RNE systems. The current state-of-the-art RNC techniques exhibit numerous heuristic traits, but are still inferior in comparison to the cognitive abilities of humans. RNC methods tend to be computationally expensive and this should be taken into consideration when developing RNE systems.

2.6 Quality metrics

Generally speaking, quality metrics quantifies the accuracy or performance of a system or a set of results. When considering Figure 2.1 the accuracy of the final output produced by the *Road Completion* component has to be quantified. Quality metrics allow this quantification to be made through a comparison of extracted roads and an accurate reference set (ground truth). These metrics are vital in the quest to develop a fully automated RNE system since they provide the means needed to compare different approaches. Harvey [60] notes that qualitative analysis is useful, but that it is burdened by subjective bias and inconsistencies. The following section considers a number of RNE quality metrics.

2.6.1 Quality metric approaches

Before describing some of the quality metrics, it seems useful to define the elements of the confusion matrix [44]. The quality of extracted roads are often quantified in terms of these elements [60, 63, 179]:

True Positive (TP): The extracted road section is consistent with the reference set.

True Negative (TN): The areas where no roads are extracted is consistent with the reference set.

False Positive (FP): The extracted road section is inconsistent with the reference set.

False Negative (FN): The areas where no roads are extracted is inconsistent with the reference set.

Heipke *et al.* [63] present a popular quality assessment method used by Bacher and Meyer [8], Doucette *et al.* [38], Jin and Davis [76], and Mena and Malpica [115], amongst others. The metric derives a quality measure by comparing extracted centrelines to a ground truth reference set. To account for the minor discrepancies that occur between extracted roads and the ground truth, a 'buffer zone' around the reference set is created. The defined quality metrics comprises the following measurements:

Completeness is a measurement of the percentage of the reference road network that has been extracted.

The completeness metric is calculated as

$$\begin{aligned} \text{completeness} &= \frac{\text{matched } L_e}{L_r} \\ &\approx \frac{TP}{TP + FN}, \end{aligned}$$

where completeness $\in [0, 1]$. L_e is defined as the length of the extracted line segment and L_r the length of the reference segment. A completeness value of one indicates that the reference network is entirely covered by the buffer area around the extracted network.

Correctness serves as a measure of the precision of the extracted road network. In effect, the percentage of the extracted road network which is covered by the ground truth is measured.

The correctness metric is calculated as

$$\begin{aligned} \text{correctness} &= \frac{\text{matched } L_e}{L_e} \\ &\approx \frac{TP}{TP + FP}, \end{aligned}$$

where correctness $\in [0, 1]$. A correctness value of 1 indicates that the extracted network is entirely covered by the buffer area around the reference network.

Quality is a measurement that combines completeness and correctness.

The quality metric is calculated as

$$\begin{aligned} \text{quality} &= \frac{\text{matched } L_e}{L_e + \text{unmatched } L_e} \\ &\approx \frac{TP}{TP + FP + FN}, \end{aligned}$$

where quality $\in [0, 1]$. A quality value of 1 indicates that the extracted network lies within the buffer around the reference network and vice versa.

Redundancy is an indication of the amount of overlap in the road network, which occurs if a given road section is extracted multiple times.

The redundancy is calculated as

$$\begin{aligned} \text{redundancy} &= \frac{L_e - \text{matched } L_r}{\text{matched } L_e} \\ &\approx \frac{2 \times TP + FP}{TP}, \end{aligned}$$

where redundancy $\in [0, \infty]$. A value of 0 indicates no overlap and is naturally the optimum.

Root mean square differences give an indication of the average distance between the reference and extracted networks, which is indicative of geometrical accuracy. Root Mean Square (RMS) is a statistical measure of the degree of variance of a given parameter.

The RMS is calculated as

$$\text{RMS} = \sqrt{\frac{\sum_{i=1}^{l} d(e_i, r)^2}{l}},$$

where RMS $\in [0, b]$. The parameter l is defined as the number of matched parts extracted, and $d(e_i, r)$ the shortest distance between the i-th piece of the reference set and the extracted roads. The optimal value for RMS is 0, indicating that the extracted road lies on top of the reference set. The maximum value depends on the size b of the buffer zone.

Gap statistics comprises the mean number of gaps per kilometre as well as the mean gap length. The mean number of gaps is calculated as $n/L_r(\text{km})$, where n is the number of gaps and L_r the length of the reference set in kilometres. To calculate the mean gap lengths, all gap lengths are added and divided by n.

Harvey [60] used the completeness, correctness, quality and redundancy metrics by Heipke *et al*, but found the following additional parameters useful in determining road quality:

Branching factor is a ratio of the correctly and incorrectly identified roads. The formula is given as

$$\text{branching factor} = \frac{FP}{TP},$$

where branching factor $\in [0, \infty]$. The objective would be to obtain a branching factor of 0. This metric is not ideally suited to solely describe the quality of the extracted roads, since the total length of road network is not considered. A partly extracted network without any FP roads would score higher than a fully extracted network with a few FP roads.

Rank distance is an additional measure of the overall quality of the extracted roads, and is computed as

$$\text{rank distance} = \sqrt{\frac{completeness^2 + correctness^2}{2}},$$

where rank distance $\in [0, 1]$.

Quality metrics need not compare vector-based data sets as is evident by the method presented by Zoej and Mokhtarzade [179]. A spectral ANN classifier is used to classify road surfaces in VHR imagery. The ground truth is manually created as a binary image, where road pixels are marked as 1 and the remaining pixels (non-road) as 0. Three metrics are created to measure classification accuracy, defined as:

Road correctness coefficient: By multiplying the classified binary image with the ground truth the Road Correctness Coefficient (RCC) is obtained, which is equivalent to the TP measure.

Background correctness coefficient: By multiplying the classified binary image with the inverse ground truth the Background Correctness Coefficient (BCC) is obtained, which is equivalent to the TN measure.

RMS: The RMS is the overall error rate of the classifier, which is the difference between the classified binary image and the ground truth.

Even though it is possible to develop raster-based quality metrics, the reference set would be more subjected to the perils of quantitative analysis. In this example, even a near-perfect road classifier would struggle to obtain a RMS error of zero.

2.6.2 Summary

Quality metrics are techniques used to assess the accuracy of results produced by RNE systems. The ability to quantify results allows comparisons to be drawn between methods

in an unbiased, consistent manner. Various methods have been proposed and include approaches based on both vector and raster data.

2.7 Optimization

Optimization is the process during which the accuracy or efficiency of a system is improved. Various aspects of a system can be optimized, including processing time, memory usage, efficient usage of external resources (such as databases or remote systems) and quality of results. The set of all possible system configurations is known as the *search space* [125]. Optimization algorithms are typically iterative processes capable of moving effectively through the search space towards a more optimal solution. During each iteration, a system parameter is modified and the effect on the system measured. The process is repeated with the modifications to the system parameter based on the measured effects of the previous iteration. This iterative process continues until a certain stopping criterion has been reached. Even though an extensive range of optimization algorithms have been developed, it is rarely the case that a truly optimal solution is reached.

During road extraction numerous input parameters are required for the algorithms within the processing chain. Optimizing these parameters has an impact on the quality of the results produced. In most RNE systems, the exact correlation between the parameter values and the final quality of the results is unknown. To obtain and optimize these parameters, the entire road extraction processing chain would have to be executed. When the system parameters are uncorrelated, it is possible to optimize each value independently. It is often the case that a correlation between parameters exists, which increases the search space dramatically. A larger search space ultimately results in an increase in the number of iterations required to find a satisfactory result. Considering a representative RNE system such as the hypothetical processing chain presented in Figure 2.1, it is clear that optimizing correlated parameters in such a system would be an extremely expensive process computationally. An efficient optimization algorithm, capable of optimizing multiple parameters simultaneously would therefore be needed.

The minimalistic Golden Section Search (GSS) algorithm is effective in finding possible areas where an optimal solution could be found. The Nelder-Mead simplex algorithm is an efficient method with the capability of optimizing multiple parameters simultaneously. A short discussion of these two methods follows.

2.7.1 Golden section search

GSS [82] is a search algorithm which finds the minimum or maximum point (extremum) within a unimodal function. In terms of optimization, the extremum would be the point in the search space where an optimal solution could be found. The method iteratively narrows the search space where the extremum is located. The algorithm name is derived from the *golden ratio* which is used in the calculation to determine the next search location.

The algorithm will be explained at the hand of the function $f(x)$ illustrated in Figure 2.12. The algorithm is initialized by specifying three points (a, b and c) within a search interval in which the extremum (in this case minimum) is expected to be found (positions 1, 2 and 3). A new point z is selected in the larger of the two sub-intervals (a, b) and (b, c). If $f(z) < f(b)$, then z replaces b. If $f(z) > f(b)$, then z replaces the nearest endpoint (either a or c). This process is repeated until the bracketing window reaches a certain specified threshold. The search bracket in Figure 2.12 begins with (1, 2, 3), which becomes (4, 2, 3), and finally ends with (4, 2, 5).

Press *et al* [131] show that the optimal interval reduction rate is achieved if the new point z is selected as a proportion of the golden ratio, and the larger sub-interval calculated from point b. The golden ration is defined as

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.618033989\ldots$$

GSS is an efficient search algorithm, but is only capable of optimizing a single parameter at a time. GSS also runs the risk of being trapped in a local minimum. GSS is ideal for

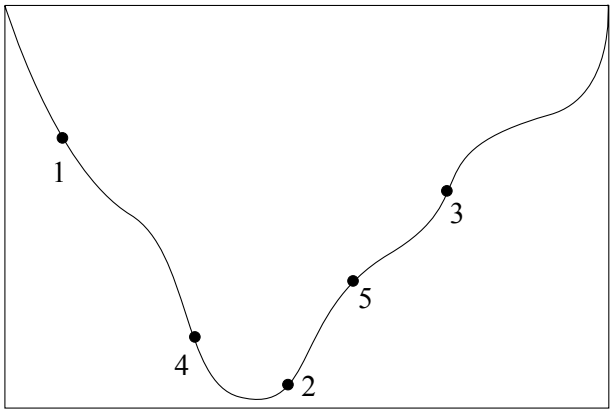


Figure 2.12: Golden section search

finding possible ranges where the extremum could be located. This information could then be used to initialize a method such as Nelder-Mead that is able to search across multiple parameters and is less susceptible to local minima.

2.7.2 Nelder-Mead

The Nelder-Mead [120] is a well-known nonlinear multidimensional unconstrained optimization algorithm. The algorithm does not use derivatives, making it ideal for optimizing non-smooth functions. The algorithm converges on non-stationary points around a local optimum and should therefore only be considered as a heuristic. The method is based on a *simplex*, which is a *convex hull* (special polytope) with $n + 1$ vertices, where n is the number of dimensions. A simplex would be a line segment in 2D and a triangle in three dimensions (3D).

The algorithm begins by initializing a simplex with each point lying on different hyperplanes. The method begins an iterative transformation process that tries to locate an extremum area. The simplest step in the process is taken by calculating the values at each point and replacing the worst point with a new point obtained by reflecting the point through the centroid of the remaining points. The new point is evaluated and if proven to be better than the current best, the point is extended further. If the new point is at a worse location than the previous one, the simplex is shrunk in the direction of a better point. The process ends when the simplex becomes smaller than a given threshold, or if the difference between the function values at each point becomes smaller than a set threshold.

The Nelder-Mead algorithm is efficient in that it requires only one or two function evaluations per step, where other approaches could require up to n evaluations. The ability of the algorithm to optimize multidimensional problems combined with its level of efficiency makes it an ideal method in determining the parameters in RNE systems.

This section has presented two optimization methods which can be used in combination to optimize individual or multiple system parameters. The following section presents a summary of the sensor types used in road extraction studies.

2.8 Sensor types

When considering some of the techniques used to perform road extraction (refer to Section 2.3) it is clear that an array of sensor types and derived data products can be

used. A mini-survey of the sensors used by the research groups (or affiliated institutions) to conduct road extraction, referenced in this thesis, was conducted.

A total of 62 research groups were identified, with some groups using up to four different sensor types over a number of studies. The results are presented in chart-form in Figure 2.13. IKONOS data is used by 20 groups and aerial imagery by 19 groups. QuickBird, SAR and SPOT (Satellite Pour l'Observation de la Terre) data is used by ten, seven and six groups respectively. LiDAR (Light Detection and Ranging) is used by three groups, and the IRS (Indian Remote Sensing Satellite) by two. Other data types include Landsat, EO-1 (Earth Observing Mission 1), KompSat EOC and KVR-1000 satellites. A number of the more frequently used sensors will now be described in more detail.

2.8.1 IKONOS

IKONOS is a commercial satellite which was launched in September 1999. The device features a panchromatic and multispectral sensor that captures data at a resolution of 1 m and 4 m respectively. The multispectral sensor captures the RGB bands as well as a Near Infrared (NIr) band. A pan-sharpened version of the data is available, where the panchromatic and multispectral data is combined to produce a 1 m resolution colour product. IKONOS provided the highest satellite commercial imagery until 2001 when QuickBird was launched. The popularity of the images obtained with this sensor is evident in the statistics.

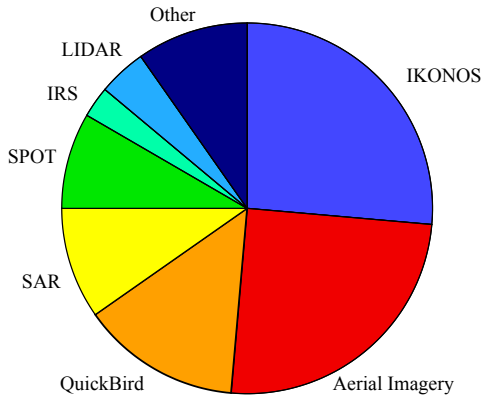


Figure 2.13: Satellite sensor types used by research groups

2.8.2 Aerial imagery

Aerial photography is one of the earliest forms of remote sensing and dates back to 1858 when the French balloonist Gaspard-Félix Tournachon took the first aerial photos of Paris. Modern aerial imagery (also referred to as orthoimagery) is captured by sophisticated sensors. A diverse range of sensors exist, which are able to collect information at various wavelengths. The sensors are typically mounted on an aircraft. The size of the equipment has also been reduced significantly, and the size of aircraft needed to carry them has consequently also been reduced. The resolution of the captured imagery depends largely on the height at which the aircraft is flying. The highest ground resolution found in the available literature was around 20 cm [9]. Aerial imagery has the highest resolution available in the visible bands, making it ideal for RNE.

2.8.3 SAR

As its name suggests, SAR is a radar-based technique that uses electromagnetic waves to determine certain characteristics of remote objects. In its simplest form, SAR calculates the distance between the sensor and the object by sending wave pulses towards the object and measuring the time it takes for the wave to be reflected. Considering that objects will reflect the wave with different intensities and polarizations, various additional characteristics can be derived. *Synthetic aperture* is a concept where multiple observations are combined to create a single, seemingly large observation. In SAR, a large antenna is emulated by emitting a series of signals from a small antenna onboard a moving vehicle, such as an aircraft or a satellite. The set of reflected signals received by the sensor are combined into a single, large radar image. The current state-of-the-art airborne systems manage a ground resolution of around 10 cm, while the RADARSAT-2 satellite has a ground resolution of 1–3 m.

2.8.4 QuickBird

QuickBird is a commercial satellite that was launched in 2001. As the IKONOS satellite, QuickBird has a panchromatic sensor and a multispectral sensor with resolutions of 64 cm and 2.4 m respectively. A pan-sharpened product is available at a 61 cm resolution at nadir and 72 cm at an offset angle of 25°. Since its launch in 2001, QuickBird has gained popularity amongst RNE researchers. QuickBird provided the highest resolution commercial satellite imagery until the launch of GeoEye-1 in September 2008. GeoEye-1

has a 41 cm panchromatic and 1.65 m multispectral sensor onboard.

2.8.5 SPOT

SPOT is a French high resolution satellite system. To date, the system utilizes five satellite instruments. SPOT-1 was launched in 1986 and the most recent SPOT-5 in 2002. SPOT-1 to SPOT-4 featured a panchromatic and visible spectrum sensor of 10 m and 20 m respectively. The SPOT-4 instrument featured a number of improvements and included an additional mid-infrared band. SPOT-5 features a panchromatic and multispectral sensor with a 2.5–5 m and 10 m resolution respectively. Only larger roads, such as freeways, are visible at the lower resolutions produced by the early SPOT sensors, but smaller roads are visible in the images produced by SPOT-5.

2.8.6 LiDAR

The optical LiDAR sensor is typically mounted on an aircraft and uses laser pulses to detect objects by measuring the properties of scattered light. It employs the same principle as radar technology, where the distance to an object is determined by the time it takes for the signal to be returned. LiDAR makes use of light rather than the radio waves used by radar. Even though LiDAR sensors are very sensitive instruments, they might not always be the ideal data type from which to extract roads. LiDAR data is more useful where the height difference between the roads and the neighbouring objects is more significant. Another point worth noting is that each data point gathered by the instrument does not represent an area, as is the case with some of the sensors that operate in the visible spectrum. Satellite mounted systems are also currently in use, but are geared towards ice (ICESat) and aerosol and cloud monitoring (CALIPSO).

2.9 Conclusion

As is evident from the topics discussed in this chapter, the process whereby roads are extracted from remotely sensed imagery is elaborate and involves a number of techniques from numerous fields within computer science. The chapter began with a brief history which was followed by a discussion on common techniques used to identify roads in RS imagery. Data fusion and the way in which it relates to RNE were discussed. A number of topology verification (RNC) approaches and their evaluation were considered. Finally,

two parameter optimization techniques and popular sensor types were considered. This review is by no means exhaustive and serves only as an introduction to commonly used RNE techniques and to highlight some of the more noteworthy works in the field. The following chapter will discuss the components used in developing a fully automated RNE system.

CHAPTER 3

Road extraction components

"Computer Science is no more about computers than astronomy is about telescopes."
— E. W. Dijkstra

Further to the background and literature review presented in the previous chapter, a number of methods are selected to form part of a dynamic road extraction system. These selected methods are implemented as autonomous components, enabling them to be stringed together in numerous image processing chain configurations. To investigate the possibility of increasing the accuracy of RNE systems, different component arrangements are tested in subsequent chapters.

Section 3.1 presents a more extensive introduction, followed by a discussion on edge detectors in Section 3.2. ACE, the geometric road extractor driven by edge detection, is presented in Section 3.3. A fast image segmentation technique based on human perceptual grouping is described in Section 3.4, while Section 3.5 discusses three spectral classifiers. The Dempster-Shafer component and the manner in which it combines the

three classifiers are discussed in Section 3.6. Section 3.7 focuses on two mathematical morphology filters used during post-processing, while Section 3.8 explains the SORM heuristic road network construction technique. The quality metric implementation is discussed in Section 3.9 and the data set used in this thesis is presented in Section 3.10. The applications and software frameworks employed are reported in Section 3.11. The chapter concludes with Section 3.13.

3.1 Introduction

Various techniques used to extract roads from RS imagery have been presented in Chapter 2. A number of these techniques were implemented and used to form part of a dynamic RNE system (refer to Section 3.11 for more implementation details). The system is based primarily on the works by Doucette *et al.* [38], and Mena and Malpica [115]. A number of additional components were also employed. These include the long range edge detector by McKinstry and Guest [110], the image segmentation by Felzenszwalb and Huttenlocher [41], the quality metric by Heipke *et al.* [63] and mathematical morphological filters [106, 138].

These methods were implemented as modular, exchangeable components. To prevent the creation of illogical connections in the processing chain, only task-specific components are allowed to be interchangeable. The generic nature of the components facilitated the necessary flexibility to test a number of different system configurations.

The subsequent chapters investigate the possibility of increasing the accuracy of RNE systems by using original extraction techniques. The following sections discuss each of these components, as well as the data and software used. The first of these sections considers two edge detection approaches, which are employed within the geometrical detector described in Section 3.3.

3.2 Edge detectors

As discussed in Section 2.3.3, edge detection is frequently used to facilitate the RNE process. One of the most popular edge detection algorithms in image processing and road extraction is proposed by Canny [23]. The Canny algorithm is also used in the ACE geometrical centrepoint detector by Doucette *et al.* [38], which was introduced in Section 2.3.1 and discussed in greater depth in Section 3.3.

As with many other edge detection approaches, the Canny algorithm considers intensity changes within a local area in digital images. A different edge detection approach, inspired by the primary visual cortex in primates, is proposed by McKinstry and Guest [110]. The primary visual cortex is abbreviated as V1. Chapter 4 considers the possibility of obtaining more accurate road extraction results by substituting Canny with V1 in the ACE algorithm [38].

Figure 3.1 depicts an example of the edges produced by the Canny and V1 algorithms. Please note that the V1 edge detector and Canny algorithm are simply referred to as *V1* and *Canny* in the remainder of this thesis. The following two sections provide a detailed description of the Canny and V1 algorithms.

3.2.1 Canny edge detector

Canny [23] defines three requirements that an accurate edge detector should fulfil. These are *few errors*, *localized edge points*, and *single edge points*. Canny seeks to formalise these requirements and consequently holds that the first derivative of a Gaussian function serves as a suitable approximation for the optimal edge detector. In practice, the first derivative of the Gaussian function is implemented by first smoothing the input image with a Gaussian filter, and subsequently computing the finite difference derivative.

Canny developed an algorithm that describes each of these requirements mathematically. His algorithm can be summarised in the following six steps:

1. Blur the image with a Gaussian filter with a specified variance σ.

2. Compute the finite difference derivative in the x and y directions to yield the

(a) Standard test image (b) Canny edges (c) V1 edges

Figure 3.1: An illustration of the edges extracted by the Canny and V1 algorithms

directional derivative approximations G_x and G_y.

3. Compute the edge magnitude as $G = \sqrt{G_x^2 + G_y^2}$, and the edge direction as

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right), \quad G_x \neq 0.$$

4. Discretize the gradient direction θ of each pixel into one of eight possible neighbouring sectors.

5. Perform non-maximum suppression on the gradient magnitude G by suppressing along the normal to the discretized gradient direction all pixels with neighbours that are not all strictly smaller than the current pixel. This effectively 'thins' the edge magnitude image to a single-pixel width.

6. Apply hysteresis to the remaining non-maximum gradient magnitudes. The hysteresis process marks a pixel as an edge pixel if its magnitude value exceed the high threshold T_{high}. The neighbours of the edge pixels are subsequently traced and labelled as edge pixels if their magnitudes remain above the lower threshold T_{low}.

The algorithm output is a bitmap of single pixel-width edges, along with the edge orientation, θ, for each of these edge pixels. The Canny edges illustrated in Figure 3.1b were created with the following parameter configuration: $\sigma = 1.5$, $T_{\text{low}} = 0.0$, and $T_{\text{high}} = 0.4$.

3.2.2 V1 Neural Network

McKinstry and Guest [110] propose an ANN-based edge detector that is geared to extract long range edges, rather than localized edges. The edge detector is modelled according to features found in V1, which is the area in a mammal's brain concerned with processing visual stimuli. These features are identified as:

Complex cells represent approximately 75% of the cells in V1 [110]. The complex cells module comprises a simple cell orientation filter layer and a complex cell response layer.

Feature maps are modelled as an arrangement of complex cells in a two-dimensional feature map.

Long-range connections are determined within the feature maps through the Self-Organizing Map Extended (SOME) algorithm [109], which is based on the Self-Organizing Maps (SOM) algorithm originally proposed by Kohonen [89].

The algorithm steps are discussed below with reference to Figure 3.2:

1. Each pixel is represented as a neuron in the input layer. The neuron output is the greyscale intensity value of the corresponding pixel.

2. Simple cells serve as an orientation filter within the complex cell module and are modelled by Gabor functions. The simple cell response is computed as

$$s_\phi = f(g_\phi * I),$$

where I is the image intensity, $*$ a two-dimensional convolution of the Gabor function (g), and the phases (ϕ) 90^o and -90^o. The Gabor function is computed as

$$g(x, y) = e^{-\pi\sigma^2(x^2+y^2)} \cos(2\pi\omega[x\cos\theta + y\sin\theta] + \phi),$$

where θ is the orientation, ω the spatial frequency, and σ the Gaussian variance.

3. The Gabor Receptive Field (GRF) is developed to model complex cell responses by computing the dot product of simple cell responses, s_ϕ (difference of Gaussian filter), and the Gabor Function, $g(x, y)$. The GRF output is defined as

$$c = f(g * s_{90} + g * s_{-90}).$$

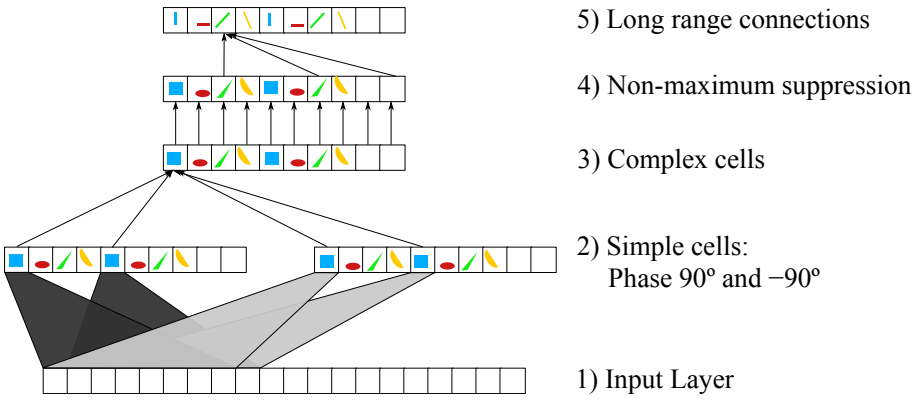


Figure 3.2: Integrated V1 model

A connection is formed only between complex and simple cells if they have corresponding orientations. This process can be considered as the step where edge detection takes place, since collinear simple cells with similar orientations are clustered together.

4. As with the Canny edge detector (refer to Section 3.2.1), non-maximum suppression is used to thin edges and find local maxima in the complex cells layer.

5. The GRF complex cells (neuron) are represented in a two-dimensional feature map. Every input pixel is used to optimize sixteen neurons to favour different line stimulus orientations within the feature map. SOME is used to detect the long range connections between these neurons. The SOME algorithm allows local connections to be made between neurons in addition to the global connections in the original SOM approach. Please refer to the original works of Kohonen [89] and McKinstry and Guest [109] for more information on these algorithms.

The result produced by V1 is a binary image displaying edges where neurons with high synaptic activity occur. Due to the nature of the SOME algorithm, high synaptic activity occurs over long range edges. This behaviour is evident when comparing the edges produced by Canny and V1 in Figure 3.1.

Considering the nature of the algorithm, it has the following three favourable traits when attempting to extract roadside edges from RS imagery:

- The ability of V1 to extract long range edges makes it ideal for identifying the long edges associated with road structures.

- V1 is not geared to detect short edges that often occur in non-road areas. Fewer short edges could translate into a reduction in the number of incorrectly detected road structures.

- When a local area edge detector is used, disturbances along the road edge, such as trees, shadows or cars, tend to result in breakages. V1 may have the ability to bridge disturbances of this nature.

This concludes the discussion of edge detection algorithms for the purpose of this thesis. The following section presents a geometrical road centrepoint detector, which is driven by both the edge location and orientation provided by an underlying edge detector.

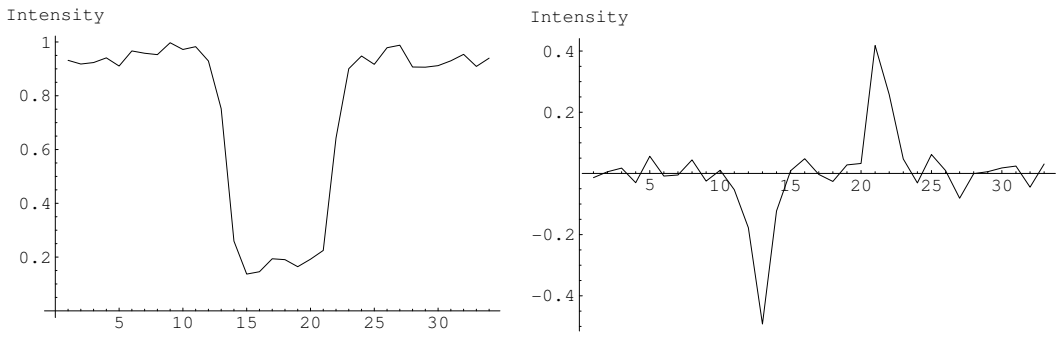
3.3 ACE

As mentioned in the discussion of automatic seeding techniques (refer to Section 2.3.1), the ACE algorithm by Doucette *et al.* [37, 38] is designed to detect road centrepoints by considering the geometrical properties of road structures. More specifically, ACE aims to detect the parallel edges of road structures. This is achieved by locating parallel edges with opposing gradients, separated by a specified width. These parallel edges with opposing gradients are often referred to as Anti-Parallel (APAR) edges. Given the narrow appearance of roads in RS imagery, high resolution data is needed to detect the road edges.

Consider the intensity profiles of a road cross section and its corresponding derivative as shown for an idealised road in Figure 3.3. The road edges are visible as two local extrema with opposing signs in the gradient plot (Figure 3.3b). ACE is designed to identify these patterns and mark a road centrepoint between the two extrema. The procedure followed by ACE is discussed in the paragraphs below.

First, a Canny edge detector is used to locate significant edges within the image, which is followed by a 3×3 Sobel filter to determine the edge orientation. The edge image is subsequently scanned both horizontally and vertically to obtain successive pixels p and q that satisfy the distance and gradient orientation conditions. Considering Figure 3.4, the conditions can be summarised as:

- The gradient at p must be approximately equal to the gradient at q plus 180°.
- If $\vec{w}$ is the vector from p to q, and $\vec{w}_{pq}$ represents a vector in the gradient direction at p with its magnitude equal to expected road width, the angle between $\vec{w}$ and



(a) Intensity profile (b) Derivative

Figure 3.3: Profile of an idealised road with noise added

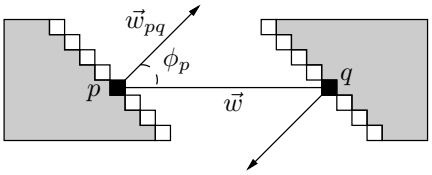


Figure 3.4: APAR Edge detection, as used in ACE

$\vec{w}_{pq}$ must be consistent with the distance between p and q, so that

$$\|\vec{w}\| \cos \phi_p \approx \|\vec{w}_{pq}\|.$$

When a pattern adhering to the above conditions is located, a road centrepoint is marked halfway between p and q.

This concludes the discussion of the ACE algorithm. The following section presents a fast image segmentation algorithm, based on human perceptual grouping. This algorithm is used in Chapter 5 in a comparison between a pixel- and segment-based spectral road classifier.

3.4 Image segmentation

Various remote sensing feature extraction approaches perform analyses on a per-pixel basis, where each pixel is analysed as an individual object in the feature space. The arrival of VHR satellite imagery introduced the option to apply object-based analysis through segmentation. Object-based analysis considers an entire segment in feature space, rather than individual pixels. The aim of segmentation is to add structure to the data, which allows for faster and more accurate analysis [142].

Segmentation algorithms attempt to identify regions with similar characteristics. These characteristics depend on the problem at hand, but are generally defined by the discontinuities and similarities of intensity values within an image [52]. Neighbouring pixels with similar intensity values are thus grouped together, while borders are created where discontinuities occur.

This study employed the graph-based colour image segmentation algorithm by Felzenszwalb and Huttenlocher [41]. The algorithm was designed to be computationally efficient and produce segments similar to those generated by human perceptual grouping. This algorithm is highly efficient with a running time amounting to $O(m \log m)$, where m is

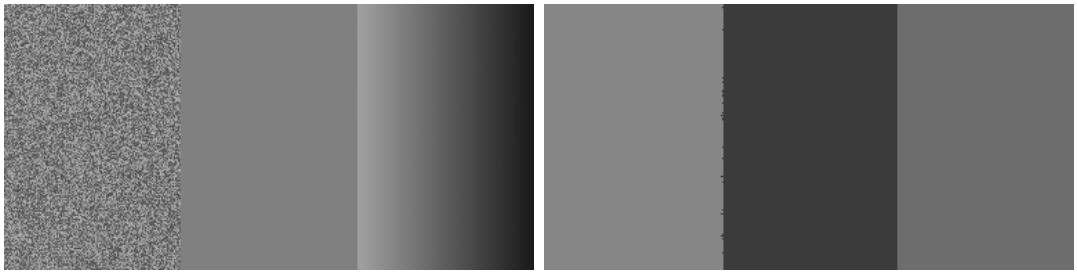
the number of pixels. The increase in computation time is nearly linear to the number of pixels in the image. In practice, the algorithm can be used to segment real-time video at a resolution of 320×240.

Felzenszwalb and Huttenlocher identified two areas where a number of image segmentation methods behave counter-intuitively. Firstly, segmentation methods tend to struggle with grouping areas where a high variability in intensity values occurs. In other words, segmentation methods tend to divide a single noisy area into smaller segments rather than grouping it into a single region with uniform noise. The second problematic area deals with the use of local decision criteria. Due to processing constraints, a number of segmentation algorithms base decisions solely on local information. As a result, the algorithm cannot produce meaningful segments, as it is only capable of analysing a limited area. To improve the segmentation algorithm's perceptual grouping abilities, it would need to base its decisions on non-local information. The use of non-local information increases the computational requirements. An adaptive non-local decision process is therefore required.

Two scenarios that explain the aforementioned issues are illustrated in Figure 3.5a. The first scenario is depicted by the noisy area to the left of the image, while the second is depicted by the uniform and the linear gradient areas to the middle and right of the image.

The first scenario illustrates an issue with high local variability. Segmentation methods typically struggle to group this area into a single segment and will thus create a number of smaller clusters. The ability of the Felzenszwalb and Huttenlocher Segmentation (FHS) approach to cluster consistent noisy areas is evident in Figure 3.5b.

The second scenario illustrates two adjacent areas where the intensity change on the border is less than the intensity change within the linear gradient area. Segmentation



(a) Synthetic input image

(b) Segmented output image ($g = 0.9$, $k = 500$)

Figure 3.5: Image segmentation

methods will usually struggle to either identify the step in intensity correctly or to divide the gradient area into sub-segments. It is thus clear that the FHS method utilises non-local information, since it is able to detect the two regions and their boundary correctly.

FHS addresses the issues discussed above by applying a set of simple rules. A measurement is introduced to define the coarseness or fineness of a segment. The algorithm iterates through all segments and divides coarse segments into smaller segments. Conversely, fine adjacent segments are merged. The algorithm stops when all segments are neither too fine nor too coarse.

Being a graph-based approach, the merging and dividing of segments deal with the classic problem of graph-cutting. Please refer to the original work of Felzenszwalb and Huttenlocher [41] for a more detailed explanation of the graph-cutting approach as well as the function that determines the coarseness or fineness of a segment.

3.5 Classification

Section 2.3.2 presented various classification approaches. The methods introduced by Mena and Malpica [115] was selected to classify imagery in Chapter 5 and 6. Mena and Malpica use three classifiers to determine the distance between image pixels and road models. The distance value is transformed into a pseudo-probability, which is assigned to the image pixel for a given road model description. The probability value indicates the likelihood of the pixel being a road pixel. The first two classifiers employ spectral information to calculate distance values, while the third combines spectral and textural information. The outputs from the three separate classifiers are merged to form a single result using Dempster-Shafer fusion (refer to Section 3.6). It is expected that the fusion of classifier outputs will produce improved results. The following three sections discuss the three classifiers in more detail.

3.5.1 Mahalanobis

The first classifier is a single pixel spectral classifier that uses the Mahalanobis distance function to calculate the separation between a candidate pixel and the training set sample in the RGB colour space. For road extraction, the training set sample is a road model containing a distribution of typical road pixels. The Mahalanobis distance allows different scales in each dimension and incorporates the correlation of variables through a covariance matrix. These traits allow for a more accurate approximation than

its Euclidean counterpart. The Mahalanobis distance is defined as

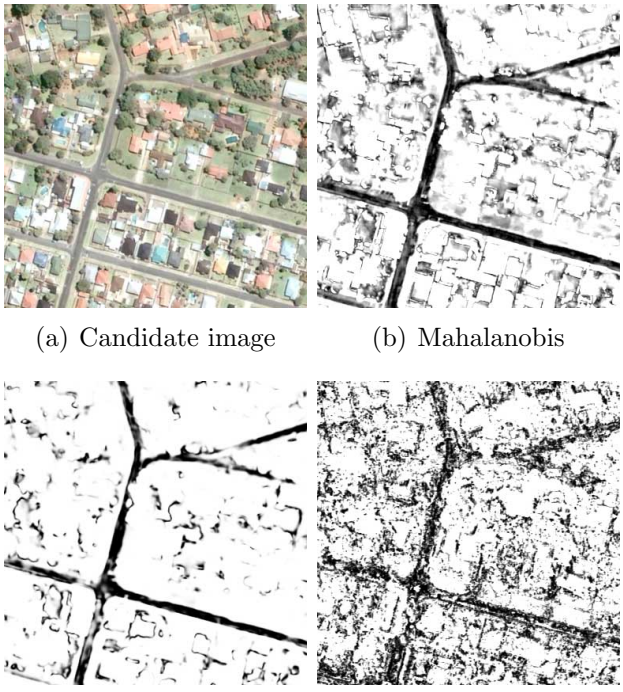
$$d = \sqrt{(\boldsymbol{m}_t - \boldsymbol{x})^t {\Sigma_t}^{-1} (\boldsymbol{m}_t - \boldsymbol{x})}. \tag{3.1}$$

The single candidate pixel is given as $\boldsymbol{x}$. The road model (training set) mean and covariance matrix are represented by $\boldsymbol{m}_t$ and Σ_t respectively. The Mahalanobis metric reduces to an Euclidean metric if Σ_t is identity.

The final Mahalanobis classifier output (d_1) is calculated as

$$d_1 = 1 - \frac{d - d_{\min}}{d_{\max} - d_{\min}}. \tag{3.2}$$

In the above calculation, d is normalized by determining the minimum ($d_{\min}$) and maximum ($d_{\max}$) distance values over the candidate set samples. This equation allows the final classifier output to represent a posterior pseudo-probability of a sample belonging to the road model set. A sample of a Mahalanobis classification is illustrated in Figure 3.6b.



(a) Candidate image (b) Mahalanobis

(c) Bhattacharyya (d) Texture cube

Figure 3.6: Spectral classification process. Darker shades represent higher probabilities in figures b–d

3.5.2 Bhattacharyya

The second classifier employs the Bhattacharyya distance function [44], which is an extension of the Mahalanobis distance. The Bhattacharyya metric measures the distance between two distributions, whereas Mahalanobis measures the distance between a point and a distribution. Mena and Malpica [115] use a 5×5 kernel, which creates a distribution of 25 sample RGB pixels. The function calculates the distance between the sample distribution and the road model distribution. The Bhattacharyya distance is defined as

$$b = \frac{1}{8}(\boldsymbol{m}_t - \boldsymbol{m}_x)^t \left(\frac{\Sigma_t + \Sigma_x}{2}\right)^{-1} (\boldsymbol{m}_t - \boldsymbol{m}_x) + \frac{1}{2}\log\frac{\frac{1}{2}|\Sigma_t + \Sigma_x|}{\sqrt{|\Sigma_t||\Sigma_x|}}. \tag{3.3}$$

The candidate and road model set means are represented by $\boldsymbol{m}_x$ and $\boldsymbol{m}_t$ respectively, while the candidate and road model set covariance matrices are represented by Σ_x and Σ_t respectively. As with the Mahalanobis classifier, the feature vector comprising the RGB bands are employed. The final distance value is scaled by Mena and Malpica as

$$d = \frac{1}{e^b}. \tag{3.4}$$

As with the Mahalanobis classifier, this distance value (d) is normalized and transformed through Equation 3.2 to resemble a pseudo-probability value. The final Bhattacharyya output is denoted by d_2. A sample of an image classified with the Bhattacharyya classifier is depicted in Figure 3.6c.

3.5.3 Texture cube

The third and final classifier is based on a multi-spectral texture analysis. As with the previous classifier, the Bhattacharyya metric is used to compute the distance between a candidate distribution and a road model set distribution. The formulation of these distributions is discussed below.

A texture cube is constructed by considering the 3 × 3 neighbourhood around a candidate centre pixel. This set of nine pixels can be visualised as a three-dimensional cube, where the height and width of the cube is represented by the 3 × 3 neighbourhood representing the height and width of the cube, while the three bands in colour space represent the depth. Six cross-sections are selected from the texture cube; three across the depth (bands) of the cube to capture the spectral characteristics and three across the face (spatial) of the texture cube to capture the textural characteristics. Although more

cross sections can be added, such additions will only introduce redundant information.

From the six cross-sections, six textural features are computed based on the work by Haralick [57]. These features are correlation, energy, entropy, maximum co-occurrence, contrast and inverse difference moment. Computing the Haralick features requires the construction of co-occurrence histograms from each of the six cross-sections. Considering that every section has nine variables, each with 256 grey levels, it is apparent that the co-occurrence histograms will be quite sparse. This sparsity issue is addressed by constructing four co-occurrence distributions for each of the cross-sections. The Haralick features are computed from the co-occurrence distributions, which results in 24-feature vectors with a dimensionality of six.

Finally, the Bhattacharyya distance between the distribution of the feature vectors of the candidate pixel and that of the road model set is computed. The initial feature vector is represented in the HSI colour space. As with the previous classifiers, the distance measure is transformed to represent a pseudo-probability of the pixel belonging to the road model set. The final texture cube output is denoted by d_3. Figure 3.6d illustrates a sample of a classification made by the texture cube classifier.

The following section on Data Fusion discusses how the results from the three classifiers are combined.

3.6 Dempster-Shafer fusion

This section sees the combination of the pseudo-probabilities from the classifiers described in Section 3.5 using the Dempster-Shafer Theory of evidence (refer to Section 2.4.2). When fusing classified road-pixels, Mena and Malpica [115] define the frame of discernment as $\Theta = \{\omega, \overline{\omega}\}$, where ω is the proposition of an observation corresponding to the *road* class and $\overline{\omega}$ its complement, also known as the *non-road* class. The resulting power set is provided as $\mathrm{P}(\Theta) = \{\emptyset, \omega, \overline{\omega}, \omega \cup \overline{\omega}\}$.

The method by which evidence is combined using DST will now be illustrated by fusing the Mahalanobis (d_1) and Bhattacharyya (d_2) values. Given the associative nature of Dempster's rule, any additional sources, such as the texture cube classifier output, can be combined sequentially.

The mass values d_1 and d_2 are indicative of the amount of evidence of the proposition (ω) that a pixel belongs to a road class. The support mass function for the Mahalanobis classifier is denoted by $m_1(\cdot)$ and the following mass functions are subsequently defined

as

$$\begin{aligned} m_1(\omega) &= d_1(1-\sigma_1) && (3.5)\\ m_1(\overline{\omega}) &= 1-d_1(1-\sigma_1)-\sigma_1 && (3.6)\\ m_1(\omega\cup\overline{\omega}) &= \sigma_1. && (3.7) \end{aligned}$$

In the above equation, d_1 is obtained with Equation 3.2 and σ_1 its variance. The empty set (Θ) is often used to calculate the support for neither of the two propositions occurring [86] and its mass function can be defined as $m(\Theta) = m(\omega\cup\overline{\omega})$. The support mass function for the Bhattacharyya classifier (d_2) is denoted by $m_2(\cdot)$ and is defined similarly to $m_1(\cdot)$, with d_2 replacing d_1 and σ_2 replacing σ_1. The fused mass function $m(\cdot)$ is obtained by applying Dempster's rule of combination. According to Equation 2.27 and 2.28, the following can be calculated:

$$\begin{aligned} m(\omega) &= \frac{m_1(\omega)m_2(\omega)+m_1(\omega)m_2(\omega\cup\overline{\omega})+m_1(\omega\cup\overline{\omega})m_2(\omega)}{1-k_{12}} && (3.8)\\ m(\overline{\omega}) &= \frac{m_1(\overline{\omega})m_2(\overline{\omega})+m_1(\overline{\omega})m_2(\omega\cup\overline{\omega})+m_1(\omega\cup\overline{\omega})m_2(\overline{\omega})}{1-k_{12}} && (3.9)\\ m(\omega\cup\overline{\omega}) &= \frac{m_1(\omega\cup\overline{\omega})m_2(\omega\cup\overline{\omega})}{1-k_{12}} && (3.10)\\ k_{12} &= m_1(\omega)m_2(\overline{\omega})+m_1(\overline{\omega})m_2(\omega). && (3.11) \end{aligned}$$

Considering Equation 2.22, Bel(ω) is defined as the amount of evidence available for the proposition ω. This allows the following inference to be made: Bel(ω) = $m(\omega)$. Referring to Equation 2.24, Pl(ω) is defined as the degree to which ω is not contradicted by the evidence. The fused belief and plausibility functions can subsequently be computed according to Equations 2.24 and 2.24, as follows:

$$\text{Bel}(\omega) = \frac{d_1(\sigma_1-1)(d_2(\sigma_2-1)-\sigma_2)-d_2\sigma_1(\sigma_2-1)}{1-d_2(\sigma_1-1)(\sigma_2-1)+d_1(2d_2-1)(\sigma_1-1)(\sigma_2-1)} \quad (3.12)$$

$$\text{Pl}(\omega) = \frac{(d_1(\sigma_1-1)-\sigma_1)(d_2(\sigma_2-1)-\sigma_2)}{1-d_2(\sigma_1-1)(\sigma_2-1)+d_1(2d_2-1)(\sigma_1-1)(\sigma_2-1)}. \quad (3.13)$$

The DST does have some shortcomings, such as its inability to model evidence accurately when a high degree of conflict between sources exists [22]. Dempster's rule of combination emphasizes agreement between sources. The normalization factor in Equations 2.27 and 3.8–3.10 uses the conflict between sources K_{12} (Equation 3.11). High K_{12} values will

cause the combination of the two masses to be scaled by a disproportionally large value (refer to Equation 2.27).

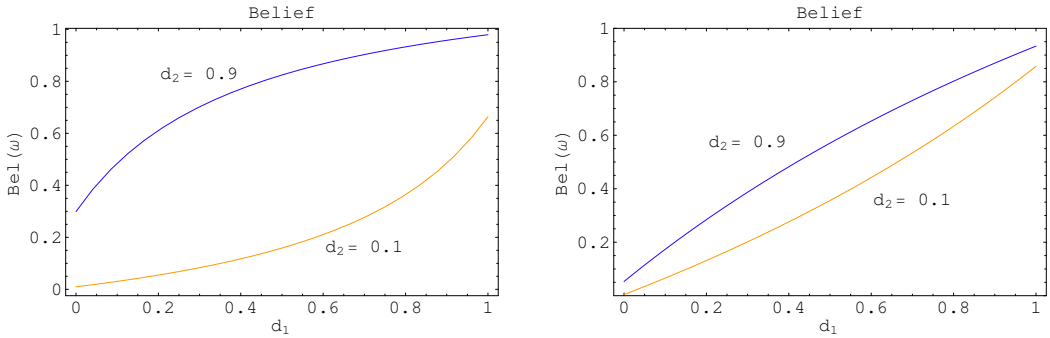
Figure 3.7 illustrates the effect that varying values for d_1 and d_2 has on the Bel(ω) value. More specifically, where σ_2 is small (Figure 3.7a), d_2 has a greater impact on Bel(ω). Conversely, where σ_2 is larger than σ_1 (Figure 3.7b), d_2 is disregarded to an extent.

Mena and Malpica [115] use the plausibility value Pl(ω) as the output of their fusion process. Figure 3.8 illustrates the manner in which the images in Figure 3.6b–d are combined through DST. By thresholding the plausibility function (Figure 3.8a), the pixels with sufficient evidence to belong to a road pattern class can be identified (Figure 3.8b).

3.7 Mathematical morphology and filtrate

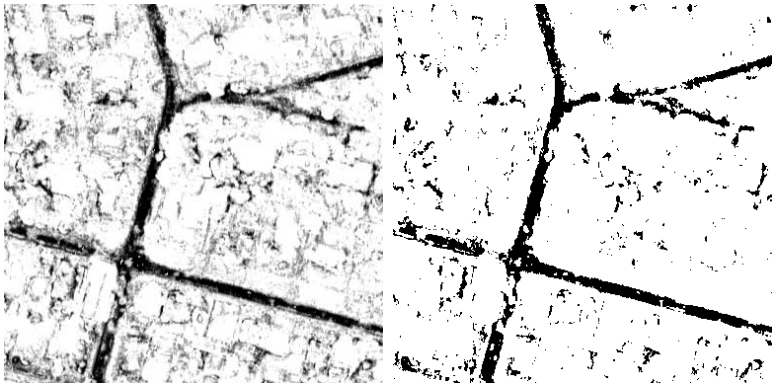
The results produced by road classifiers, such as the algorithms described in Section 3.5, or the road plausibility values produced the Dempster-Shafer fusion (refer to Section 3.6) can be thresholded. The thresholding process will produce a binary image comprising two classes, namely *road* and *non-road*. These binary images typically contain a number of artefacts which can be reduced through filtering.

Section 2.3.5 introduced mathematical morphology and several basic filtering operators. The following two sections discuss two additional operations used by the road extraction system proposed in the subsequent chapters.



(a) $\sigma_1 = 0.1$, $\sigma_2 = 0.1$ (b) $\sigma_1 = 0.1$, $\sigma_2 = 0.6$

Figure 3.7: Dempster-Shafer fusion: A plot of Bel(ω) for different values of σ_2



(a) Plausibility values (b) Thresholded result

Figure 3.8: Dempster-Shafer fusion

3.7.1 Connected component filter

The concept of *connected components* is rooted in graph theory and is defined as a sub-graph within an undirected graph in which all vertices are connected with a path. In image processing, a connected component is a set of pixels where each pixel has at least one immediate neighbouring pixel with the same value. The purpose of a connected component filter is to remove or suppress objects larger or smaller than a specified threshold.

This study used the mathematical morphological *binary attribute opening* operation as a connected component filter [19]. Binary attribute opening is defined as the *trivial opening* of a *connected opening* [153]. Consider the binary image $X \subseteq \mathbf{M}$ with the domain $\mathbf{M}$.

The connected opening Γ_x at pixel $\boldsymbol{x}$ is defined as

$$\Gamma_x(X) = \begin{cases} C & \text{if } \boldsymbol{x} \in X, \\ \emptyset & \text{otherwise.} \end{cases} \tag{3.14}$$

C is the connected component of which pixel x is a sub-component. The trivial opening Γ_T with a certain criterion T is defined as

$$\Gamma_T(C) = \begin{cases} C & \text{if } C \text{ satisfies } T, \\ \emptyset & \text{otherwise.} \end{cases} \tag{3.15}$$

The criterion T can be any restriction on a real-world attribute of the component. The

binary attribute opening Γ^T of set X with the criterion T is defined as

$$\Gamma^T(X) = \bigcup_{x \in X} \Gamma_T(\Gamma_x(X)). \tag{3.16}$$

The criterion used to remove or suppress objects smaller than a specified threshold is defined as

$$T(C) = |C| \geq \lambda, \tag{3.17}$$

where $|C|$ denotes the cardinality of C and λ a certain threshold.

The connected component filter is typically used during post-processing to remove small clusters of pixels. Figure 3.9 illustrates two connected component filters with $\lambda = 50$ and $\lambda = 100$ applied to a binary image obtained by thresholding the output from a Mahalanobis classifier.

3.7.2 Closing operation

The *dilation* ($\oplus$) and *erosion* ($\ominus$) operations, introduced in Section 2.3.5, can be used in different combinations to construct additional mathematical morphology operators. The closing operator is an example of such a combination, and the closing of a given set A by a kernel B is obtained by a dilation ($\oplus$) of A by B and an erosion of the result with B. Drawing from Equations 2.3 and 2.4, the closing operation can be defined as

$$A \bullet B = (A \oplus B) \ominus B. \tag{3.18}$$

(a) Binary image (b) $\lambda = 50$ (c) $\lambda = 100$

Figure 3.9: Connected component filter with varying minimum component sizes

This operation can be used to fill small holes within larger components. Figure 3.10 depicts the closing operation on the image presented in Figure 3.9b with the disc-shaped kernel with a radius (r) of 2 and 4.

3.8 Self-organizing road map

Section 2.5 considered a number of approaches aimed at constructing a topologically correct road network. The approach by Oddo *et al.* [122] is considered ideal, as it circumvents the inelegant vectorization process. The SORM algorithm by Doucette *et al.* [38, 39] builds on the work by Oddo *et al.* and presents a SORM variant that is able to construct networks from disparate sources, which further contributes to the appeal of the method. These sources includes both classified binary pixels and centreline seeds, which are depicted in Figures 3.11a and 3.11b respectively. The original SORM algorithm [39] is designed to construct a road network topology from a classified binary image. In this thesis, SORM is adapted to receive this image from the classification and fusion process proposed by Mena and Malpica [115] (refer to Sections 3.5 and 3.6). The centreline seed SORM approach obtains its input from the ACE algorithm (refer to Section 3.3) originally proposed by Doucette *et al.* [38].

The main function of SORM is to link probable road pixel clusters — be it classified road pixels or centreline seeds — by creating a set of vectors that runs through the centre of each segment. Moreover, where the segments have similar orientations, vectors running through nearby segments are linked. SORM is also used to remove redundant, isolated pixel clusters that do not form part of the road network. A brief description of

(a) Disc-shaped kernel $r = 2$ (b) Disc-shaped kernel $r = 4$

Figure 3.10: Closing filter with varying disc-shaped kernel sizes

(a) Classified binary pixels (b) Centreline seeds

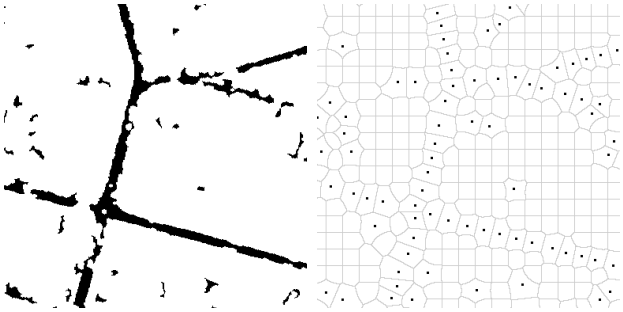
Figure 3.11: Disparate sources from which SORM can construct a road topology

the adapted SORM implementation follows.

The binary images produced by a classifier (refer to Section 3.5) or a fusion process (refer to Section 3.6) tend to contain a number of discrepancies. These discrepancies include isolated, misclassified road pixels and small gaps within road segments. A connected component filter (refer to Section 3.7.1) is used to remove small isolated clusters of pixels, while a closing operation filter (refer to Section 3.7.2) is used to fill small gaps. The result of the thresholding and filtering process on Figure 3.8 is presented in Figure 3.12a.

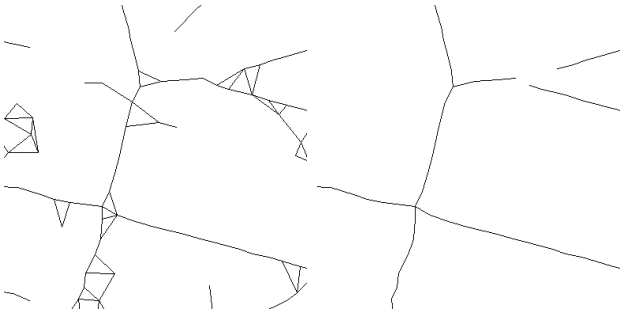
After morphological filtering, an evenly spaced grid is created across the image, where every point on the grid represents a cluster centroid. After initialising the clusters, SORM updates the centroids by moving them towards the nearest grouping of road pixels. A median clustering approach is followed, which reduces the effect of outliers. The cluster centroids are updated iteratively either until no further updates occur or until a specified number of iterations are reached. Clusters with unchanged centroids are marked as dead and not considered in subsequent processing. The remaining active cluster centroids are illustrated in the Voronoi diagram in Figure 3.12b.

A Relative Neighbourhood Graph (RNG) is created from the active cluster centroids (Figure 3.12c). The brute-force algorithm is computationally expensive and has a running time of $O(n^3)$. The Urquhart method [154] provides a high quality approximation of the RNG and can be extracted in linear time from the Delaunay triangulation of the centroid points, which has an extraction time of $O(n^2)$ [4]. The RNG is preferred to the Minimum Spanning Tree (MST), since the RNG allows cycles within the network. The MST has a single-linkage tree structure, which might result in valid links being discarded



(a) Mathematical morphology filtering

(b) Voronoi diagram of active centroids



(c) Relative neighbourhood graph

(d) Final output

Figure 3.12: SORM process

at this early stage in the process.

After obtaining a network structure, six features are computed for every possible edge pair combination in the network. An edge connected to two other edges will be evaluated for each pair combination. The six features comprise the edge pair's angles and Eigen vectors, the percentage of road pixels underneath the edge in question, and an edge length ratio. A Multi-Layer Perceptron (MLP) is used to filter out edges that violate typical road network behaviour.

The MLP was trained with a steepest gradient descent algorithm (using backpropagation to compute the gradient) on a manually created data set that illustrates both valid and invalid road network edge pairs. A learning rate of 0.3 and a momentum factor of 0.2 were used during training. The optimal MLP structure comprises six input nodes,

Table 3.1: MGE for different MLP architectures

Hidden units	**4**	**5**	**6**	**7**	**8**
MGE (%)	3.21	3.16	3.16	3.12	3.16

seven hidden nodes and one output node, with both the hidden- and output nodes using sigmoidal activation functions. The optimal architecture was determined by executing 10 runs of a 5-fold cross-validation experiment, and recording the mean generalisation error (Table 3.1). A higher-level filter (employing a fuzzy controller) was applied to check all edge pair combinations, discarding edges that are unlikely to appear in road networks. The remaining edges are used to construct a minimal spanning tree using Kruskal's algorithm [91]. The final result (Figure 3.12d) yields a graph where each edge represents the road centreline.

This concludes the discussion of the SORM algorithm. The following section considers the metrics used to quantify the quality of the centreline road network produced by SORM.

3.9 Quality metric

The quality metrics by Heipke *et al.* [63, 164] were used in this thesis to quantify the output from the road extraction system. The metric derives quality measures by comparing extracted centrelines to a ground truth reference set. Section 2.6 describes the quality metrics in detail. To simplify calculations, a raster-based approximation is implemented where morphological dilation [52] is used to create the buffer zone. In the original work by Heipke *et al.*, six quality metrics are defined; three of which (*redundancy*, *root mean square difference*, and *gap statistics*) are ancillary measurements. In this thesis, the final accuracy of the extracted centrelines is determined using the remaining metrics, namely *correctness*, *completeness*, and *quality*.

The following section considers the data set used for the experiments conducted in this thesis.

3.10 Data

Pan-sharpened orthorectified QuickBird images, acquired in January 2005, are used as the data source for the experiments in this thesis. The images were acquired over the Gauteng province of South Africa. The orthorectification process projects the data into the Universal Transverse Mercator (UTM) coordinate system along the 29° central meridian, using the current World Geodetic System (WSG 84) as the geodetic datum.

The QuickBird satellite uses the Ball Aerospace Global Imaging System 2000 (BGIS

2000) with the following sensors:

Panchromatic: The sensor has a single black and white band with a ground resolution of 61 cm at nadir and a spectral range of 445–900 μm.

Multispectral: The sensor has a ground resolution of 2.4 m at nadir and has four bands, namely blue (450–520 μm), green (520–600 μm), red (630–690 μm) and near infrared (760–900 μm).

The final pan-sharpened product combines the spatial information from the panchromatic band with the spectral information of the multispectral bands into a spectrally and spatially enhanced colour composite with a ground resolution of 61 cm. The composite can be created by selecting various combinations of the multispectral bands and the data used in this thesis was created with the *red*, *green* and *blue* bands. The final product was delivered as a mosaic consisting of multiple GeoTIFF (version 1.0) files. Each band had a radiometric depth of 8-bits; the original 11-bit data was not available for this study. The software packages and frameworks used to implement the RNE system and its components are considered in the next section.

3.11 Software

A combination of software packages and frameworks were used in this thesis to create the components within the image processing chain. The majority of the development was done in C++ using the Open Source Software Image Map[1] (OSSIM) system on a Linux platform. Currently funded by a number of USA government agencies, OSSIM is a sophisticated platform designed specifically for remote sensing, image processing and GIS related tasks. The framework design allows generic components to be stringed together into an image processing chain. The following components were implemented within the OSSIM framework: the Canny edge detector, the ACE geometrical centreline seed detector, the three classifiers by Mena and Malpica and the Dempster-Shafer fusion module.

Another package used is the Olena framework[2], which is created by the EPITA Research and Development Laboratory situated in Le Kremlin-Bicêtre, Paris, France. An image processing framework with a multitude of functionalities, Olena was used

[1] http://www.ossim.net
[2] http://olena.lrde.epita.fr

in this study mainly for its built-in mathematical morphology functions. Some of the components created in Olena include the connected component filter, closing operation, dilation and quality metric.

The C++ source code for the V1 algorithm by McKinstry and Guest and the image segmentation algorithm by Felzenszwalb and Huttenlocher were obtained directly from the authors. The SORM algorithm was implemented in C++ and a Java-based MLP filter was used. The MLP was optimized with the Weka package[3] which is created by the Computer Science Department of the University of Waikato, New Zealand. Finally, a number of Python and shell scripts were used to fulfil certain minor ad-hoc functionalities.

3.12 Test system specifications

All experiments were conducted on a single machine with the following specifications:

- **CPU:** Intel Core 2 Duo 2.4 GHz
- **Memory:** 4 GB, DDR2, 667 MHz
- **Hard drive:** 200 GB, 7200 rpm, SATA
- **Operating system:** Ubuntu 8.04, 32-bit

3.13 Conclusion

The literature review and background (Chapter 2) proposed numerous techniques to be used to construct an RNE system. A few of these techniques were selected for this thesis and implemented within a dynamic road extraction system. This chapter discussed these techniques in more detail. The most significant components include the following: the ACE and SORM algorithms by Doucette *et al.* [38, 39], the classification and fusion process by Mena and Malpica [115], the edge detector by Canny [23], the V1 edge detector by McKinstry and Guest [110], the quality metric by Heipke *et al.* [63], and various mathematical morphological filters.

The following chapter considers the possibility of improving the road extraction accuracy of an ACE-based system by replacing the Canny algorithm with the V1 long range edge detector.

[3]http://www.cs.waikato.ac.nz/ml/weka

CHAPTER 4

Long Range Edge Detection

"However beautiful the strategy, you should occasionally look at the results."
— Winston Churchill

Human beings posses the ability to perceive relationships between image objects that are not necessarily adjacent with relative ease. This ability enables the detection of elongated features, such as roads and rivers, in remotely sensed imagery. This chapter presents a road extraction system and compares the performance of a long range edge analysis method [110] to the renowned Canny algorithm [23]. Long range edge detection is ideally suited to road extraction applications, since the edges of these elongated features will be detected while disregarding certain shorter edges in non-road areas. This ultimately results in more true positive and less false positive detections. Long range detection has a further advantage in its ability to bridge certain occlusions that might occur along the roadside.

This chapter presents a more detailed introduction in Section 4.1, followed by a

description of the processing chain in Section 4.2. Section 4.3 discusses the experimental design, while the experimental results and discussion are presented in Section 4.4. Section 4.5 provides the concluding remarks.

4.1 Introduction

Edge detection or gradient analysis is a very popular technique in road extraction literature. As indicated in Section 2.3.3, 129 of the 217 road extraction studies surveyed employ edge detection at some stage within the image processing chain. The algorithms presented in these studies can either be classified according to the number of assumptions made regarding the input images or by the number of input parameters required.

The road extraction system presented by Doucette *et al.* [38] was selected, since it makes very few initial assumptions about the appearance of roads within high resolution imagery and achieves results comparable to the current state-of-the-art RNE systems. The ACE algorithm discussed in Section 3.3 is used by Doucette *et al.* in a preliminary stage to detect road centreline seeds. ACE identifies seeds by considering the intensity gradient of opposing road edges. The low-level Canny edge detector discussed in Section 2.3.3 is used to calculate the intensity gradient and to detect the road edges. It should be noted that the Canny method operates in a local area, which has the adverse effect of producing disjointed edges when encountering occlusions or sections with poor image contrast. ACE will naturally fail in cases where an opposing edge is missing; consequently an edge detector with the ability to bridge these problematic areas could result in an improvement in ACE's performance.

Human beings are capable of detecting roads with a high degree of accuracy by combining analytical capabilities with prior knowledge of road structure characteristics. One of the analytical techniques used by humans is *long range edge analysis* facilitated through the primary visual cortex [110]. Despite fragmentation occurring in certain cases, this technique enables humans to create long range connections. McKinstry and Guest [110] propose a long range neural network-based edge detector called V1 (refer to Section 3.2.2), which is modelled after the primary visual cortex. According to the edge coherence metric by Kitchen and Rosenfeld [85], V1 outperforms Canny.

V1 may not only be able to detect the road edges with a higher level of accuracy and produce better road seeds through ACE, but may also reduce the number of spurious seeds found in non-road areas. Images containing for example urban, suburban or

industrial regions display a large number of short edges in non-road areas. Canny will consequently extract these edges irrespective of the length, which results in ACE detecting spurious road seeds. V1 is less sensitive, since it is geared towards detecting longer edges, which may translate into fewer erroneous road seeds produced by a V1-based ACE.

In summation, the objective of this chapter is to investigate the possibility of improving the performance of the ACE algorithm by replacing Canny with V1. The proposed road extraction system and its sub-processes are presented in the following section.

4.2 Road extraction pipeline

The proposed road extraction system is modelled on the system by Doucette *et al.* [38]. The system is organized in the form of an image processing chain, consisting of a number of interchangeable autonomous components. Figure 4.1 presents a flow chart that illustrates the manner in which these components are joined together.

The following list provides a brief overview of each layer within the flow chart:

1. A candidate image is provided as an input to the system. This candidate image is

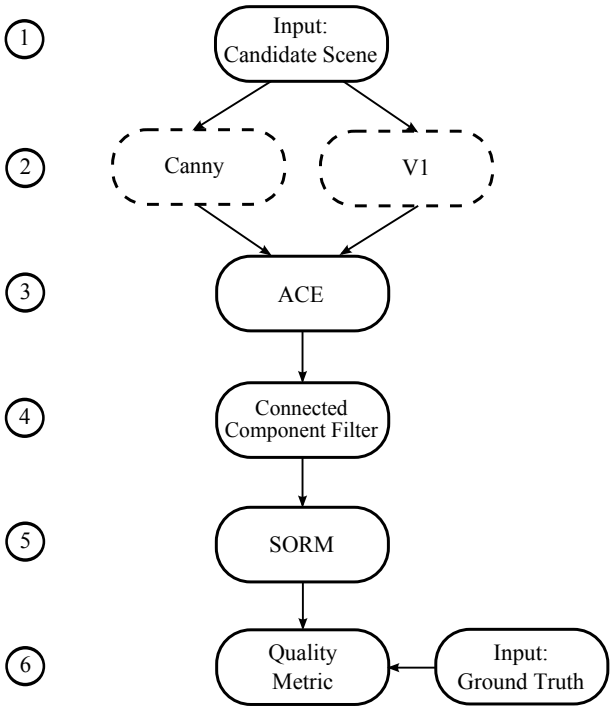


Figure 4.1: Data flow representation of road network extraction process

the scene from which the road network should be extracted. Both the Canny and V1 edge detectors expect greyscale images as input, as these methods base their detections on changes in intensity. Candidate images containing more than one band are converted automatically into a single-banded greyscale image.

2. The second layer applies either of the two edge extractors, depending on the method followed. The Canny process produces a two-band 8-bit image. The first band marks the locations of edge pixels with the value 255, while the rest of the image is set to 0. The second band contains the edge gradient information for the candidate image. The V1 process produces a single 8-bit edge image that contains the locations of the edge pixels only. The gradient information for the V1-based approach is obtained with the same method used in Canny's algorithm. Canny and V1 are described in greater detail in Sections 3.2.1 and 3.2.2 respectively.

3. The ACE algorithm (refer to Section 3.3) aims to find the centrepoints between the edges of elongated objects. ACE considers parallel edges with opposing gradients, which are also known as APAR edges. A minimum and maximum distance threshold is used to consider APAR edges that are separated by the width of typical road structures. ACE can also be configured to consider either all objects with opposing gradients or objects that are brighter or darker than their neighbouring pixels. Asphalt roads tend to be darker, while concrete and dirt roads appear brighter. By constraining the direction of the opposing gradients, the search space is reduced and the number of erroneous detections reduced. The algorithm produces a single-band image containing the locations of the probable centreline seeds.

4. ACE is solely concerned with detecting APAR edges, regardless of whether these edges represent road or non-road objects. As a result, the output from ACE may contain centreline seeds for non-road objects with APAR edges. Road objects typically appear as elongated structures with long parallel edges. The ACE output for these elongated structures is a set of connected seed points along the centre of the object. Since these points are connected, isolated seeds and clusters can be removed to reduce the number of spurious seeds. The connected component filter (refer to Section 3.7.1) will remove all objects below a specified size.

5. The remaining centreline seeds are used to create a road network topology through the SORM algorithm (refer to Section 3.8). The seeds are sorted into a self-organized mapping and a road vector topology extracted through a fuzzy controller

and neural network grouping algorithm. During this process, SORM attempts to connect the road seeds into a logical road network and discards seeds that cause counterintuitive results.

6. The metric by Heipke *et al.* [63] (refer to Section 3.9) is used to determine the quality of the extracted road topology. The SORM output and the ground truth of the candidate scene is used as input for the component. The three values produced by the metric, namely *completeness*, *correctness* and *quality*, are used to consider the accuracy of the results. The completeness indicates how many of the roads (length) in the candidate scene were extracted, while correctness indicates the precision at which they were extracted. The quality measure combines completeness and correctness values into a single accuracy percentage for the extracted road network.

The proposed system is fully automated and requires few input parameters. The parameters and the optimisation of their values are discussed in Section 4.3.3.

4.3 Experimental design

Various aspects regarding the system implementation and experimental design are presented in this section. A detailed discussion of the data set is provided, followed by several considerations regarding the system implementation. The section concludes with a discussion on the parameter selection process.

4.3.1 Data set

Thirty-five scenes were selected from the pan-sharpened QuickBird data set described in Section 3.10. Each candidate scene had a pixel dimensionality of 512×512 with a ground resolution at nadir of 61 cm, which translates to a window size of 312×312 m. The scenes were selected to be representative of the real-world scenarios that a road extraction application would have to deal with. To this end, a variety of roads with different characteristics were selected to illustrate the strengths and weaknesses of the system.

The ACE algorithm makes certain assumptions regarding the structure of roads and their appearance within an image. The first assumption is that the minimum and maximum road widths are known in advance. Considering that remotely sensed

image pixel sizes are fully specified via projection metadata, it is possible to satisfy this assumption. In real world examples, scenes often contain roads with varying widths. To replicate real-world scenarios, scenes containing roads ranging from single carriageways to three-lane highways were included in the data set.

ACE makes two additional assumptions, which are tested by defining four scene classes to illustrate the system's behaviour. These classes are defined by the amount of occlusion as well as the level of contrast between the road and the non-road areas. Table 4.1 provides a breakdown of these classes as well as the number of scenes selected within each class. A sample from each class is depicted in Figure 4.2 for clarity.

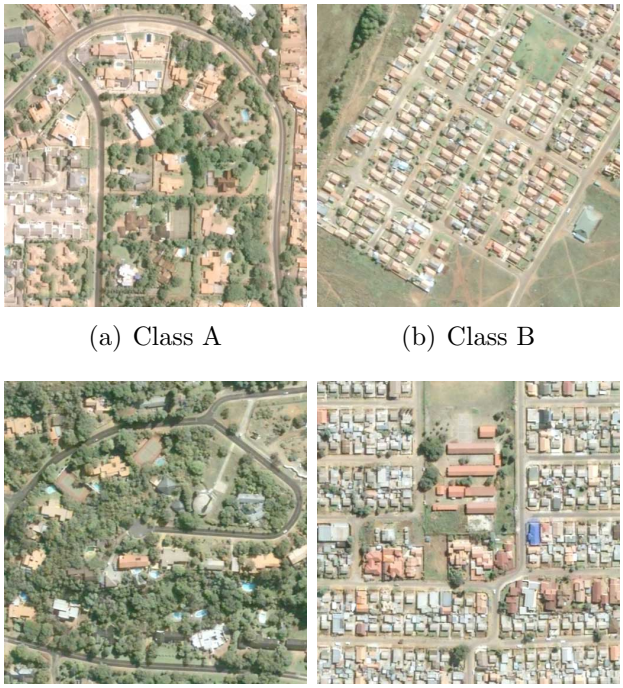
The first assumption is that opposing road edges will be visible contiguously. This assumption rarely holds in real world scenarios where trees, shadows or parked vehicles often occlude or cover roadsides. These occlusions cause fragmentation along one or both road edges, which results in ACE failing to detect an opposing edge. Scenes with both opposing roadsides clearly visible and those with occlusion were considered in the data set selection. Images in classes A and B have no or little occlusion, while the images in classes C and D display a higher degree of occlusion.

The second assumption is that the contrast between the road surface and its immediate surroundings will be prominent enough to detect the edge. This is not consistently the case in real world scenarios, especially in rural and lower income areas where roads are partly covered by dirt. Images in classes A and C demonstrate a visible contrast, while roads in classes B and D are less prominent. It is recognized that ACE would fare poorly in low contrast scenarios and would typically not be employed in such areas. A smaller number of poorly contrasted scenes were therefore included in the overall data set.

Once roads have been extracted from the candidate images, the quality metric component will measure the accuracy of the results. The extracted roads are compared to the corresponding image's ground truth, which was delineated manually from the candidate image. A ground truth sample is provided in Figure 4.3.

Table 4.1: Explanation of image classes. The value in parentheses indicates the number of scenes in each class

	Good contrast	**Poor contrast**
Little occlusion	`Class A (12)`	`Class B (5)`
More occlusion	`Class C (13)`	`Class D (5)`



(a) Class A (b) Class B

(c) Class C (d) Class D

Figure 4.2: A sample from each scene class

(a) Candidate image (b) Ground truth

Figure 4.3: A sample of a candidate input image and its corresponding ground truth

This concludes the discussion of the data set used in this chapter. A few implementation considerations are mentioned in the following section.

4.3.2 Implementation

A few issues regarding the system implementation should be considered. The pan-sharpened QuickBird data comprise RGB colour images. As both edge detection algorithms operate on greyscale imagery, the original data had to be converted. The greyscale is by definition the luminosity of an image and is obtained by considering only the intensity (I) band of an HSI image. The I band is calculated as

$$I = \frac{1}{3}(R + G + B).$$

Another issue regarding the implementation should be noted: The ACE algorithm is designed to detect opposing edges of road objects, which are either brighter or darker than the non-road areas. This difference in intensity has an impact on the gradient orientation of the opposing edges under consideration. A parameter indicating the orientation of the opposing edge gradients are included in the ACE implementation used in this thesis. The road surface intensity in relation to the non-road areas is explicitly specified through this parameter. By considering a specific gradient orientation, the search space for opposing road edges is greatly reduced. This reduction should translate into a decrease in the number of spurious road centreline seeds extracted.

A final point to note is that the original system proposed by Doucette *et al.* [38] includes a spectral classification feedback loop (SSRC). The SSRC extracts road training samples from the topology created by the SORM algorithm. The training samples are used to classify a scene with the Bayesian ML algorithm spectrally. The final road topology is obtained by applying ACE and SORM to the classified image. The SSRC process has been excluded from the system proposed in this chapter, as it is unlikely to have a considerable impact when determining the effectiveness of the edge detector used. The next section considers the system parameter values used during experimentation.

4.3.3 Parameter selection

The road extraction system described in Section 4.2 has various parameters that should be set to appropriate values to produce good results. The selection of these parameters are discussed according to the flow of the road extraction pipeline illustrated in

Figure 4.1.

The first components in the chain are the Canny and V1 edge detectors. It was noted in Section 3.2.1 that the Canny algorithm expects three parameters, namely the Gaussian filter variance σ, and the high and low thresholds T_{high} and T_{low} used within the hysteresis process. The V1 algorithm was used as described by McKinstry and Guest [110] without any adjustments.

The next component in chain is the ACE algorithm. As discussed in Section 4.3.1, the ACE algorithm requires three parameters, namely road minimum and maximum width, and road gradient orientation. The selection of these parameters is discussed in the following paragraphs.

When using the ACE algorithm an implicit assumption is made that the road widths are known beforehand. The road widths are defined by selecting a minimum and maximum threshold of typical road structures within a data set. Considering that the data set was acquired over the Gauteng province, the guidelines set forth by the Gauteng Department of Transport and Public Works (Gautrans) [155], provides an appropriate reference point for estimating typical road widths. According to Gautrans, suburban roads should have a lane width of 3.5 m, resulting in a total road width of 7 m. Highways should have a lane width of 3.7 m and a shoulder of 2.5 m. The minimum road width would therefore be 5.2 m with a maximum width of 9.9 m for a dual carriage highway. Scenes containing highways with more than two lanes were not considered. Roads in low income areas are required to have a width of between 4–6 m.

As mentioned in Section 2.8, the data set of pan-sharpened QuickBird data have a GSD of 61 cm at nadir. When considering this GSD, it is expected to find road widths of 11.4 pixels (7 m) for suburban roads, 8.5–16.2 pixels (5.2–9.9 m) for dual lane highways and 6.6–9.8 pixels (4–6 m) for low-income areas. The minimum road width in Gauteng should therefore be 6.6 pixels and the maximum 16.2 pixels. To confirm these figures, 60 road width samples were taken across 12 scenes in the data set. To ensure that the actual road width was measured, samples were taken from road sections without any occlusions. The histogram in Figure 4.4 confirms the initial assumption with the largest fraction falling between 6–14 pixels.

Certain trade-offs should be considered when selecting the minimum and maximum road width thresholds. Opting for a narrow range would reduce the system's ability to detect different road widths and consequentially, diverse road types. A narrow range would, however, have the advantage of increasing the system accuracy when detecting specific road types. Conversely, a wide range would allow multiple road widths to be

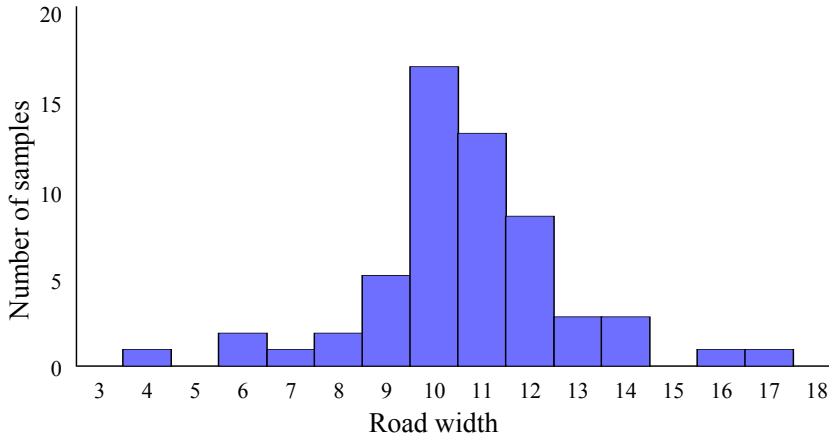


Figure 4.4: Histogram of road widths

detected, but at the cost of accuracy for specific road types.

In light of the above, the minimum and maximum road width thresholds were set to 6 and 14 respectively. This range should allow the system to detect the majority of the roads, while maintaining an acceptable level of accuracy.

The final parameter required by the ACE algorithm is the road gradient orientation, which reduces the search space by only considering structures in RS images that are either darker or lighter than the surrounding objects. This gradient orientation parameter was set manually per scene, depending on the intensity of the road relative to the non-road.

The next component in the road extraction pipeline is the connected component filter, which requires a single parameter. As described in Section 3.7.1, the minimum connected component size ($cc_{\min}$) is used to remove small objects from a binary image. The $cc_{\min}$ parameter was determined by testing a range of values from 0 to 13, by executing the full processing chain and determining the effect on the final quality of the results. As with the road widths threshold selection, the system was tested on 12 images from the data set. The quality metric component defined in Section 3.9 produced the final system output. The resulting median quality values for each component size over the 12 images for each $cc_{\min}$ value are displayed in Table 4.2. The correctness and completeness values are also included to illustrate the interaction between the metrics. Based on these results, a $cc_{\min}$ values for both the Canny and V1 variants were selected by considering the quality value. A value of 10 was selected for the Canny variant and a value of 1 was chosen for the V1 variant.

The final component in the road processing pipeline requiring an input parameter is the SORM algorithm (refer to Section 3.8). SORM initializes its cluster centroids as an

Table 4.2: Extraction results of ACE variants for various minimum connected component filters. The column headings are: **cor** denotes *correctness*, **com** denotes *completeness*, and **qual** denotes *quality*

	Canny			**V1**		
$cc_{\min}$	**cor**	**com**	**qual**	**cor**	**com**	**qual**
0	0.519	0.287	0.236	0.808	0.373	0.320
1	0.468	0.389	0.278	0.773	0.399	0.356
2	0.468	0.465	0.311	0.895	0.374	0.352
3	0.538	0.534	0.357	0.931	0.344	0.328
4	0.627	0.549	0.366	0.931	0.333	0.317
5	0.662	0.575	0.381	0.927	0.334	0.323
6	0.660	0.556	0.398	0.924	0.278	0.264
7	0.661	0.518	0.389	0.917	0.275	0.265
8	0.689	0.519	0.411	0.927	0.267	0.256
9	0.702	0.526	0.416	0.923	0.239	0.228
10	0.752	0.511	0.450	0.976	0.239	0.231
11	0.793	0.506	0.432	0.955	0.206	0.202
12	0.794	0.486	0.416	0.949	0.194	0.190
13	0.817	0.476	0.414	0.946	0.194	0.190

evenly spaced grid. The width of this grid spacing could have an impact on the quality of the results, since too few clusters would cause road sections to be missed, while too many clusters could cause multiple clusters to converge on a single road section. In addition to the quality of the results, the number of clusters also effect the running time of the algorithm.

In the original work by Doucette *et al.* [39], this cluster width parameter is calculated as a function of the nominal road width and minimum expected road length. Doucette *et al.* apply the SORM algorithm to two HYDICE images, each with a GSD of 1.0 m. The first image has a resolution of 300×300 pixels, and the second a resolution of 510×404 pixels. A grid spacing of 7×7 and 10×8 is used respectively. This spacing translates to a ground sampling spacing of 42.86×42.86 m for the first scene and 51×50.5 m for the second. Considering the 61 cm GSD of the QuickBird imagery, the grid spacing by Doucette *et al.* translates into a spacing of 70.27×70.27 pixels and 83.61×82.79 for the images used in this study.

To this end, the impact of the grid spacing width was evaluated for the QuickBird data set. As with the $cc_{\min}$ a range of values was tested on 12 images from the data set by running the full road extraction pipeline and calculating the quality metrics for

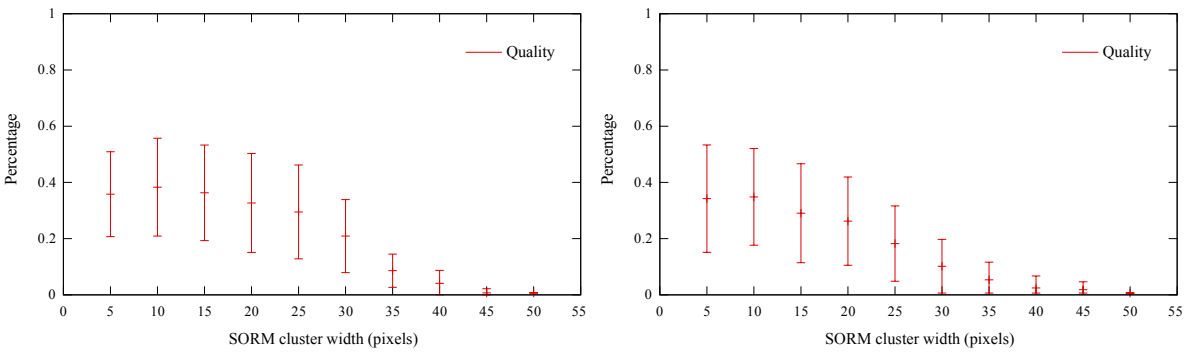
every evaluation. Increments of 5 were used to test a range of grid widths from 5 to 50. The results are illustrated in Figure 4.5. Error bars were used to illustrate the mean quality of the results over the 12 evaluations. The error (ε) was selected at one standard deviation from the mean. It is clear that the grid spacing used by Doucette *et al.* would be too large for the imagery used in this study and by considering both flavours of ACE, the final grid spacing was selected at 10×10 pixels.

4.4 Results and discussion

The system described in Section 4.2 was executed for both ACE variants for each of the 35 scenes in the data set. Figures 4.6 and 4.7 illustrate sample outputs from the Canny and V1 approaches respectively. Please see Section A.1 in Appendix A for more sample images. A visual comparison between the edge extractor outputs in Figures 4.6a and 4.7a reveals Canny as suited for detecting edges in the local neighbourhood, while V1 is geared towards detecting longer edges. The Canny ACE variant appears to have a higher degree of background noise than the V1 ACE approach. When evaluating the centreline seeds (Figures 4.6b and 4.7b), the Canny approach appeared to have a higher number of correctly detected seed points, but also a higher degree of background noise. This result is repeated for the connected component (Figures 4.6c and 4.7c) and consequently also for the SORM output (Figures 4.6d and 4.7d).

The quality metrics derived for Figures 4.6d and 4.7d are presented in Table 4.3 and confirm the above observations. In this instance, the Canny approach achieved higher values than the V1 variant over the three metrics.

The final median quality values over each road class are displayed in Table 4.4. The



(a) Canny ACE (b) V1 ACE

Figure 4.5: Error bars of ACE variants for various SORM cluster widths

(a) Canny edges (b) Canny ACE

(c) Canny ACE $cc_{\min}$=7 (d) Canny SORM (final)

Figure 4.6: The Canny approach applied to the sample scene

(a) V1 edges (b) V1 ACE

(c) V1 ACE $cc_{\min}$=7 (d) V1 SORM (final)

Figure 4.7: The V1 approach applied to the sample scene

Table 4.3: A comparison of the median quality values for the sample scenes illustrated in Figures 4.6d and 4.7d

Metric	Canny	V1
correctness	0.985	0.925
completeness	0.628	0.532
quality	0.623	0.510

Table 4.4: Comparison of median quality values by image classes for the Canny and V1 approaches over all scenes. No statistically significant difference was detected between algorithms and classes when applying the Mann–Whitney U test using a 5% confidence level

	Canny			V1		
Class	cor	com	qual	cor	com	qual
Class A	0.643	0.546	0.459	0.760	0.392	0.339
Class B	0.189	0.056	0.051	0.098	0.021	0.017
Class C	0.738	0.369	0.339	0.722	0.290	0.268
Class D	0.548	0.424	0.314	0.677	0.230	0.215

Canny approach obtains higher median quality values than its V1 counterpart, over all four classes by 3.4–12%. The collective results displayed in Table 4.5 reaffirm the stratified results, with the Canny approach having a higher median quality value by 8.3%.

The *Mann–Whitney U* test was applied with a 5% confidence level to the collective results over all classes to evaluate whether a significant difference between the performance of the Canny and V1 approaches exists. The p-values for the correctness, completeness and quality values ranged between 0.1–0.28. The null hypothesis, namely that there is no difference between the performance of the Canny and V1 implementations, cannot be rejected at a 5% confidence level.

The same statistical test was applied to each of the classes individually to determine

Table 4.5: Comparison of the collective median quality values over all classes

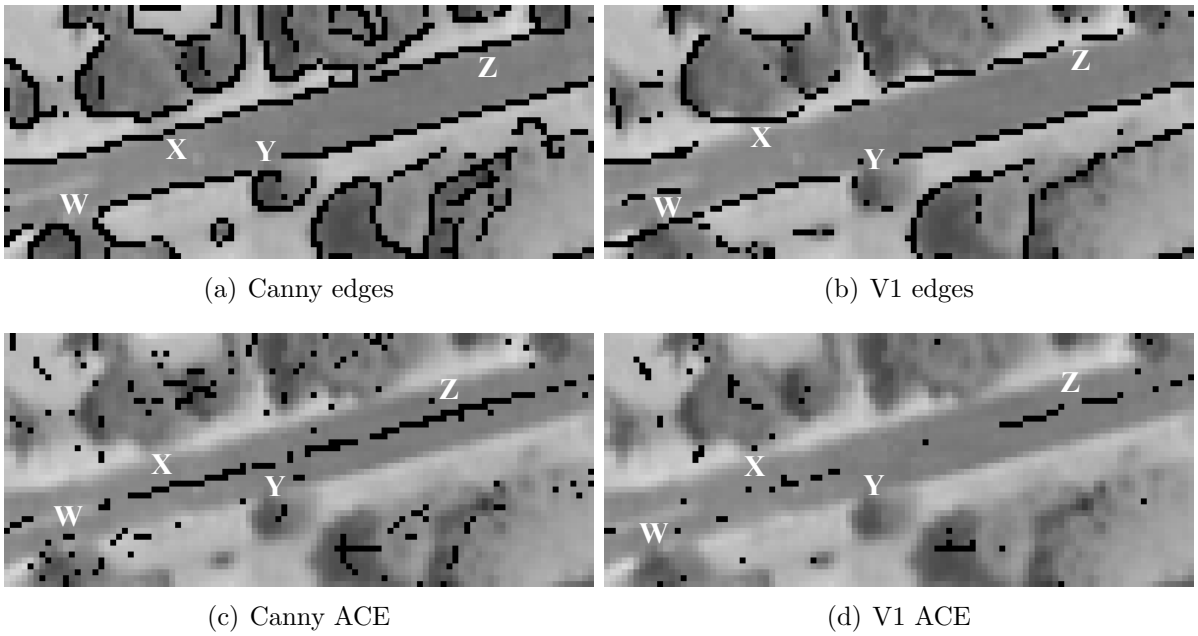
Metric	Canny	V1
correctness	0.656	0.698
completeness	0.424	0.244
quality	0.304	0.221

whether there is any considerable difference between the approaches on the stratified Bonferroni results. The p-values obtained over the three quality metrics and four classes ranged between 0.16–0.691. As a result, the null hypothesis can once again not be rejected at a 5% confidence level.

Figure 4.8 provides a detailed comparison of the Canny and V1 approaches. A visual comparison of both figures 4.8a and b illustrates that Canny detects more edges than V1, which is more inclined to extract a single edge where two or more parallel Canny-edges are in close proximity. When considering the edges extracted around the roadsides, four areas of interest have been identified and marked as points W, X, Y and Z in Figure 4.8. The remainder of this section presents a discussion of the edges produced by the two algorithms, which is followed by an examination of the effect each algorithm has on ACE.

At point W, the foliage adjacent to the road prevented Canny from detecting the entire road edge. Rather than continuing along the roadside, Canny veered away from the road surface. V1 was able to circumvent this problem and managed to extract a greater portion of the roadside than Canny. Closer scrutiny of point X reveals a rapid change in intensity in the area between the roadside, the sidewalk and vegetation. Although the separation was only a pixel wide at the nearest point, Canny was able to detect each of the changes individually. Rather than extracting an edge for each change in intensity, V1 opted to extract a single edge, causing a breakage in the edge along the roadside and pulling the foliage-edge slightly towards the road surface. As with point W, vegetation covers part of the roadside at point Y. Both Canny and V1 struggled to detect the road edge at this point, which resulted in an edge breakage. With regards to point Z, Canny managed to identify the roadside, but V1 pulled the road edge towards the nearby shrubbery which resulted in a breakage. This concludes the discussion of the edges produced by Canny and V1. The section concludes in the following paragraph by considering the results produced by ACE for both edge detector variants.

Figures 4.8c and 4.8d illustrate the output produced by ACE for the Canny and V1 approaches respectively. The ACE algorithm experienced difficulty in detecting the centreline pixels from the input provided by Canny at point W. Although V1 managed to extract the road edges on either side, ACE failed to extract the centreline pixels. The road intensity profile at point W (I_a) and an idealized profile (I_b) are depicted in Figures 4.9a and 4.9b. The edge points on intensity profile I_b have opposing gradients, while the edge points on I_a have similar gradients. Given the similar orientations of the edge points at point W, ACE was unable to identify the centreline seeds. The Canny variant was able to detect the centreline at point X, since both roadsides were identified.



(a) Canny edges (b) V1 edges

(c) Canny ACE (d) V1 ACE

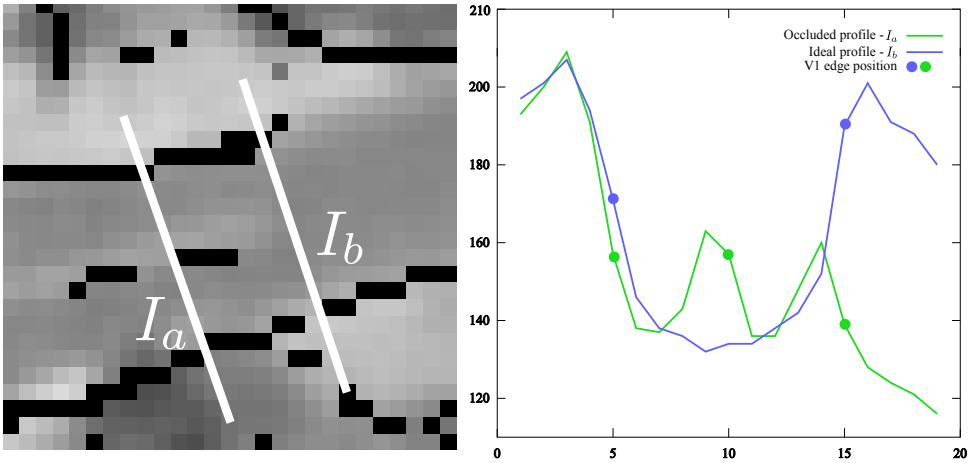
Figure 4.8: Detailed comparison of Canny and V1 in the sample scene

The breakage caused by V1 proved too severe and prevented ACE from detecting the entire centreline segment. Although the foliage-edge was pulled closer to the road, the underlying intensity gradient was in the wrong direction, which caused ACE to fail. Moving onto point Y, a slight breakage is visible in the Canny-ACE variant which is caused by the absence of an opposing edge. V1 detected few centreline seeds, since gaps occurred on either side of the road. Finally, Canny was able to detect the centreline at point Z, but the breakage in the V1 edges resulted in a breakage in the ACE output. Although the remaining V1 edges were pulled upward and away from the area where the most rapid change in intensity occurred, ACE was still able to detect the centreline pixels.

4.5 Conclusion

This chapter investigated the possibility of improving the quality of road extraction, by substituting the well known Canny edge detection algorithm [23] with the V1 neural network algorithm [110] in a system based on the work of Doucette *et al.* [38].

V1 is modelled after the human primary visual cortex, which possesses the ability to extract long-range edges. Despite published evidence claiming that V1 has superior edge coherence when compared to Canny's algorithm [110], this dominance does not appear



(a) Paths consider for profile (b) Intensity profile

Figure 4.9: Road intensity profile over V1 edges

to translate into a higher road extraction quality.

V1's substandard performance could be attributed to one of the following three factors:

- V1 might not possess the sensitivity required to detect the roadsides with the same accuracy as Canny. As indicated in the previous section, V1 has a tendency to extract a single edge where parallel edges occur in close proximity.

- The intensity gradient over the V1 edge pixels might not have the correct orientation. As a result, ACE fails when testing for opposing edges, even in instances where the roadsides have been detected accurately.

- The road width thresholds used for the ACE algorithm were determined by considering the change in intensity along the roadsides. The V1 edges might not be detected in the area where the change in intensity occurs and different width thresholds might be required.

Addressing these factors might increase the performance of the V1 approach. When considering the empirical results presented in this chapter, it seems unlikely that the V1 variant of the ACE algorithm would be able to produce better results than the Canny alternative.

CHAPTER 5

Object-based Analysis

"The real problem is not whether machines think but whether men do."

— Burrhus Frederic Skinner

Various remote sensing feature extraction approaches perform analysis on a per-pixel basis, where each pixel is analysed as an individual object in the feature space. The arrival of very high-resolution satellite imagery, such as the QuickBird imagery used in this study, introduced the option of segmenting an image to apply Object-Based Analysis (OBA). OBA considers an entire segment rather than individual pixels in feature space. The aim of segmentation is to add structure to the data, which allows for faster and more accurate analysis. This chapter presents a system configuration designed to investigate the possibility of obtaining a higher road extraction quality with OBA as opposed to Pixel-Based Analysis (PBA).

This chapter provides an introduction in Section 5.1, followed by a description of the processing chain in Section 5.2. Section 5.3 discusses the experimental design, while the

experimental results and discussion are provided in Section 5.4. The concluding remarks are presented in Section 5.5.

5.1 Introduction

PBA is a method where each pixel in an image is examined on an individual basis. Although PBA is a popular approach in feature extraction literature, the introduction of VHR imagery allows for analysis to be conducted on an object rather than a pixel basis. Detecting objects is usually performed using a segmentation algorithm, which aims to capture the characteristic relationship between the pixels that comprise an object in feature space. This chapter evaluates the accuracy of road extraction systems using OBA as opposed to PBA on VHR imagery.

A system is proposed which facilitates a comparison of the quality measures obtained through OBA and PBA road extraction. The system presented in Chapter 4 is adapted to extract road centrelines from images produced by spectral classifiers. The values within these road probability images represent a pixel or object's likelihood of being part of a road. The ACE component is replaced with the classification process by Mena and Malpica [115], which is implemented for both OBA and PBA. The FHS algorithm [41] described in Section 3.4 is used to detect objects within the RS images. The proposed road extraction system and its sub-processes are presented in the following section.

5.2 Road extraction pipeline

Section 4.2 discussed a road extraction system based on the work of Doucette *et al.* [38]. In this chapter, the system is modified to accommodate both an object and pixel-based classification process. The modified system is depicted in Figure 5.1. A brief description of each system component follows:

1. A candidate image and a set of road models are provided as input to the system. The candidate image is the scene from which roads should be extracted. A road model comprises a set of pixels that models the spectral characteristics of roads in the candidate image.

2. When the OBA approach is followed, the candidate image and road models are segmented with the FHS algorithm (refer to Section 3.4). Each segment produced

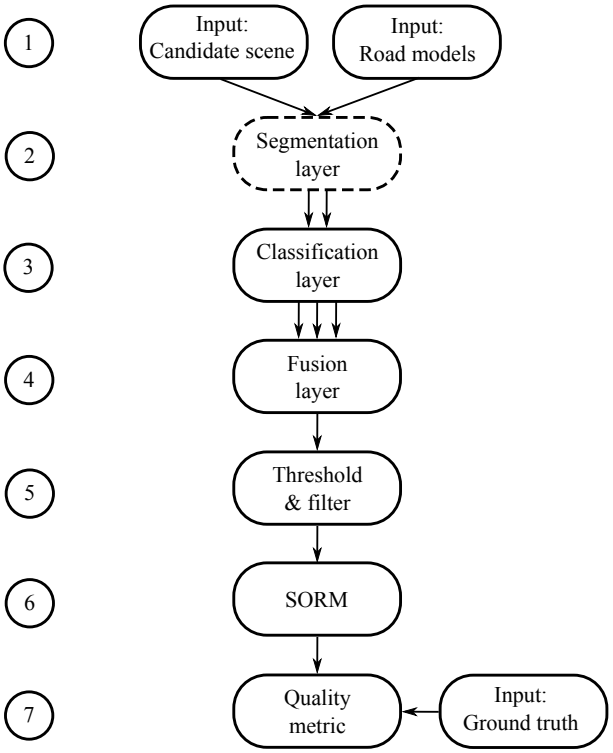


Figure 5.1: Road extraction pipeline

by the FHS method is considered as an object in feature space. The reasoning behind segmenting the road model is discussed in Section 5.3.2. No segmentation step is present in the PBA processing chain.

3. Using the road models provided, the classification layer applies the three classifiers to the candidate image from the layer above. Each classifier generates individual pseudo-probability values of the likelihood of an object or pixel forming part of the road class. Please refer to Sections 3.5.1–3.5.3 for detailed information on these classifiers.

4. The three outputs from the classification layer are combined into a single output in the fusion layer. The Dempster-Shafer fusion algorithm used is described in Sections 2.4 and 3.6.

5. A binary image is created by thresholding the results obtained from the fusion layer. The thresholding process results in all objects or pixels being classified in either the *road* or *non-road* class. Thresholded images tend to contain small clusters of isolated pixels as well as smalls gaps between road segments. A number of

these discrepancies are removed with a mathematical morphology filtering process. This process comprises a connected component filter (refer to Section 3.7.1) and a closing operation (refer to Section 3.7.2).

6. Using the SORM algorithm (refer to Section 3.8), road centrelines are extracted from the remaining road segments in the binary image.

7. Finally, the quality of the results is determined using the quality metric by Heipke *et al.* [63], as discussed in Sections 2.6 and 3.9.

5.3 Experimental design

Various aspects regarding the system implementation and experimental design are presented in this section. The section starts with a detailed review of the data set, followed by several implementation aspects. The section concludes with a discussion of the parameter optimisation process and colour spaces.

5.3.1 Data set

Fifty scenes were selected for the data set to be used during experimentation in Chapter 5 and Chapter 6. Each of these candidate images had a pixel dimensionality of 512×512. The scenes were selected to be representative of the real world scenarios that a road extraction application would have to deal with. To this end, a variety of roads from multiple land cover types were selected. The data set also included a variety of different road surfaces. The road visibility ranged from partly occluded to clearly visible. A different combination of images from the data set presented in Chapter 4 was used, as the class definitions in that chapter were selected to test specific aspects of the ACE algorithm.

The SORM algorithm was optimized for single and dual carriageways. Areas with wider roads, such as national roads or roads in dense urban areas were not included. Although SORM can be optimized to extract wider roads, the inclusion of additional road types was considered to have little effect on the objective of this study.

The road network ground truth and a set of road models were extracted for each candidate image. The set of road models contained a single model for every type of road in the image. If an image contained, for example, both dirt and asphalt roads, a road

model was created for each surface type. The road centreline ground truth was digitized manually from the original candidate image.

5.3.2 Implementation

Two points regarding the use of road models with the segmentation layer should be noted. Firstly, both the candidate image and its road models were segmented when executing OBA. Segmentation of the road models resulted in a set of sub-road models being created. Each object in the candidate image was compared to every sub-road model. The comparison resulted in a set of pseudo-probability measures for every object within the candidate image. The pseudo-probability represents the likelihood of a candidate image object forming part of the given sub-road model. The road model providing the closest representation of the candidate image object was used for classification.

Secondly, in the original work by Mena and Malpica [115], a median filter with a 3×3 window is used to smooth the initial candidate image. As information was discarded in the process, it was decided not to apply a median filter in this study.

5.3.3 Parameter selection

Various system parameters at each process level can be optimized to increase the extraction accuracy. These parameters include:

- Segments shape and size parameters for the FHS algorithm;
- Closing SE disc radius and minimum connected component sizes in the filtering step;
- Cluster width used to initialize the SORM cluster centroids; and
- The probability threshold value used to create the binary image.

To optimize these parameters, a fitness function was used to calculate the quality obtained by the RNE system for a given image. The fitness function accepts the parameters in the aforementioned list and produces the quality metrics (discussed in Section 2.6 and 3.9) as output. These metrics were used to gauge the effect a set of parameters has on the quality of the results.

The fitness function executed the system described in Section 5.2 end-to-end. It is clear that a single fitness function evaluation is a computationally expensive operation.

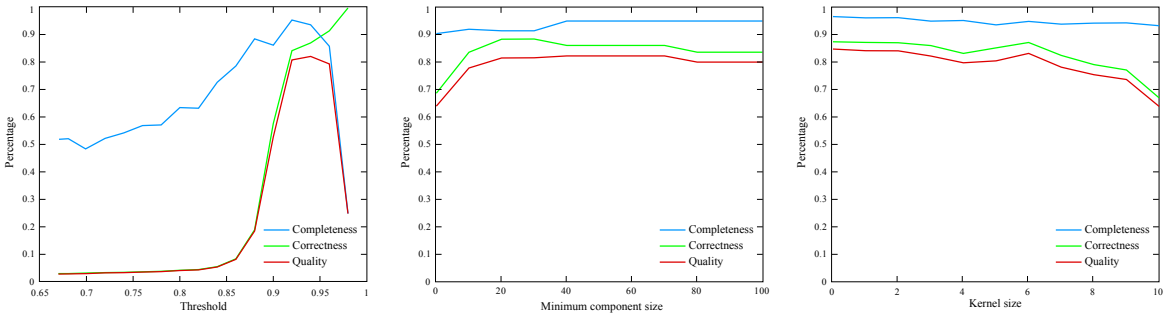
A caching component was included to guard against repetition of operations already performed. Even though caching saved a significant amount of processing time, the fitness function remained computationally expensive.

To optimize the set of input parameters in a timely fashion, an efficient optimization algorithm, with the ability to optimize multiple parameters simultaneously, was required. Two optimization algorithms, namely GSS and Nelder-Mead, were used in conjunction to fulfil these requirements. The application of these two methods is discussed below.

Firstly, the GSS (refer to Section 2.7.1) algorithm was employed to evaluate each parameter individually to determine the broad range where a global minimum could be found. GSS was able to cover a large search space efficiently [82].

The ranges determined by the GSS operation were refined further using the second optimization method, namely the Nelder-Mead optimization with relaxed parameters (refer to Section 2.7.2). The Nelder-Mead approach was selected as it requires a minimal number of function evaluations and is able to search across multiple dimensions. A further advantage of Nelder-Mead is that it is not overly sensitive to local minima [120].

The quality metrics for varying parameter values are illustrated in the graphs depicted in Figure 5.2 for a single image. These graphs were obtained by evaluating the parameters on the x-axis (*probability threshold*, the *minimum connected component size* and the *closing SE size*) at consistent intervals. The quality metrics are computed using a single PBA Mahalanobis classifier. The graphs illustrate the behaviour of the quality metrics as well as the sensitivity of these parameters. It is clear that thresholding has a considerable effect on the quality of final extracted roads, while the effect of the minimum connected component and closing SE is less significant.



(a) Threshold limit (b) Connected component size (c) Closing operation size

Figure 5.2: Quality assessment of varying parameter values for a single candidate scene using the PBA Mahalanobis classifier

When optimizing the parameters for a system, there is a risk of overfitting the parameters to the sample data set. Overfitting occurs when noise or random error is modelled rather than the underlying patterns in the sample data. This results in suboptimal parameters to be selected, which will cause a reduction in the performance of the system when applied to other data sets. The data set was partitioned into a training set of 20 images and a test set of 30 images to guard against overfitting. The parameter optimization was performed on the training set and the results evaluated against the test set. Parameter optimization was repeated for each individual experiment as any changes within the structure of the processing chain could have an impact on the interaction between the parameters. More detail on the optimization process is given in Section 5.4.

5.3.4 Colour spaces

The three classifiers discussed in Sections 3.5.1–3.5.3 all calculate the distance between two distributions. These distributions are defined within the dimensions of a given colour space. Different colour spaces provide different representations of image data. Considering that the mapping between some of these representations is non-linear, they will have an effect on the distance calculations, and influence the performance of the classifiers.

Mena and Malpica achieved the best results when using the RGB colour space for both the Mahalanobis and Bhattacharyya-based approaches and HSI for the texture cube approach. To ensure comparable results, this study followed the same colour space configuration. HSI images were created from the original RGB scenes using the following calculations:

$$\begin{aligned} H &= \begin{cases} \cos^{-1}\left(\frac{(r-g)+(r-b)}{2\sqrt{(r-g)^2+(r-b)(g-b)}}\right) & \text{if } b < g \\ 2\pi - \cos^{-1}\left(\frac{(r-g)+(r-b)}{2\sqrt{(r-g)^2+(r-b)(g-b)}}\right) & \text{otherwise} \end{cases} \\ S &= 1 - 3\min(r,g,b) \\ I &= \frac{1}{3}(R+G+B). \end{aligned}$$

where $H \in [0, 2\pi]$, $S \in [0, 1]$ and $I \in [0, 1]$.

5.4 Results and discussion

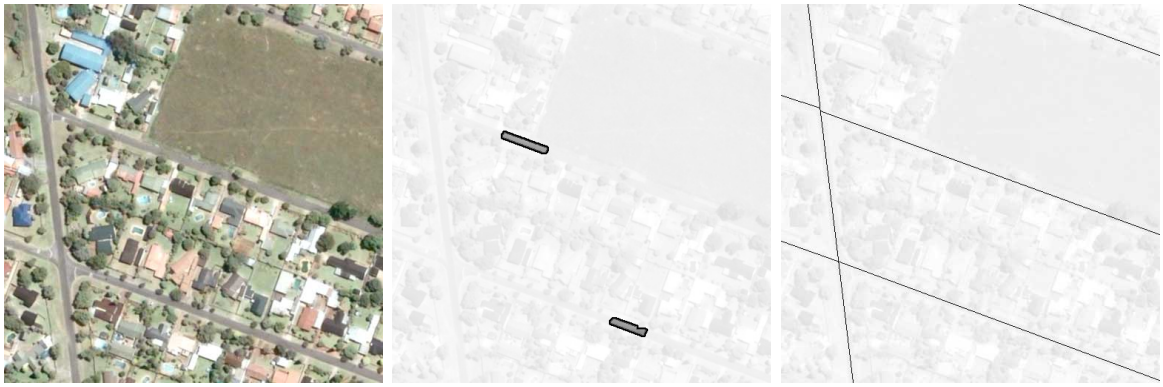
The experiments conducted in this chapter focused mainly on investigating the possibility of increasing the accuracy of road extraction results by using OBA instead of PBA. Further experimentation considered the contribution of fusing multiple classifiers as opposed to using a single classifier. A comparison of the OBA and PBA approaches is presented in the first subsection, which is followed by a comparison of OBA and PBA when only a single classifier is used.

5.4.1 Object analysis experiment

The first experiment investigated whether an increase in quality could be obtained by employing OBA instead of PBA. The OBA approach entails executing the system discussed in Section 5.2 with the segmentation layer enabled, while the PBA approach is executed without the segmentation layer. An in-depth discussion of the experiment and the results follows below.

Figure 5.3 depicts a single sample from the 50 scenes used during experimentation. The system is presented with three images per scene, namely the *candidate image*, its *road models* and the *road network ground truth*. Depending on the classifier used, the candidate image and its accompanying road models are either RGB or HSI images. The ground truths were extracted manually and presented to the system as a greyscale image, where centreline pixels have a value of 255 and the remaining pixels a value of 0. Please see Section A.2 in Appendix A for more sample images.

A sample of the intermediate results of the PBA approach is depicted in Figure 5.4.

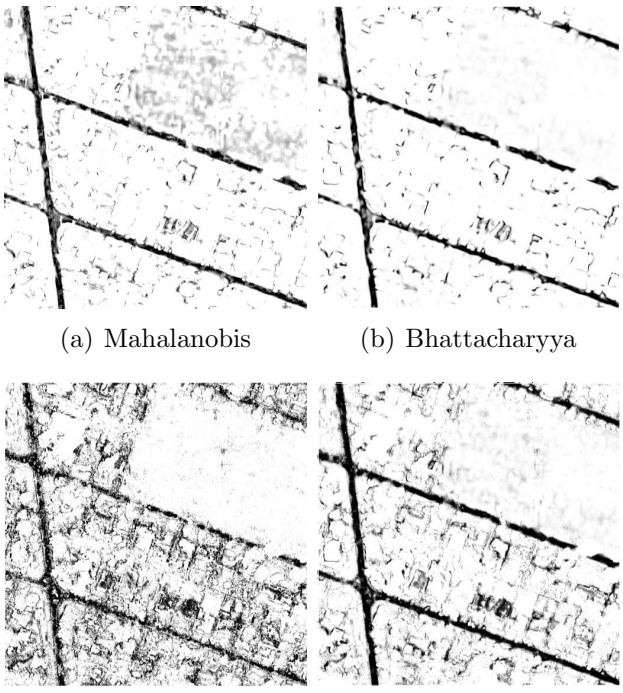


(a) Candidate image (b) Road models (c) Ground truth

Figure 5.3: A sample of the images used as system input

A visual comparison of the Mahalanobis (Figure 5.4a) and Bhattacharyya (Figure 5.4b) classifiers reveals comparable results. Both Mahalanobis and Bhattacharyya determine the spectral distance in RGB between a candidate pixel and a road model. The Mahalanobis approach detected long homogeneous sections with high probability values, while smaller isolated sections with high probability values are also visible. The blurred appearance of the Bhattacharyya can be attributed to the 5×5 kernel used to sample the distribution of values around the pixel under consideration. The texture cube classifier (Figure 5.4c) also detected long homogeneous sections, but these areas appear somewhat fragmented. Numerous smaller isolated sections are also visible. Even though the distance function is calculated from a 3×3 kernel, the overall texture cube results appear noisy. The Bhattacharyya and texture cube classifiers were somewhat more successful than the Mahalanobis approach in disregarding the open field at the top right of the scene.

At this stage in the process, the outputs from the three different classifiers were combined with Dempster-Shafer fusion (Figure 5.4d). The objective of fusing information from disparate sources is to achieve a higher accuracy by capitalizing on each classifier's distinctive feature extraction ability. DST required an uncertainty value for



(a) Mahalanobis (b) Bhattacharyya

(c) Texture cube (d) Fused output

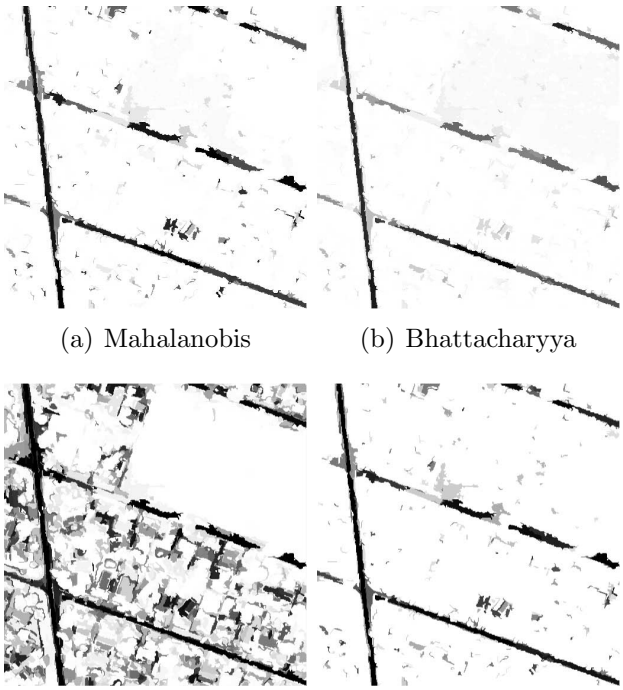
Figure 5.4: Pixel classifier output

each classifier, which serves as a weighting factor during fusion. This uncertainty value was obtained by testing the classifiers' ability to produce accurate classifications. This ability was determined by applying the classifier to the road models it was based on. Such an application should produce probability values close to one, as the model is classifying the same data it is modelled upon. The standard deviation of the probability value serves as an indication of the classifiers' confidence in making accurate predictions. A low standard deviation is indicative of a higher confidence. Due to the variety of road surfaces in the data set, each classifier's uncertainty is calculated individually for every candidate scene.

Figure 5.5 presents a sample of the intermediate results from the OBA approach. In Figure 5.5a the Mahalanobis approach detected elongated segments with high probability values. Isolated segments and sections with lower probability values are also apparent. Similarly, the Bhattacharyya approach (Figure 5.5b) detected elongated segments and isolated sections with lower probability values. The texture cube approach assigned high probability values to numerous segments (Figure 5.5c). The overall result appears noisier than the previous classifiers. The fusion of the three OBA classifiers is illustrated in Figure 5.5d. The fused result appears to have slightly more noise than the Mahalanobis and Bhattacharyya classifiers, but succeeded to suppress much of the background noise created by the texture cube approach.

After the classifier outputs were fused, thresholding was employed to divide the image into a road and non-road class. Mathematical morphology was used to remove some isolated road clusters and to close a number of gaps between road segments. Figure 5.6a illustrates a sample of the filtered PBA image. Once the filtering process had been completed, the SORM algorithm was used to extract the road centrelines (Figure 5.6b). For this sample scene, the system was able to detect the road centrelines with a quality value of 86.3%. SORM failed to detect two short sections at the top and right of the image, which illustrates its limitations in bridging gaps between clusters. As a result, various road sections were discarded, which resulted in the centreline being too short.

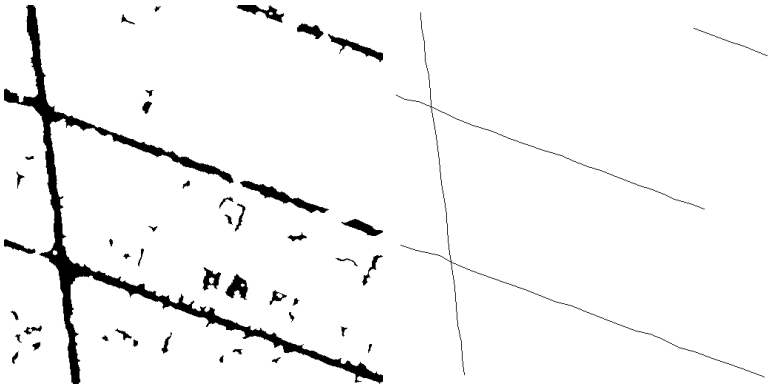
The OBA followed the same process as PBA with the fused result being thresholded and filtered before running the SORM algorithm. The filtered OBA image and the corresponding SORM output are illustrated in Figure 5.7. In this instance, SORM managed an extraction quality value of 64.4%. Noteworthy are the two gaps towards the middle and top of the image as well as the peculiar behaviour at the intersections to the left of the image. The two gaps can once again be attributed to SORM's limited ability to breach gaps. The behaviour at the intersections is a result of the linking process,



(a) Mahalanobis (b) Bhattacharyya

(c) Texture cube (d) Fused output

Figure 5.5: Object classifier output



(a) Thresholding and filtering (b) SORM centrelines

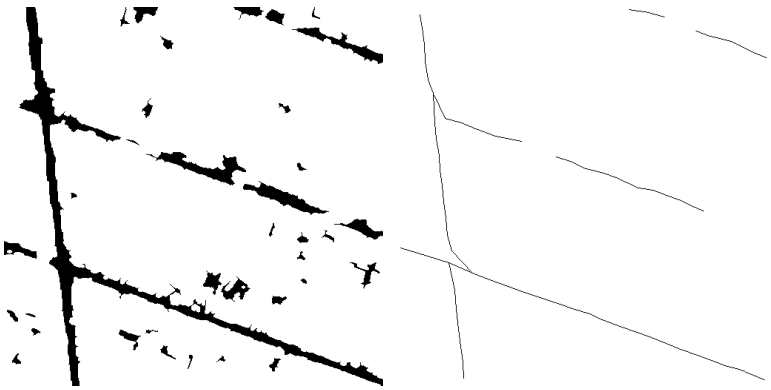
Figure 5.6: Resulting centrelines for the PBA approach

which focuses on local connections rather than considering the network holistically.

The training and test sets were obtained by partitioning a random permutation of the 50 scenes in a 40%/60% ratio. To reduce bias, ten cross-validation rounds [88] were selected with a constant training and test set partition size. To reduce optimization processing time, constant values were assigned to system parameters that were deemed not to have a considerable effect on the quality measures. The following parameters were assigned constant values:

- The FHS shape parameter (g) was set to 0.38 and the size parameter (k) was set to 65.
- In the thresholding and filtering component, the closing SE was selected as a disc with a radius of 2 pixels, while the minimum connected component size (cc_{min}) was set to 30.
- The SORM algorithm's cluster width was set to 20 pixels.

Table 5.1 presents the quality measures obtained over the ten cross-validation rounds for the test set. The mean values were calculated as the total mean of the quality metrics over all ten cross-validation rounds. The total standard deviation values were computed using the law of total variance and comprise two components, namely the mean standard deviation and the standard deviation of the means. The mean standard deviation was calculated as the mean of the ten round's standard deviation values. The standard deviation of the means was calculated as the standard deviation of the ten cross-validation round's mean values.



(a) Thresholding and filtering (b) SORM centrelines

Figure 5.7: Resulting centrelines for the OBA approach

Table 5.1: Results for PBA and OBA with the fusion of the three classifiers

	Mean (Total standard deviation)		
Type	**Correctness**	**Completeness**	**Quality**
PBA	0.838 (0.234)	0.735 (0.085)	0.644 (0.129)
OBA	0.834 (0.131)	0.734 (0.162)	0.640 (0.063)

When considering the test set quality values, the PBA performed marginally better than the OBA. The PBA achieved a quality of 64.4%, while the OBA managed 64.0%. The PBA system had a total standard deviation of 12.9%, compared to the 6.3% obtained by OBA. The difference in the total standard deviation quality values is more pronounced than the difference in mean values, indicating that the OBA was the more consistent of the two approaches. Taking the total standard deviation into account, the correctness and completeness values of the two approaches did not differ substantially. Considering both the mean and total standard deviation values, it is evident that the two approaches are comparable.

To evaluate these findings statistically, the Mann–Whitney U test was applied with a 5% confidence level to the collective results over all ten cross-validation rounds to evaluate the null hypothesis. In this instance, the null hypothesis is the proposition that no considerable difference between the results produced by the OBA and PBA algorithms exists. The p-values for the correctness, completeness and quality values ranged between 0.28–0.92. This implies that the null hypothesis is not rejected at a 5% confidence level. This outcome is supported by close proximity of the mean quality values and the spread of the distributions, as suggested by the total standard deviation.

It should be noted that the presented results are only applicable to the relevant system configuration and should not be generalized for all feature extraction approaches. Employing a different segmentation algorithm in the OBA approach might produce different results. A qualitative evaluation of the results produced by the FHS method found the segments to agree well with subjective human interpretation. Given this finding and the range of the total standard deviation values, it is unlikely that an alternative segmentation algorithm will produce significantly better results than the PBA method.

A final noteworthy observation regarding the three classifiers is the visually noticeable commonality between the spectrally classified images. The objective of fusion is to combine information for dissimilar sources, with the aim of obtaining more accurate

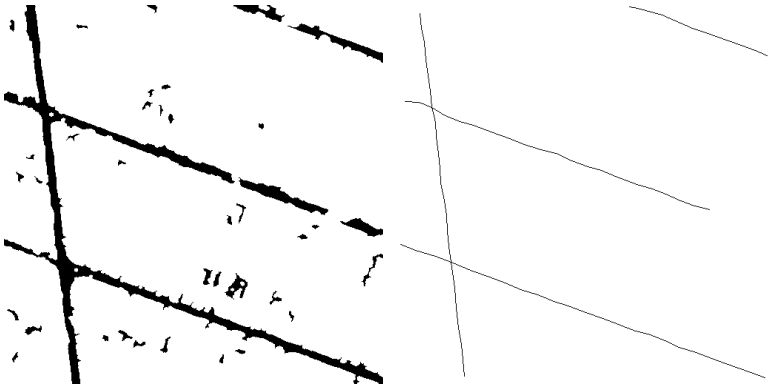
results. When fusing the three classifier outputs, the disparity between input sources might be too low to produce considerable improvements in accuracy. The contribution by the data fusion process to road extraction quality is investigated in the next experiment.

5.4.2 Fusion experiment

The second experiment investigated whether the fusion of classifiers contributed to an increase in the accuracy of the overall system. This experiment evaluated each of the classifiers separately for both PBA and OBA, and subsequently compared the results to the fusion approach in the previous experiment.

As with the previous experiment, ten cross-validation rounds were used. The mean results are depicted in Table 5.2. The Mahalanobis PBA achieved the highest quality measure with a value of 62.9% and a total standard deviation of 16.4%. Figure 5.8 depicts a sample from this approach. The Mahalanobis OBA yielded a lower quality measure of 57.0% and total standard deviation of 7.01%. The PBA and OBA Bhattacharyya approaches produced marginally lower results, obtaining quality measures of 59.6% and 61.9% respectively. The PBA and OBA texture cube method produced poorer results than the Mahalanobis and Bhattacharyya variants. The texture cube total standard deviation values were also high in comparison to that of the other classifiers.

As with the previous experiment, the results are evaluated statistically with the Mann–Whitney U test, applied at a 5% confidence level to the collective results over all ten cross-validation rounds. The Bonferroni correction [17] was used to control the probability of *type I errors* (α), since multiple pairwise tests were performed. Table 5.3



(a) Thresholding and filtering (b) SORM centrelines

Figure 5.8: PBA Mahalanobis approach

Table 5.2: Results for PBA and OBA when using individual classifiers without data fusion

	Mean (Total standard deviation)		
Type	**Correctness**	**Completeness**	**Quality**
PBA Mahalanobis	0.792 (0.224)	0.747 (0.097)	0.629 (0.164)
OBA Mahalanobis	0.772 (0.372)	0.698 (0.309)	0.570 (0.071)
PBA Bhattacharyya	0.799 (0.359)	0.708 (0.283)	0.596 (0.121)
OBA Bhattacharyya	0.810 (0.274)	0.727 (0.168)	0.619 (0.115)
PBA Texture Cube	0.675 (0.280)	0.574 (0.316)	0.454 (0.166)
OBA Texture Cube	0.739 (0.302)	0.587 (0.321)	0.487 (0.145)

depicts a matrix of the p-values obtained from pairwise significance tests on the quality value. An asterisk is used to indicate cases where a statistically significant difference was observed. The significance tests confirm that approaches with quality values closer to each other do not differ significantly. For example, the two texture cube approaches obtained noticeably lower performance values in comparison to the other approaches and were marked as significantly different.

A point worth noting is that the Bhattacharyya OBA obtained statistically different results than the Mahalanobis OBA (refer to Table 5.2 and Table 5.3). Although the Mahalanobis and Bhattacharyya distance functions are calculated in a similar fashion, the Bhattacharyya distance function includes additional information in the form of a covariance matrix of the sample distribution. The usage of this additional information

Table 5.3: P-values obtained for multiple pairwise statistical significance tests on the quality values. The results are obtained with Mann–Whitney U test at a Bonferroni corrected 5% confidence level. The fused approach are abbreviated as **F**, Mahalanobis as **M**, Bhattacharyya as **B**, and texture cube as **T**

	PBA F	OBA F	PBA M	OBA M	PBA B	OBA B	PBA T	OBA T
PBA F	1.000	0.604	0.869	2.08E-10*	2.68E-04*	0.018	2.20E-16*	2.20E-16*
OBA F	0.604	1.000	0.908	3.55E-09*	2.93E-03	0.081	2.20E-16*	2.20E-16*
PBA M	0.869	0.908	1.000	1.06E-08*	1.57E-03*	0.070	2.20E-16*	2.20E-16*
OBA M	2.08E-10*	3.55E-09*	3.55E-09*	1.000	7.15E-03	7.57E-05*	5.27E-12*	2.76E-07*
PBA B	2.68E-04*	2.93E-03	2.93E-03*	7.15E-03*	1.000	0.205	2.20E-16*	2.96E-12*
OBA B	0.018	0.081	0.070	7.57E-05*	2.05E-01	1.000	2.20E-16*	2.22E-16*
PBA T	2.20E-16*	2.20E-16*	2.20E-16*	5.27E-12*	2.20E-16*	2.20E-16*	1.000	0.053
OBA T	2.20E-16*	2.20E-16*	2.20E-16*	2.76E-07*	2.96E-12*	2.22E-16*	0.053	1.000

allowed the Bhattacharyya OBA to obtain higher quality values than its Mahalanobis OBA counterpart.

The processing times of the various approaches are also worth noting. Table 5.4 depicts the processing time of each approach for a single cross-validation round, which comprises thirty candidate scenes. The test system configuration is discussed in Sections 3.12. The OBA offered an increase in processing speed on all approaches other than the Mahalanobis variant. This increase in speed can be attributed to the following three main factors:

- The Bhattacharyya distance function (refer to Equation 3.3) requires a mean and covariance matrix for a candidate sample and the road model. The OBA calculated a single mean and covariance matrix for a segment of pixels, while the PBA calculated these matrices separately for each individual pixel.
- When following the OBA, the Bhattacharyya distance calculation itself was also only calculated once for a segment of pixels. With the PBA, the distance function was executed for each individual pixel.
- The FHS algorithm is highly efficient, adding little additional processing time to the system.

The OBA Mahalanobis did not offer an increase in speed over its PBA counterpart as the Mahalanobis distance function (refer to Equation 3.1) requires no mean or covariance calculations for the candidate sample, and the distance function itself is less computationally intensive than the Bhattacharyya function. The Mahalanobis OBA offered little gains in speed by processing a segment of pixels, while the addition of a segmentation algorithm added to the total processing time.

Finally, by comparing the results from both experiments (Tables 5.1 and 5.2) the fusion PBA and fusion OBA achieved the highest quality value with a measures of 64.4% and 64.0% respectively. The Mahalanobis PBA obtained the highest quality

Table 5.4: Processing times for the respective OBA and PBA road extraction systems

	Time in minutes and seconds			
Approach	**Fused classifier**	**Mahalanobis**	**Bhattacharyya**	**Texture cube**
PBA	44:48	12:46	15:24	40:02
OBA	36:21	13:10	14:09	37:34

out of the single classifier based approaches with a measure of 62.9%. Although the fusion approaches yielded results with the highest quality, the increase in complexity and computational time should be considered. The benefit of data fusion might be more evident in cases where greater disparities between the sources exist and could be a topic of further studies.

5.5 Conclusion

This chapter presented a road extraction system to test both pixel and object-based analysis. The system was based primarily on the following approaches: the image segmentation method by Felzenszwalb and Huttenlocher [41], the classification system by Mena and Malpica [115], the road topology construction technique by Doucette *et al.* [38, 39], and finally the quality metric by Heipke *et al.* [63, 164].

The examination of the empirical results illustrated comparable accuracies between the pixel and object-based approaches. The pixel-based experiment yielded a mean extraction quality of 64.4% and total standard deviation of 12.9%, while the object-based approach obtained 64.0% and 6.3% respectively. The lower total standard deviation value obtained by the object-based approach implies that more consistent results were produced during experimentation.

Further analyses indicated that a higher extraction quality was obtained by fusing the results from multiple classifiers as opposed to using a single classifier. Even though a multiple-classifier system achieved a higher extraction quality, the increase in complexity and computational time should be considered. The single classifier Mahalanobis based approach obtained a quality measure of 62.9%, against the 64.4% of the multiple-classifier based system.

CHAPTER 6

Fully Automated Road Extraction using Spectral Classification

"The science of today is the technology of tomorrow."

— Edward Teller

The system presented in Chapter 5 can be considered a semi-automated system, since the road models used for classification were extracted manually. By adapting the system presented in Chapter 4 into a road model generator, a fully automated road extraction system based on spectral classification is proposed in this chapter.

This chapter provides an introduction in Section 6.1, followed by a description of the processing chain in Section 6.2. Section 6.3 examines the experimental design, while the experimental results and a discussion thereof are provided in Section 6.4. The concluding remarks are presented in Section 6.5.

6.1 Introduction

Although various road extraction methods and systems have been presented in Chapter 2, few of these could be regarded as fully automated approaches. While a number of systems have reached a high level of automation, additional input from an operator or existing GIS database is still required at certain stages within the processing chain. The system presented in this chapter aims to extract road centrelines by using no inputs other than a candidate image.

The proposed system can be considered as a two-part process. The first process generates road models (pattern classes) with an adapted version of the geometric road extraction system presented in Chapter 4. This Automated Road Model Generator (ARMG) builds on the SSRC feedback loop proposed by Doucette *et al.* [38] by using an iterative approach to select multiple road models. The second process extracts the road centrelines with the PBA Mahalanobis system presented in Chapter 5. The proposed road extraction system and its sub-processes are presented in the following section.

6.2 Road extraction pipeline

As per Section 6.1, aspects of the two road extraction systems presented in Sections 4.2 and 5.2 were integrated into a fully automated road extraction system. The automated system incorporates two key components, namely a road model generator (ARMG) and a road centreline extractor (PBA Mahalanobis and SORM).

The fully automated system builds on the SSRC process proposed by Doucette *et al.* [38]. The SSRC process starts by extracting a road topology with ACE and SORM. Road models are extracted at each node in the road topology by using a 3×3 window. The *k-means* segmentation algorithm [103] is applied to the entire scene and segments with dissimilar spectral signatures from the road models are selected as non-road models. Based on these models, the Bayesian ML algorithm [59] is employed to spectrally classify all pixels as either road or non-road, which results in the creation of a binary image. The process can be repeated iteratively by reapplying ACE and SORM to the binary image.

The fully automated system proposed in this chapter differs from the SSRC algorithm in the following regards:

- The SSRC system iteratively applies the entire system to a candidate scene. The system proposed in this chapter iteratively generates road models, but has a single road centreline extraction phase.

- An entire line segment is used to create an initial road model rather than the 3×3 windows used in the SSRC algorithm.
- Doucette *et al.* use the k-means algorithm to derive models for the non-road areas. This process is excluded from the system proposed in this chapter as the non-road class could contain a diverse range of features, which is difficult to model accurately.
- The Mahalanobis classifier is used in this study rather than the Bayesian ML algorithm used by Doucette *et al.*
- During the SSRC process, ACE is reapplied to spectrally classified images produced by the Bayesian ML algorithm. SORM is subsequently used to extract roads from the centreline seeds produced by ACE. The fully automated system proposed in this study reapplies SORM directly to the spectrally classified image produced by the Mahalanobis classifier.

The remainder of this section discuses the road model generator and the road centreline extractor. The ARMG's iterative process flow is depicted in Figure 6.1. A description of each component within the processing chain follows.

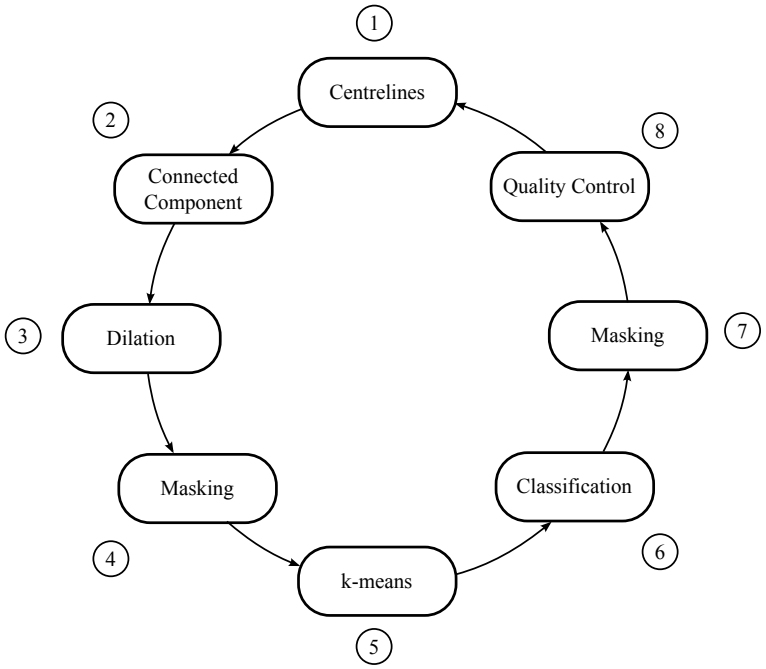


Figure 6.1: The iterative process chain followed by the ARMG

1. The method starts by obtaining road centrelines from a candidate image with the Canny-based ACE system discussed in Section 4.2. The road centrelines and background are depicted in a raster image as having values of 255 and 0 respectively. During subsequent iterations, road centrelines are obtained from the masking component illustrated in step 7.

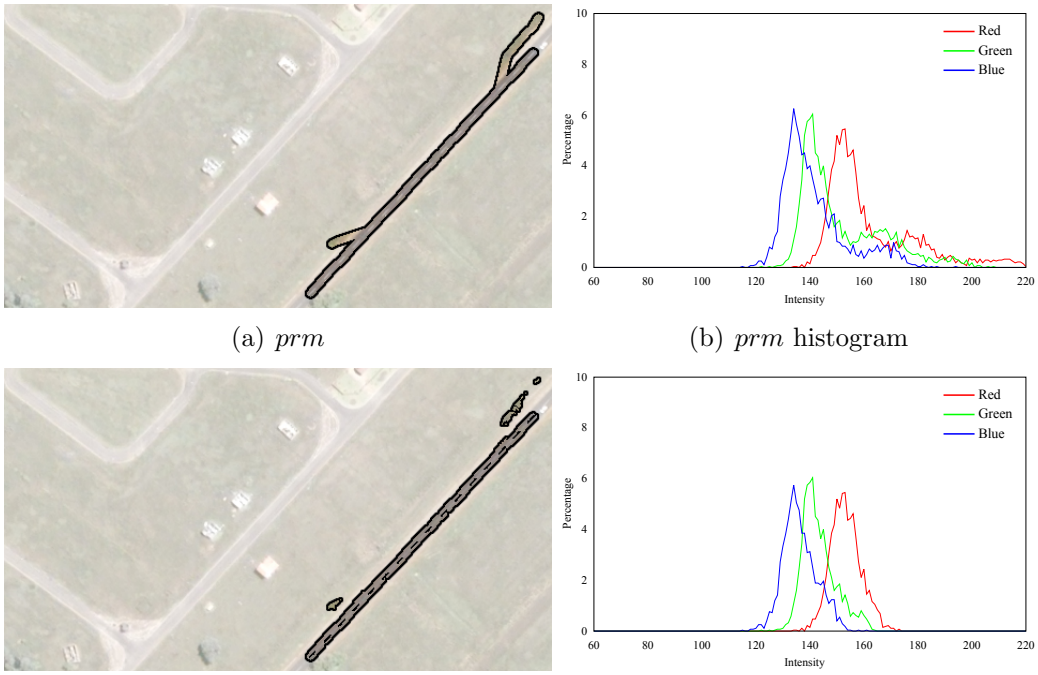
2. A connected component filter is used to select the largest road centreline. The size of the largest connected component (lcc) is evaluated against a predefined threshold (lcc_{min}), which guards against the extraction of a road model with a small distribution size. This regulating process is discussed in more detail in Section 6.3.2.

3. A disc shaped SE with a radius of 3 is used to dilate the lcc. The dilation process defines the image region from which the road model is extracted. The resulting raster image values are binarized, assigning a value of 1 to the dilated lcc and 0 to the background.

4. The original candidate image is multiplied by the dilated lcc image. This masking process causes the dilated lcc image to be populated with the original input scene's underlying spectral values. The resulting image can be considered as a *preliminary road model* (prm).

5. The prm may contain non-road objects, such as trees, cars, road markings or sidewalks. To remove non-road objects, an assumption is made that the majority of the pixels in prm belong to road models. The k-means algorithm is used to create two segments based on the prm's spectral values. The largest of these segments is selected as the final *refined road model* (rrm). Figure 6.2a illustrates a scenario where non-road areas form part of the prm. The k-means filter produced the rrm depicted in Figure 6.2c. In this instance, the filtering process succeeds in removing a number of non-road pixels. When comparing the corresponding histograms in Figures 6.2b and 6.2d, it is clear that pixels with higher intensity values (such as road markings) have been removed.

6. A candidate image may contain a variety of road types, such as dirt, concrete or asphalt. To produce models for all surface types within the candidate image, the ARGM process chain will have to be repeated. All the centrelines which might produce redundant road models will have to be masked. The mask is obtained



(a) *prm* (b) *prm* histogram

(c) *rrm* (d) *rrm* histogram

Figure 6.2: Road model filtering with k-means

by applying the classification process (Section 5.2), based on the extracted road model (step 5), to the original candidate image. The classification process produces a binary image, where pixels considered as road pixels have a value of 255 and the remaining pixels a value of 0. A threshold is introduced to reject road models that resulted in an overclassification of road pixels in the candidate scene (Section 6.3.2).

7. The binary image obtained from the previous step is used to mask the centrelines that might result in redundant road models being created. This is achieved by subtracting the binary image from the current centreline image.

8. The extracted road model may contain too many divergent spectral values, which may result in the classifier producing inaccurate results (step 6). A quality control process evaluates whether the *lcc* detected in step 2 was removed in the preceding masking step. The percentage of the *lcc* that was removed is calculated and a threshold is introduced to discard road models that were not sufficiently covered by the mask (Section 6.3.2).

The aforementioned ARMG process is repeated until no more road models are detected and the remaining models are kept as valid road models.

The second key component is the road centreline extractor, which is a slightly modified and simplified version of the system used in Chapter 5. This system is illustrated in Figure 6.3 and comprises the following components:

1. The candidate image and the automatically generated road models (obtained with ARMG) are used as the system input.

2. A spectral Mahalanobis classifier is employed in the classification layer. The classifier is applied to the candidate image for every road model extracted by the ARMG process, resulting in multiple classified images.

3. Each of the spectrally classified images is thresholded into a binary image. As with the system described in Section 5.2, a connected component filter and closing operation is applied to the binary images.

4. The filtered binary images are fused into a single image by applying a union operation.

5. The SORM algorithm is used to extract the final road topology from the fused binary image.

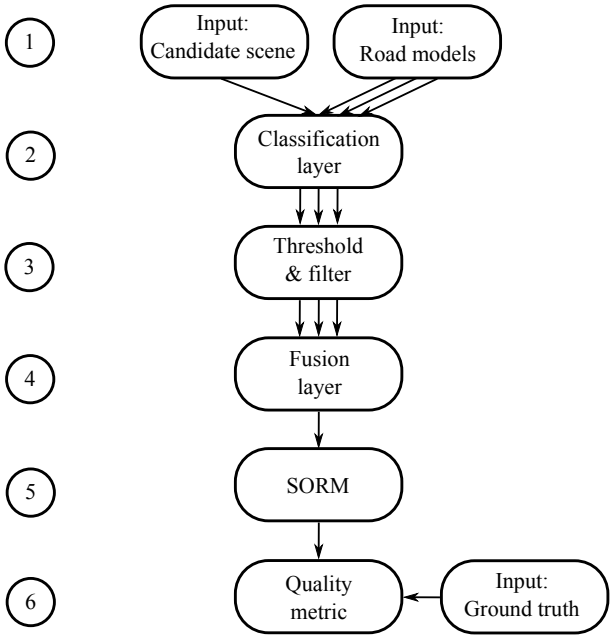


Figure 6.3: Road extraction pipeline

6. The quality of the results is determined using the quality metric by Heipke *et al.* [63], as discussed in Sections 2.6 and 3.9.

6.3 Experimental design

This following section presents various aspects regarding the system implementation and experimental design. A discussion of the data set is provided, followed by a number of considerations regarding the system implementation. The section concludes with a discussion on the parameter optimization process.

6.3.1 Data

The data set used in Chapter 5 (Section 5.3.1) was reused in this experiment, which allows for an original candidate image comparison between a system driven by manually extracted road models, and a system using automatically extracted road models.

6.3.2 Implementation

The component labelling algorithm by Chang *et al.* [24] was used to find the *lcc* from the centreline image (step 2 in Figure 6.1). Given its efficiency and the fact that it runs in linear time, the approach by Chang *et al.* is considered appealing. The algorithm was adapted slightly to count the size of each component after the labelling phase has been completed. The detected *lcc* was retained and the remaining components removed.

Three thresholds were introduced in the ARMG in an attempt to reject invalid road models. The first threshold (lcc_{min}) was used to reject road models based on the *lcc* component size (step 2 in Figure 6.1). A small connected component would result in a small road model distribution, which might bring about an inaccurate representation of the underlying road surface. When a *lcc* was extracted with a size smaller than lcc_{min}, the ARMG process was ended as the remaining components would naturally also be smaller than lcc_{min}.

The second threshold disregarded road models that caused the Mahalanobis algorithm to classify large portions of the candidate image as road pixels (step 6 in Figure 6.1). Seeing that road structures are typically thin, elongated objects, the total road surface area are relatively small in relation to the remaining non-road regions. The percentage of

pixels marked as road pixels within a scene was calculated. If this road pixel percentage (rpp) reached above a certain threshold value, the proposed road model was rejected.

The third threshold was defined within the quality control component in step 8 and considers the percentage of the lcc that was masked in step 7. This percentage will be referred to as *coverage*. If the coverage was lower than the specified threshold, the proposed road model was rejected.

6.3.3 Parameter selection

The two-staged system proposed in this chapter (Section 6.2) utilizes the systems described in Sections 4.2 and 5.2 and the selection of their parameters was described in Sections 4.3.3 and 5.3.3 respectively. In addition to these parameters, three thresholds were introduced in the ARMG component (Section 6.3.2) to regulate the automated road model generation process. Numerous combinations of threshold values were found to produce acceptable road models. Applying the optimization techniques discussed in Section 2.7 to these threshold values resulted in little improvement in the quality of the final results. Constant threshold values were consequently specified. The threshold value for $lcc_{\min}$ was set at 50 pixels, the rpp threshold at 85% and the coverage threshold at 80%.

6.4 Results and discussion

This section discusses the results obtained using the system proposed in Section 6.2 in three parts. The first part presents a stepwise examination of the ARMG process for a sample scene. The second part gives an analysis of the results produced by the PBA Mahalanobis road extraction system (Figure 6.3). The section concludes with a discussion of certain limitations of the model generator.

Figures 6.4 and 6.5 illustrate an example of the output obtained during the automated model generation process. Each image is numbered for ease of reference. The process starts by extracting centrelines from the candidate image with the Canny-based ACE system (Section 4.2). The largest component (lcc_0) was identified, dilated and masked to produce the first preliminary road model (prm_0) in image 5. The k-means filter was applied to prm_0 to create a refined road model (rrm_0). Using rrm_0, the Mahalanobis classifier was applied to the original candidate image and through thresholding a binary image was produced (image 7). The binary image was subtracted from the centreline

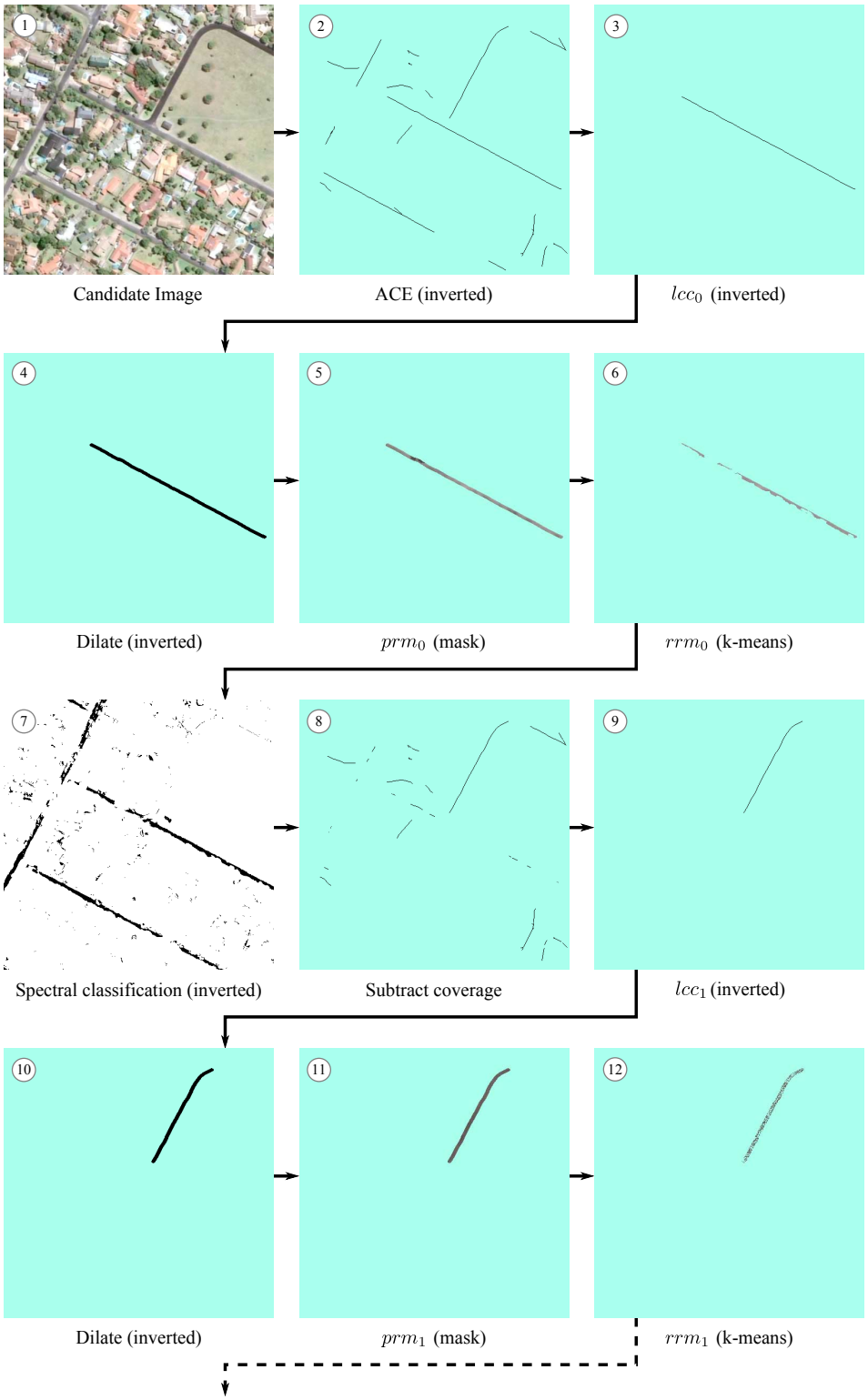


Figure 6.4: ARMG applied to a sample scene

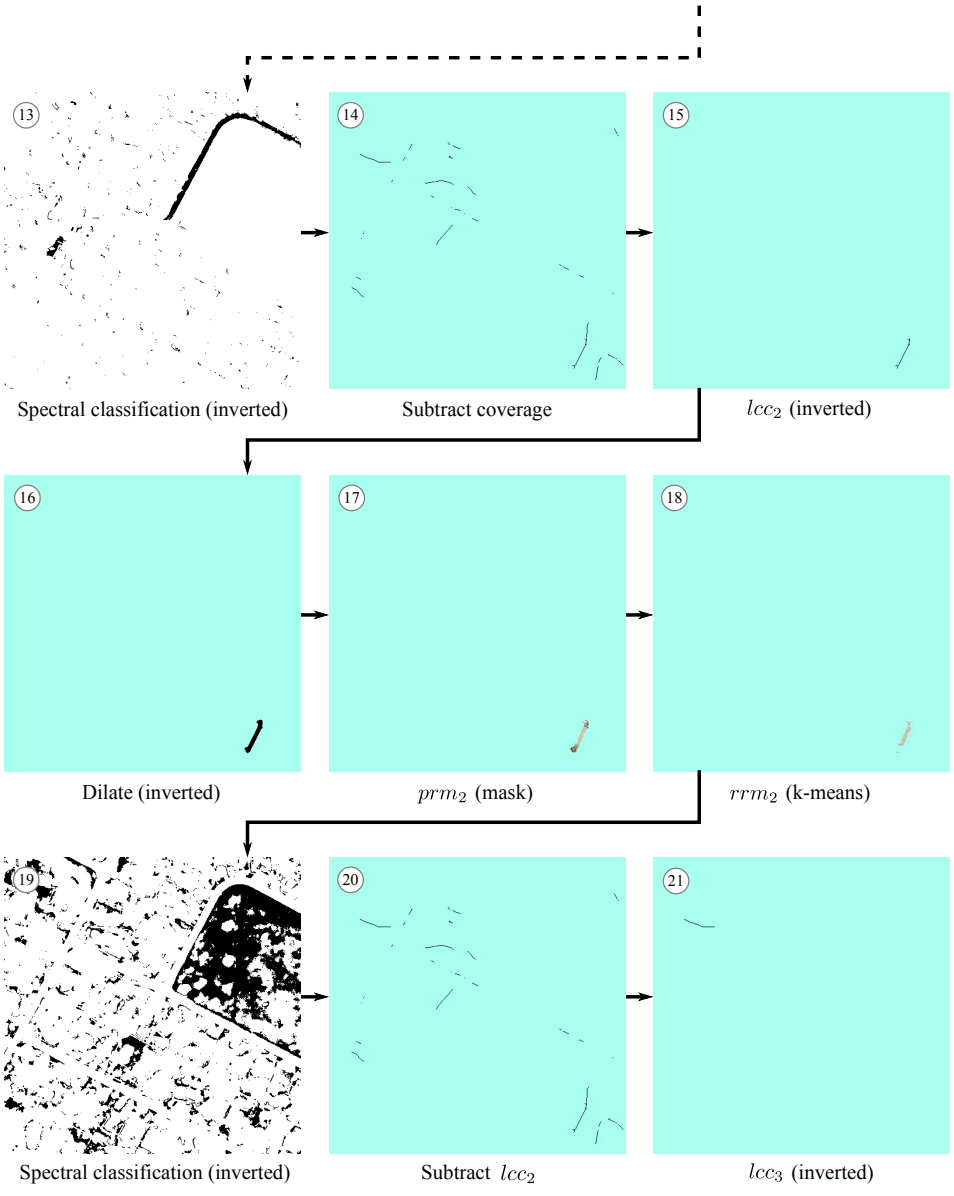
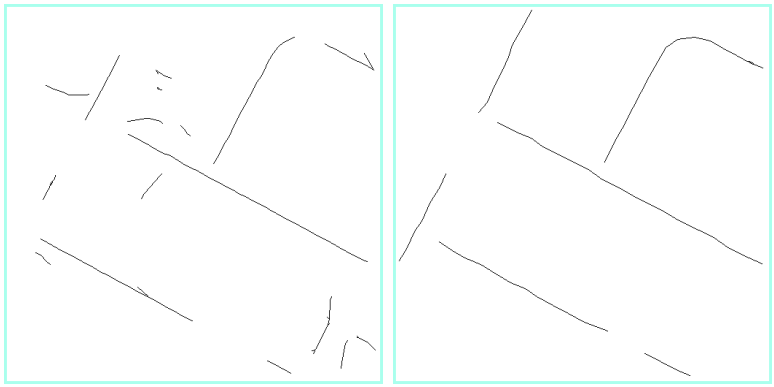


Figure 6.5: ARMG applied to a sample scene (continued)

image (image 2), which resulted in the masked centreline image (image 8). The coverage was calculated using image 8, and rrm_0 was retained as a valid model. The ARMG process was repeated (images 8–14) and rrm_1 (image 12) extracted and deemed a valid model. The model produced in the third iteration (rrm_2 in image 18) was rejected, since the coverage of lcc_2 (image 15) by the spectrally classified binary image (image 19) was below the specified 80% threshold. The ARMG process was stopped during the fourth iteration, since lcc_3 (image 21) was smaller than the specified threshold size. The final output produced by the Canny-based ACE system and the fully automated system for this sample scene is depicted in Figure 6.6. In this instance, it appears as if the output produced by the fully automated system had both a higher correctness and completeness. Please see Section A.3 in Appendix A for more sample images.

The ARMG algorithm was applied to the 50 scenes in the data set and failed to find valid models for six of the scenes. To obtain quality scores for these models, the first model produced was included in the final road model set, even though this model was deemed to be incorrect by the heuristics (refer to Section 6.2).

The road model's ability to facilitate road extraction was tested with the PBA Mahalanobis system illustrated in Figure 6.3. The cross-validation procedure followed in Section 5.4 was repeated with the same training and test partitioning sizes of 40% and 60% respectively. This allowed for a comparison between the quality measures obtained with a system driven by manually and automatically extracted road models. The final results are displayed in Table 6.1. The table displays both the results obtained from the road models extracted for all 50 scenes as well as the 44 scenes from which valid road



(a) Canny-based ACE system (b) Fully automated system

Figure 6.6: Road topologies extracted from a sample scene with the Canny-based ACE system and the fully automated system

models were extracted. Please see Section 5.4.1 for more details on the calculation of the mean and total standard deviation values.

The final test mean quality measure obtained was 48.5% with a total standard deviation of 23.9%. When considering scenes with valid road models only, the quality was higher at 53.4% and the total standard deviation marginally lower at 20.6%. Referencing the results obtained with the manually extracted road models in Table 5.2, the PBA Mahalanobis system extracted roads with a quality measure of 62.9% and a total standard deviation of 16.4%. The fully automated system was therefore unable to improve on the quality of the results produced by a system driven by manually extracted models. The total standard deviation also indicates that the fully automated system was less consistent.

The impact of the k-means filter was tested by repeating the experiment with the filter omitted. Using the preliminary road models only, valid models were found for 47 scenes, i.e. three more than the approach using refined models. The results obtained from the PBA Mahalanobis system are presented in Table 6.2. A test quality of 54.3% was obtained over all 50 scenes. Considering the valid road models only, a test quality of 56.3% was obtained. These results reveal that the k-means filter did not have the desired effect and caused a reduction in the quality of the results.

As in Chapters 4 and 5, the results are evaluated statistically with the Mann–Whitney U test, applied at a 5% confidence level to the collective results over all ten cross-validation rounds. This statistical significance is used to confirm or reject the null hypothesis, which states that there is no considerable difference between the results produced by the various aforementioned algorithms. As multiple pairwise tests were performed, the Bonferroni correction [17] was used to adjust the *type I error* (α). Table 6.3 depicts a matrix of p-values obtained from pairwise significance tests conducted on the final quality values. An asterisk is used to indicate cases where a statistical significant difference was observed. Even though the mean quality values

Table 6.1: Results for the PBA Mahalanobis system, based on the automatically derived road models (rrm). Compare with Table 5.2

	Mean (Total Standard Deviation)		
Type	**Correctness**	**Completeness**	**Quality**
All scenes	0.713 (0.301)	0.584 (0.267)	0.485 (0.239)
Scenes with valid models	0.776 (0.233)	0.638 (0.227)	0.534 (0.206)

Table 6.2: Results for the PBA Mahalanobis system, based on the preliminary automatically derived road models (*prm*)

	Mean (Total Standard Deviation)		
Type	**Correctness**	**Completeness**	**Quality**
All scenes	0.711 (0.258)	0.700 (0.189)	0.543 (0.209)
Scenes with valid models	0.736 (0.232)	0.717 (0.156)	0.563 (0.192)

differ noticeably, the significance tests show that only the results produced by the *rrm* (using all scenes) and the *prm* (using the valid road models only) approaches differ. This behaviour can be attributed to the wide spread of the quality value distributions, as suggested by the total standard deviation values.

The apparent lower scores obtained by the k-means filtered approach will be explained with reference to the example illustrated in Figure 6.4. Table 6.4 displays the variance within each RGB band for the road models prm_0, rrm_0, prm_1 and rrm_1. The values suggest that the k-means filter narrowed the distribution of prm_0 considerably in rrm_0. The difference in variance between prm_1 and rrm_1 was less noticeable.

Figure 6.7 represents the spectral histograms of prm_0, rrm_0, prm_1 and rrm_1. When comparing Figures 6.7a and 6.7b, it is clear that a portion of the spectral information towards the lower intensity range in prm_0 was removed. In this instance, the rrm_0 could be too specific which ultimately results in the reduction of the classifier's generalization ability. A comparison of Figures 6.7c and 6.7d reveals no considerable change in the spectral range between prm_1 and rrm_1, which agrees with the model's corresponding variance values. In this instance, the filtering process merely reduced the model's distribution size, which is not necessarily a desirable effect.

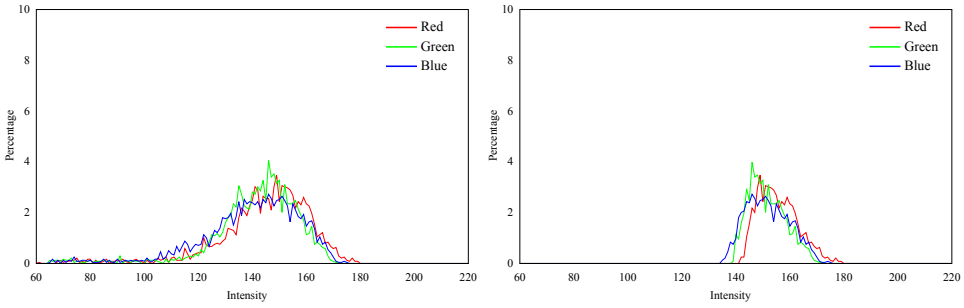
The results obtained by spectrally classifying the original candidate image using the road models prm_0, rrm_0, prm_1 and rrm_1 are depicted in Figure 6.8. When compar-

Table 6.3: P-values obtained for multiple pairwise statistical significance tests at a Bonferroni corrected 5% confidence level using the Mann–Whitney U test.

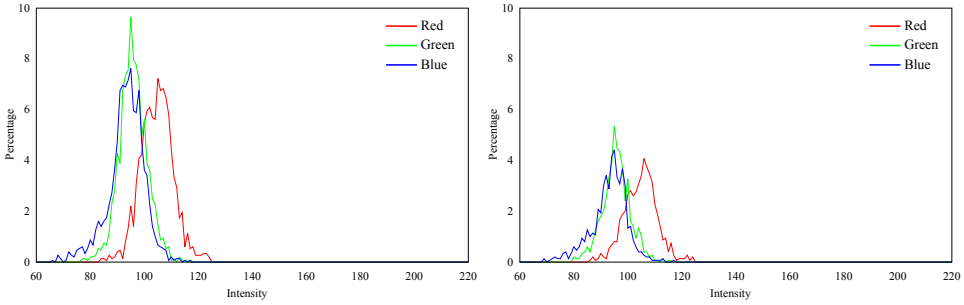
	rrm (all)	rrm (valid)	prm (all)	prm (valid)
rrm (all)	1.000	0.029	0.029	0.002*
rrm (valid)	0.029	1.000	0.942	0.400
prm (all)	0.029	0.942	1.000	0.323
prm (valid)	0.002*	0.400	0.323	1.000

Table 6.4: Variance of the spectral values for the automatically extracted road models (Figure 6.4)

	Variance		
Model	**Red**	**Green**	**Blue**
prm_0	303.16	272.11	329.46
rrm_0	53.49	45.81	64.38
prm_1	36.49	27.61	45.07
rrm_1	36.63	28.26	45.06



(a) prm_0 (b) rrm_0



(c) prm_1 (d) rrm_1

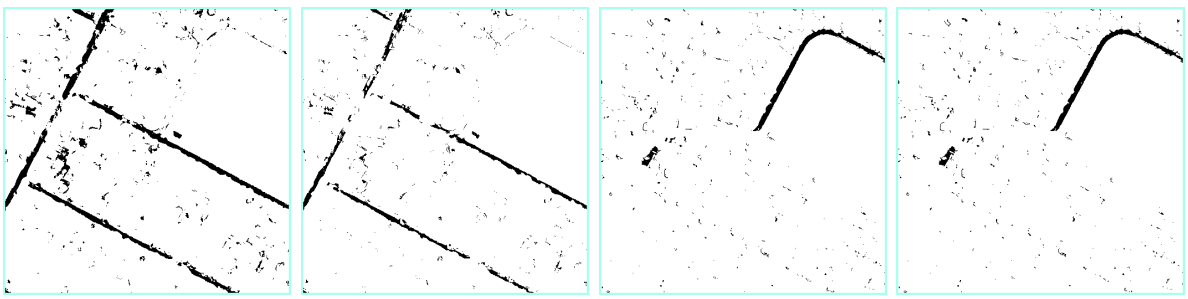
Figure 6.7: Histograms of the road models automatically extracted from a sample scene (Figure 6.4)

ing Figures 6.8a and 6.8b visually, it appears as if the constricted rrm_0 distribution translated into a constricted classification. More breaks are evident in the roads in Figure 6.8b which result in a decrease in completeness and quality values. A visual comparison of Figures 6.8c and 6.8d reveals little change, since the difference in spectral variance between prm_1 and rrm_1 is less prominent. As a result, it appears as if the lower scores obtained by the k-means filtered approach can in part be attributed to the disproportionate narrowing of the spectral distribution, which results in fewer road pixels being extracted. The refinement also resulted in fewer valid road models being extracted, which could impact the final scores.

6.5 Conclusion

This chapter presented a fully automated road extraction system, requiring only a candidate scene as input. The system incorporates the road extraction approaches described in Chapters 4 and 5. These approaches were mainly based on the work by Doucette *et al.* [38], and Mena and Malpica [115]. The key feature within the proposed fully automated system was an automated road model generator, which was inspired by the SSRC feedback loop that Doucette *et al.* suggested.

Two automated system configurations were tested. The first incorporated a k-means filtering component, which aimed to produce refined road models, while the second omitted the filtering component. The first configuration yielded a final extraction quality of 48.5%, while the second unfiltered approach managed 54.3%. The poorer quality produced by the k-means filtered approach is likely due to the amount of information discarded during filtration. The total standard deviation of the quality values ranged



(a) prm_0 (b) rrm_0 (c) prm_1 (d) rrm_1

Figure 6.8: Spectral classification using the road models automatically extracted from a sample scene (Figure 6.4)

between 19.2–23.9%, which suggests that the system produced irregular results.

The same system, driven by manually extracted road models, obtained a quality of 62.9% with total standard deviation values ranging from 13.0–18.8% (refer to Section 5.4.2). The lower results obtained by the automated system can in part be ascribed to the ACE algorithm's inability to find suitable centreline seeds for a number of scenes, which results in the selection of poor road models. A more advanced filtering mechanism is needed to remove non-road objects from the road model without reducing the spread of the road's spectral distribution.

CHAPTER 7

Conclusion and Future Work

"Science never solves a problem without creating ten more."

— George Bernard Shaw

7.1 Summary

The aim of this study was threefold. The first objective was to develop a flexible semi-automated road extraction system. Secondly, the study sought to integrate a variety of algorithms within the road extraction system, with the goal of developing a clearer understanding of the benefits of using these methods in such a system. Finally, the study aimed to create a fully automated system with the only external input being a sample image.

The first objective of developing a generic system capable of extracting roads was accomplished and the system was employed in three experiments in Chapters 4–6. The

system provided the necessary flexibility to test a diverse number of system configurations. These configurations were discussed in Sections 4.2, 5.2 and 6.2.

The second goal was achieved by integrating a variety of algorithms within the road extraction system. These algorithms are listed here and are categorized according to the functionality they perform.

Edge detectors: The Canny [23] and V1 [110] edge detectors were tested within a geometrical road detector. Please see Chapter 4 for more information.

Geometrical detector: The ACE algorithm [38] was used to detect road centreline seeds by considering the parallel edges of road structures and the orientation of the underlying intensity gradients of these parallel edges (Chapters 4 and 6).

Spectral classifiers: The Mahalanobis, Bhattacharyya, and texture cube spectral classifiers proposed by Mena and Malpica [115] were used to identify road objects by classifying images spectrally (Chapters 5 and 6).

Segmentation: The colour segmentation algorithm by Felzenszwalb and Huttenlocher [41] was used to detect objects in remotely sensed images (Chapter 5). The k-means clustering algorithm [103] was used in an attempt to remove non-road objects from road models (Chapter 6).

Filters: Various mathematical morphology filters [106, 138] were employed. The key functions fulfilled by these filters included the detection and removal of connected components, as well as the filling of gaps and linking of objects with the closing operation (Chapters 4–6).

Data fusion: Dempster-Shafer fusion [35, 141] was used to combine multiple classifier outputs into a single result (Chapter 5).

Road network construction: The SORM algorithm by Doucette *et al.* [38, 39] was used to heuristically construct a road network topology from ACE centreline seeds or binarized classifier output images (Chapters 4–6).

The final objective of developing a fully automated road extraction system was achieved in Chapter 6, where road models were automatically generated and used to drive a road extraction system that employed a Mahalanobis spectral classifier.

In addition to the key objectives discussed in this section, an extensive literature review of the current state-of-the-art was presented in Chapter 2. This review serves as

an update to the 2003 review by Mena [114]. The thesis findings are summarized in the following section.

7.2 Conclusion

A number of experiments and results were discussed in Chapters 4–6 and the outcomes are reviewed in this section.

Chapter 4 considered improving RNE results by replacing the Canny algorithm with the V1 edge detector in the ACE geometrical detector. The V1 algorithm is designed to detect long range edges, making it ideally suited to detect the edges of road structures. It was found that the system employing the Canny edge detection algorithm produced more accurate results, but the two approaches were considered comparable. The edges extracted by the V1 edge detector were not ideally suited to the ACE algorithm and more accurate results might be obtained if a different geometrical detector was used.

An RNE system driven by spectral classifiers was presented in Chapter 5. The objective of the experiment was to test whether an increase in road extraction accuracy could be obtained by classifying segments (OBA) rather than individual pixels (PBA). It was found that the OBA and PBA produced comparable results. A different segmentation algorithm might provide an increase in accuracy, but it seems unlikely that such an increase would be significant. The RNE system presented in Chapter 5 combined the outputs from three spectral classifiers by employing Dempster-Shafer fusion. An additional experiment investigated whether the fusion of classifiers contributed to an increase in the accuracy of the overall system. It was found that the fusion process improved the results marginally.

In Chapter 6, a fully automated system was proposed comprising two components. The first component extracted road models automatically and the second constructed the final road network topology. The second component employed the PBA Mahalanobis road extraction system described in Chapter 5, but used the automatically generated road models rather than manually extracted models. The fully automated system produced lower quality scores at a lower consistency than the system driven by manually extracted road models. The interaction between the road models and the final extracted results are complex and various factors contributed to the lower results obtained by the fully automated system.

7.3 Future work

The following sub-sections provide a set of possible extensions and further investigations based on the range of topics and methods covered in this thesis.

7.3.1 Iterative execution

The fully automated system presented in Chapter 6 comprised two key components of which the first automatically detected spectral road models, and the second extracted a road topology based on the detected road models. The first component is executed iteratively to detect all road types within a candidate scene, but the entire end-to-end system is only executed once per scene. This system was derived from the SSRC feedback loop proposed by Doucette *et al.* [38]. The SSRC algorithm repeats the entire road extraction pipeline iteratively and obtains significant increases in the quality of the results over the first two iterations. By executing the system presented in Chapter 6 iteratively, the quality of the results could potentially be improved.

7.3.2 Diverse classifiers

The experiment conducted in Section 5.4.2 showed that the results produced by a road extraction system can be improved by fusing the outputs from multiple classifiers as opposed to using a single classifier. Considering that the classifiers used in Chapter 5 were all based on spectral features, the improvement was marginal and greater improvements might be obtained when combining more diverse methods. Additional methods such as the texture-based ATS algorithm by Dial *et al.* [36], the geometrical curvilinear detector by Steger [147], or the template matching SVM classifier by Mei *et al.* [112] could be considered.

7.3.3 Masking non-road regions

In image processing, masking is the process whereby unwanted features are removed from an image and not considered during further processing. This technique can also be applied to RNE problems by masking non-road regions. Masking non-road regions would help to reduce the number of spurious road detections, and result in an increase in the processing speed. By masking non-road regions, the constraints used in this thesis (Sections 4.3.3, 5.3.3 and 6.3.3) could be relaxed, which might translate into a higher

road network completeness and quality. As roads are typically not characterized by their vegetative or water properties, masks can be derived from indices such as the Normalized Difference Vegetation Index (NDVI) or the Normalized Difference Water Index (NDWI).

7.3.4 Scalability

The data sets used in this thesis were composed of images with a pixel dimensionality of 512×512. If the methods presented in this thesis were to be used in real-world applications, their scalability would first have to be tested on images with larger pixel dimensionalities. As more variety is introduced when considering larger regions, numerous new problems could be introduced. An example of such a problem would be to find applicable road models within a scene. A single road model is unlikely to be valid for widespread regions and a mechanism would be required to associate a road model to an applicable region. Alternatively, the methods used in this thesis can be applied to overlapping tiles, followed by an algorithm to merge the tiles into a single output.

7.3.5 SORM enhancements

The SORM algorithm employed in this study determined the validity of centreline vectors by considering a vector in relation to its immediate connected neighbours. A more robust approach would be to consider the vector network holistically. Such an approach is proposed by Géraud and Mouret [47] where Markov random fields are defined over a raster-based connected graph. An energy model of a road network is defined and the network construction is treated as a global energy minimization problem. In addition, SORM could be enhanced further by accepting input data with continuous values.

As illustrated in Chapter 5, SORM extracted road topologies from spectrally classified images. These classified images were binarized, which resulted in an explicit distinction being made between road and non-road regions. SORM could be extended to accept the original classified images, which contained values of the likelihood of a pixel being part of a road object. As more information is available, SORM might be able to make more informed decisions which could translate into improved results. The cumbersome process of finding a globally optimal binarization threshold will also be avoided.

7.3.6 Semi-automated extraction comparison

An investigation of the possible advantages of using a human operator during certain stages of processing should provide valuable insights into areas where further research is required. An example would be to use an operator to determine the initial road seeds, as opposed to using an automatic seeding algorithm. Such a comparison could clarify whether the automatic seeding is robust enough to extract roads accurately, in spite of poor initialization points; or whether the initial seed points are causing the rest of the system to falter.

Bibliography

[1] P. Agouris, P. Doucette, and A. Stefanidis. Spatiospectral cluster analysis of elongated image regions. In *Proceedings of the International Conference on Image Processing*, 2001.

[2] P. Agouris, S. Gyftakis, and A. Stefanidis. Dynamic node distribution in adaptive snakes for road extraction. In *Proceedings of Vision Interface*, 2001.

[3] V. Amberg, M. Spigai, M. Coulon, and P. Marthon. Structure extraction from high resolution SAR data on urban areas. In *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, 2004.

[4] D. V. Andrade and L. H. Figueiredo. Good approximations for the relative neighbourhood graph. In *Proceedings of 13th Canadian Conference on Computational Geometry*, 2001.

[5] G. Arfken. *Hermite Functions*. Academic Press, 1985.

[6] B. Ayalew, R. Gomez, and W. R. O. Carrasco. Pavement management using hyperspectral imagery. In *Proceedings of SPIE — The International Society for Optical Engineering*, 2003.

[7] U. Bacher and H. Mayer. Automatic road extraction from IRS satellite images in agricultural and desert areas. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35:1055–1060, 2004.

[8] U. Bacher and H. Mayer. Automatic road extraction from multispectral high resolution satellite images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36:29–34, 2005.

[9] E. Baltsavias, L. O'Sullivan, and C. Zhang. Automated road extraction and updating using the Atomi system — performance comparison between aerial film, ADS40, IKONOS and QuickBird orthoimagery. In *Proceedings of the 20th ISPRS Congress*, 2004.

[10] A. Baumgartner, C. Steger, H. Mayer, and W. Eckstein. Multi-resolution, semantic objects, and context for road extraction. In *Proceedings of Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*, 1997.

[11] A. Baumgartner, C. Steger, C. Wiedemann, H. Mayer, W. Eckstein, and H. Ebner. Update of roads in GIS from aerial imagery: Verification and multi-resolution extraction. *International Archives of Photogrammetry and Remote Sensing*, 31:53–58, 1996.

[12] L. Bentabet, S. Jodouin, D. Ziou, and J. Vaillancourt. Road vectors update using SAR imagery: A snake-based method. *IEEE Transactions on Geoscience and Remote Sensing*, 41:1785–1803, 2003.

[13] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1945.

[14] A. Bhattacharyya, I. H. Jermyn, X. Descombes, and J. Zerubia. Computing statistics from a graph representation of road networks in satellite images for indexing and retrieval. In *Proceedings of CompIMAGE — Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*, 2006.

[15] U. Bhattacharyya and S. K. Parui. An improved backpropagation neural network for detection of road-like features in satellite imagery. *International Journal of Remote Sensing*, 18:3379–3394, 1997.

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

[17] C. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.

[18] A. Bovik. *Handbook of image and video processing*. Elsevier Academic Press, 2005.

[19] E. Breen and R. Jones. Attribute openings, thinnings and granulometries. *Computer Vision and Image Understanding*, 64:377–389, 1996.

[20] D. E. Brown and J. Marin. Learning vector quantization for road extraction from digital imagery. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1995.

[21] X. Cai, A. Sowmya, and J. Trinder. Learning to recognise roads from high resolution remotely sensed images. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2005.

[22] F. Campos and S. Cavalcante. An extended approach for Dempster-Shafer theory. In *Proceedings of the IEEE International Conference on Information Reuse and Integration*, 2003.

[23] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–714, 1986.

[24] F. Chang, C. Chen, and C. Jen. A linear-time component labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93:206–220, 2004.

[25] J. Chanussot, G. Mauris, and P. Lambert. Fuzzy fusion techniques for linear features detection in multitemporal SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 37:1292–1305, 1999.

[26] C. Chen, C. Shahabi, and C. A. Knoblock. Utilizing road network data for automatic identification of road intersections from high resolution color orthoimagery. In *Proceedings of the Second Workshop on Spatio-Temporal Database Management*, 2004.

[27] T. Chen, J. Wang, and K. Zhang. A wavelet transform based method for road centerline extraction. *Photogrammetric Engineering & Remote Sensing*, 70:1423–1431, 2004.

[28] E. Christophe and J. Inglada. Robust road extraction for high resolution satellite images. In *IEEE International Conference on Image Processing*, 2007.

[29] D. L. Civco and Y. Waung. Classification of multi-spectral, multitemporal multi-source spatial data using artificial neural networks. In *Proceeding of the ASPRS 1994 Annual Convection*, 1994.

[30] C. Cortes and V. N. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[31] I. Couloigner and T. Ranchin. Mapping of urban areas: A multiresolution modeling approach for semi-automatic extraction of streets. *Photogrammetric Engineering & Remote Sensing*, 66:867–874, 2000.

[32] A. P. Dal-Poz, G. M. D. Vale, and R. B. Zanin. Automatic extraction of road seeds from high-resolution aerial images. *Annals of the Brazilian Academy of Sciences*, 77:509–520, 2005.

[33] F. Dell'Acqua, P. Gamba, and G. Lisini. Extraction and fusion of street networks from fine resolution SAR data. In *Proceedings of the International Geoscience and Remote Sensing Symposium*, 2002.

[34] F. Dell'Acqua, P. Gamba, and G. Lisini. Road extraction aided by adaptive directional filtering and template matching. In *Proceedings of the ISPRS joint conference URBAN2005*, 2005.

[35] A. P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, Series B(30):205–247, 1968.

[36] G. Dial, L. Gibson, and R. Poulsen. IKONOS satellite imagery and its use in automated road extraction. In *Proccedings of the Workshop on Automated Extraction of Man-Made Objects from Aerial and Space Images*. A.A. Balkema Publishers, Lisse, The Netherlands, 2001.

[37] P. Doucette. *Automated Road Extraction from Aerial Imagery by Self-Organization*. PhD thesis, Department of Spatial Information Science and Engineering, University of Maine, 2002.

[38] P. Doucette, P. Agouris, and A. Stefanidis. Automated road extraction from high resolution multispectral imagery. *Photogrammetric Engineering & Remote Sensing*, 70(12):1405–1416, Dec. 2004.

[39] P. Doucette, P. Agouris, A. Stefanidis, and M. Musavi. Self-organized clustering for road extraction in classified imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(5–6):347–358, 2001.

[40] R. Duba and P. Hart. *Pattern classification and scene analysis*. John Wiley & Sons, 1973.

[41] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[42] R. Fiset, F. Cavayas, M. Mouchot, B. Solaiman, and R. Desjardins. Map-image matching using a multi-layer perceptron: the case of the road network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53:76–84, 1998.

[43] P. Fua and Y. G. Leclerc. Model driven edge detection. *Machine Vision and Applications*, 3:45–56, 1990.

[44] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[45] P. Gamba, F. Dell'Acqua, G. Lisini, and G. Trianni. Improved VHR urban area mapping exploiting object boundaries. *IEEE Transactions on Geoscience and Remote Sensing*, 45:2676–2682, 2007.

[46] D. Geman and B. Jedynak. Detection of roads in satellite images. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, 1991.

[47] T. Géraud and J. Mouret. Fast road network extraction in satellite images using mathematical morphology and Markov random fields. *EURASIP Journal on Applied Signal Processing*, 16:2503–2514, 2004.

[48] M. Gerke. Automatic quality assessment of GIS road data using aerial imagery - a comparison between bayesian and evidential reasoning. *International Archives of Photogrammetry and Remote Sensing*, 36:171–177, 2005.

[49] L. Gibson. Finding road networks in IKONOS satellite imagery. In *Proceedings of ASPRS 2003 Conference*, 2003.

[50] G. A. Giraldi, E. Strauss, and A. A. Oliveira. Boundary extraction method based on Dual-T-Snakes and dynamic programming. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2000.

[51] P. Gong and J. Wang. Road network extraction from airborne digital camera images: A multi- resolution comparison. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, 1997.

[52] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (Third Edition)*. Prentice Hall, 2008.

[53] A. Grote and C. Heipke. Road extraction for the update of road databases in suburban area. In *Proceedings of the 21st ISPRS Congress*, 2008.

[54] A. Gruen and H. Li. Road extraction from aerial and satellite images by dynamic programming. *Journal of Photogrammetry & Remote Sensing*, 50:11–20, 1995.

[55] D. Guo, A. Weeks, and H. Klee. Segmentations of road area in high resolution images. In *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, 2004.

[56] D. Hall and G. Ball. Isodata: A novel method of data analysis and pattern classification,. Technical report, Stanford Research Institute, Menlo Park, CA, 1965.

[57] R. M. Haralick. Statistical and structural approaches to texture. In *Proceedings of the IEEE*, 1979.

[58] J. B. Harley and D. Woodward. *The History of Cartography*. Humana Press, 1987.

[59] J. Harris and H. Stocker. *Maximum Likelihood Method*. New York: Springer-Verlag, 1998.

[60] W. Harvey. Performance evaluation for road extraction. In *Proceedings of the ISPRS Workshop on 3D Geospatial Data Production: Meeting Application Requirements*, 1999.

[61] D. Haverkamp. Extracting straight road structure in urban environments using IKONOS satellite imagery. *Optical Engineering*, 41:2107–2110, 2002.

[62] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer. Comparison of edge detectors. a methodology and initial study. *Computer Vision and Image Understanding*, 69:38–54, 1998.

[63] C. Heipke, H. Mayer, and C. Wiedemann. Evaluation of automatic road extraction. In *Proceedings of the ISPRS Conference*, 1997.

[64] J. Heller and K. Pakzad. Scale dependent adaptation of object models for road extraction. In *Proceedings of the ISPRS Workshop CMRT*, 2005.

[65] O. Hellwich, I. Laptev, and H. Mayer. Extraction of linear objects from interferometric SAR data. *International Journal of Remote Sensing*, 23:461–475, 2002.

[66] A. Hendry, S. Quegan, and J. Wood. The visibility of linear features in SAR images. In *Proceedings of the International Geoscience and Remote Sensing Symposium*, 1988.

[67] S. Hinz and A. Baumgartner. Automatic extraction of urban road networks from multi-view aerial imagery. *Journal of Photogrammetry & Remote Sensing*, 58:83–98, 2003.

[68] S. Hinz, A. Baumgartner, H. Mayer, C. Wiedemann, and E. Heinrich. Road extraction focussing on urban areas. In *Remote Sensing and Data Fusion over Urban Areas, IEEE and ISPRS Joint Workshop*, 2001.

[69] J. Hu, A. Razdan, J. C. Femiani, M. Cui, , and P. Wonka. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Transactions On Geocence And Remote Sensing*, 45:4144–4157, 2007.

[70] X. Hu and C. V. Tao. A reliable and fast ribbon road detector using profile analysis and modelbased verification. *International Journal of Remote Sensing*, 26:887–902, 2005.

[71] X. Huang and L. Zhang. Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. *International Journal of Remote Sensing*, 30:1977–1987, 2009.

[72] G. Hui, L. Jilin, and L. Yaya. Road extracting based on texture analysis. In *Proceedings of the 16th International Conference on Artificial Reality and Telexistence–Workshops*, 2006.

[73] A. K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[74] A. K. Jain, R. P. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.

[75] B. Jeon, J. Jang, and K. Hong. Map-based road detection in spaceborne synthetic aperture radar images based on curvilinear structure extraction. *Optical Engineering*, 39:2413–2421, 2000.

[76] X. Jin and C. H. Davis. An integrated system for automatic road mapping from high-resolution multi-spectral satellite imagery by information fusion. *Information Fusion*, 6:257–273, 2005.

[77] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

[78] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[79] A. Katartzis, H. Sahli, V. Pizurica, and J. Cornelis. A model-based approach to the automatic extraction of linear features from airborne images. *IEEE Transactions on Geoscience and Remote Sensing*, 39:2073–2079, 2001.

[80] T. Keaton and J. Brokish. A level set method for the extraction of roads from multispectral imagery. In *Proceedings of the Applied Imagery Pattern Recognition Workshop*, 2002.

[81] R. Ketting and D. Landgrebe. Classification of multispectral image data by extraction and classification of homogeneous objects. *IEEE Transactions on Geoscience Electronics*, 14:19–26, 1976.

[82] J. Kiefer. Sequential minimax search for a maximum. In *Proceedings of the American Mathematical Society*, 1953.

[83] T. Kim, S. Park, M. Kim, S. Jeong, and K. Kim. Tracking road centrelines from high resolution remote sensing images by least squares correlation matching. *Photogrammetric Engineering & Remote Sensing*, 70(12):1417–1422, Dec. 2004.

[84] R. Kindermann. *Markov Random Fields and Their Applications*. 1980.

[85] L. Kitchen and A. Rosenfeld. Edge evaluation using local edge coherence. *EEE Transactions on Systems, Man, and Cybernetics*, 11:597–605, 1981.

[86] L. A. Klein. *Sensor and Data Fusion Concepts and Applications*. The International Society for Optical Engineering press, 1993.

[87] L. A. Klein. *Sensor and Data Fusion: A Tool for Information Assesment and Decision Making*. The International Society for Optical Engineering press, 2004.

[88] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[89] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 2001.

[90] G. Koutaki and K. Uchimura. Automatic road extraction based on cross detection in suburb. In *Proceedings of The International Society for Optical Engineering*, 2004.

[91] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, 1956.

[92] L. I. Kuncheva. *Fuzzy Classifier Design*. Springer-Verlag, 2000.

[93] S. Lanser and W. Eckstein. A modification of Deriche's approach to edge detection. In *11th International Conference on Pattern Recognition*, 1992.

[94] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner. Automatic extraction of roads from aerial images based on scale space and snakes. *Machine Vision and Applications*, 12:23–31, 2000.

[95] K. Laws. *Texture Image Segmentation*. PhD thesis, Department of Engineering, University of Southern California, 1980.

[96] H. Y. Lee, H. Lee, T. Kim, and W. Park. Towards knowledge-based extraction of roads from 1 m-resolution satellite images. In *Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, 2000.

[97] K. Lee and H. Ryu. Automatic circuity and accessibility extraction by road graph network and its application with high-resolution satellite imagery. In *Proceedings of the International Geoscience and Remote Sensing Symposium*, 2004.

[98] C. Li, D. Xia, Y. Wang, and H. Zou. Satellite sensing image analysis by integrating neural network with knowledge reasoning technique. In *Proceedings of SPIE Image Processing and Pattern Recognition in Remote Sensing*, 2003.

[99] X. Lin, J. Zhang, Z. Liu, and J. Shen. Integration method of profile matching and template matching for road extraction from high resolution remotely sensed imagery. In *Proceedings of the International Workshop on Earth Observation and Remote Sensing Applications*, 2008.

[100] G. Lisini, C. Tison, D. Cherifi, F. Tupin, and P. Gamba. Improving road network extraction in high resolution SAR images by data fusion. In *Proceedings of the Committee on Earth Observation Satellites Working Group on Calibration and Validation: Synthetic Aperture Radar Workshop*, 2004.

[101] H. Liu, J. Li, and M. A. Chapman. Automated road extraction from satellite imagery using hybrid genetic algorithms and cluster analysis. *Journal of Environmental Informatics*, 1:40–47, 2003.

[102] Y. Luo, Y. Xue, and S. Zhong. Road extraction from IKONOS image using grid computing platform. In *Proceedings of the Geoscience and Remote Sensing Symposium*, 2005.

[103] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *The Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[104] J. Mao and A. K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25:173–188, 1992.

[105] D. Marr and E. Hildreth. Theory of edge detection. In *The Proceedings of the Royal Society of London*, 1980.

[106] G. Matheron. *Random Sets and Integral Geometry*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York etc., 1975.

[107] H. Mayer, I. Laptev, A. Baumgartner, and C. Steger. Automatic road extraction based on multi-scale modeling, context, and snakes. *International Archives of Photogrammetry and Remote Sensing*, 32:106–113, 1997.

[108] D. J. McKeown and J. Denlinger. Cooperative methods for road tracking in aerial imagery. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, 1988.

[109] J. L. McKinstry and C. C. Guest. Self-organizing map develops V1 organization given biologically realistic input. In *Proceedings of the IEEE International Conference on Neural Networks*, 1997.

[110] J. L. McKinstry and C. C. Guest. Long range connections in primary visual cortex: A large scale model applied to edge detection in gray-scale images. In *Proceedings of IJCNN*, Washington, DC., 2001.

[111] T. Mei, D. Li, and Q. Qin. Application of knowledge based watershed transform approach to road detection. In *Proceedings of the SPIE International Symposium on Multispectrum Image Processing and Pattern Recognition*, 2005.

[112] T. Mei, F. Li, Q. Qin, and D. Li. Road extraction from remote sensing image using support vector machine. In *Proceedings of SPIE — Third International Symposium on Multispectral Image Processing and Pattern Recognition*, 2003.

[113] J. Mena. Automatic vectorization of segmented road networks by geometrical and topological analysis of high resolution binary images. *Knowledge-Based Systems*, 19:704–718, 2006.

[114] J. B. Mena. State of the art on automatic road extraction for GIS update: A novel classification. *Pattern Recognition Letters*, 24:3037–3058, 2003.

[115] J. B. Mena and J. A. Malpica. An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. *Pattern Recognition Letters*, 26:1201–1220, 2005.

[116] A. Mohammadzadeh, A. Tavakoli, and M. V. Zoej. Automatic linear feature extraction of Iranian roads from high resolution multi-spectral satellite imagery. In *Proceedings of the 10th International Society for Photogrammetry and Remote Sensing Congress*, 2004.

[117] M. Mokhtarzade, M. Zoej, and H. Ebadi. Automatic road extraction from high resolution satellite images using neural networks, texture analysis, fuzzy clustering and genetic algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:549–556, 2008.

[118] M. Mokhtarzade and M. V. Zoej. Road detection from high-resolution satellite images using artificial neural networks. *International Journal of Applied Earth Observation and Geoinformation*, 9:32–40, 2007.

[119] V. S. Nalwa and T. O. Binford. On detecting edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:699–714, 1986.

[120] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[121] R. Nevatia and K. Babu. Linear feature extraction and description. *Computer Graphics and Image Processing*, 13:257–269, 1980.

[122] L. A. Oddo, P. Doucette, and P. Agouris. Automated road extraction via the hybridization of self-organization and model based techniques. In *Proceedings of the 29th Applied Imagery Pattern Recognition Workshop*, 2000.

[123] P. P. Ohanian and R. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25:819–833, 1992.

[124] T. J. Patterson. Hierarchical spatial spectral processing for road mapping. In *Proceedings of SPIE – Volume 4725, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VIII*, 2002.

[125] E. S. Peer. A serendipitous software framework for facilitating collaboration in computational intelligence. Master's thesis, Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, 2004.

[126] T. Peng, I. H. Jermyn, V. Prinet, J. Zerubia, and B. Hu. Urban road extraction from VHR images using a multiscale approach and a phase field model of network geometry. In *Proceeding of the Urban Remote Sensing Joint Event*, 2007.

[127] R. Péteri and T. Ranchin. Multiresolution snakes for urban road extraction from IKONOS and QuickBird images. In *Proceedings of the 23rd Symposium of the European Association of Remote Sensing Laboratories*, 2003.

[128] R. Péteri and T. Ranchin. Urban street mapping using QuickBird and IKONOS images. In *Proceedings of the International Geoscience and Remote Sensing Symposium*, 2003.

[129] C. Poullis, S. You, and U. Neumann. Linear feature extraction using perceptual grouping and graph-cuts. In *Proceedings of the 5th annual ACM International Symposium on Advances in Geographic Information Systems*, 2007.

[130] C. Poullis, S. You, and U. Neumann. A vision-based system for automatic detection and extraction of road networks. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 2008.

[131] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[132] A. Reggiani, S. Cattaneo, M. Janic, and P. Nijkamp. Combined freight transport in Europe: Policy issues and forecast scenarios with reference to the Alpine case. In *Proceedings of the 11th International Symposium of the PRSCO Secretariat*, 1999.

[133] C. J. Robinove. Earth resources satellites and integrated resource surveys. In *Proceedings of the Symposium on Remote Sensing*, pages 47–56, Pretoria, South Africa, 1973.

[134] J. Rocha and M. Queluz. Integration of census data, remote sensing and GIS techniques for land-use and cover classification. In *Proceedings of SPIE — The International Society for Optical Engineering*, 2002.

[135] M. Rochery, I. H. Jermyn, and J. Zerubia. Higher order active contours. *International Journal of Computer Vision*, 69:27–42, 2006.

[136] R. Ruskoné and S. Airault. Toward an automatic extraction of the road network by local interpretation of the scene. In *46th Photogrammetric Week, Stuttgart*, 1997.

[137] S. Sarkar and K. Boyer. Perceptual organization in computer vision: A review and a proposal for a classificatory structure. *IEEE Transactions on Systems, Man, and Cybernetics*, 23:382–399, 1993.

[138] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, England, 1982.

[139] A. K. Shackelford and C. H. Davis. Fully automated road network extraction from high-resolution satellite multispectral imagery. In *Proceedings of the International Geoscience and Remote Sensing Symposium*, 2003.

[140] A. K. Shackelford and C. H. Davis. Urban road network extraction from high-resolution multispectral ddata. In *Proceedings of the 2nd GRSS and ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, 2003.

[141] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[142] L. G. Shapiro and G. C. Stockman. *Computer Vision*. New Jersey, Prentice-Hall, 2001.

[143] J. Shen, X. Lin, Y. Shi, and C. Wong. Knowledge-based road extraction from high resolution remotely sensed imagery. In *Proceedings of the Congress on Image and Signal Processing*, 2008.

[144] J. Sherman, J. Spencer, J. Preisser, W. Gesler, and T. Arcury. A suite of methods for representing activity space in a healthcare accessibility study. *International Journal of Health Geographics*, 4:24, 2005.

[145] M. Song and D. Civco. Road network extraction using SVM and image segmentation. *Photogrammetric Engineering & Remote Sensing*, 70:1365–1371, 2004.

[146] C. Steger. Extracting curvilinear structures: A differential geometric approach. In *Proceedings of the Fourth European Conference on Computer Vision*, 1996.

[147] C. Steger. An unbiased detector of curvilinear structures. Technical report, Forschungsgruppe Bildverstehen (FG BV), Informatik IX, 1996.

[148] C. Steger, C. Glock, W. Eckstein, H. Mayer, and B. Radig. Model-based road extraction from images. In *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, 1995.

[149] H. Tesser and T. Pavlidis. Roadfinder front end: An automated road extraction system. In *Proceedings of the International Conference on Pattern Recognition*, 2000.

[150] K. Treash and K. Amaratunga. Automatic road detection in grayscale aerial images. *Journal of Computing in Civil Engineering*, 14:60–69, 2000.

[151] O. Tuncer. Fully automatic road network extraction from satellite images. In *Proceedings of the 3rd International Conference on Recent Advances in Space Technologies*, 2007.

[152] F. Tupin, B. Houshmand, and M. Datcu. Road detection in dense urban areas using SAR imagery and the usefulness of multiple views. *IEEE Transactions On Geoscience and Remote Sensing*, 40:2405–2414, 2002.

[153] F. Tushabe and M. H. F. Wilkinson. Image preprocessing for compression: Attribute filtering. In *Proceedings of the World Congress on Engineering and Computer Science*, 2007.

[154] R. Urquhart. Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, 15:173–187, 1982.

[155] J. van der Merwe. Typical plans for road design. Technical report, Gauteng Department of Transport and Public Works, 1998.

[156] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[157] L. Vincent. Morphological area openings and closings for grey-scale images. In *Proceedings of the Workshop Shape in Picture: Mathematical Description of Shape in Gray-Level Images*, 1992.

[158] G. Vosselman and J. de Knecht. Road tracing by profile matching and Kalman filtering. In *Proceedings of the Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, pages 265–274, 1995.

[159] Y. Wan, S. Shen, Y. Song, and S. Liu. A road extraction approach based on fuzzy logic for high-resolution multispectral data. In *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, 2007.

[160] J. Wang, P. M. Treitz, and P. J. Howarth. Road network detection from SPOT imagery for updating geographical information systems in the rural-urban fringe. *International Journal of Geographical Information Science*, 6:141–157, 1992.

[161] R. Wang and Y. Zhang. Extraction of urban road network using QuickBird pan-sharpened multispectral and panchromatic imagery by performing edge-aided post-classification. In *ISPRS Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modeling and Analysis*, 2003.

[162] C. Wiedemann and H. Ebner. Automatic completion and evaluation of road networks. *International Archives of Photogrammetry and Remote Sensing*, 33:979–986, 2000.

[163] C. Wiedemann, C. Heipke, H. Mayer, and S. Hinz. Automatic extraction and evaluation of road networks from MOMS-2P imagery. *International Archives of Photogrammetry and Remote Sensing*, 32:285–291, 1998.

[164] C. Wiedemann, C. Heipke, H. Mayer, and O. Jamet. Empirical evaluation of automatically extracted road axes. In *Empirical Evaluation Techniques in Computer Vision*, 1998.

[165] C. Wiedemann and S. Hinz. Automatic extraction and evaluation of road networks from satellite imagery. *International Archives of Photogrammetry and Remote Sensing*, 32:95–100, 1999.

[166] D. Xiong. Automated road extraction from high resolution images. Technical report, U.S. Department of Transportation, University Consortium on Remote Sensing in Transportation, 2001.

[167] J. Yang and R. S. Wang. Classified road detection from satellite images based on perceptual organization. *International Journal of Remote Sensing*, 28:4653–4669, 2007.

[168] F. Ye, L. Su, and J. Tang. Automatic road extraction using particle filters from high resolution images. *The Journal of China University of Mining & Technology*, 16:490–493, 2006.

[169] J. Youn and J. S. Bethel. Adaptive snakes for urban road extraction. In *Proceedings of the 10th International Society for Photogrammetry and Remote Sensing Congress*, 2004.

[170] C. Zhang. Towards an operational system for automated updating of road databases by integration of imagery and geodata. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58:166–186, 2004.

[171] C. Zhang, E. Baltsavias, and A. Gruen. Knowledge-based image analysis for 3D road reconstruction. In *Proceedings of the 21st Asian Conference on Remote Sensing*, 2000.

[172] C. Zhang, S. Murai, and E. Baltsavias. Road network detection by mathematical morphology. In *Proceedings of the ISPRS Workshop on 3D Geospatial Data Production: Meeting Application Requirements*, 1999.

[173] Q. Zhang and I. Couloigner. Automatic road change detection and GIS updating from remotely-sensed imagery. *Geo-Spatial Information Science*, 7:89–95, 2004.

[174] Q. Zhang and I. Couloigner. A framework for road change detection and map updating. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34:729–734, 2004.

[175] Q. Zhang and I. Couloigner. Benefit of the angular texture signature for the separation of parking lots and roads on high resolution multi-spectral imagery. *Pattern Recognition Letters*, 27:937–946, 2006.

[176] H. Zhao, J. Kumagai, M. Nakagawa, and R. Shibasaki. Semi-automatic road extraction from high-resolution satellite image. In *Proceedings of the ISPRS Conference on Photogrammetric Computer Vision*, 2002.

[177] S. Zheng, J. Liu, W. Shid, and G. Zhue. An automatic road segmentation algorithm using one-class SVM. In *Proceedings of the 14th International Conference on Geoinformatics*, 2006.

[178] J. Zhou, W. F. Bischof, and T. Caelli. Robust and efficient road tracking in aerial images. *International Archives of Photogrammetry and Remote Sensing*, 36:35–40, 2005.

[179] M. V. Zoej and M. Mokhtarzade. Road detection from high resolution satellite images using artificial neural networks. In *Proceedings of the 10th International Society for Photogrammetry and Remote Sensing Congress*, 2004.

APPENDIX A

Additional Result Samples

Additional sample results for the experiments conducted in Chapters 4 - 6 are presented in this appendix. For consistency, the sample results are obtained by applying each experiment to the candidate scenes depicted in Figure A.1. The terms *correctness*, *completeness* and *quality* will be used to describe the results. Please refer to Section 2.6 for the definition of these concepts.

Section A.1 presents results for the Canny and V1 system described in Section 4.2. The pixel and object-based results for the fusion based system discussed in Section 5.2 are illustrated in Section A.2. The results for the fully automated system presented in Section 6.2 are included in Section A.3.

(a) (b)

(c) (d)

Figure A.1: Sample scenes

A.1 Long range edge detection

Figure A.2 depicts the ACE output for the Canny-based system discussed in Section 4.2. The overall impression of the ACE results is that road centrelines were extracted to a degree, but the images include numerous incorrectly detected centreline seeds.

(a) (b)

(c) (d)

Figure A.2: ACE results for the Canny-based system

The final extracted road topologies for the Canny-based ACE results depicted in Figure A.2 are presented in Figure A.3. Given the ACE input, SORM was able to discard the majority of incorrectly detected centreline seeds, but a few short spurious sections remained. The system was not able to detect all roads centrelines and it is doubtful if the quality of the final road topologies is high enough to be incorporated in operational GIS systems.

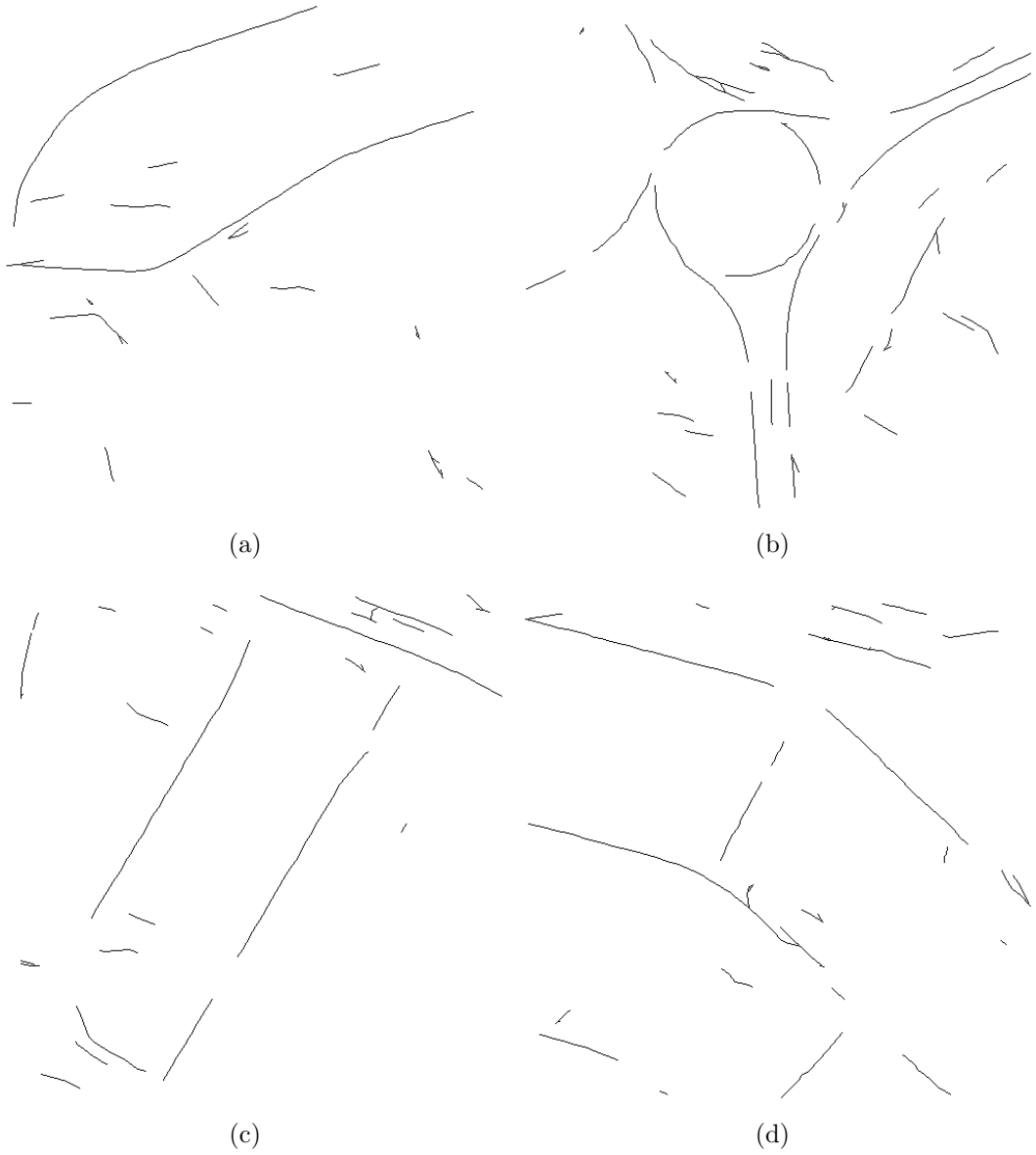
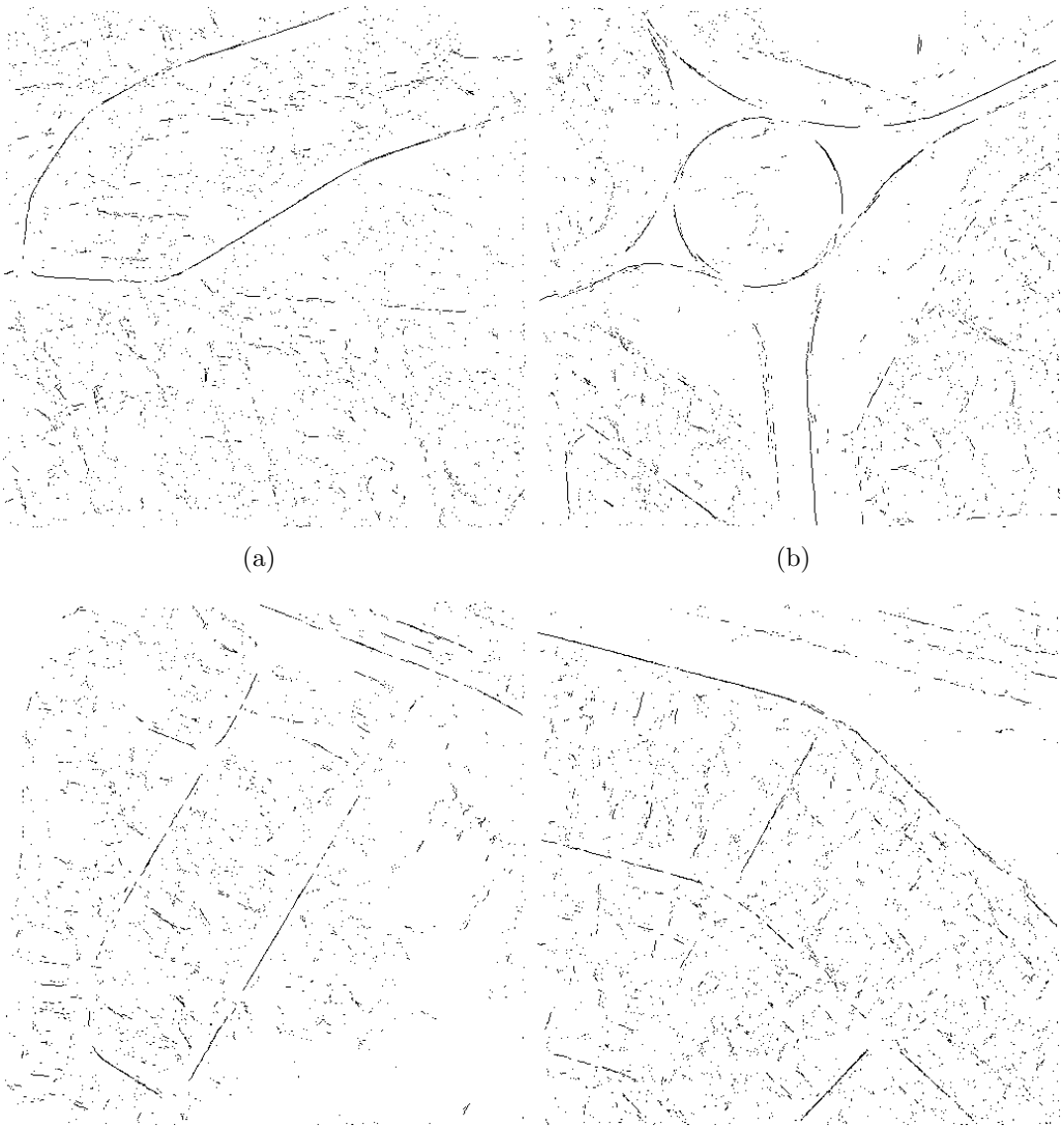


Figure A.3: SORM results for the Canny-based system

The ACE output for the V1-based system discussed in Section 4.2 is illustrated Figure A.4. When compared to the Canny-base results in Figure A.2, the V1-based ACE results appear to have less incorrectly detected centreline seeds, but also fewer correctly detected centreline seeds.

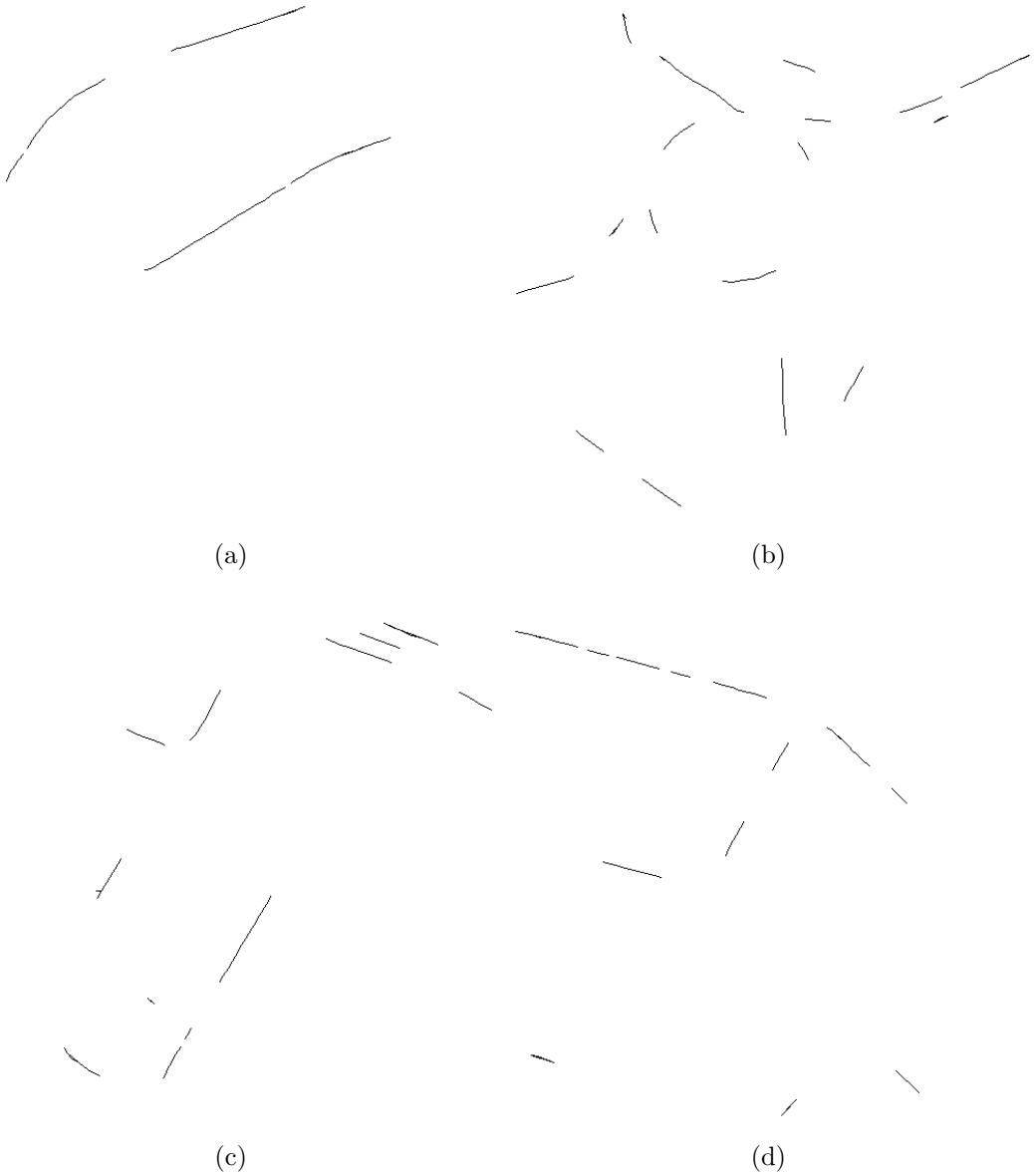


(a) (b)

(c) (d)

Figure A.4: ACE results for the V1-based system

The lower number of road seeds detected in the V1-based ACE results translated into fewer roads in the final road topology, as illustrated in Figure A.5. The results appear to have very few incorrectly detected centrelines, but the gaps within the final network is considerably larger than the result produced by the Canny-based system (refer to Figure A.3).



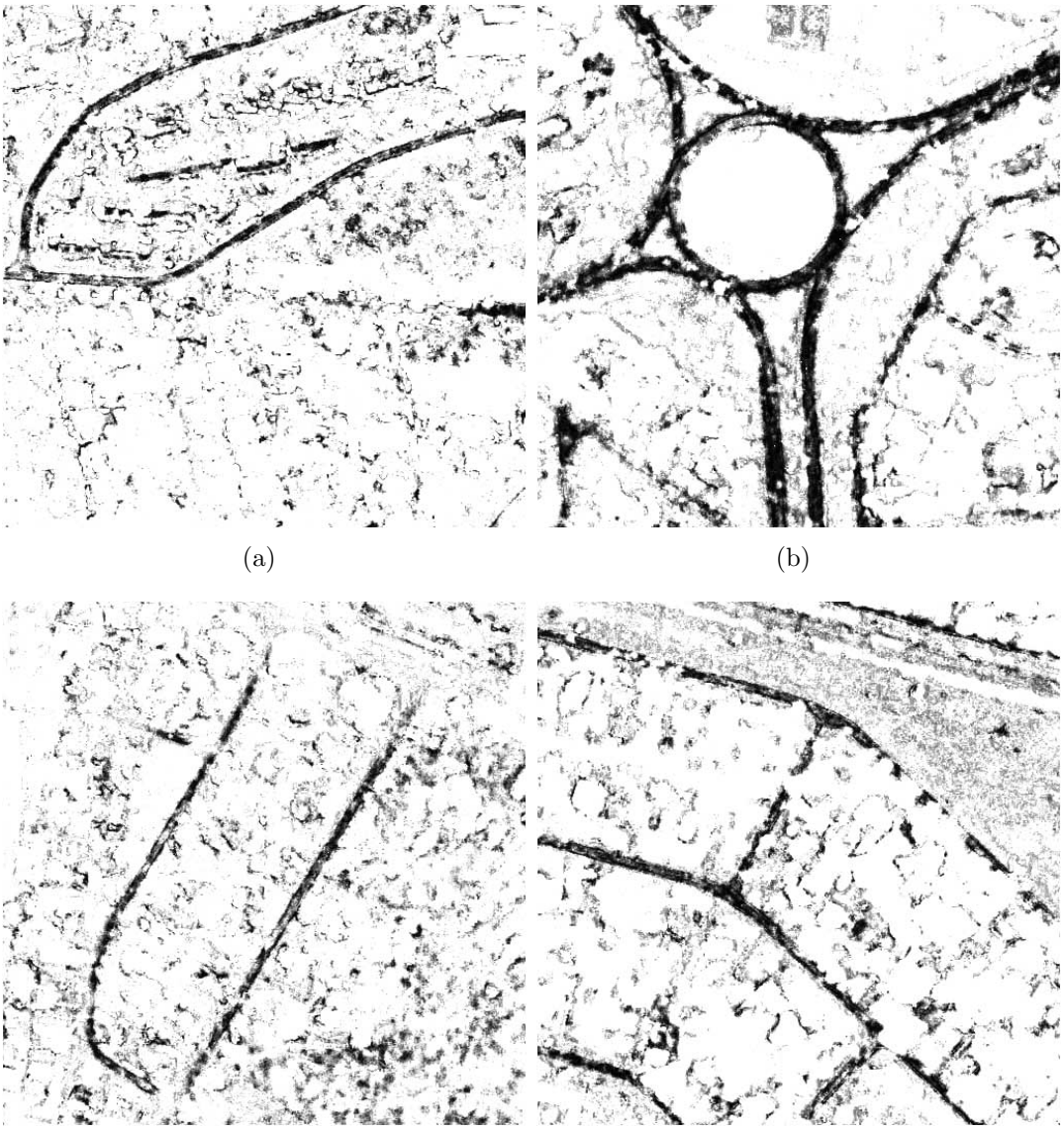
(a) (b)

(c) (d)

Figure A.5: SORM results for the V1-based system

A.2 Object-based analysis

The images illustrated in Figure A.6 were obtained by fusing the results obtained from three spectral classifiers (refer to Sections 3.5 and 3.6). The pixels in each scene were classified individually. The results are varied with certain roads clearly visible and contiguous, while other roads had a considerable number of breakages, or were entirely omitted.

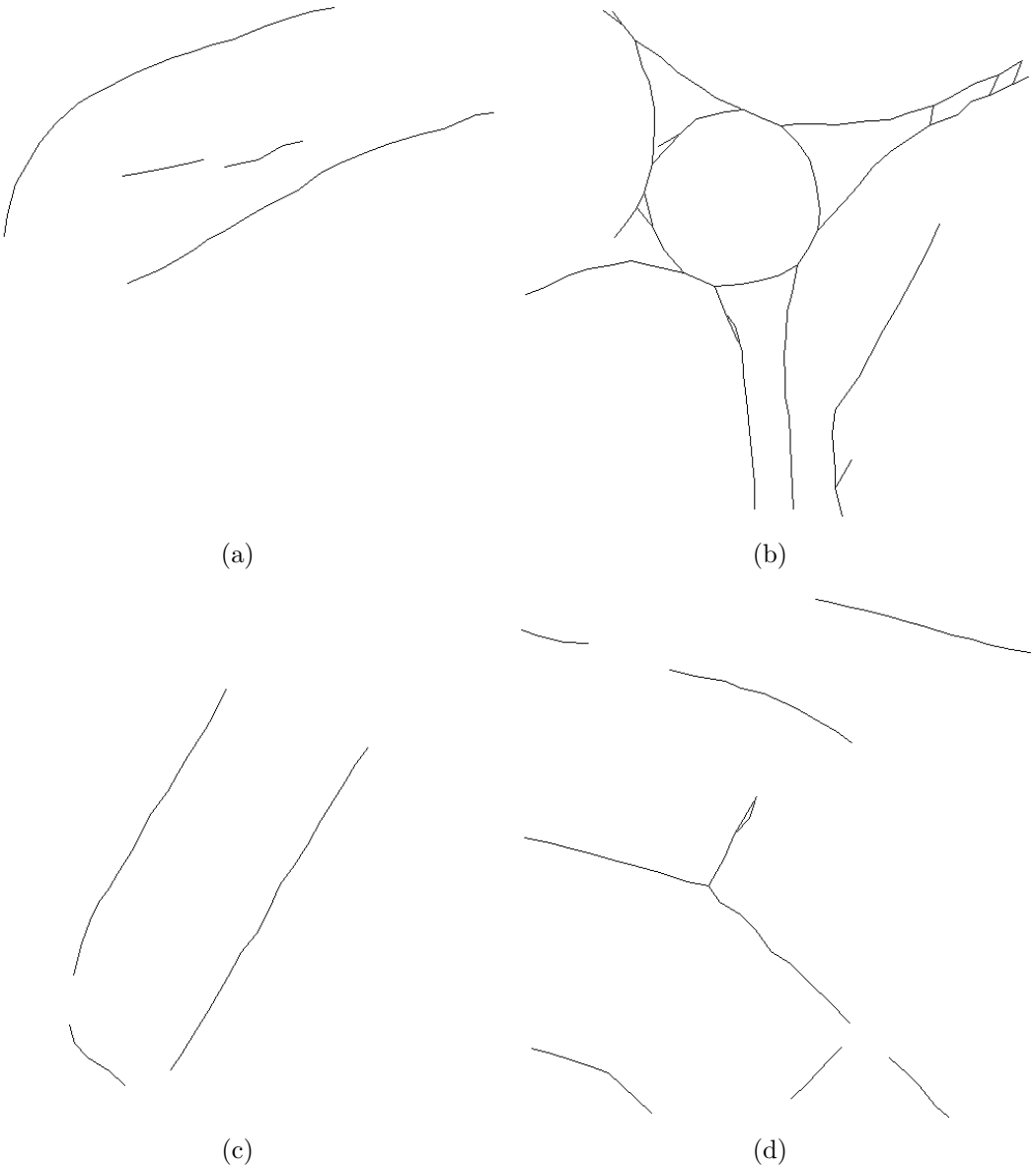


(a) (b)

(c) (d)

Figure A.6: Fusion of three spectral classifiers using PBA

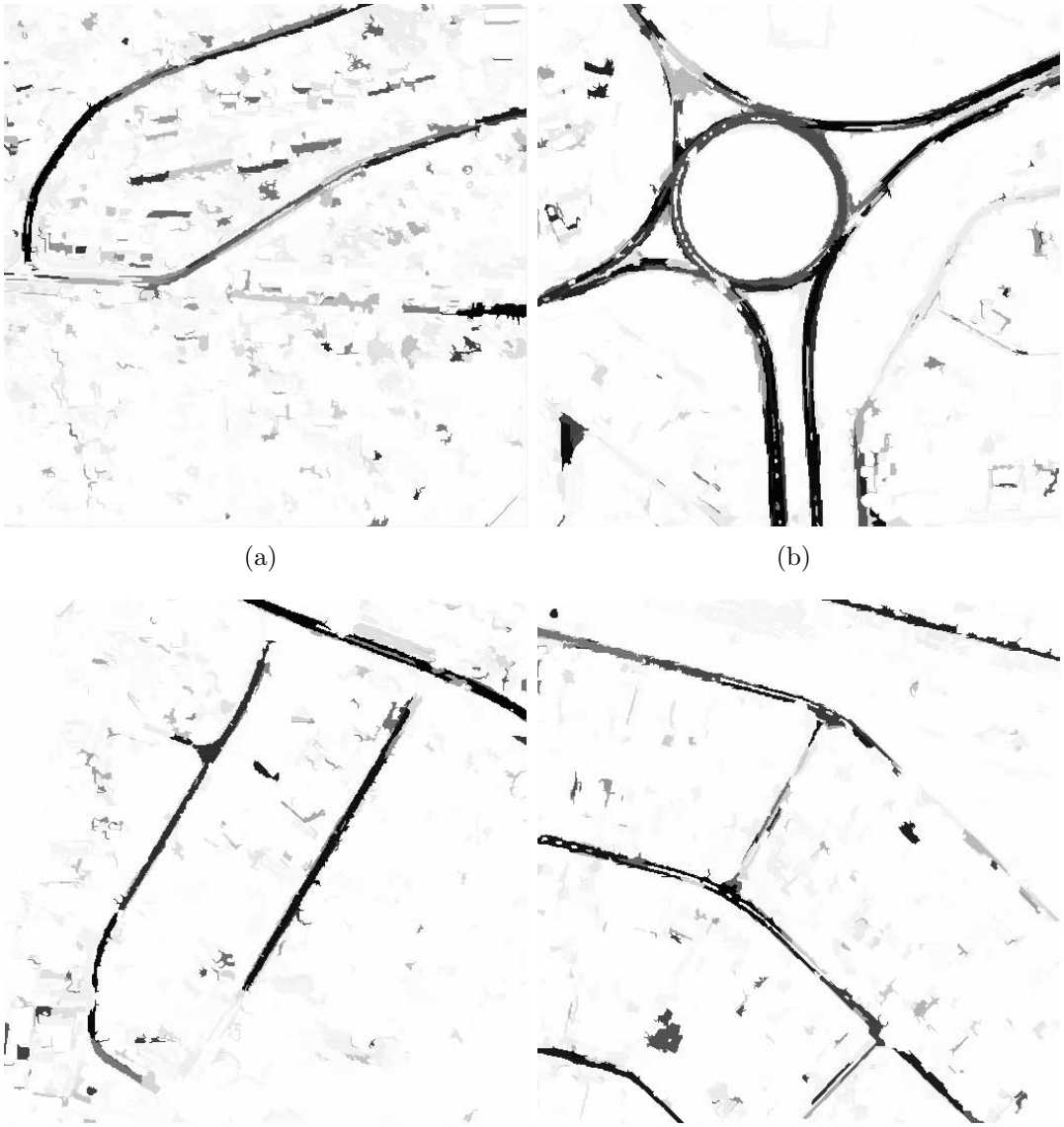
The final extracted topologies for the spectrally classified images (refer to Figure A.6) are depicted in Figure A.7. It appears as if very little spurious roads were extracted, which implies a high correctness. The networks are not fully complete and missing roads, and gaps are evident. The SORM algorithm exhibits some peculiar behaviour at junctions and intersections in Figure A.7b.



(a) (b) (c) (d)

Figure A.7: SORM results for the PBA system

The images in Figure A.8 were obtained with an object-based classification process. The objects were created with the FHS algorithm described in Section 3.4 and the classification process used to create the images in Figure A.6 was adapted to classify objects. When comparing the object-classified images in Figure A.8 to the pixel-classified images in Figure A.6, the object-based approach resulted in fewer false classifications in the non-road regions. Both approaches struggled to find all roads.



(a) (b)

(c) (d)

Figure A.8: Fusion of three spectral classifiers using OBA

The road networks extracted from the object-based approach (refer to Figure A.9) appears more aesthetically pleasing than those produced with the pixel-based approach (refer to Figure A.7). The object-based approach only extracted a single false road section, which appears in Figure A.9a. The road networks appear to be less complete than those obtained with the pixel-based approach (refer to Figure A.7).

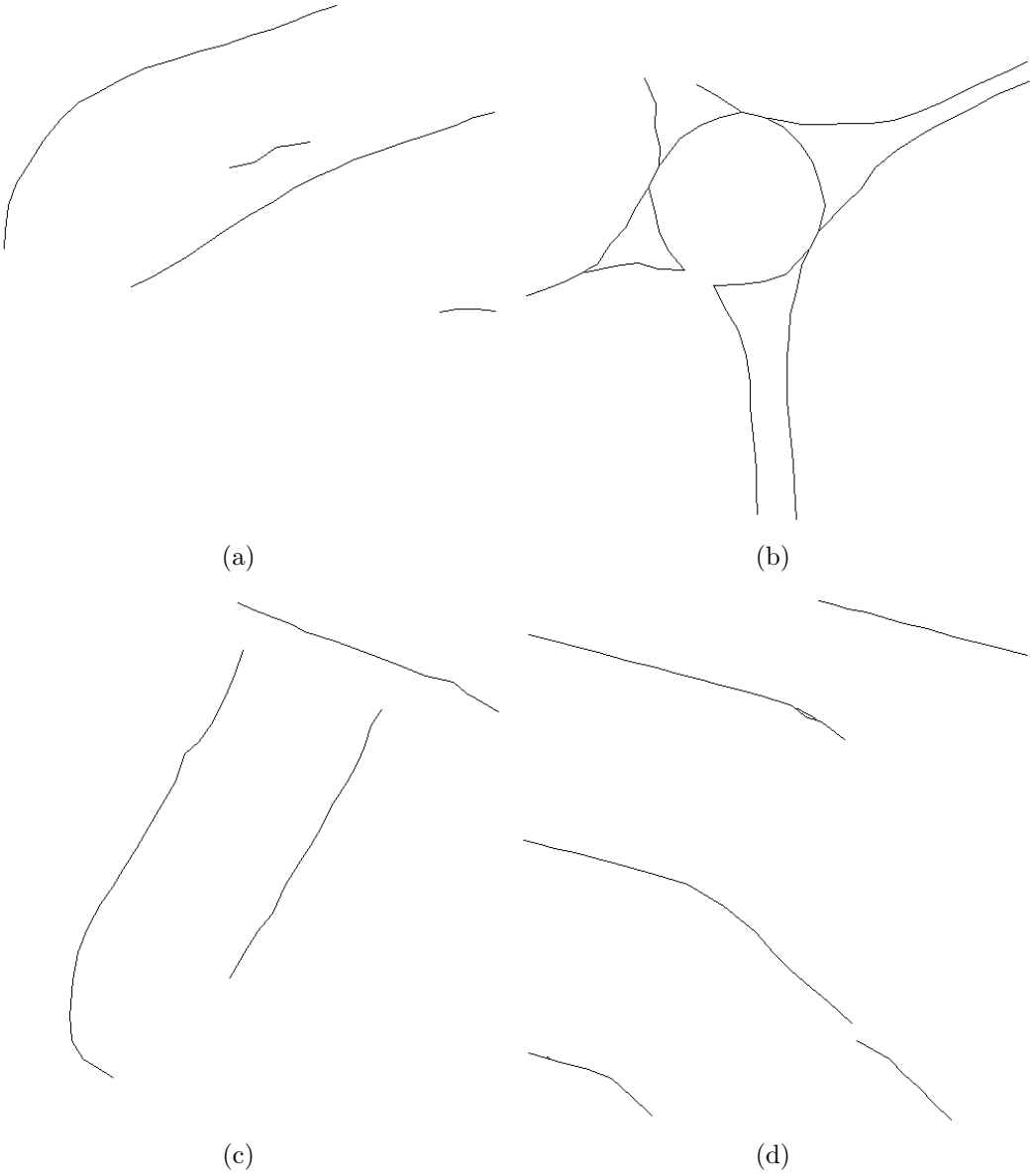
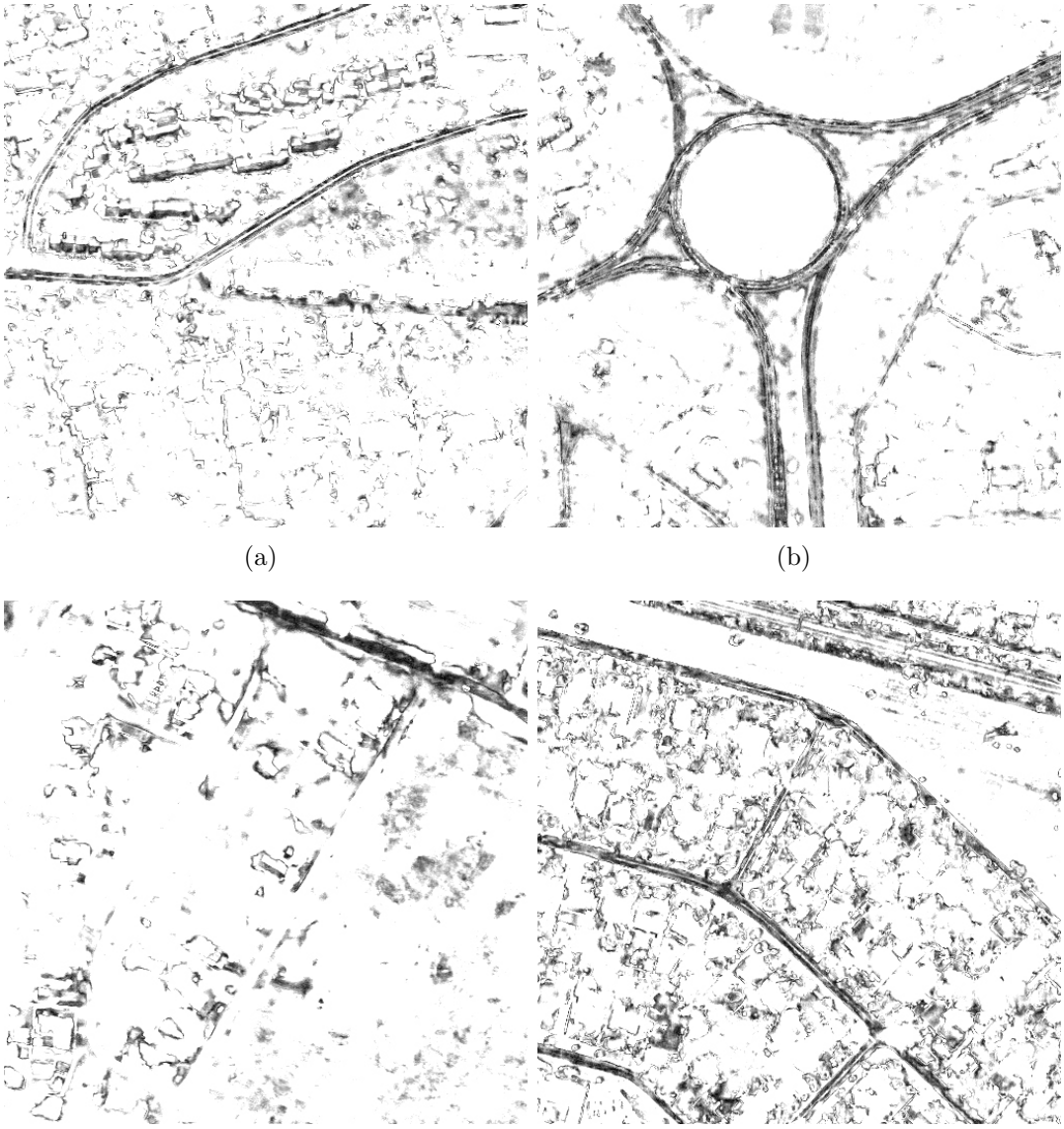


Figure A.9: SORM results for the OBA system

A.3 Fully automated road extraction using spectral classification

The classification process described in Section 6.2 was used to create the images in Figure A.10. The classification process struggled to find roads in Figure A.10c, and a considerable amount of background noise is visible in Figure A.10d.



(a) (b)

(c) (d)

Figure A.10: Fully automated spectral classification (including the k-means filter)

The overall quality of the extracted road networks depicted in Figure A.11 is poor. The networks in Figures A.11a and A.11d were partly extracted, but contained a considerable number of incorrectly detected roads. Figure A.11b had few incorrectly detected roads, but the network is incomplete. The quality of the roads extracted in Figure A.11c is very low, since large road portions are missing and a two spurious road sections are evident.

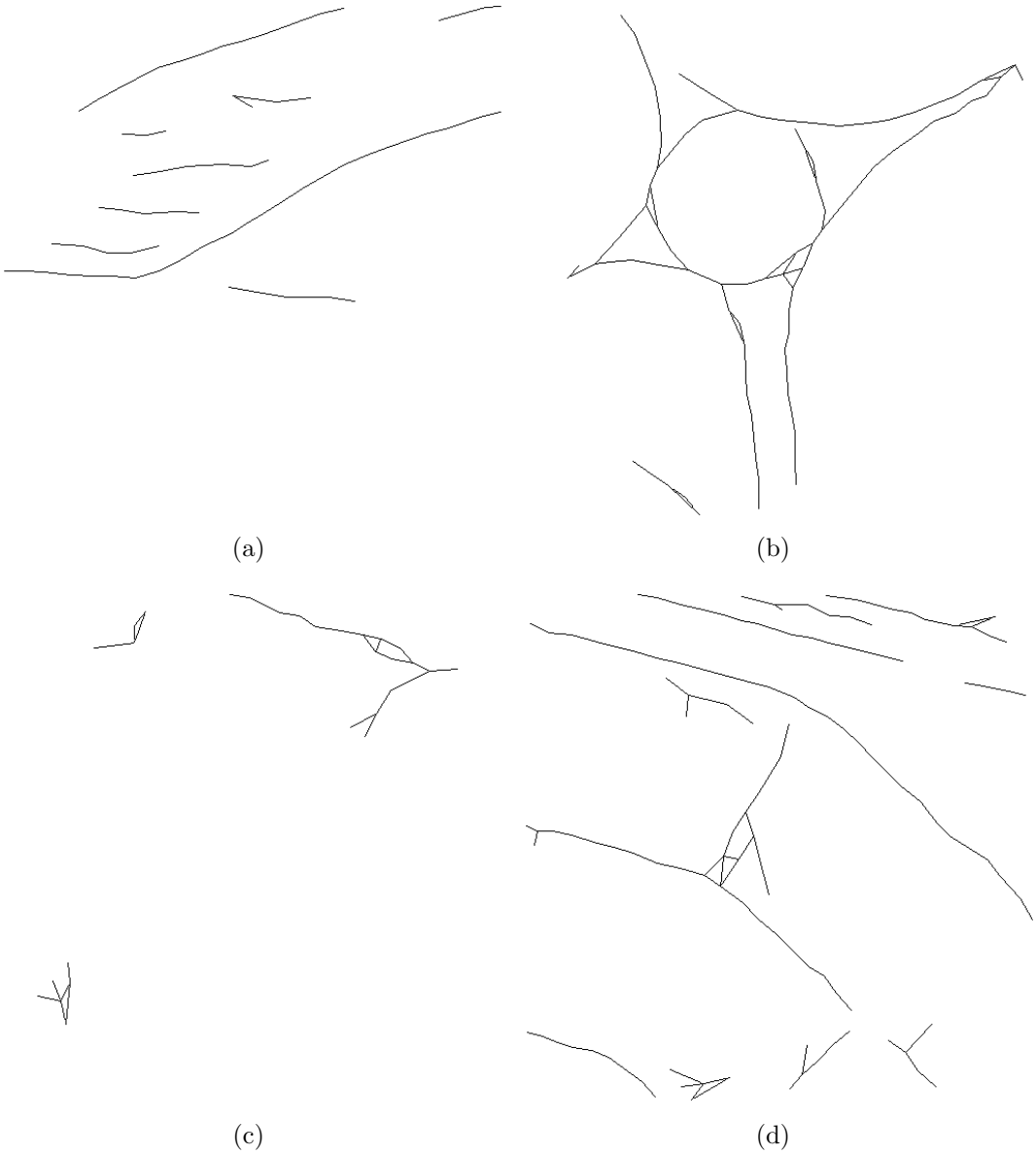
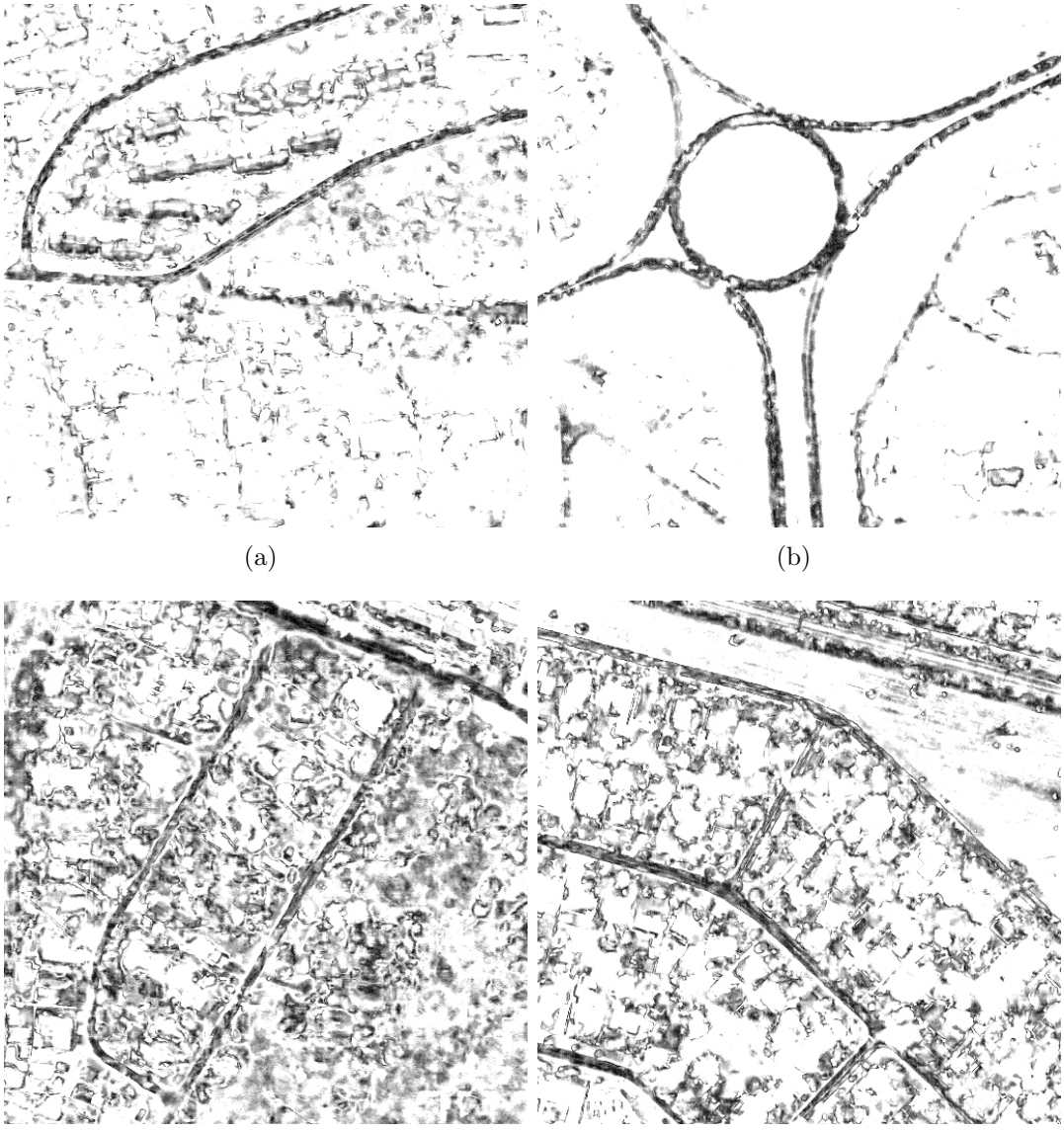


Figure A.11: SORM results for the fully automated system (including the k-means filter)

The classification process described in Section 6.2 included a k-means filtering component, which was removed to produce the images presented in Figure A.12. Chapter 6 showed that the addition of the filter did not improve the results of the final extracted road networks. These findings are not reflected in all the sample images provided, with Figures A.12c and A.12d containing many misclassified road pixels. This misrepresentation of the overall results can be attributed to the inconsistent nature of the system.



(a) (b)

(c) (d)

Figure A.12: Fully automated spectral classification (excluding the k-means filter)

The final extracted road networks for the images in Figure A.12 are depicted in Figure A.13. Figure A.13a is comparable to the networks produced by the filtered approach in Figure A.11a. Figure A.13b appears more aesthetically pleasing than Figure A.11b, but has fewer correctly extracted roads. SORM struggles to detect roads from images, such as Figures A.12c and A.12d, which have many misclassified road pixels. In these instances SORM produced peculiar road networks (refer to Figures A.13c and A.13d).

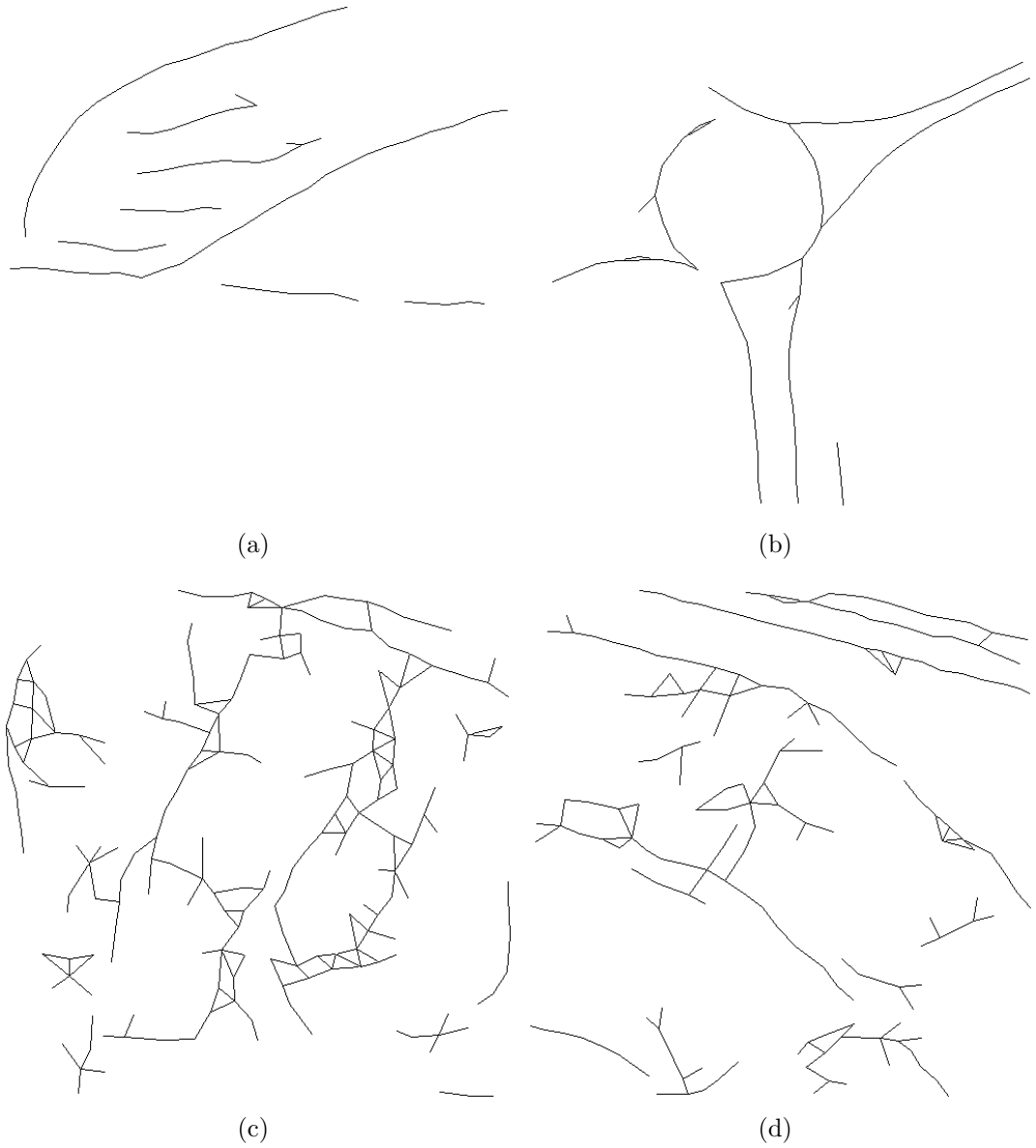


Figure A.13: SORM results for the fully automated system (excluding the k-means filter)

Glossary

2D Two Dimensional, 29
3D Three Dimensional, 65

ACE Anti-parallel Centreline Extraction, 11
ANN Artificial Neural Networks, 14
APAR Anti-Parallel, 75
ARMG Automated Road Model Generator, 128
ATS Angular Texture Signature, 19

BCC Background Correctness Coefficient, 62
BGIS 2000 Ball Aerospace Global Imaging System 2000, 90

CART Classification and Regression Trees, 18
CI Computation Intelligence, 54

DIP Digital Image Processing, 4
DP Dynamic Programming, 28
DSM Digital Surface Model, 21
DST Dempster-Shafer Theory, 50

EO-1 Earth Observing Mission 1, 66

FHS Felzenszwalb and Huttenlocher Segmentation, 78

GA Genetic Algorithm, 57
Gautrans Gauteng Department of Transport and Public Works, 101
GBT Generalized Balanced Ternary, 36
GDPA Gradient Directional Profile Analysis, 26
GIS Geographic Information Systems, 1
GPS Global Positioning System, 2
GSD Ground Sampling Distance, 26
GSS Golden Section Search, 63

HSI Hue Saturation Intensity, 41
HT Hough Transform, 27
HVS Human Vision Systems, 3

IRS Indian Remote Sensing Satellite, 66
ISODATA Interactive Self-Organizing Data Analysis Technique, 12

LiDAR Light Detection and Ranging, 66

ML Maximum Likelihood, 15
MRA Multi-Resolution Analysis, 33
MSA Multi-Scale Analysis, 33
MST Minimum Spanning Tree, 58

NDVI Normalized Difference Vegetation Index, 20
NDWI Normalized Difference Water Index, 146
NIr Near Infrared, 66

OBA Object-Based Analysis, 110
OSSIM Open Source Software Image Map, 91

PBA Pixel-Based Analysis, 110

RBF Radial Basis Function, 17
RCC Road Correctness Coefficient, 62
RGB Red, Green and Blue, 41
RMS Root Mean Square, 61
RNC Road Network Construction, 54
RNE Road Network Extraction, 3
RNG Relative Neighbourhood Graph, 88
ROC Receiver Operating Characteristics, 16
RS Remotely Sensed, 2

SAR Synthetic Aperture Radar, 12
SE Structuring Element, 29
SOM Self-Organizing Map, 73
SOME Self-Organizing Map Extended, 73
SORM Self-Organized Road Mapping, 22
SPOT Satellite Pour l'Observation de la Terre, 66
SSRC Self-Supervised Road Classification, 22
SSWHT Spatial Signature Weighted Hough Transform, 28
SVM Support Vector Machine, 17

UTM Universal Transverse Mercator, 90

V1 Primary visual cortex, 71
VHR Very High Resolution, 16

WSG 84 World Geodetic System, 90