AUTOMATIC MULTI-SCALE SEGMENTATION OF HIGH SPATIAL RESOLUTION SATELLITE IMAGES USING WATERSHEDS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

KEREM ŞAHİN

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2013

Approval of the thesis:

AUTOMATIC MULTI-SCALE SEGMENTATION OF HIGH SPATIAL RESOLUTION SATELLITE IMAGES USING WATERSHEDS

submitted by **KEREM ŞAHİN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** ________________

Assoc. Prof. Dr. İlkay Ulusoy
Supervisor, **Electrical and Electronics Engineering Dept., METU** ________________

Examining Committee Members

Prof. Dr. Uğur Halıcı
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Gözde B. Akar
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Engineering Dept., METU ________________

Asst. Prof. Dr. Alptekin Temizel
Work Based Learning Studies Dept., METU ________________

Date: 21.01.2013

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name	: Kerem Şahin
Signature	:

ABSTRACT

AUTOMATIC MULTI-SCALE SEGMENTATION OF HIGH SPATIAL RESOLUTION SATELLITE IMAGES USING WATERSHEDS

Şahin, Kerem
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. İlkay Ulusoy

January 2013, 77 pages

Useful information extraction from satellite images for the use of other higher level applications such as road network extraction and update, city planning etc. is a very important and active research area. It is seen that pixel-based techniques becomes insufficient for this task with increasing spatial resolution of satellite imaging sensors day by day. Therefore, the use of object-based techniques becomes indispensable and the segmentation method selection is very crucial for object-based techniques. In this thesis, various segmentation algorithms applied in remote sensing literature are presented and a segmentation process that is based on watersheds and multi-scale segmentation is proposed to use as the segmentation step of an object-based classifier. For every step of the proposed segmentation process, qualitative and quantitative comparisons with alternative approaches are done. The ones which provide best performance are incorporated into the proposed algorithm. Also, an unsupervised segmentation accuracy metric to determine all parameters of the algorithm is proposed. By this way, the proposed segmentation algorithm has become a fully automatic approach. Experiments that are done on a database formed with images taken from Google Earth® software provide promising results.

Keywords: Watershed segmentation, multi-scale segmentation, automatic segmentation, high spatial resolution satellite images, object-based classification

ÖZ

YÜKSEK UZAYSAL ÇÖZÜNÜRLÜKLÜ UYDU GÖRÜNTÜLERİNİN WATERSHED KULLANILARAK ÇOK ÖLÇEKLİ OTOMATİK BÖLÜTLENMESİ

Şahin, Kerem
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. İlkay Ulusoy

Ocak 2013, 77 sayfa

Yol ağı çıkarımı ve güncellenmesi, şehir planlama v.b. gibi yüksek seviye uygulamaların kullanımı için uydu görüntülerinden işe yarar bilgi çıkarımı çok önemli ve aktif bir araştırma alanıdır. Uydu görüntüleme algılayıcılarının günden güne artan uzaysal çözünürlüğü ile, piksel tabanlı yöntemlerin bu iş için yetersiz kaldığı görülmüştür. Bu yüzden, nesne tabanlı yöntemlerin kullanımı kaçınılmaz hale gelmiştir ve bölütleme yöntemi seçimi, nesne-tabanlı yöntemler için çok kritiktir. Bu tezde, uzaktan algılama literatüründe uygulanmış çeşitli bölütleme algoritmaları sunuldu ve bir nesne tabanlı sınıflandırıcının bölütleme adımı olarak kullanabileceği watershed ve çok ölçekli bölütleme tabanlı bir bölütleme yöntemi önerildi. Önerilen bölütleme yönteminin her adımı için, alternatif yaklaşımlarla niteliksel ve niceliksel karşılaştırmalar yapıldı. En iyi performansı sağlayanlar, önerilen algoritmaya dahil edildi. Ayrıca, algoritmanın tüm parametrelerine karar vermek için bir denetlenmeyen bölütleme doğruluk metriği önerildi. Bu sayede, önerilen bölütleme algoritması tamamen otomatik bir yaklaşım haline geldi. Google Earth® yazılımından alınan görüntülerle oluşturulan veritabanı üzerinde gerçekleştirilen deneyler gelecek vaat eden sonuçlar sağladı.

Anahtar Kelimeler: Watershed bölütlemesi, çok ölçekli bölütleme, otomatik bölütleme, yüksek uzaysal çözünürlüklü uydu görüntüleri, nesne tabanlı sınıflandırma

To my lovely wife Eda

ACKNOWLEDGEMENTS

I would like to thank my supervisor Assoc. Prof. Dr. İlkay Ulusoy for her supervision, support and tolerance throughout the development and improvement of this thesis.

I would like to thank Prof Dr. Aydın Alatan, Prof Dr. Uğur Halıcı, Prof Dr. Gözde Akar and Assist. Prof. Dr. Alptekin Temizel for serving in my committee and sharing their opinions.

I would like to also express my thanks to my friends, my colleagues and ASELSAN A.Ş. for their precious support and fellowship.

I am grateful to TÜBİTAK for their financial support during my graduate education.

Finally, I would like to thank my family and Eda for their love, support and patience over the years. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Image segmentation can be defined as partitioning an image into non-overlapping, homogenous segments/objects such that combination of any two adjacent segments/objects is heterogonous. The aim to incorporate a segmentation method to an image processing framework could be very distinct depending on the problem domain Image segmentation is seen as an auxiliary process to landscape change detection and land use/ land cover classification in remote sensing problems [6]. Especially, it is crucial for object-oriented classification methodology that is heavily used in remote sensing image processing field. In object-oriented classification, objects found after an initial segmentation step are classified rather than pixels [8].

High spatial resolution satellite image classification is a highly complex and challenging task because of two reasons. Firstly, the increase in spatial resolution of satellite sensors makes pixel-based classification techniques inefficient because of the growing intra-class spectral variability of different classes. The huge amount of spatial details present in those images necessitates a different look at the problem compared with low or middle resolution satellite images. Some of very popular high spatial resolution satellite imaging sensors and their spatial resolutions and launch dates are shown in Table 1-1. Second major problem of high spatial resolution satellite image classification is the inadequacy of spectral resolution. There is an inverse relation between spatial resolution and spectral resolution of imaging sensors [7]. Considering these two reasons, it can be easily said that a classification method that only takes spectral values of individual pixels into account will be insufficient. Therefore, object-based methods that consider both spectral and spatial properties of pixels by using segments would be more appropriate for high spatial resolution satellite image classification problem.

Table 1-1 Some of very popular high spatial resolution satellite sensors and their features

Satellite	*Spatial Resolution*	*Launch Date*
IKONOS	4m/multi-spectral, 1m/panchromatic	1999
QuickBird	2.4m/multi-spectral, 0.6m/panchromatic	2001
GeoEye-1	1.65m/multi-spectral, 0.41m/panchromatic	2008
WorldView-2	1.8m/multi-spectral, 0.46m/panchromatic	2009

Object-based classification is built on the observation that important semantic information is included in image objects rather than pixels [1]. Pixel-based classification methods do not take into account contextual information which is very important in high spatial resolution satellite image classification. In object-based classification, after the creation of segments, there will be so many features such as spectral, spatial, structural, textural etc. to use for classification. A successful classification result needs feature values that represent the class type well and this implicitly needs segments that represent the specific objects. Therefore, the success of an object-based classification system heavily depends on its segmentation step.

In this thesis, a segmentation method for high spatial resolution satellite images that can be used as an initial segmentation step of an object-based classifier is aimed with minimal user input, maximal segmentation accuracy and maximal execution speed. For this purpose, firstly, the segmentation algorithms that are applied in remote sensing literature are presented in a categorized way. Then, we decide to apply a hybrid segmentation algorithm that goes into both watershed model and multi-resolution model of segmentation categories by considering the design goals of this thesis. After deciding the main category of segmentation algorithm, we present the alternative approaches for different steps of the algorithm and we decide on an alternative approach for every step of the algorithm by applying qualitative and quantitative analyzes. Finally, after constructing the proposed algorithm, we propose a method that is based on an unsupervised segmentation accuracy evaluation metric to make proposed segmentation approach fully automatic.

The contributions of this thesis can be listed as below:

- A new region merging procedure that is a hybrid form of two different region merging algorithms is proposed to use as the multi-scale segmentation algorithm part of the proposed algorithm.
- A procedure that is based on an unsupervised segmentation accuracy evaluation metric is proposed to make the segmentation approach fully automatic.

This thesis is composed of five chapters excluding this chapter.

- In Chapter 2, generally used segmentation methods and trends in remote sensing literature are presented.
- In Chapter 3, pre-processing methods, watershed segmentation methods, multi-scale segmentation methods and segmentation accuracy and performance measures that are evaluated in this thesis are presented.
- In Chapter 4, the proposed approach is presented by putting example results of the implementation using the database images 1-3.
- In Chapter 5, the experiments are presented. Strategies and concerns while choosing the right algorithm from the set of alternative approaches for the specific purpose of the thesis is explained in detail. Also, the comparison of proposed approach of this thesis with similar algorithms in the previous literature is done.
- In Chapter 6, the whole thesis is summarized and possible future works are proposed.

CHAPTER 2

A SURVEY OF SEGMENTATION TECHNIQUES USED IN REMOTE SENSING

In this chapter of the thesis, generally used segmentation algorithms in the remote sensing literature are examined.

Image segmentation can be defined as partitioning an image into non-overlapping, homogenous on its own (with respect to preferred homogeneity criterion) segments/objects such that combination of any two adjacent segments/objects is heterogonous (with respect to the same homogeneity criterion).

[9] gives a formal definition of the segmentation problem. If we suppose that a true image T is corrupted by a noise factor of N, observed image O could be formulated as in Formula 2-1.

$$O(p) = T(p) + N(p), p \in C \tag{2-1}$$

$C = \{(x,y) : 1 \leq x \leq L_c, 1 \leq y \leq L_r\}$ denotes the spatial coordinates of a pixel where L_r is the row count and L_c is the column count. Also, presuming the piecewise constantness of T, we can write the ideal segmentation of C as in Formula 2-2.

$$\Delta_K(C) = \{R_1, R_2, \dots, R_K\} \text{ such that } C = \cup_{k=1}^{K} R_k \tag{2-2}$$

where:

- $R_m \cap R_n = \emptyset, \forall m, n \in \{1,2,\dots,K\}$, for $m \neq n$,
- $R_m, \forall m \in \{1,2,\dots,K\}$, is connected,
- $T(p) = Z_m$, if $p \in R_m$, $\forall m \in \{1,2,\dots,K\}$, where Z_m is a constant, and
- $Z_m \neq Z_n$, if R_m and R_n are adjacent.

After applying a segmentation algorithm to the observed image, if the resulting count of segments K_0 is greater than K, the ideal segment count, the segmentation result is called as over-segmented, otherwise, under-segmented. The over-segmentation case is a minor one comparing with the under-segmentation case because over-segmentation can be recovered using region merging and classification steps [7].

Segmentation algorithms can be grouped by various ways. While [7] groups segmentation algorithms by the level of operations applied, [6] categorizes them by considering the image analysis approach models.

2.1 Methods Categorized by the Level of Operations

[7] divides segmentation algorithms into four different categories as below:

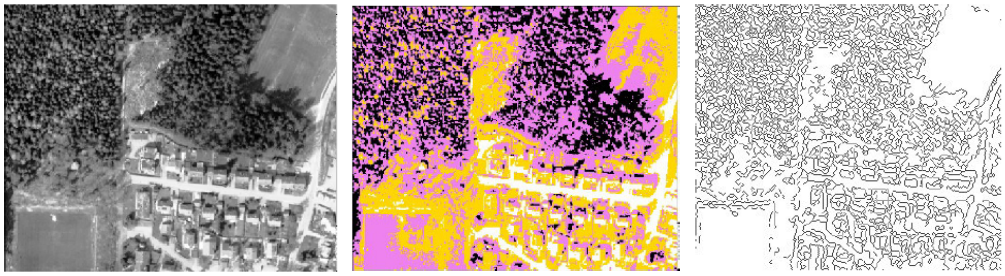
- Point-based algorithms: pixel level operations
- Edge-based algorithms: edge level operations
- Region-based algorithms: region level operations
- Combined(hybrid) algorithms

2.1.1 Point-based Segmentation Approaches

These approaches are also called as histogram-based segmentation methods. Because, in these methods, one or more than one global threshold is found looking at the histogram of the image and partitions are obtained after applying this/these threshold/thresholds. In this type of methods, because of using global information and discarding local information, contextual information could not be used in the segmentation. Therefore, it is necessary to use a complementary algorithm to incorporate contextual information following segmentation.

In [9], histogram is described as a Gaussian mixture pdf (probability density function) and segmentation problem is defined as finding parameters of mixture components and assigning pixels to corresponding components.

Point-based approaches are not much appropriate for high spatial resolution remote sensing image segmentation because intra-class variance of objects in high spatial resolution is very high. This impropriety can be clarified with an example segmentation taken from [7] that is shown in Figure 2-1. Also, this class of methods needs supervision about the number of segments beforehand and this is a major drawback of these approaches.



(a) Original Image (b) Point-based seg. result (c) Contours

Figure 2-1 An example satellite image (a), its segmentation result with four classes (b) with a point-based segmentation method, contours found (c) for edge-based segmentation [7]

2.1.2 Edge-based Segmentation Approaches

This type of methods is used for boundary extraction of objects in the image. After boundary extraction, segments can be generated by filling the areas whose boundaries are known.

There are two steps of object boundary extraction as below:

1. Edge detection
2. Generation of closed contours

First, edges of image are extracted using discontinuity property by edge detection algorithms (Sobel, Prewitt etc. are the most known). After edge detection step, obtained contours are not closed most of the time [4]. To attain closed object boundaries, an auxiliary approach of contour closing is necessary.

In [4], the optimal edge detector algorithm which is an edge-based segmentation method is used. This method includes three main steps. First, the image is filtered by the Canny-Deriche operator to obtain the gradient image. Second, to keep the consistent boundaries, a hysteresis thresholding is applied. Eventually, contours in the image are closed in the way that gives best count. The count in a way is computed by summing the gradient values on this way.

Also, the watershed segmentation that is based on image morphology can be put in the category of edge-based segmentation approaches [4].

High susceptibility to noise and the lack of successful generation of closed and one-pixel wide contours [9] are the main drawbacks of edge-based segmentation algorithms that prevent their use in remote sensing field. An example result of contour generation with an edge-based segmentation method can be seen in Figure 2-1. We can easily observe the high density of edges that will be resulted in over-segmentation afterwards in textured regions like forest area.

2.1.3 Region-based Segmentation Approaches

In this type of techniques, regions which are defined as spatially connected sets of pixels and homogenous with respect to a predefined homogeneity measure are found. There are three main region-based segmentation methods as shown below:

- Region growing/merging
- Splitting
- Split and merge

In region growing and merging techniques, segments at hand (found after an initial segmentation step) are merged with spatially neighboring and similar segments, with respect to the homogeneity measure. These segments could be one pixel segments. In splitting techniques, whole image is taken as one heterogeneous region at first. Then, until all fragments are homogenous with respect to chosen homogeneity criterion, every heterogeneous image region is tessellated into four pieces in a rectangular manner. In split and merge type methods, different from splitting techniques, when the splitting process is over, a merging process is started to merge similar neighboring regions. The main drawback of splitting and split and merge methods is the boundary corruption. They always tend to produce rectangular shaped boundaries. The most critical point in the techniques described above is a good homogeneity measure [9].

2.1.4 Hybrid Segmentation Approaches

Combining previously discussed segmentation approaches to eliminate their deficiencies while using on their own and trying to present an advanced solution to the segmentation problem is common in the literature. Most of the time, edge and region-based segmentation techniques are used in conjunction [9].

In [11], after initial partitioning of the input image with morphological watershed transformation, a fast region merging algorithm that uses region adjacency graph for handling similarity and spatial neighborhood of segments is applied to get the final result. [11, 12, 40, 14] also use a similar hybrid segmentation technique which is based on watershed transform and region merging.

2.2 Methods Categorized by the Image Analysis Approach

[6] classifies segmentation algorithms as shown below:

- Object-Background Model
- Markov Random Field (MRF) Model
- Fuzzy Model
- Neural Model
- Multi-resolution Model
- Watershed Model

In general, approaches in the literature are using combination of two or more models to decrease the shortcomings of each other and obtain a superior result.

2.2.1 Object-Background Model

[6] asserts that the methods in this category are using the conjecture of uniformness of background and spectrally well-separated objects on this background. Because of considering only individual pixels, these methods are mostly using spectral features in the image. A histogram is used to show spectral distribution. Therefore, the algorithms that belong to this category generally use a histogram thresholding approach.

2.2.2 Markov Random Field (MRF) Model

In this model, it is assumed that the observed image is the degraded version of the true image that can be modeled with a Markov or Gibbs random field and the aim is to estimate the true image using an optimality criterion [9]. Maximum a Posteriori (MAP) Probability criterion is the most known and used criterion of optimality when estimating the true image in this type of approaches [24] and there are 3 generally used estimation algorithms which are Expectation Maximization (EM), Stochastic Expectation Maximization (SEM) and Iterated Conditional Modes (ICM) [10].

[24] uses a segmentation approach that incorporates Markov Random Field model for color textured images. First, it extracts textural features of regions belonging to different classes selected by the user using Gray Level Co-occurrence Matrix (GLCM) method and finds the statistical mean feature as the most useful one by examining the class separability. Then, it accepts the feature matrix of the image as a degraded version of the true image and models it as a Markov Random Field to estimate the unknown image labels (true image). Estimation step employs ICM algorithm using MAP criterion.

The reason that makes MRF model a charming approach in high spatial resolution satellite image analysis is its ability to use the contextual information in the image. However, it is not appropriate for the applications that require low execution times and automaticity. It uses complex calculations that need high execution times [6] and needs prior information about the true image [9].

2.2.3 Fuzzy Model

Fuzzy model utilizes fuzzy clustering for segmentation. In traditional clustering, each point is belonging to a specific cluster. On the other hand, in fuzzy clustering, each point could belong to more than one cluster with a certain probability value corresponding to each cluster. These probabilities are called point's cluster coefficients. Most fuzzy segmentation methods present in the literature are based on Fuzzy Center Means (or Fuzzy C-Means or FCM) clustering and fuzzy thresholding [19].

[18] defines steps of FCM clustering approach as shown below:

1. Determine the cluster count.
2. Assign each point's cluster coefficients in a random manner.
3. Calculate the centroid of each cluster and recalculate cluster coefficients of each point using newly found centroids.
4. Iterate step 3 until convergence (successive coefficient change is no more than a specified threshold).

In [15], a supervised fuzzy logic system is applied with the aim of improving object-oriented classification results. In this paper, first, an initial segmentation in a fine scale is performed and primitive segments are obtained. Next, this segmentation results (primitive segments) are used to train the fuzzy logic system to procure the best segmentation parameters.

In [16], FCM clustering approach is used to group segments into clusters as a final step of a hybrid multi-scale segmentation algorithm. First, a coarse scale version of original image is partitioned to initial coarse regions. Next, those coarse regions' equivalents are found in the original image and these equivalents are segmented again on their own. Finally, achieved segmentation results are clustered using FCM.

In [17], FCM is used to segment texture feature images obtained with different sized local windows (3*3, 5*5 and 7*7 neighborhoods).

The main drawbacks of this method are its complexity and the need of cluster count information (by considering the FCM case).

2.2.4 Neural Model

In computer vision algorithms, robustness to random noise and producing real time response are the most important properties and neural networks provide us both [19]. Neural networks try to simulate human learning process. In other words, depending on the inputs, it models a system and learns to produce meaningful outputs after training stage. Because of training, neural networks are known as supervised approaches most of the time. However, there are unsupervised ones like Self-Organizing Maps (SOM) [6].

In [20], the feed-forward back-propagation multi-layer perceptron (MLP) is stated as the most used neural network method in the remote sensing literature.

In [21], SOM learning is incorporated in an unsupervised color image segmentation system. This system is composed of two steps. First, color levels of pixels in the image are projected to a selected set of prototypes using SOM learning, this step is called color reduction. Then, in second step called color clustering, using prototypes found in previous step, optimal clusters are searched by Simulated Annealing (SA).

[22, 23] make use of Pulse-Coupled Neural Network (PCNN) method. This method is an unsupervised neural network approach and embodies spatial neighborhood information [6]. [23] improves traditional PCNN algorithm that [22] uses, applies this method on IKONOS imagery for image segmentation and obtains superior results.

2.2.5 Multi-resolution Model

In the remote sensing field, every object has its own scale [3]. This model puts forward the claim that an object could be segmented best in its own scale.

There are two different strategies of multi-scale image segmentation [16]. Bottom-up approach starts with fine scale image and moves up to coarser levels. On the other hand, top-down approach starts with coarse scale image and goes to finer levels.

[16] employs a top-down multi-scale segmentation approach and uses two scale levels of an image obtained by wavelet decomposition with a Daubechies's wavelet. First, the coarse resolution image is segmented by clustering the sub-images obtained after wavelet decomposition. Then, segments found are projected to original image level and corresponding regions are again segmented in their own using a region growing approach.

In [25], a bottom-up multi-scale segmentation approach is employed using region merging. At the finest scale (original image), every pixel is considered as a different segment. This approach is the same with the one of commercial remote sensing image analysis software eCognition® by Trimble Navigation Ltd. [3]. Homogeneity criteria used in region merging step is also same with [3].

[13] applies a multi-resolution model based technique using wavelet decomposition. After decomposing the original image into selected scale level, approximation image is segmented by the watershed transform, then; this segmented image is projected to zero level.

2.2.6 Watershed Model

In this model, the image is first converted to the gradient image. Next, gradient image is interpreted as a topographical surface in which intensity values of pixels show elevations of corresponding points. Then, depending on the used watershed transform, this surface is filled with water that is either effusing out from lowest altitude [26] or pouring from highest altitude [27].

Watershed segmentation generally gives over-segmented results and because of this, it is not used alone in a general image segmentation system. An auxiliary method along with watershed segmentation is used either as a pre-processing step or as a post-processing step. There is a considerable texture in high spatial resolution satellite images. These high textured images make the over-segmentation problem worse. Noise is an increasing factor, too. As a solution, smoothing methods and/or region merging methods are generally used in literature as a pre-processing and/or post-processing, respectively.

For decreasing the over-segmentation effect of classical watershed approach, [12] operates on a floating point based rainfalling watershed algorithm [27]. Also, it applies a non-linear peer group filtering method to smooth the original image as a pre-processing step for the same purpose and a fast multiscale region merging method as a post-processing step.

[13] applies watershed segmentation to an approximation image of a lower resolution obtained by wavelet transform. Therefore, the low-pass filtering inherent in wavelet decomposition is a bit decreasing the extreme values in the gradient image. Also, it performs an adaptive thresholding operation on the gradient image.

Although watershed segmentation is a well-known and popular approach, it is relatively new in remote sensed image segmentation field compared to other models and it is promising when used along with multi-resolution model because it produces over-segmented results [6].

CHAPTER 3

BACKGROUND METHODS

In this part of the thesis, the algorithms found in the literature that are evaluated during the construction of proposed approach of this thesis are described in detail.

3.1 Pre-processing

The watershed segmentation algorithm that constitutes one of the major blocks in this thesis generally leads over-segmentation due to noise and high texture in the images. It is necessary to use one or more pre-processing steps to decrease noise and to smooth high textured regions in the image.

[12] states that the uniform application of filters such as median and Gaussian filters throughout the whole image is the main reason behind their erroneous results. Therefore, Chen et al. [12] and Feng et al. [31] use a non-linear peer group noise filter proposed by [30] as a pre-processing method before watershed segmentation. This method is detailed in Section 3.1.1. Besides peer group filtering, we also evaluate edge preserved smoothing filter. This algorithm is detailed in Section 3.1.2. Giving intensity gradient of the input image as input to the watershed algorithm is very common. Jing et al. [37] propose a different approach called homogeneity image (H-image) to overcome the shortcomings of intensity gradient image. These two different types of pre-processing method are detailed in Sections 3.1.3 and 3.1.4, respectively.

3.1.1 Peer Group Filter

This filter is a color image impulse noise reduction and smoothing filter. In this method, every pixel's corresponding peer group pixels within a window are found by calculating color distances. Then, pixel's color value is changed with a weighted average of color values of its peer group pixels. If the pixel is found to be an impulse noise, then the true peer group is found by using remaining pixels. Instead of averaging over all window pixels, incorporating a peer group is the source of success of this algorithm. Deng et al. [30] states that the algorithm is composed of two major parts which are classification of peer group pixels and replacement.

In the first step, peer group elements in a square window centered on the pixel which is at the image position n are classified. Thinking of the $x_0(n)$ as the color vector of the pixel, the color distances between this pixel and the other pixels in the window can be calculated by using Euclidean distance as in Equation 3-1.

$$d_i(n) = \|x_0(n) - x_i(n)\| \; where \; i = 0, \dots, w^2 - 1 \tag{3-1}$$

In Equation 3-1, w shows the width of the window and i shows the pixel numbers that are assigned in an order of ascending distance. The peer group of pixel n, $p(n)$, could be represented as in Equation 3-2.

$$p(n) = \{x_i(n)\} \; where \; i = 0, \dots, s(n) - 1 \tag{3-2}$$

In Equation 3-2, $s(n)$ represents the peer group size of the pixel positioned at n. It is necessary to find a proper $s(n)$ that is adaptive to local statistics for every pixel. For this purpose, Fisher's linear discriminant is incorporated into the algorithm. Fisher's linear discriminant tries to minimize intra-class distance while maximizing inter-class distance in a two-class situation. However, the pixels in the window could constitute more than two classes. By supposing peer group elements and non-peer group elements as constituting two classes, the problem at hand is converted to a two-class problem.

By making use of Euclidean color distances found, Fisher's criterion is calculated for every possible peer group size and the peer group size that gives maximum value is found to be as optimal $s(n)$ as shown in Equation 3-3.

$$s(n) = \operatorname{argmax}_i J(i) \tag{3-3}$$

In this equation, $J(i)$ shows the Fisher's criterion for $s(n) = i$ that can be formulated as in Equations 3-4, 3-5 and 3-6. In Fisher's criterion formula, nominator represents inter-class distance and denominator represents intra-class variance.

$$J(i) = \frac{|a_1(i) - a_2(i)|^2}{s_1^2(i) + s_2^2(i)}, \ where\ i = 1, \dots , w^2 \tag{3-4}$$

$$a_1(i) = \frac{1}{i}\sum_{j=0}^{i-1} d_j(n) \ and\ s_1^2(i) = \sum_{j=0}^{i-1}\left|d_j(n) - a_1(i)\right|^2 \tag{3-5}$$

$$a_2(i) = \frac{1}{w^2 - i}\sum_{j=i}^{w^2-1} d_j(n) \ and\ s_2^2(i) = \sum_{j=i}^{w^2-1}\left|d_j(n) - a_2(i)\right|^2 \tag{3-6}$$

Before the classification of peer group elements, pixels are analyzed to eliminate impulse noise effects by incorporating first order differences of color distances, $d_i(n)$, as formulated in Equation 3-7.

$$f_i(n) = d_{i+1}(n) - \ d_i(n) \tag{3-7}$$

To check for impulse noise existence within a window, first and last m (half of window size) $f_i(n)$s are compared with a pre-defined threshold t. If the color difference $f_k(n)$ at a sequence k exceeds the threshold t, the pixels that correspond to $i \leq k$ when k is in first m or $i > k$ when k is in last m are discarded and the remaining pixels are used for peer group estimation.

The second step of this algorithm is replacement of pixel color value. Using peer group $p(n)$, pixel color value at a position n is changed with the new value found with the Formula 3-8.

$$x_{new}(n) = \frac{\sum_{i=0}^{s(n)-1} w_i p_i(n)}{\sum_{i=0}^{s(n)-1} w_i} \tag{3-8}$$

In Formula 3-8, w_i is the standard Gaussian (mean 0 and standard deviation 1) weight of each peer group member i depending on its position in the window.

3.1.2 Edge Preserved Smoothing Filter

In edge preserved smoothing filter (EPSF), also called as adaptive mean filter, the smoothing amount of every pixel is determined using local statistics of the corresponding pixel as in peer group filtering approach. In this thesis, we evaluate the EPSF which is proposed by [32]. Nikolaou et al. [43] applies this method in the area of document image processing.

The format of a 3*3 EPSF can be seen in Equation 3-9.

$$EPSF = \frac{1}{\sum_{i=1}^{8} c_i}\begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & 0 & c_5 \\ c_6 & c_7 & c_8 \end{bmatrix} \tag{3-9}$$

To overcome the impulse noise, center pixel of filter is set to 0. The filter coefficients, c_i, are found using Equation 3-10.

$$c_i = (1 - d_i)^s \ where\ s \geq 1 \tag{3-10}$$

d_i in Equation 3-10 is the Manhattan color distance between corresponding pixel and the center pixel that can be calculated using Equation 3-11. With increasing color distance d_i between corresponding pixel and center pixel, corresponding filter coefficient c_i decreases. Therefore, the effect of corresponding pixel on the result decreases. s in Equation 3-10 is the scale factor that controls the amount of smoothing. With increasing s, the effect of low color distant pixels on the result increase while the effect of high color distant pixels decrease and hence, smoothing decreases.

$$d_i = \frac{\left|R_{a_c} - R_{a_i}\right| + \left|G_{a_c} - G_{a_i}\right| + \left|B_{a_c} - B_{a_i}\right|}{3*255} \tag{3-11}$$

Equation 3-11 is for the example case of 8-bit RGB color images. The denominator of this equation is for normalization purpose and d_i takes values between 0 and 1.

This filter can be extended to any window size and multi-spectral images.

3.1.3 Multi-spectral Image Gradient

Watershed transform interprets its input (generally an edge map) as a topographical relief and tries to find the local maximums (watershed lines) of this relief. Therefore, it is common to obtain gradient magnitude image of the input image before applying watershed transform. For a grayscale image I, gradient images in x and y directions and gradient magnitude image could be acquired as in Equations 3-12 and 3-13, respectively.

$$\nabla I = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix} = \begin{pmatrix} I_x \\ I_y \end{pmatrix} \tag{3-12}$$

$$I_m = \sqrt{I_x^2 + I_y^2} \tag{3-13}$$

In these formulas, I_x corresponds to the gradient image in the direction of x, I_y corresponds to the gradient image in the direction of y and I_m represents the gradient magnitude image. In grayscale images, gradient magnitude value of a point shows the strongest intensity change at that point.

Multi-spectral image gradient algorithm is always used with a Sobel filter in this thesis to find horizontal and vertical edges. Sobel filter is shown as in Equation 3-14.

$$Sobel\ Filter = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{3-14}$$

On the other hand, we can't directly apply gradient operator to the multi-spectral images, because gradient operator is applied to the scalar functions. In multi-spectral images, every pixel is a vector that contains spectral values of different bands. In [41], it is explained and proved that the gradient magnitude image of a multi-spectral image could be found from maximum eigenvalues of J^TJ matrix where J is the Jacobian matrix of the input image. Jacobian matrix for an m-band multi-spectral image I can be seen in Equation 3-15 and corresponding J^TJ matrix can be seen in Equation 3-16.

$$J = \begin{bmatrix} \frac{\partial I_1}{\partial x} & \frac{\partial I_1}{\partial y} \\ \vdots & \vdots \\ \frac{\partial I_m}{\partial x} & \frac{\partial I_m}{\partial y} \end{bmatrix} = \begin{bmatrix} I_{1x} & I_{1y} \\ \vdots & \vdots \\ I_{mx} & I_{my} \end{bmatrix} \tag{3-15}$$

$$J^TJ = \begin{bmatrix} ({I_{1x}}^2 \ + \ \dots \ + \ {I_{mx}}^2) & (\ I_{1x} * I_{1y} \ + \ \dots + \ I_{mx} * I_{my}) \\ (\ I_{1x} * I_{1y} \ + \ \dots + \ I_{mx} * I_{my}) & ({I_{1y}}^2 \ + \ \dots \ + \ {I_{my}}^2) \end{bmatrix} \tag{3-16}$$

For the special case of RGB images, J^TJ matrix looks like as in Equation 3-17.

$$J^TJ = \begin{bmatrix} ({R_x}^2 + {G_x}^2 + {B_x}^2) & (R_x * R_y + G_x * G_y + B_x * B_y) \\ (R_x * R_y + G_x * G_y + B_x * B_y) & ({R_y}^2 + {G_y}^2 + {B_y}^2) \end{bmatrix} \tag{3-17}$$

J^TJ matrix has two eigenvalues for every point in the input image and larger of those eigenvalues corresponds to the strongest spectral change and the square of gradient magnitude [41]. If we express J^TJ matrix as shown in Equation 3-18, maximum eigenvalue of this matrix can be calculated with the Equation 3-19.

$$J^TJ = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{3-18}$$

$$\lambda_{max} = \frac{(A_{11}+A_{22}) + \sqrt{(A_{11}+A_{22})^2 - 4*(A_{11}*A_{22} - A_{12}*A_{21})}}{2} \tag{3-19}$$

3.1.4 Homogeneity Image (H-image)

In this method [37], homogeneity level of local windows is extracted using a homogeneity criterion. These homogeneity levels constitute the H-Image. While a high value corresponds to a region boundary (less homogeneous), a low value corresponds to a region interior (more homogenous).

To extract the homogeneity level of a window, the spectral differences of individual pixels with respect to the central pixel are incorporated. If we call the center pixel as $p_c = (x_c, y_c)$ and another pixel in the window as $p_i = (x_i, y_i)$, we can define the vector directed from p_c to p_i as in Equation 3-20.

$$\boldsymbol{p_{c-i}} = (x_i - x_c, y_i - y_c) \tag{3-20}$$

If we add the spectral difference into consideration, we can define a new vector that reflects the spectral change in a specific direction independent from the spatial distance as in Equation 3-21.

$$\boldsymbol{d_i} = (I_i - I_c)\frac{\boldsymbol{p_{c-i}}}{\|\boldsymbol{p_{c-i}}\|} \tag{3-21}$$

In Equation 3-21, I_i shows the intensity value of pixel p_i and I_c shows the intensity value of center pixel, p_c. If we sum all $\boldsymbol{d_i}$**s** over a local window as shown in Equation 3-22, we can get an idea about the total spectral variation or texture in that window.

$$\boldsymbol{d} = \textstyle\sum_i \boldsymbol{d_i} \tag{3-22}$$

Finally, by changing the center pixel of the window with $\|\boldsymbol{d}\|$, we get the H value of that pixel. The image that is composed of H values is called H-image. In above formulas, we present the H-image computation for a gray-scale image. However, the n-band multi-spectral extension of this method is straightforward by combining H-images resulted from different spectral bands as shown in Equation 3-23.

$$H = \sqrt{H_1^2 + \cdots + H_n^2} \tag{3-23}$$

3.2 Watershed Segmentation

Watershed segmentation is a morphological edge-based image segmentation method. It has been used in various segmentation problems and it has multiple versions in the literature.

[35] divides watershed segmentation algorithms into two major groups which are flooding (immersion) simulation based and rainfalling simulation based by their approach to extract watershed lines. The most popular example of flooding type watershed segmentation algorithms is the one proposed by Vincent and Soille [26]. [9, 40] incorporate this algorithm in their method. [27, 35] which are proposed by De Smet et al. are the two examples of rainfalling watershed segmentation algorithms. In [35], it is stated that the algorithms of [27] and [35] are theoretically the same except for the additional connected component labeling step in [35] to merge individual pixels of plateau regions. However, the algorithm in [35] is shown to be faster than the one in [27].

As a result, we use and evaluate two prominent methods which are Vincent-Soille flooding watershed algorithm [26] and the fast rainfalling watershed algorithm [35] in this thesis.

3.2.1 Vincent-Soille Flooding Watershed Algorithm

In this algorithm, as for the other flooding based watershed algorithms, the progressive immersion of the topographic relief into the water is simulated. By the application of this immersion, relief starts to fill with water from the minimum altitude points. Based from the local minimums of the relief, ponds start to appear with the progressive flooding. When two or more ponds start to intersect, a dam (watershed) between them is generated. The pixels that belong to a specific pond are the catchment basin of the corresponding minimum. The dam pixels form the watershed lines. In Figure 3-1, the progressive flooding (from left to right) and watershed pixel creation mechanism of this algorithm is depicted on a topographical relief. Vertical lines correspond to watershed pixels in this figure.

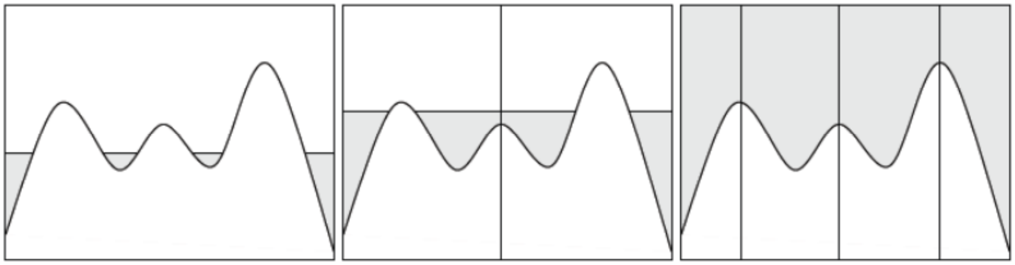


Figure 3-1 Progressive flooding and creation of watersheds in Vincent-Soille [35]

Looking from the implementation side, the algorithm contains two steps that are sorting and flooding. Sorting of image pixels with respect to their intensities is done first. Then, a hash table whose entries are intensity levels of the image and which returns the locations of pixels at the specified intensity level (entry) is formed by scanning the image. In the flooding step, by scanning intensity sorted image pixels and by considering neighborhood information of the same intensity level pixels, catchment basins are constructed.

3.2.2 Fast Rainfalling Watershed Algorithm by De Smet et al.

Topographical relief that is immersed with water in the case of flooding type watershed algorithms is exposed to the rain in the case of rainfalling type watershed algorithms. Rain drops that are accumulated at the same local minimum form a segment in this type of watershed algorithms. Therefore, steepest descent of a point (pixel) determines its segment. This phenomenon is shown in Figure 3-2 [35]. By using a drowning threshold, small local maximums that are corresponding to the weak edges could be eliminated. Instead of drowning threshold, relative drowning threshold which is drowning threshold divided by maximum value in the input image could also be applied.

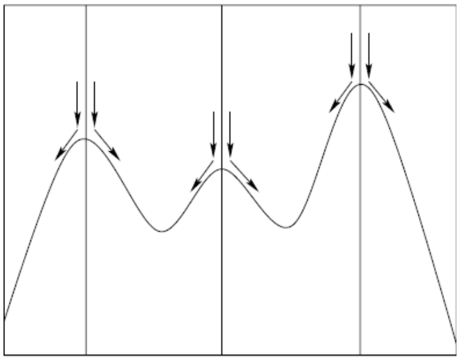


Figure 3-2 Rains follow the path of steepest descent [35]

Implementation of the rainfalling watershed algorithm proposed by [35], also used in this thesis, can be divided into five major steps as detailed below:

0. Initially, two data structures that represent label image which represents the respective segment of each pixel and local minima image which indicates the local minima pixels are created. While L denotes the label image, M denotes the local minima image. Label of every pixel is initialized with the video scanning order of it (left to right, top to bottom). M is initialized with all 1s. Also, drowning threshold, DT, and neighborhood size (4 or 8 for two-dimensional images), NS, are initialized.

1. In the first step, all pixels in the input image are visited in the order of left to right and top to bottom. At every pixel, steepest descent direction check is applied depending on DT. There are three different cases for each pixel in this step and these are shown in Equation 3-24. In this formula I shows the input image, i shows the current pixel and i_{SN} shows the label of the pixel that is neighbor to the current pixel in the direction of the steepest descent.

$$L(i) = \begin{cases} i, if\ I(i) < DT \\ i, if\ I(i) \geq DT\ and\ no\ steep.desc.(local\ min.) \\ i_{SN}, M(i) = 0, if\ I(i) \geq DT\ and\ a\ steep.desc. \end{cases} \quad (3\text{-}24)$$

2. The label image is traversed in the video scanning order to find the local minimum that corresponds to each pixel. At every pixel i, by tracking $L(i)$s (steepest descents) iteratively, the corresponding local minimum ($M(j)$ =1) is found and $L(i)$ is changed as $L(i) = j$. This step actually simulates the rainfalling behavior.

3. A connected component labeling algorithm with the neighborhood size given by DT is performed on the local minima image M. This step is added into this algorithm to remove a major shortcoming of over-segmented plateau regions in the method proposed by [27].

4. A once more traversal of the label image to carry the newly found local minima information in step 3 is performed. For every pixel i, $L(i)$ is updated as $M(L(i))$.

3.3 Multi-scale Segmentation

In a high spatial resolution satellite image, while cars, individual trees, narrow roads and small-sized buildings etc. could be classified as micro-objects, parking lots, forests, main roads and big-sized buildings etc. could be classified as macro-objects. According to [3], every object has its own scale in remote sensing images that it is best extracted. Keeping this in mind, we can say that micro-objects are best extracted in higher scales and macro-objects are best extracted in lower scales if the original image is accepted as the highest scale image. Macro-objects are more related with the aim of this study because we are interested in land cover types.

There are mainly two types of multi-scale segmentation algorithms in the literature which are wavelet based and region merging based. While wavelet-based algorithms can be labeled as coarse-to-fine type, region merging-based ones are fine-to-coarse type multi-scale segmentation algorithms. In general, wavelet based algorithms perform segmentation at a lower scale and project the segmentation result to the original image scale while trying to preserve object boundaries. In region merging based algorithms, regions found after an initial segmentation are iteratively merged trying to reach to the desired scale level. On the other side, wavelet-based algorithms are computationally less complex than region merging based algorithms at the possible expense of corrupted object boundaries.

In this thesis, we evaluate two wavelet based [13, 11] and two region merging based [12, 14] multi-scale segmentation algorithms for best results.

3.3.1 Multi-scale Segmentation based on Wavelet Decomposition

In this type of techniques, input image is analyzed at multiple resolutions by using wavelet decomposition. General flow of this type of algorithms can be seen in Figure 3-3. First, the segmentation scale at which segmentation algorithm is applied is selected. Then, by using wavelet transform (WT), the lower scale version of the original image is obtained and segmented. Finally, the segmented image is projected to the original image scale, mostly by using Inverse Wavelet Transform (IWT).

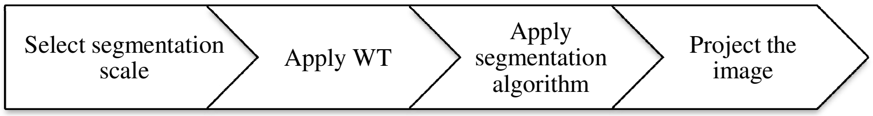


Figure 3-3 General flow of MSS algorithms that are based on wavelet decomposition

Wavelet decomposition which is proposed by [38], unlike other multi-resolution decomposition techniques like Gaussian and Laplacian pyramids, is a complete representation in that the original image can be reconstructed without a loss [11]. In Figure 3-4, wavelet decomposition process for a two-dimensional image can be seen.

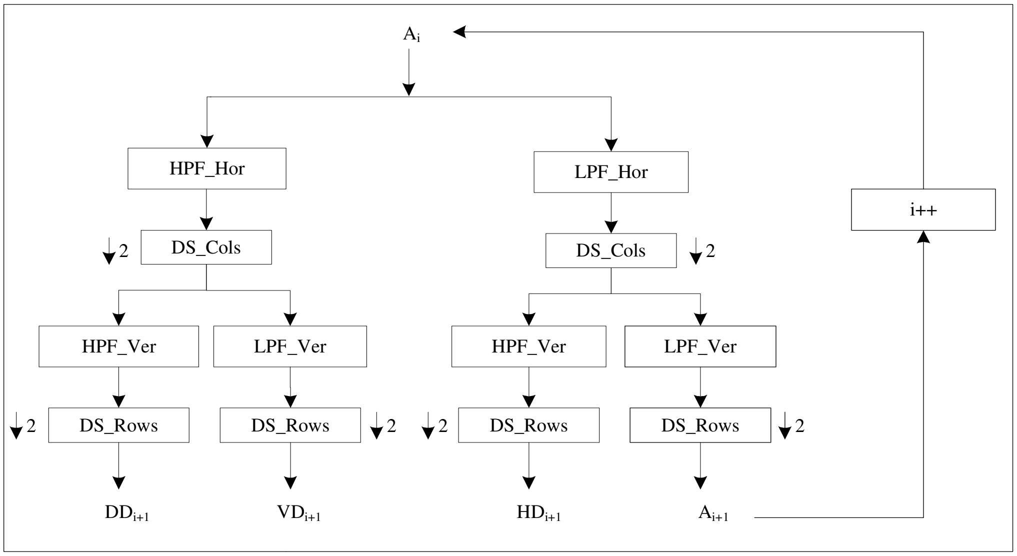


Figure 3-4 The flow of wavelet decomposition for a two-dimensional image

In Figure 3-4, the process starts with level $i = 0$ where A_0 corresponds to the original image. At each level i, the input image A_i is first convolved with low-pass (LPF_Hor) and high-pass (HPF_Hor) filters in horizontal direction and a column-wise down-sampling (DS_Cols) is applied to filtered images. Then, these images are convolved with low-pass (LPF_Ver) and high-pass (HPF_Ver) filters in vertical direction and a row-wise down-sampling (DS_Rows) is applied to filtered images. At the output of this process, there are four different images, DD_{i+1}, VD_{i+1}, HD_{i+1} and A_{i+1}, that are half-sized with respect to A_i. Each output image contains details (high frequency parts) of input image A_i in different directions. DD_{i+1} contains diagonal details, VD_{i+1} contains vertical details, HD_{i+1} contains horizontal details and A_{i+1} which is also called as the approximation image is the low-pass filtered version of the input image A_i and contains no details.

In wavelet decomposition, HPF_Hor and LPF_Hor are transposed versions of HPF_Ver and LPF_Ver, respectively and these filters are associated with a mother wavelet [38]. Both wavelet-based multi-scale segmentation algorithms [13, 11] that are evaluated in this thesis incorporate Haar wavelet as the mother wavelet. It is also a good choice considering the aims of this study because of its low computational complexity [13]. The high-pass and low-pass filter definitions associated with the Haar wavelet are shown in Equation 3-25.

$$HPF\_Hor = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \qquad LPF\_Hor = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{3-25}$$

[13] and [11] that go in this category of multi-scale segmentation techniques are evaluated in this thesis. They mainly differ in their application of projection to return back to the original image scale which is the last step of Figure 3-3.

3.3.1.1 Projection Method of Kim & Kim [11]

In [11], the coarsest scale image (segmentation level s is manually selected) is segmented by marker-controlled watershed algorithm. Then, a region merging algorithm that benefits from wavelet coefficients is performed on the segmented image. Finally, the region merged image M is projected as shown in Figure 3-5. We are only interested in the projection step of the algorithm of [11].

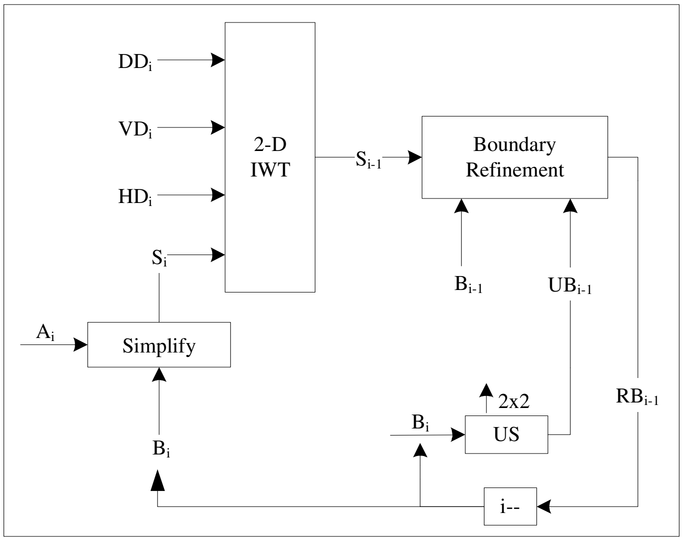


Figure 3-5 The flow diagram of projection algorithm proposed in [11]

As can be seen in Figure 3-5, the projection method is based on the application of 2-D IWT. It is an iterative approach where boundaries are refined at every iteration (scale level i). First, the region merged image A_i is simplified using the watershed line image B_i where $i = s$(segmentation scale) at start. B_i contains a different label for each region. Simplified image S_i is obtained by averaging the intensity values of A_i over the corresponding region in B_i and setting intensity value of each pixel in the corresponding region to this average. After this operation, except for watershed lines, S_i is piece-wise constant. For pixels corresponding to the watershed lines, the intensity values of A_i are directly copied to S_i. Then, a 2-D IWT is applied by incorporating the detail images DD_i, VD_i, HD_i and the simplified image S_i and the result is the finer scale simplified image S_{i-1}. Finally, by utilizing S_{i-1}, watershed line image of scale level $i-1$, B_{i-1}, which is the segmentation result of A_{i-1} (approximation image of scale level $i-1$) and upsampled watershed line image of scale level i, UB_{i-1}, boundaries and labels of the simplified image are refined as follows:

0. UB_{i-1} represents for possible labels of the image S_{i-1}. However, at the watershed lines which are not anymore one-pixel wide, there could be mis-segmented pixels.

1. Find the label that has maximum count in the portions of image UB_{i-1} corresponding to each segment of image B_{i-1}.

2. Assign the label found in step 1 to the corresponding region of B_{i-1} and obtain RB_{i-1}. RB_{i-1} is mainly an image whose boundaries are sourced from B_{i-1} and labels are sourced from UB_{i-1}.

3. RB_{i-1} has more segments than B_{i-1}, however, the label count is same. Adjacent regions that have same label could be merged.

After refinement of boundaries, this whole process presented above is iterated as refined labels image RB_{i-1} is the new labels image.

3.3.1.2 Projection Method of Jung [17]

In [17], after manual selection of segmentation scale s, the approximation image at that scale, A_s, is obtained using Haar wavelet decomposition. Then, the gradient magnitude image of A_s which is acquired using Prewitt edge detector is exposed to an adaptive thresholding algorithm proposed by [17]. Finally, watershed transform is applied to the thresholded gradient magnitude image and segmented image is projected to the original scale using the algorithm shown in Figure 3-6.

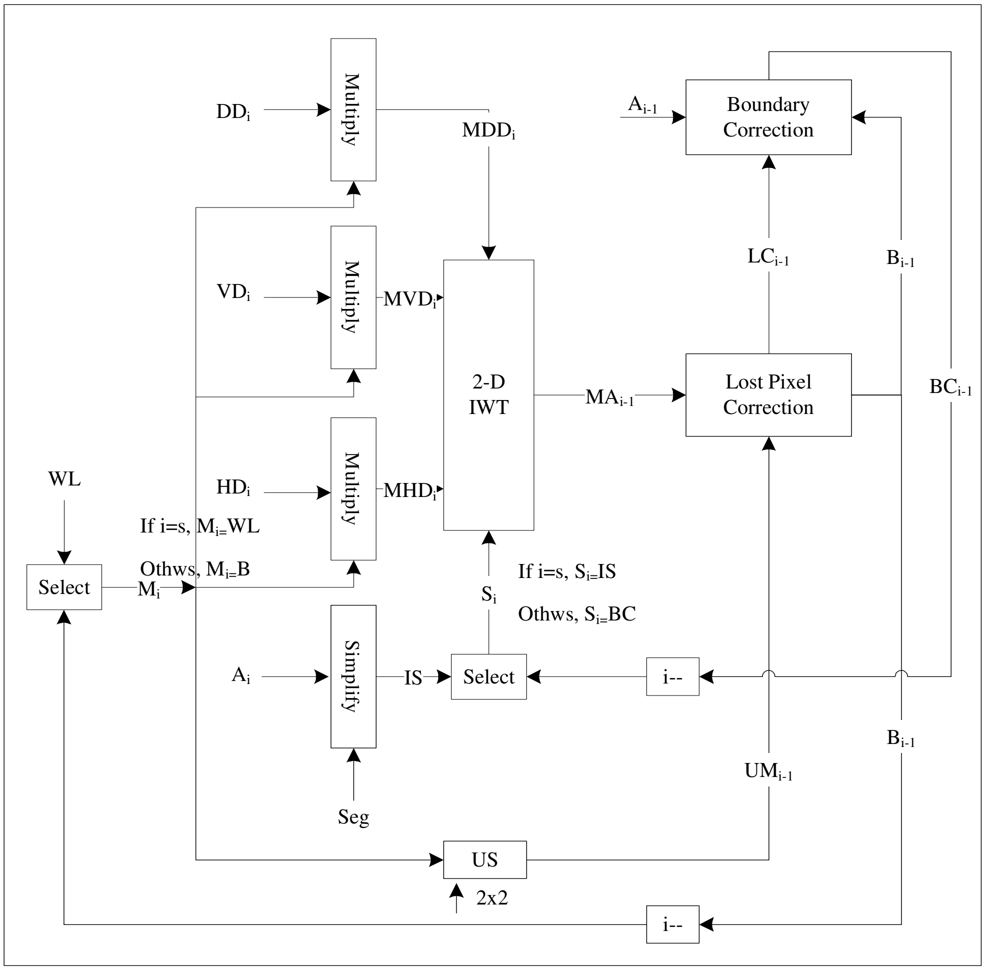


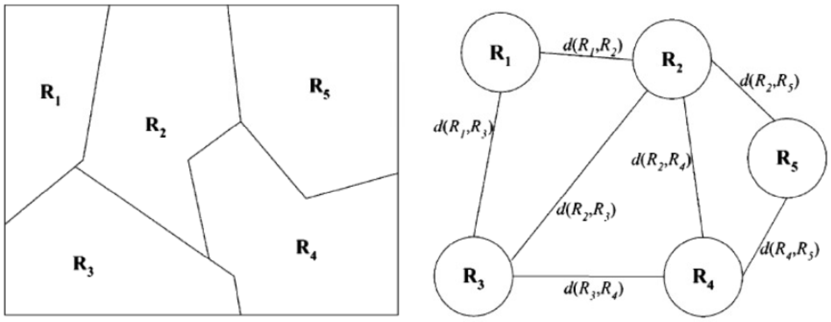
Figure 3-6 The flow diagram of projection algorithm proposed in [17]

In this projection method, the approximation image is tried to be projected to the original resolution iteratively using IWT. At every level of wavelet reconstruction, boundaries of objects (which are possible to be corrupted during IWT) are corrected using corresponding level's approximation image. Firstly, the approximation image A_i at the segmentation scale ($i = s$) is simplified using the segmentation result Seg. The same approach used in projection method of Kim&Kim [11] is followed for obtaining the simplified result IS from A_i and Seg. DD_i, VD_i, HD_i and IS are incorporated to obtain one upper scale approximation image by using 2-D IWT. However, to prevent filtered details throughout the region interiors to be involved back in, detail images are multiplied with a mask image WL. WL is a binary image that takes value of 1 on watershed lines and 0 otherwise. A modified approximation image at level $i - 1$, MA_{i-1}, that is piece-wise constant except for the upsampled watershed lines UM_{i-1}. These pixels are called lost pixels and we can be sure that their locations are defined by UM_{i-1} because the support of Haar wavelet is 2 and every detail coefficient influences a 2*2 window in the upper scale image [17]. In lost pixel correction step of Figure 3-6, every lost pixel's neighbors in an 8-connected manner are searched and lost pixel's intensity value is updated with the one that minimizes intensity difference. The output of this step, LC_{i-1}, is a true piece-wise constant image. Finally, to retain the true boundary information, the pixels that are belong to the object boundaries (B_{i-1}) are updated with the corresponding ones in true approximation image A_{i-1} in the boundary correction step. Now, simplified and corrected result of the level $i-1$, BC_{i-1}, is at hand. To arrive at the original image resolution, this whole process should be repeated until $i=0$ as shown in Figure 3-6.

3.3.2 Multi-scale Segmentation based on Region Merging

We use and evaluate two different region merging algorithms in this thesis. They mainly differ in their order of merging and stopping criterion.

In both region merging algorithms, a region adjacency graph (RAG) is generated to hold the segmentation information at any scale. This graph holds the neighborhood information of segments found after watershed segmentation and is mainly composed of nodes, N, edges, E, and distances, D. An example segmentation and the corresponding RAG is shown in Figure 3-7. Nodes, N, correspond to segments in our case. Edges, E, are present between adjacent segments (nodes) and distances, D, represent the distances between adjacent nodes. In our case, a distance corresponds to the merging cost of two adjacent segments and is calculated by the Equations 3-26 to 3-30 for both of the algorithms.



(a) (b)

Figure 3-7 An example segmentation and its RAG (a) Segmentation with region labels (b) RAG corresponding to (a) [11]

The merging cost that these two algorithms incorporate is the one which is also used in eCognition® software [3] and its formulation is as follows:

$$f = w_{spectral}h_{spectral} + (1 - w_{spectral})h_{shape} \tag{3-26}$$

$$h_{spectral} = \sum_i w_i(n_m\sigma_{m,i} - (n_1\sigma_{1,i} + n_2\sigma_{2,i})) \tag{3-27}$$

$$h_{shape} = w_{compactness}h_{compactness} + (1 - w_{compactness})h_{smoothness} \tag{3-28}$$

$$h_{smoothness} = n_m\frac{l_m}{b_m} - (n_1\frac{l_1}{b_1} + n_2\frac{l_2}{b_2}) \tag{3-29}$$

$$h_{compactness} = n_m\frac{l_m}{\sqrt{n_m}} - (n_1\frac{l_1}{\sqrt{n_1}} + n_2\frac{l_2}{\sqrt{n_2}}) \tag{3-30}$$

The merging cost, f, is a mixed score that takes both spectral heterogeneity change, $h_{spectral}$, and shape heterogeneity change, h_{shape}, into account in a region merge operation. The parameters of the Equations 3-26 to 3-30 are as follows:

- In Equation 3-26, $f$is the merging cost of two regions, region 1 and region 2. $h_{spectral}$ and h_{shape} are the spectral heterogeneity change and the shape heterogeneity change resulting from the merge operation, respectively. $w_{spectral}$ is the weight of $h_{spectral}$ when combining $h_{spectral}$ and h_{shape} for the merging cost calculation.

- In Equation 3-27, the difference of the standard deviation of merged region and the total standard deviation of constituting regions for each spectral band, i, are summed for the calculation of $h_{spectral}$. w_i shows the weight of band i. n_m, n_1 and n_2 represent the areas of merged region, region 1 and region 2, respectively. $\sigma_{m,i}$, $\sigma_{1,i}$ and $\sigma_{2,i}$ are the standard deviations of merged region, region 1 and region 2, respectively, for band i.
- In Equation 3-28, two shape heterogeneity change metrics are combined by linear weighting to form h_{shape}. $h_{compactness}$ and $h_{smoothness}$ are the segment compactness and boundary smoothness changes resulting from the merge operation, respectively. $w_{compactness}$ is the weight of $h_{compactness}$ when combining $h_{compactness}$ and $h_{smoothness}$ for the h_{shape} calculation.
- In Equations 3-29 and 3-30, n_m, n_1 and n_2 represent the areas of merged region, region 1 and region 2, respectively. l_m, l_1 and l_2 show the perimeters of merged region, region 1 and region 2, respectively. b_m, b_1 and b_2 show bounding box perimeters of merged region, region 1 and region 2, respectively.

3.3.2.1 First Algorithm [12]

After the generation of RAG for a watershed segmented image, first algorithm [12] searches the nodes (segments) which have the minimum area throughout the image and after that, it examines the edges connected to these regions to find the one that has minimum distance. Then, it merges the regions that are at a distance of that minimum to each other and updates its data structures. This procedure iterates until the minimum segment area in the image exceeds the size threshold. The flowchart of this algorithm can be seen in Figure 3-8. In this flowchart, C_{min} shows the current minimum segment area in the image and $SizeThr$ is the stopping criterion of the algorithm. As a result, it can be said that this algorithm merges the most similar segments that have areas smaller than $SizeThr$.

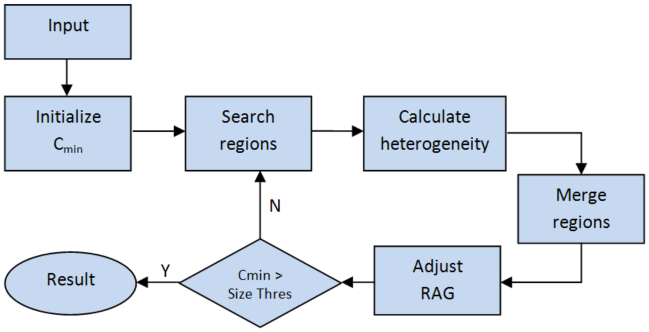


Figure 3-8 The flow chart of the first region merging based multi-scale segmentation algorithm [12]

3.3.2.2 Second Algorithm [14]

On the other hand, second algorithm [14] does not consider segment sizes as in the first algorithm and searches for the edge that has minimum distance throughout the image. Then, it merges the corresponding regions that are at a distance of that minimum to each other and updates its data structures. This procedure iterates until the minimum merging cost in the image exceeds a threshold. The flowchart of this algorithm can be seen in Figure 3-9. In this flowchart, $Scale\ threshold$ is the stopping criterion of the algorithm. As a result, it can be said that this algorithm merges the most similar segments that have similarity smaller than $Scale\ threshold$.

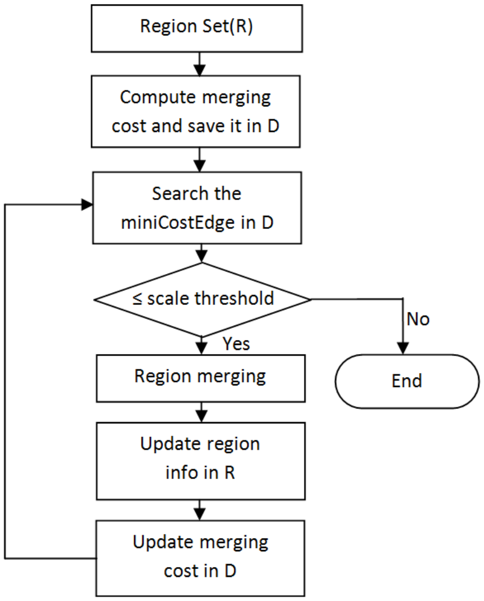


Figure 3-9 The flow chart of the second region merging based multi-scale segmentation algorithm [14]

$$v(x, y) = max \tag{3-31}$$

3.4 Segmentation Accuracy and Performance Measures

Segmentation accuracy can be assessed by qualitatively (visual analysis) or quantitatively. Qualitative assessment methods need an expert look at the segmentation results to decide the truthfulness of outputs. Quantitative assessment methods (empirical methods as called in [4]), on the other hand, evaluates segmentation results by using some statistical measurements. Quantitative assessment methods can be divided into two categories as supervised and unsupervised. In supervised methods, a ground truth image obtained at a pre-segmentation step is used as reference to evaluate the segmentation accuracy. In unsupervised methods, measures that simulate human approach of evaluation of segmentation correctness such as correct segment boundaries, intra-segment homogeneousness etc. could be used to assess segmentation accuracy. Carleer et al. [4] name supervised category of quantitative analysis as empirical discrepancy methods and unsupervised category as empirical goodness methods and prefer to use empirical discrepancy methods in their case. Kim and Kim [11], Sun et al. [2] and Johnson et al. [39], on the other hand, evaluate their segmentation results by performing empirical goodness methods. We are using both empirical discrepancy and empirical goodness methods in this thesis.

3.4.1 Unsupervised Measures

Empirical goodness function that is proposed by [33] and incorporated in [11] can be seen in Formula 3-32. The lower the output of this function, the better the segmentation accuracy is.

$$F(I) = \sqrt{N} \times (\textstyle\sum_{k=1}^{N} \frac{D_k^2}{\sqrt{A_k}}) \times \frac{1}{S} \tag{3-32}$$

In Formula 3-32:

- I is the input image.
- N is the segment count obtained after segmentation.
- D_k is the sum of Euclidean distances between spectral vectors of pixels of kth region of segmented image and corresponding region of simplified image. Simplified image, as

discussed in Section 3.3.1.1, is obtained by averaging the intensity values of input image over each segment and setting intensity value of each pixel in a segment to intensity average of that segment.

- A_k is the pixel count or area of kth region.
- S is the size of the input image I.

This function evaluates segmentation results in local and global level at the same time. $\sqrt{N}$ punishes global over-segmentation, D_k^2 and $\sqrt{A_k}$ encourage intra-segment homogeneity and big-sized regions, respectively. $\frac{1}{S}$ term is here for normalization.

Kim and Kim [11] apply measures such as the segment count, PSNR (Peak Signal to Noise Ratio) and computation time beside the goodness measure. PSNR is a parameter calculated between two images and is mainly used in image compression field as a compression efficiency measure. If it is high, the loss in compression is low and it means that the similarity between the compressed image and the original image is high. After converting segmented image to simplified version, PSNR value can be used as an accuracy measure of segmentation. The Mean Square Error (MSE) that is an inversely proportional parameter with PSNR can be formulated as in Formula 3-33.

$$MSE = \frac{1}{rc}\left(\sum_{x=0}^{r-1}\sum_{y=0}^{c-1}|I(x,y) - I_s(x,y)|^2\right) \tag{3-33}$$

In Formula 3-33:

- r and c show row and column count of the image, respectively.
- I and I_s represent original and simplified image, respectively.

In the formula of MSE, $|I(x,y) - I_s(x,y)|$ difference is an absolute intensity difference in the case of grayscale images and a Euclidean distance of color vectors in the case of multispectral images. MSE is a quantitative representation of total squared error between two images. PSNR, on the other hand, is an indicator of peak error in decibel units. Formula 3-34 represents PSNR equation.

$$PSNR = 10\log_{10}\left(\frac{F^2}{MSE}\right) \tag{3-34}$$

In Formula 3-34, F stands for the maximum achievable difference that can be obtained by using $|I(x,y) - I_s(x,y)|$. For example, it is 255 in the case of 8-bit gray-scale images.

Sun et al. [2] incorporate spatial autocorrelation analysis and intra-segment variance for the selection of optimum segmentation scale in their multiscale segmentation method. In this thesis, they are used as a segmentation accuracy evaluation method. Spatial autocorrelation analysis supplies information about the intersegment autocorrelation or seperability of segments. To obtain intersegment autocorrelation, [2] handles Moran's I index which is proposed by [34]. Its formula can be seen in Formula 3-35.

$$I = \frac{N}{\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}}\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}w_{ij}(X_i-\bar{X})(X_j-\bar{X})}{\sum_{i=1}^{N}(X_i-\bar{X})^2} \tag{3-35}$$

In Formula 3-35:

- N is the segment count.
- w_{ij} shows the spatial weight of ijth region pair. In [2], it is used as an adjacency measure. If regions are adjacent w_{ij} is 1, otherwise 0. When i and j are equal, w_{ij} is 0.
- X represents the selected feature of segments. In [2], the spectral means of segments are selected. X_k and $\bar{X}$ represents the mean color of region k and the mean color of whole image, respectively. $X_i - \bar{X}$ difference is calculated by simple scalar difference in the case

of grayscale images and by Euclidean distance of color vectors in the case of multispectral images.

Moran's I index represents the level of separation between adjacent segments' means. A high Moran's I index value corresponds to high spatial autocorrelation, in another say, a bad segmentation.

[2] also makes use of a measure that reveals intrasegment variance or homogeneity. The formulation of this measure can be seen in Formula 3-36.

$$V = \frac{\sum_{k=1}^{N}(A_k v_k)}{\sum_{k=1}^{N} A_k} \qquad (3\text{-}36)$$

In Formula 3-36:

- A_k shows the area of kth segment.
- v_k shows the variance of kth segment.

The intra-segment variance, V, formulated as above is the average of variances of each segment that are weighted with segment sizes. By this way, the effects of small-sized segments on the result are decreased and a more stable result is obtained. By the way, a better segmentation result produces lower intrasegment variance.

In [2], it is stated that while Moran's I index decreases, intra-segment variance increases with increasing segmentation scale. Therefore, the optimum segmentation scale is found from a set of different scale values by combining these two measures.

Johnson et al. [39] also incorporate Moran's I and intra-segment variance for the selection of optimum segmentation scale in their multiscale segmentation method as with Sun et al. [2]. In [39], after Moran's I and intra-segment variance are calculated independently for all spectral bands by employing the previous equations, a goodness measure that incorporates both is proposed. First, to evaluate both Moran's I and intra-segment variance equally, they are normalized to 0-1 range as in Equation 3-37.

$$X_{normalized,b,i} = \frac{X_{b,i} - X_{min,b}}{X_{max,b} - X_{min,b}} \qquad (3\text{-}37)$$

In Equation 3-37, min and max values of corresponding measure X (Moran's I or intra-segment variance) are found from a set of values (for different is) as $X_{min,b}$ and $X_{max,b}$, respectively where b stands for spectral band. This set of values in the case of [39] is the set of measurements found after region merging segmentation with different scale thresholds i. Then, to be able to benefit from both measures, they are combined in the way as seen in Equation 3-38 and the result is called as Global Score ($GS_{b,i}$) for spectral band b and scale threshold i.

$$GS_{b,i} = V_{normalized,b,i} + MI_{normalized,b,i} \qquad (3\text{-}38)$$

In Equation 3-38, $V_{normalized,b,i}$ shows the normalized value of intra-segment variance and $MI_{normalized,b,i}$ shows the normalized value of Moran's I for spectral band b and scale threshold i. Finally, global scores of each band are averaged as shown in Equation 3-39 and an average global score that collects information of different spectral bands, GS_i, is formed for scale threshold i.

$$GS_i = \frac{1}{Nb} \times \sum_{b=1}^{Nb}(GS_{b,i}) \qquad (3\text{-}39)$$

After finding GS_i for every scale level i, Johnson et al. [39] choose the i that has minimum GS_i as the best segmentation scale. Because, low Moran's I and intra-segment variance values that are properties of a good segmentation produce lower global scores.

Chabrier et al.[42] also propose an unsupervised segmentation evaluation metric called Zeboudj's contrast (Zeb contrast) based on the internal and external contrasts of the regions calculated around the pixels. Chabrier et al. define the internal contrast (I_i) and external contrast (E_i) of a region i (R_i) as shown in Equations 3-40 and 3-41.

$$I_i = \frac{1}{A_i} \times \sum_{s \in R_i} \max \{c(s,t), t \in W(s) \cap R_i\} \tag{3-40}$$

$$E_i = \frac{1}{l_i} \times \sum_{s \in F_i} \max \{c(s,t), t \in W(s), t \notin R_i\} \tag{3-41}$$

In these equations, $W(s)$ shows the neighborhood of a pixel s. A_i and l_i show the area and perimeter length of R_i, respectively. F_i corresponds to the pixels at the border of R_i. c represents the contrast operator for corresponding pixels.

By using I_i and E_i, the contrast of R_i is calculated as in Equation 3-42. Finally, the Zeb contrast of a segmented image is calculated as in Equation 3-43.

$$C(R_i) = \begin{cases} 1 - \frac{I_i}{E_i}, if\ 0 < I_i < E_i \\ E_i, if\ I_i = 0 \\ 0, otherwise \end{cases} \tag{3-42}$$

$$C_z = \frac{1}{A} \times \sum_i A_i C(R_i) \tag{3-43}$$

Zeb contrast metric measures the homogeneity of segment interiors by the internal contrast (I_i) and the heterogeneity between segments by the external contrast (E_i). Finally, higher values of Zeb contrast correspond to better segmentation results.

Another unsupervised segmentation evaluation metric proposed by Zhang et al. [43] is based on the entropy. Entropy of a segment represents the level of disorder within the corresponding segment. The entropy of a region R_j is formulated as in Equation 3-44.

$$H(R_j) = -\sum_{m \in V_j} \frac{L_j(m)}{S_j} \log \frac{L_j(m)}{S_j} \tag{3-44}$$

In this equation, m shows a specific luminance value throughout all possible luminance values (V_j) in the region j. $L_j(m)$ represents the count of pixels that have luminance value m in region j and S_j shows the total pixel count (area) of region j. By using the entropy of a region found before, the expected region entropy of an image I can be formulated as in Equation 3-45.

$$H_r(I) = \sum_{j=1}^{N} \left(\frac{S_j}{S_I}\right) H(R_j) \tag{3-45}$$

Expected region entropy serves as a measure of intra-segment homogeneity of a segmentation. Low expected region entropy corresponds to better intra-segment uniformness. Because expected region entropy has a bias towards over-segmentation, a balancing factor that is called as layout entropy is proposed by Zhang et al. [43] whose formulation is in Equation 3-46.

$$H_l(I) = -\sum_{j=1}^{N} \frac{S_j}{S_I} \log \frac{S_j}{S_I} \tag{3-46}$$

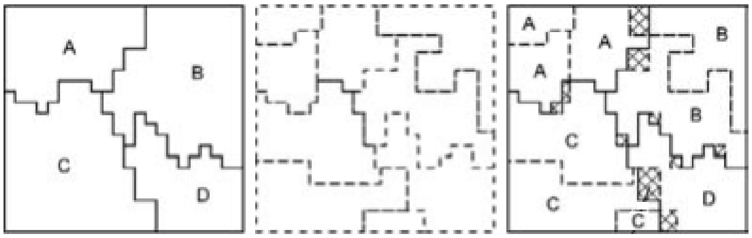
Finally, these two entropy evaluations are combined as in Equation 3-47 for achieving an optimum segmentation result. Because one term ($H_r(I)$) has a bias towards over-segmentation and the other term has a bias towards under-segmentation ($H_l(I)$), E produce optimum values when these two terms are balanced as in the formulation. This formulation of total entropy produces lower values for better segmentation results.

$$E = \; H_r(I) + H_l(I) \tag{3-47}$$

In this thesis, F, GS, Zeb contrast and E metrics are used during the performance comparison of the proposed approach with the similar approaches in the literature by using the test images that do not have ground truths. Also, we use and evaluate F and GS metrics as automatic parameter selection mechanisms for our proposed segmentation approach and call them Goodness1 (G1) and Goodness2 (G2). In [2], [11] and [39], these metrics are used for the determination of best segmentation scale or resolution. However, we evaluate the usefulness of these parameters when determining all parameters of the algorithms that are included in the proposed segmentation approach.

3.4.2 Supervised Measures

Supervised accuracy evaluation metrics produce results by comparing segmentation results with manually segmented results (ground truths). An example of this evaluation is shown in Figure 3-10. In Figure 3-10c, wrongly segmented pixels are shown as shaded.



(a) (b) (c)

Figure 3-10 Example of a supervised segmentation evaluation (a) true segmentation result, (b) found segmentation result, (c) mis-segmented pixels [4]

We apply supervised segmentation accuracy evaluation metrics proposed by [4] as shown in Equations 3-48 and 3-49.

$$Ev1 = \frac{\left[\left(\sum_{i=1}^{N_{ref}} \sum_{j=1}^{N_{ref}} c_{ij}\right) - \sum_{k=1}^{N_{ref}} c_{kk}\right]}{\sum_{i=1}^{N_{ref}} \sum_{j=1}^{N_{ref}} c_{ij}} \times 100 \tag{3-48}$$

$$Ev2 = \frac{\sum_{j=1}^{N_{ref}} \left(\frac{\left(\sum_{i=1}^{N_{ref}} c_{ij}\right) - c_{jj}}{n_{jref}} \times 100 \right)}{N_{ref}} \tag{3-49}$$

In Equations 3-48 and 3-49, N_{ref} and n_{jref} are the true segment count of the image and the pixel count of true segment j found from ground truth, respectively and c_{ij} is the ijth element of the confusion matrix C.

An example of a confusion matrix for four true segments case can be seen in Table 3-1. Confusion matrix has a size of $N_{ref} \times N_{ref}$ where N_{ref} is the true segment count. An element of a confusion matrix, N_{XY}, shows the number of pixels that have true segment label Y and found segment label X. Before generating confusion matrix of a found segmentation result, a conversion is applied to the segmentation result such that its segment count and segment labels are the same with its ground truth. Due to the difficulty of having an ideal segmentation result, found segment count mostly differs from the true segment count. Therefore, the label of the true segment that is included most by a found segment becomes the new label of the corresponding found segment. As a result of this approach, it is obvious that the segmentation error will only be sourced from the segments that include the boundaries of true segments.

Table 3-1 Confusion matrix for a segmentation having true segment count of 4

	True Segment Label			
Found Segment Label	N_{AA}	N_{AB}	N_{AC}	N_{AD}
	N_{BA}	N_{BB}	N_{BC}	N_{BD}
	N_{CA}	N_{CB}	N_{CC}	N_{CD}
	N_{DA}	N_{DB}	N_{DC}	N_{DD}

In the light of above discussions, it can be easily said that $Ev1$ is a global measure of pixel segmentation error throughout the whole image. However, it is insufficient for providing an object-based segmentation error. For this reason, [4] proposes $Ev2$ measure which is simply an average of pixel segmentation errors of all segments in the image.

CHAPTER 4

PROPOSED APPROACH

In this chapter, the proposed method of this thesis for segmentation of high spatial resolution satellite images is expressed in detail. In the following sections, a detailed and comparative analysis about the reason behind choosing the corresponding method used in each step of the algorithm and the effects on the result could be found. The steps of proposed segmentation method in this thesis can be seen in Figure 4-1. The rest of this chapter will contain these parts in detail.

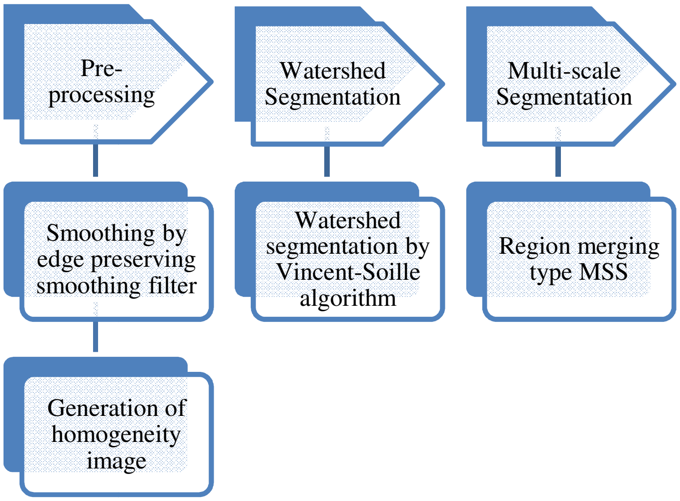

Pre-processing
Watershed Segmentation
Multi-scale Segmentation
Smoothing by edge preserving smoothing filter
Watershed segmentation by Vincent-Soille algorithm
Region merging type MSS
Generation of homogeneity image

Figure 4-1 Steps of proposed segmentation method in this thesis

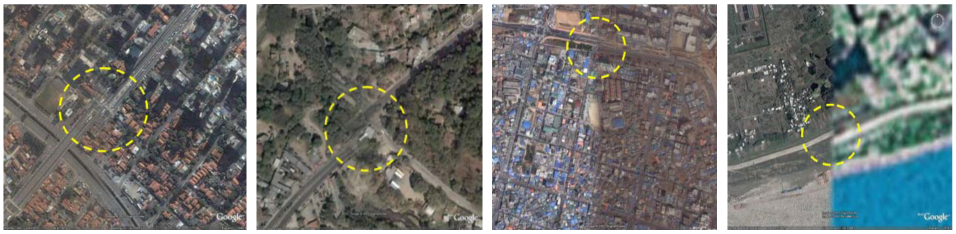
In Section 4.1, the way we obtain satellite imagery from a specific satellite at a specific resolution and at a specific time using Google Earth® software of Google Inc. is explained in detail. In Section 4.2, the watershed-based region merging type MSS (multi-scale segmentation) approach that we propose is described. In Section 4.6, the unsupervised segmentation accuracy evaluation metric for parameter selection of the algorithm that we propose is described.

4.1 Input Image Selection

Today, high spatial resolution satellite images are abundant with many satellites and many sensors on these satellites; however, they are not easily accessible by research community. Images taken from satellites are at a price. DigitalGlobe Inc. which is the owner of QuickBird and WorldView 1-2 satellites and GeoEye Inc. which is the owner of IKONOS, OrbView 2-3 and GeoEye 1-2 satellites are the major vendors in high spatial resolution satellite imagery industry.

In year 2005, freely available software Google Earth® created by Google Inc. started to provide satellite imagery at no cost but reduced spatial and spectral resolutions and without the infrared band [29]. Google Earth® database is composed of images taken at different times and supplied from many different satellite and aerial image vendors including but not limited to DigitalGlobe Inc. and GeoEye Inc. Because of this multi-source and multi-temporal environment, Google Earth® obviously applies geo-registration to images for unifying them. In this way, Google Earth® allows analysis of everywhere on the earth in a continuity. However, this process could not give best results all the time. Examples of geo-registration errors found in Google Earth® that are taken from [29] can be seen in Figure 4-2.

This type of errors is easily discernible by following linear objects like roads. While geo-registration problems seen in Figure 4-2a-b-c are owed to multi-temporal images taken from same satellite, the one in Figure 4-2d is because of two different satellites (The left half is taken with QuickBird and the right half is taken with Landsat) [29].



(a) (b) (c) (d)

Figure 4-2 Examples of Google Earth® geo-registration errors [29]

Considering deficiencies of Google Earth® stated above, a scene that includes image taken from only one satellite and at only one time should be searched. Actually, with enhanced database of Google Earth® from 2005 till today, it is not a hard task to find a scene that meets these criteria. By enough zooming and positioning, we could easily obtain a scene that is taken from only one satellite at only one time. However, the necessary zoom level (Google Earth® calls this as "Eye Alt") corresponding to the spatial resolution of the scene is still an unknown. Although vendor and time tag of source image are available in Google Earth®, Google Inc. chooses not to share spatial and spectral resolution of the source images [29].

Finally, by encountering "GeoEye Featured Image Gallery" of Google Earth®, we are also becoming able to get rid of spatial resolution uncertainty. Selected set of interesting places on earth are included in "GeoEye Featured Image Gallery". These places are not abundant but the set is continually growing. The scenes in this gallery carry satellite source, time tag and spatial resolution information.

4.2 Segmentation Model Selection

To solve remote sensing image segmentation problem, numerous segmentation methods are employed in the literature. These methods are grouped into different classes from two different perspectives in Chapter 2. In the following paragraphs, we try to decide the best segmentation model considering the design goals stated in Chapter 1.

From the operation level perspective, there are four categories of segmentation algorithms which are point-based, edge-based, region-based and hybrid algorithms. Point-based approaches are applied to an image globally by discarding the local information and they need supervision about the segment count of the image beforehand. Therefore, edge-based and region-based methods are more preferable over point-based methods. However, edge-based segmentation methods are very susceptible to noise because of their working principle that is based on intensity gradient and region-based segmentation methods, especially splitting techniques, produce corrupted (rectangular-like) boundaries. Hence, a hybrid approach that combines edge-based and region-based techniques to overcome their individual deficiencies is necessary.

From the image analysis approach perspective, there are six types of segmentation algorithms which are object-background model, MRF model, fuzzy model, neural model, multi-resolution model and watershed model. Object-background model is actually corresponding to point-based approaches in the previous perspective. MRF model incorporates an assumption that the observed image is the corrupted version of the true image and tries to estimate the true image using optimization techniques. However, it suffers from extensive computation times and has the disadvantage of the necessity of supervision about the true image. Fuzzy model performs fuzzy clustering for segmentation. FCM is the most known and applied fuzzy clustering method [19], however; its computational burden and supervised nature (cluster count information) makes it a bad choice for us. Neural networks (neural model) try to model a system simulating human learning activity. Because of the required training stage, they contradict our design goals. Multi-resolution models (multi-scale) try to provide the segmentation of objects in their intrinsic scale. Because every object has its own scale in remote sensing images [3], the inclusion of multi-resolution models into proposed method of this thesis may be reasonable. Watershed models tackle the segmentation problem using watershed algorithm. Watershed algorithm is an unsupervised and fast approach. This algorithm is an edge-based segmentation method and therefore, it produces over-segmented results. However, it is generally used along with other segmentation models.

By the light of above comparisons, it appears that we need a hybrid (edge and region-based) and a multi-scale segmentation approach. Combining watershed and multi-scale models would be a good selection towards satisfying our design goals. However, the over-segmentation problem caused from watershed model should be handled with pre-processing or post-processing approaches.

As a result, the flow that we will be following in the rest of this section can be seen in Figure 4-3. This figure shows the construction phases of our watershed-based multi-scale segmentation approach. At every phase, we compare two or more methods for that step in qualitative and quantitative manners. Quantitative comparison uses the accuracy and performance measures that are introduced in Section 3.4. After the evaluation of comparison results, the best method for that phase is selected. By using this step-wise optimal approach, we find a sub-optimal set of algorithms that constitute our proposed segmentation method.

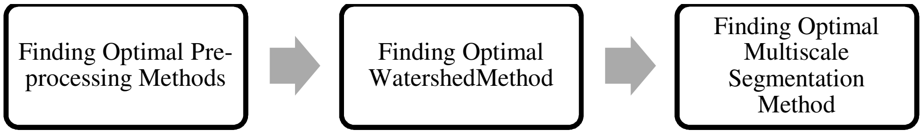

Finding Optimal Pre-processing Methods
Finding Optimal WatershedMethod
Finding Optimal Multiscale Segmentation Method

Figure 4-3 Phases of the construction of the proposed algorithm of this thesis

4.3 Pre-processing

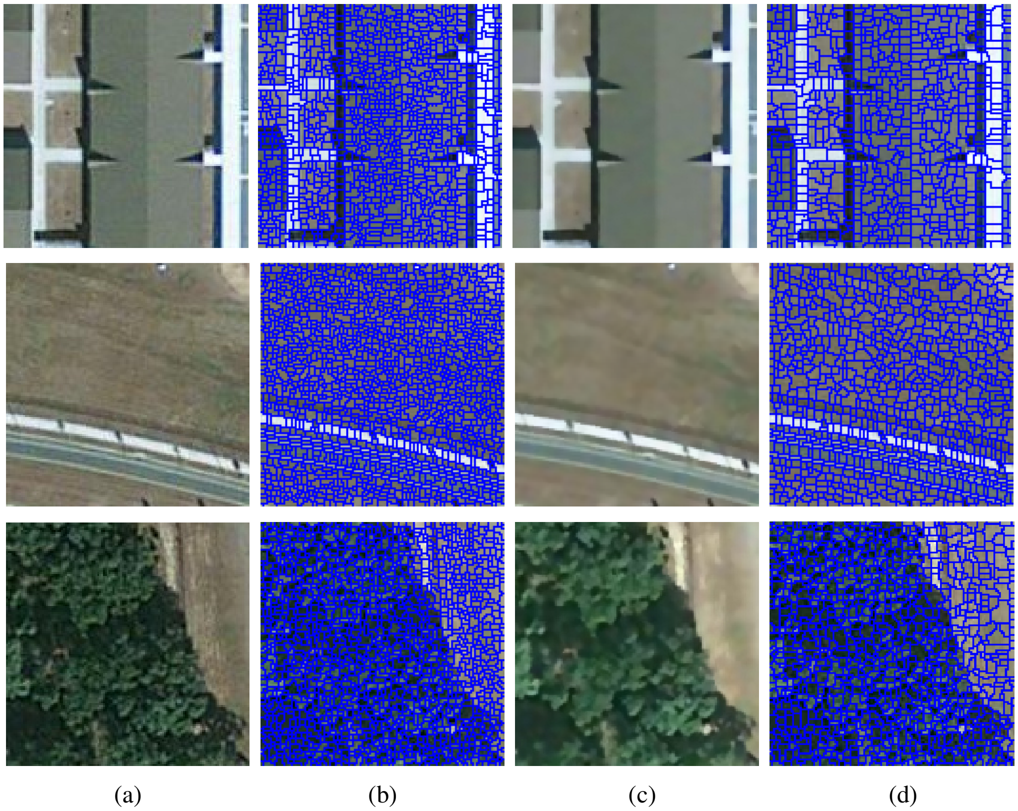
Direct gradient magnitude image input to watershed in natural images always results in over-segmentation. This is because of noisy and textured nature of those images. Therefore, smoothing of the input image before watershed application becomes a must.

4.3.1 Pre-filtering by EPSF

We evaluate the effect of the use of PGF [30] and EPSF [32] smoothing/de-noising filters before watershed on the segmentation accuracy and performance in Section 5.2. In that section, it is shown that although the segmentation accuracy resulted from these two algorithms is more or less the same, EPSF is better than PGF in terms of the over-segmentation handling capability and computational complexity. Hence, we incorporate EPSF algorithm to our proposed approach; however, we made a little modification on the filter coefficients as shown in Equation 4-1 for a 3*3 filter example. While Nikolaou et al. [32] propose to use 0 for center pixel to remove impulse noise; we choose to use 1 for center pixel with the assumption that high spatial resolution satellite images are free of impulse noise.

$$EPSF = \frac{1}{(\sum_{i=1}^{8} c_i) + 1} \begin{bmatrix} c_1 & c_2 & c_3 \\ c_4 & 1 & c_5 \\ c_6 & c_7 & c_8 \end{bmatrix} \qquad (4\text{-}1)$$

Also, the smoothing parameter,s, of EPSF algorithm is found to have an inconsiderable effect on the segmentation accuracy except for the very low values of it. Therefore, we propose to use a constant $s = 10$ for all test images as also proposed by Nikolaou et al. [32]. From the experiments done on Section 5.2, we come up with a window size,w, of 5 for test images 1 to 3. We present the watershed segmentation results of test images 1 to 3 after the application of EPSF with corresponding window sizes and the watershed segmentation results of images without preprocessing applied in Figure 4-4.



(a) (b) (c) (d)

Figure 4-4 The effect of EPSF on the segmentation results (a) Original test images, (b) Segmentation of (a), (c) EPSF applied results of (a), (d) Segmentation of (c)

From the visual results shown in Figure 4-4c, the success of EPSF is apparent in smoothing the region interiors and preserving the boundaries. Comparing Figure 4-4b with Figure 4-4d, it can also be seen that over-segmentation is reduced a little by the application of EPSF. Because of still high over-segmentation, it is hard to assess the change in the segmentation accuracy from Figure 4-4b to Figure 4-4d.

4.3.2 Homogeneity Image Generation

H-Image method [37] which is an alternative approach to gradient magnitude image is evaluated and compared with MSGM method [41] in Section 5.3. In that section, H-Image method is found as superior of MSGM in the sense that it decreases over-segmentation while preserving real object boundaries as with MSGM with the expense of computational complexity. It is also shown in that section that although MSGM and H-Image produce nearly same results for test images 1 and 2, H-Image gives superior results for test image 3. While MSGM algorithm looks for sequential intensity change of pixels throughout the image, H-Image looks for the homogeneity of windows centred around the considered pixel. Therefore, while MSGM produces extensive number of edges in for example forestry regions, H-Image handles these regions well and reduces the over-segmentation as seen from the results of test image 3 in Section 5.3. In Section 5.3, optimum window sizes of H-Image method are found as $w = 3$ for test images 1 and 2 and $w = 7$ for test image 3. H-Images and their corresponding segmentation results are shown in Figure 4-5.

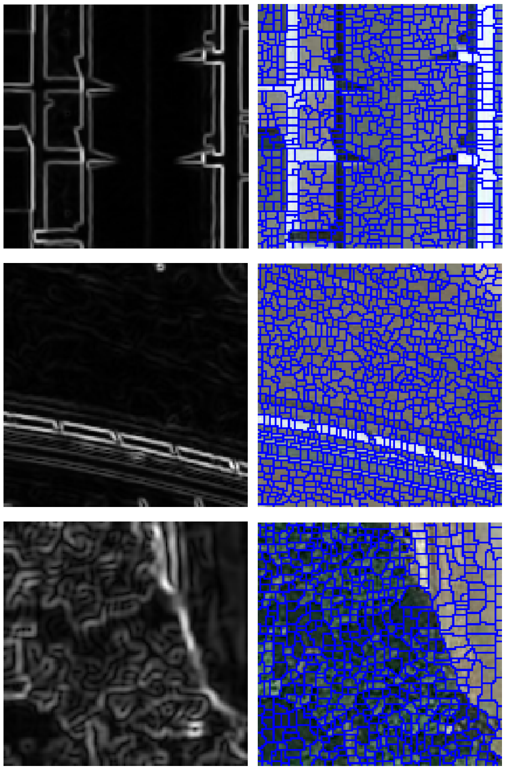


Figure 4-5 H-Images and their segmentation results (a) H-Images of test images 1 to 3 (b) Segmentation results of (a)

From the visual results shown in Figure 4-5b, no big change can be discernible except for the test image 3 by comparing the results with the ones in Figure 4-4d which are found by using MSGM.

4.4 Watershed Segmentation

We evaluate two different watershed algorithms from two different classes in this thesis. These are Vincent-Soille's algorithm [26] which is a flooding type watershed algorithm and De Smet et al.'s algorithm [35] which is a rainfalling type watershed algorithm. We compare the segmentation results of these two algorithms in Section 5.1 for test images 1 to 8 and found that Vincent-Soille's algorithm is superior to De Smet et al.'s algorithm in terms of segmentation accuracy. Actually, De Smet et al.'s algorithm is highly successful in reducing the over-segmentation in the results by its relative drowning threshold, rdt, however, it is not good at preserving object boundaries and segmentation accuracy.

4.5 Multi-scale Segmentation by Region Merging

In Sections 5.4, 5.5 and 5.6, we compare and evaluate different approaches to multi-scale segmentation (MSS) and conclude that second region merging based MSS algorithm [12] provides better results than first region merging based MSS algorithm [14]. In both of the region merging algorithms that we evaluate, we apply the heterogeneity criterion that is also incorporated in eCognition® software [3]. This criterion measures spectral and structural heterogeneity change resulted from merging of two regions and structural part of this criterion considers smoothness of segment boundaries and compactness of the segments. As found in Section 5.5, inclusion of structural part of the criterion doesn't change the results so much possibly because of extreme over-segmentation present in the results. Therefore, we prefer not to include structural part of the criterion to the heterogeneity criterion. The heterogeneity criterion that we incorporate in this thesis turns out to be a criterion as shown in Equation 4-2. The parameters in this formula are explained in Section 3.3.2 in detail.

$$f = \sum_i w_i(n_m \sigma_{m,i} - (n_1 \sigma_{1,i} + n_2 \sigma_{2,i})) \tag{4-2}$$

The two algorithms differ in their merging order and stopping criterion. First method arranges merging list in the increasing order of segment size and merging cost. In other words, merging occurs from small-sized segments to large-sized segments and the priority of merging at the same-sized segments is the one with lower merging cost. Second method, on the other hand, arranges merging list in the increasing order of merging cost by not considering segment sizes.

Considering accuracy and performance results obtained in Section 5.5, we realize that those two region merging algorithms are not sufficient in their own. While first one incorporates only a size-based stopping criterion, second one incorporates only a heterogeneity-based stopping criterion. While first one achieves a segmentation result in that minimum segment size is the stopping criterion, second one achieves a segmentation in that any two segments are different from each other at least as the stopping criterion. However, in an ideal segmentation, we need both of these properties. Therefore, we propose a hybrid region merging method that uses these two region merging algorithms in a cascaded form as shown in Figure 4-6. In other words, first, we apply first region merging algorithm and give the output to second region merging algorithm as input.

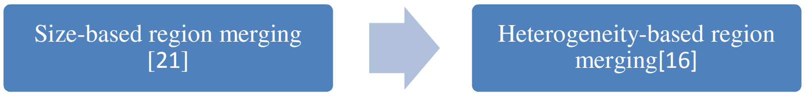

Size-based region merging [21]
Heterogeneity-based region merging[16]

Figure 4-6 The process flow of proposed region merging algorithm in this thesis

The segmentation results with no region merging applied, with first region merging applied, with second merging applied and with proposed region merging applied cases are shown in Figure 4-7 for test images 1 to 3 by applying the parameters found in Section 5.5.

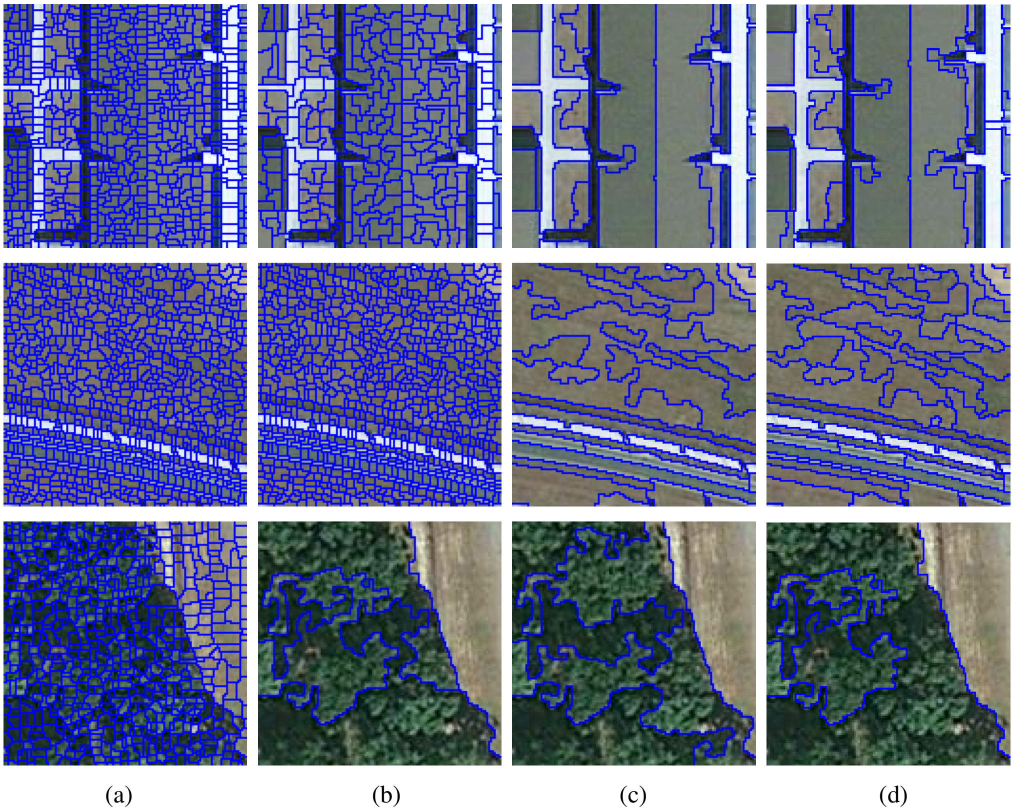


Figure 4-7 Segmentation results of different region merging algorithms for test images 1-3 (a) No region merging, (b) First region merging, (c) Second region merging, (d) Proposed region merging

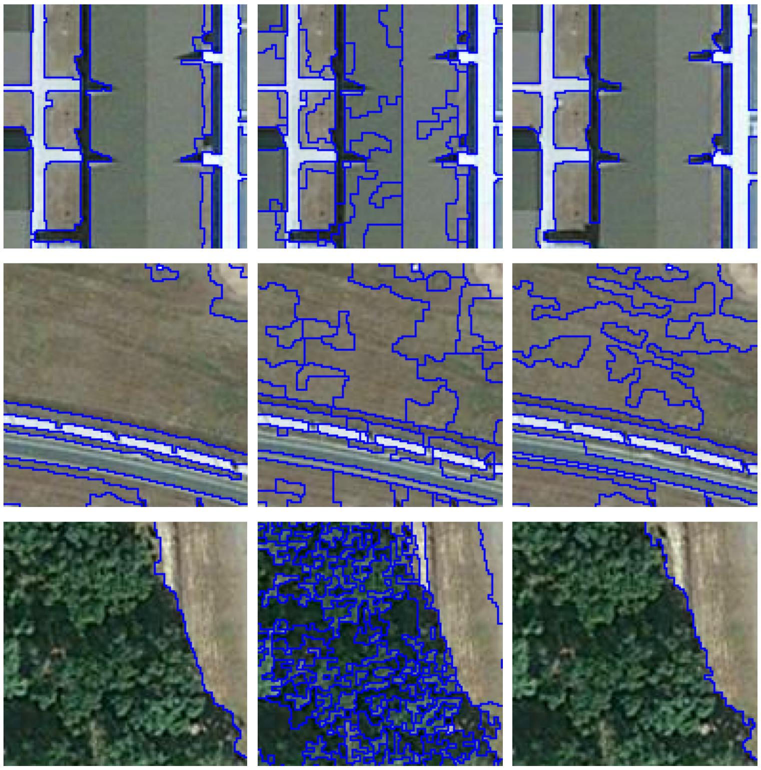
As discussed in Section 5.5 and as shown in Figure 4-7, newly proposed hybrid algorithm improves over-segmentation handling capability of RM1 and RM2, and it has lower computational complexity than other two region merging algorithms; however, its segmentation quality is slightly less than the others. Because of the similarity of the results of RM2 and RM3, we may think that first region merging that is incorporated in RM3 is unnecessary. Let's think a situation like that there exist a parking lot and cars parked in it. Actually, an optimum segmentation of this scene is dependent on the application. If the segmentation of individual cars in the parking lot is necessary, the segmentation approach must handle a lower minimum segment size. Otherwise, if the segmentation of the parking lot as a whole is necessary, the segmentation algorithm must handle a higher minimum segment size. Because we want to deal with macro objects in the image (not a car but a parking lot, not a tree but a forest), a pre-merging of segments that have sizes lower than a threshold is necessary. First region merging of the proposed algorithm provides us the application of this threshold. By labeling the ground truth image according to our needs (e.g. segments are major objects in our case) and analyzing the best results of evaluation measures, this threshold is found heuristically.

4.6 Automatization of the Segmentation

Up to this point, we selected parameters of every algorithm in an attempt to maximize segmentation accuracy obtained with segmentation accuracy evaluation metrics ($Ev1$ and $Ev2$) and minimize the over-segmentation. Actually, application of supervised evaluation measures to natural images is problematic because of the deficiency of being sure about the true class of each pixel. This is eminently observed when generating ground truth images that are made use of in this thesis.

For this reason, two unsupervised goodness measures are compared and evaluated in 5.7. First goodness measure which is specified in Equation 3-32 evaluates the segment count and intra-segment variance of the segmentation with the assumption that an ideal segmentation has minimum segment count and homogenous segments. Second goodness measure tries to find the optimum point in that sum of Moran's I and intra-segment variance of the segmentation is lowest. Moran's I is also an unsupervised measure that reflects inter-segment correlation and therefore, its lowness means the non-existence of over-segmentation. It also could mean under-segmentation; however, the intra-segment variance is here to balance this.

In Figure 4-8, the segmentation results obtained by following different segmentation accuracy measures are shown. From these visual comparisons, an apparent advantage of $G2$ measure over $G1$ stands out. It produces nearer results to manual segmentations at the expense of a decrease in segmentation accuracy. For the segmentation of test image 3, $G2$ produces almost the same result with the manual segmentation; however, $G1$ produces a high over-segmented result because of its nature of favoring spectrally homogeneous regions and high texture content of forest. This over-segmentation effect of $G2$ could be seen in all three test images. On the other hand, $G1$ is prone to under-segmentation in spectrally similar regions as in test image 1. Because the color of roof and soil are similar, it gives under-segmented outcomes.



(a) Manual (b) $G1$ (c) $G2$

Figure 4-8 Segmentation results obtained by following different measures

As a result, although the under-segmentation effect of it, incorporating $G2$ measure with the purpose of determining segmentation parameters to our proposed approach is fair enough considering the quantitative results found in Section 5.7. By this way, our proposed segmentation approach becomes a fully automatic one.

CHAPTER 5

EXPERIMENTS

In this chapter, experimental works that are done to decide best method for each step of the segmentation approach shown in Figure 5-1, to decide best unsupervised segmentation goodness metric that makes segmentation approach fully automatic, to determine the quality of this approach and to compare it with other approaches present in the literature are presented.

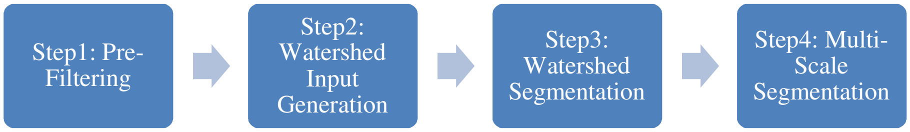

Step1: Pre-Filtering
Step2: Watershed Input Generation
Step3: Watershed Segmentation
Step4: Multi-Scale Segmentation

Figure 5-1 The flow of segmentation approach proposed in this thesis

For these experiments, a database as shown in Table 5-1 is constructed. All images in the database are taken from Google Earth$^{®}$ software. The test images 1-8 are relatively less complex and composed of less segments than test images 9-20. The ground truths of test images 1-8 are generated manually but they are only used for performance evaluation and comparison purposes. Ground truth generation for remaining test images (test images 9-20) is omitted because of the unreliable behavior of manual ground truth generation process at the segment boundaries in such complex images. Those images are used for visual segmentation quality assessment of the proposed approach. All experiments are done on MATLAB$^{®}$ 7.10 software.

Table 5-1 Satellite image database that is constructed and used in this thesis

Image Size	*Image Count*	*Image No*	*Ground Truth*
128*128	3	1-3	Available
256*256	5	4-8	Available
512*512	2	9-10	Not Available
1024*1024	10	11-20	Not Available
Total	20	1-20	-

This chapter can be divided into three parts:

- In Sections 5.1 to 5.6, we try to find the best alternative method for each step of the segmentation approach by using supervised segmentation accuracy evaluation metrics and performance metrics such as segment count, computation time and PSNR. At the end of this part, the proposed segmentation approach is constructed.
- After deciding the best alternative methods for each step in the first part, all input parameters of the proposed segmentation approach are found by using unsupervised segmentation accuracy evaluation metrics in Section 5.7. Also, the segmentation results found in a supervised manner in the first part are compared with the fully automatic segmentation results and then; the best unsupervised segmentation accuracy evaluation metric is decided in this section. At that point, decided metric becomes a part of the proposed segmentation approach and the proposed approach becomes a fully automatic one.
- Finally, in Section 5.8, the proposed segmentation approach is compared with the similar approaches present in the literature.

5.1 Comparison of Watershed Algorithms

In this section, we evaluate two different watershed algorithms by following the steps shown in Figure 5-2. In this figure, RF stands for rainfalling watershed algorithm detailed in Section 3.2.2 and VS stands for Vincent-Soille flooding watershed algorithm detailed in Section 3.2.1. First, we apply the gradient magnitude operation on the input image using the Multi-Spectral Image Gradient algorithm detailed in Section 3.1.3 that incorporates Sobel filter. Then, watershed algorithms are applied and segmentation results are compared by using both visual analysis and the accuracy and performance measurements. For both watershed algorithms, we apply 8-neighborhood.

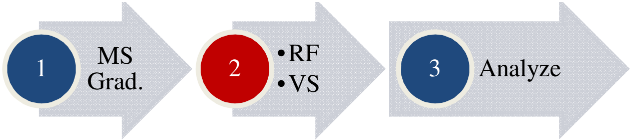
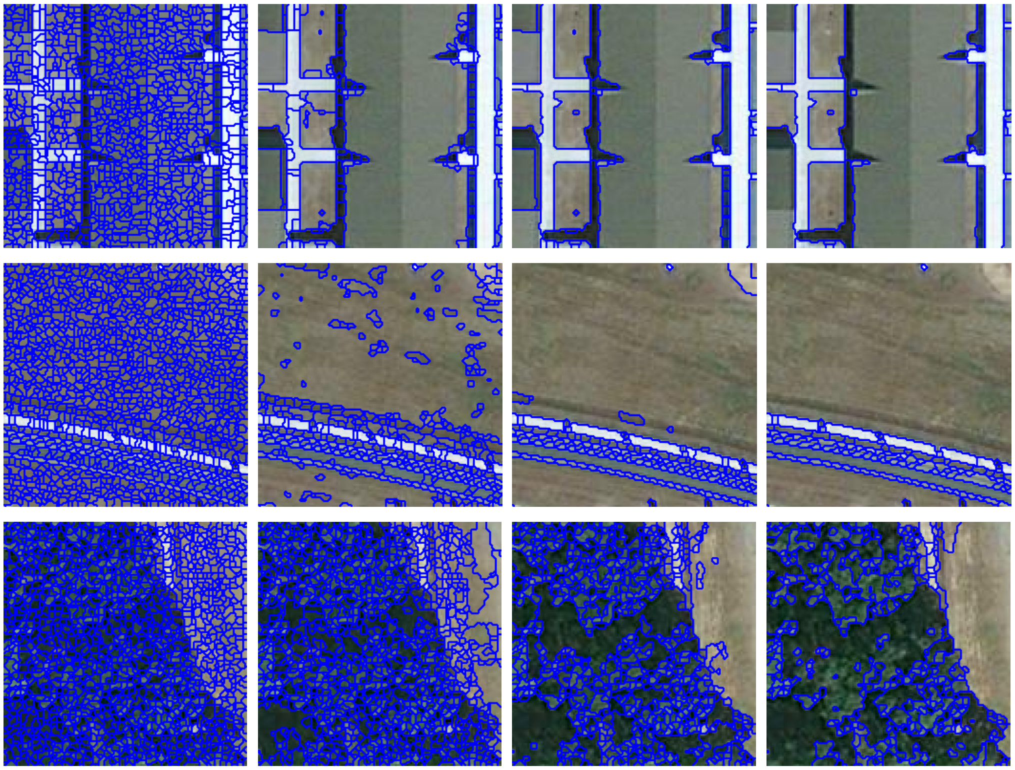


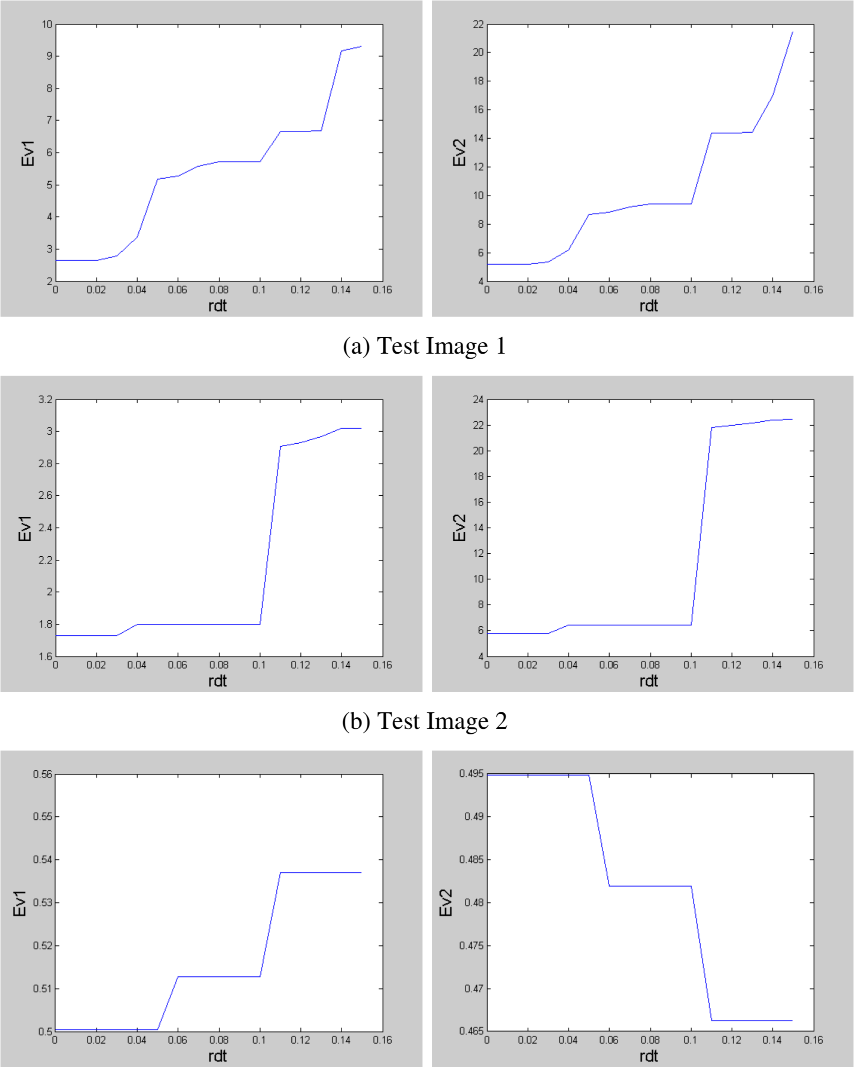
Figure 5-2 The method applied for finding the best watershed algorithm

While RF has one input parameter (relative drowning threshold parameter), VS has none. Therefore, we have to find the input parameter that provides best RF result before comparing RF with VS. We begin with finding the best rdt(relative drowning threshold) parameter for the RF algorithm which is discussed in Section 3.2.2 in detail. With increasing rdt, more local minimums become drowned. This means, for our case, that small gradients become discarded. This can be called in a way as region merging; however, instead of regions' properties, just the edge magnitude between corresponding regions is considered. To be able to observe the effect of increasing rdt on the RF segmentation result, we apply RF to test images 1-3 with 16 different rdts that are extending from 0 to 0.15 with steps of 0.01. Visual results for rdts 0.00, 0.05, 0.10 and 0.15 for these test images could be seen in Figure 5-3. Also, the supervised accuracy evaluation measurements obtained for all rdts are shown in Figure 5-4. The direct consequence of increasing rdt is decreasing segment count and computational complexity because of less local minimums. By not considering accuracy measurements, the results in Figure 5-3 look quite appealing indeed. One can think that increasing rdt improves visual segmentation quality by pretty much decreasing the over-segmentation. Although this opinion is correct until some point (this point differs for different test images), the increase of rdt beyond that point decreases segmentation accuracy as seen from accuracy measurements in Figure 5-4. These points are $rdt = 0.02$ for test image 1, $rdt = 0.03$ for test image 2 and $rdt = 0.05$ for test image 3. While $Ev1$ measurements remain constant up to these points and therefore segmentation accuracies are preserved, the over-segmentation effects are reduced. $Ev2$ measurements are parallel with the $Ev1$'s except for the test image 3. However, the variations in $Ev1$ and $Ev2$ measurements in the case of the test image 3 are very low and this inconsistency could be negligible.



(a) $rdt = 0$ (b) $rdt = 0.05$ (c) $rdt = 0.10$ (d) $rdt = 0.15$

Figure 5-3 RF segmentation results of test images 1-3 obtained for shown rdts



(a) Test Image 1

(b) Test Image 2

(c) Test Image 3

Figure 5-4 $Ev1$ and $Ev2$ measurements of RF segmentation results of test images 1-3 for rdts 0.00 to 0.15 with steps of 0.01

After finding the optimal rdts for test images 1-3, we are now able to compare VS and RF segmentation algorithms. The visual comparison of these two algorithms by the test images 1-3 can be seen in Figure 5-5. In this figure, simplified images, as discussed in Section 3.3.1.1, are obtained by averaging the intensity values of input image over each segment and setting intensity value of each pixel in a segment to intensity average of that segment. Because of the extreme over-segmentation present in the segmentation results, it is quite hard to decide which algorithm has better segmentation quality from segmented and simplified images. However, in simplified form of test image 1 by RF, the corruption in the boundaries of buildings draws attention and this obviously shows us the existence of a problem of co-location of RF segment boundaries with the real object boundaries.

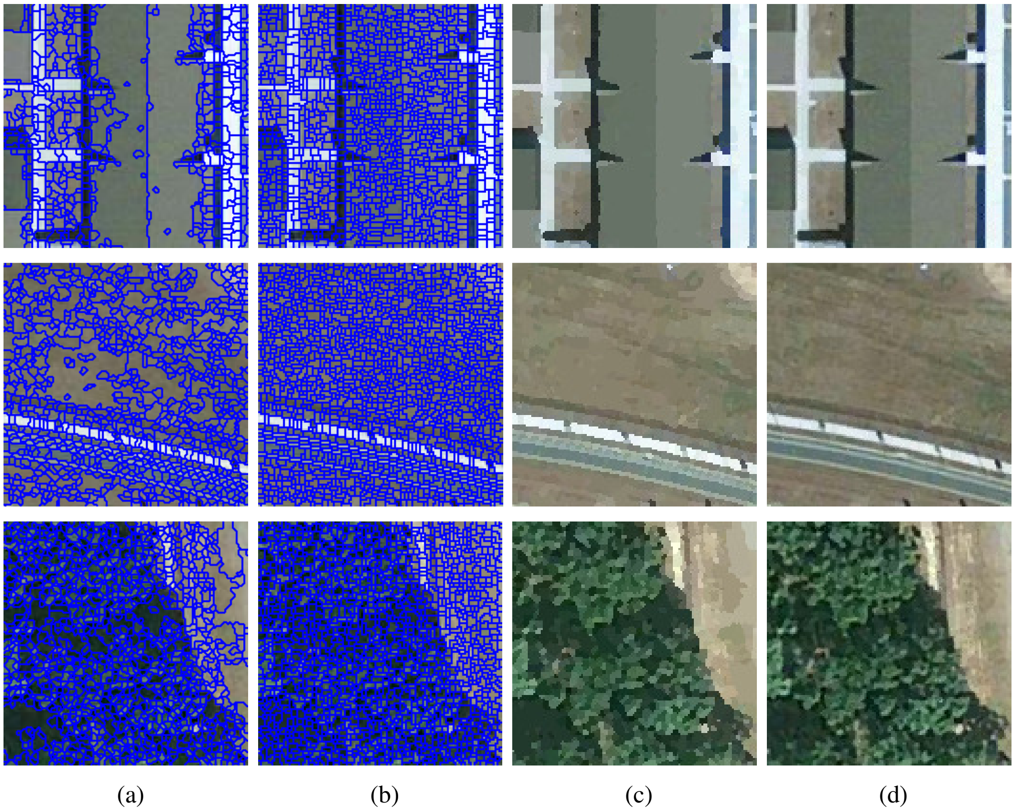


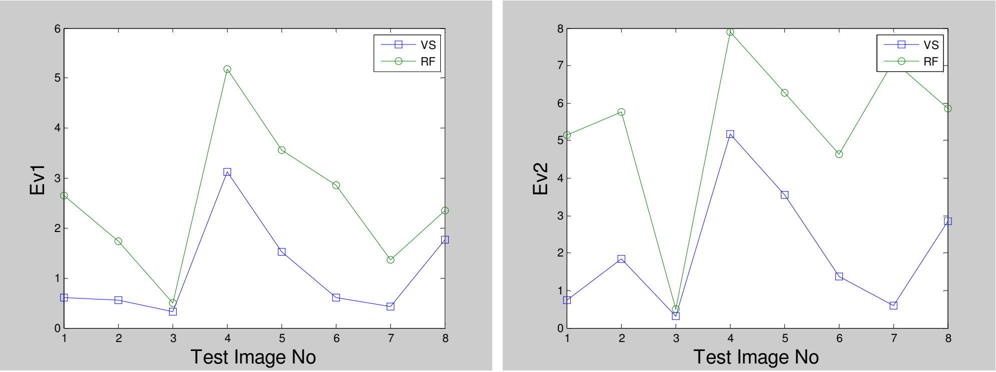
Figure 5-5 Visual comparison of the results of RF and VS algorithms for test images 1-3 (a) RF segmented results with optimal rdts, (b) VS segmented results, (c) Simplified image of (a), (d) Simplified image of (b)

To compare the segmentation results in a more objective manner, we make use of the accuracy and performance measures and the results of these analyses are shown in Table 5-2 for test images 1-3. In this table, better accuracy results are shown in bold. As seen in this table, supervised accuracy evaluation measurements ($Ev1$ and $Ev2$) show an apparent advantage of VS over RF. PSNR measurements reflect the better intra-segment homogeneity in VS results and this supports our visual findings of corrupted boundaries in RF results previously. VS is seen as slightly less computationally complex; however, this contradicts with the findings of [35]. This contradiction may be due to the optimized implementation of VS in MATLAB® software. While RF is our own implementation, VS is directly performed with the built-in function of MATLAB® software.

Table 5-2 Quantitative comparison of the results of RF and VS segmentations for test images 1-3

Test Image	*Algorithm*	*Seg. Cnt.*	*Time (sec)*	*PSNR (dB)*	*Ev1*	*Ev2*
1	VS	1220	0.04	31.92	**0.61**	**0.73**
	RF	552	0.09	26.90	2.64	5.16
2	VS	1562	0.04	32.87	**0.58**	**1.69**
	RF	1029	0.10	29.70	1.73	5.77
3	VS	1541	0.04	25.86	**0.32**	**0.31**
	RF	1246	0.11	22.79	0.50	0.50

Finally, to support our previous findings about the superiority of VS over RF, we measure the accuracy of segmentation results of test images 4-8 in the database and share the results in Figure 5-6. For these test images, optimum rdt values are found by using the same procedure explained previously. The accuracy measurements obtained from every test image show that VS is superior to RF.



(a) $Ev1$ vs. Test Image No

(b) $Ev2$ vs. Test Image No

Figure 5-6 Quantitative comparison of the results of RF and VS segmentations for test images 1-8

As a result, although RF reduces the over-segmentation enormously, VS comes into prominence for the optimal watershed algorithm from the perspective of quantitative segmentation accuracy.

5.2 Comparison of Smoothing/De-Noising Filters

In this section, we evaluate the effect of two different noise removing/smoothing filters which are Peer Group Filter (PGF) and Edge Preserved Smoothing Filter (EPSF) on the segmentation results.

In the first part of this section, each filter's input parameters that give best segmentation results according to both qualitative and quantitative analysis are found. Then, best results offered from the two filters are compared. Throughout these comparisons, VS watershed algorithm and Multi-Spectral Image Gradient algorithm that incorporates Sobel filter are applied. This process could be seen in Figure 5-7.

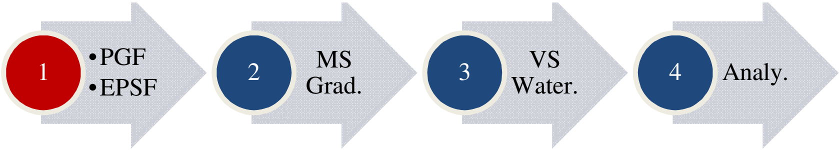
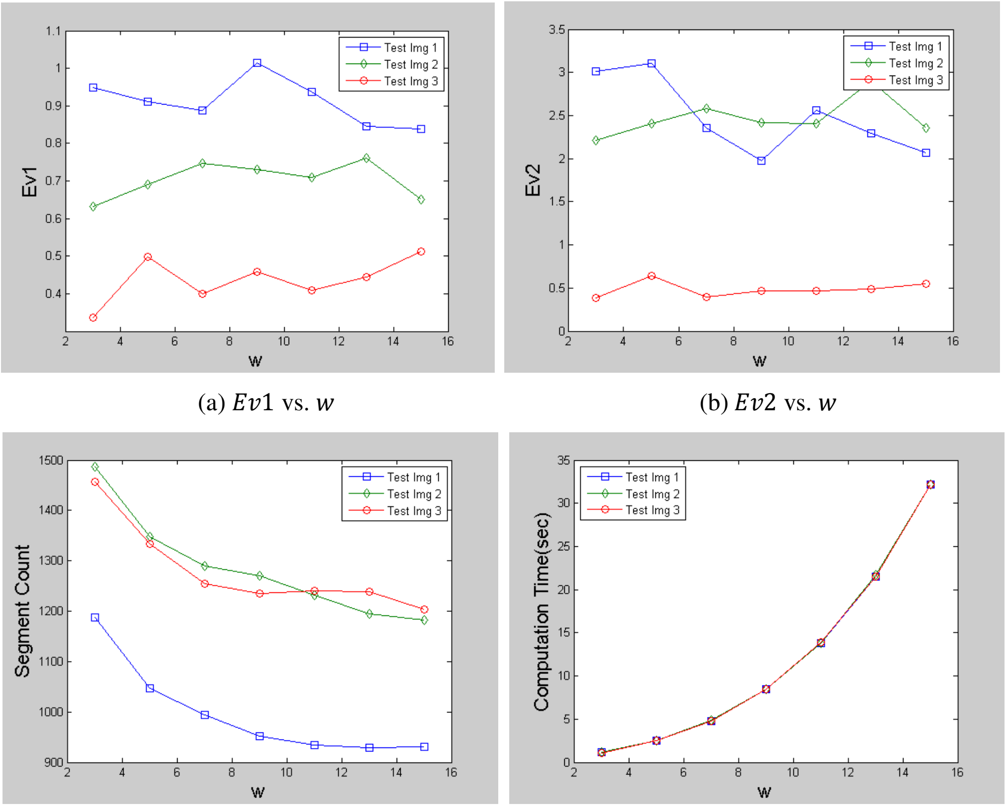


Figure 5-7 The method applied for finding the best smoothing/de-noising filter

We begin with finding the best parameter set for the PGF algorithm. PGF algorithm has two input parameters which are window size, w, and impulse noise threshold, t. We make an assumption that the input image is free of impulse noise. This assumption is logical in a sense that the high spatial resolution satellite images are generally impulse noise free. This threshold may be useful in the case of SAR images which are rich of speckle noise that is a type of impulse noise. Even in the case of SAR images, handling of a constant threshold is a bit risky because its value should be dependent on the level of noise present in the image. Therefore, adjusting the threshold needs an adaptive approach which should be aware of the noise level in the image or needs supervision. As a result, we set $t = 0$ to discard impulse noise threshold.

In Figure 5-8, we try to show the effect of PGF window size on the accuracy and performance of the segmentation results for ws extending from 3 to 15. Considering Figure 5-8a-b, it can be said that window size of PGF has a minor effect on the segmentation accuracy. In PGF algorithm, a peer group that is composed of spectrally similar pixels with respect to the central pixel of the window is constructed for every pixel of the image. Then, the average of the intensity values of the pixels in the peer group weighted by a standard 2D Gaussian becomes the filtered value of the corresponding pixel. Therefore, while window size increases, even if the peer group starts to include the pixels that are spatially distant to the central pixel, the effect of those pixels on the average will become minimized because of Gaussian weights. As a result, the results in Figure 5-8a-b are expected. At this point, we prefer choosing the window size that reduces the over-segmentation most. Therefore, we come up with $w = 15$ for test images 1-3 as can be seen from Figure 5-8c.



(a) $Ev1$ vs. w

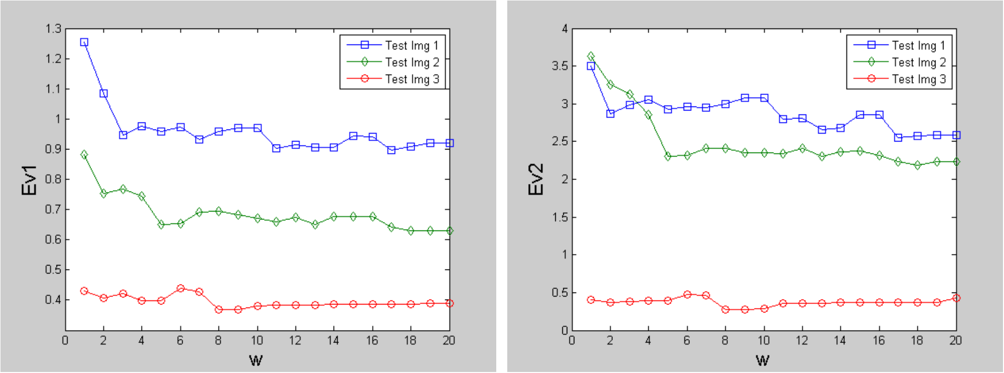
(b) $Ev2$ vs. w

(c) Segment Count vs. w

(d) Computation Time vs. w

Figure 5-8 $Ev1$, $Ev2$, segment count and computation time measurements of PGF pre-processed segmentation results of test images 1-3 for PGF's ws 3 to 15

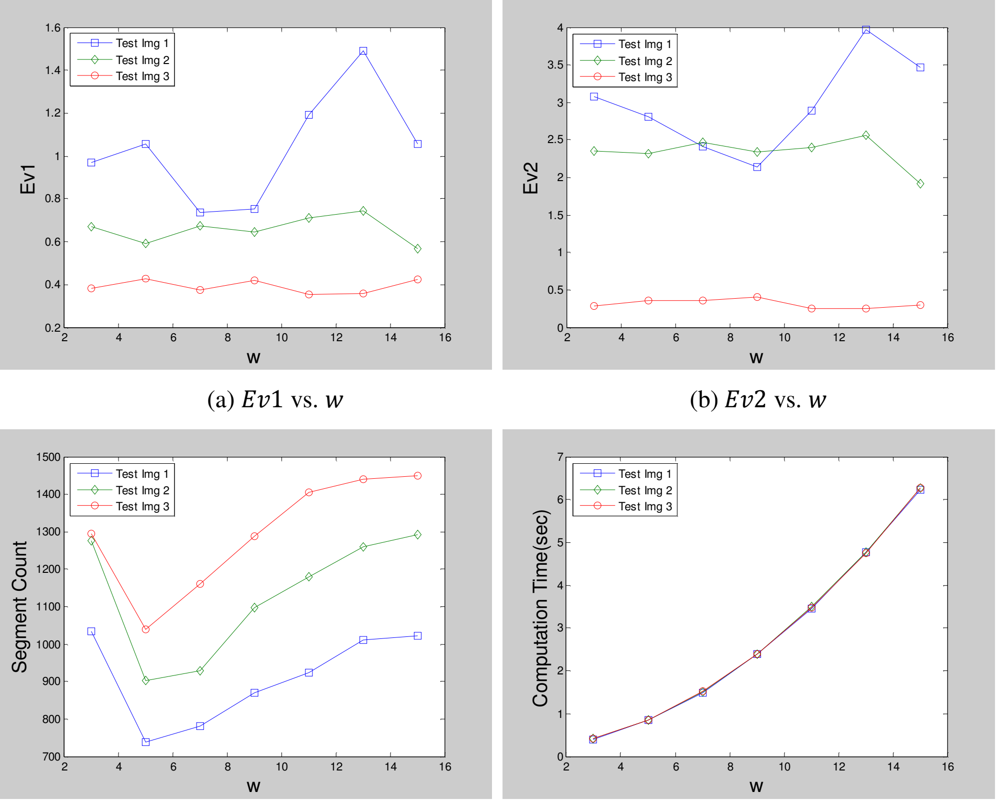
We continue with finding the best parameter set for the EPSF algorithm. EPSF algorithm has two input parameters which are window size, w, and scale or smoothing factor, s. EPSF is also called as an adaptive mean filter. Different from a classical mean filter, the coefficients of the filter are not unique throughout the image but depend on the local statistics. Spectrally similar pixels to the central pixel in a window take more weight and s could be thought of a multiplier of these weights. With increasing s, the weights of similar pixels increase more and more and smoothing of the image decreases. In [32], it is seen that a fixed $s = 10$ gives a good performance on all tests. To check the validity of this finding, we analyze the accuracy of segmentation results for s extending from 1 to 20 and $w = 3$ for test images 1-3. As seen from the results in Figure 5-9, despite the fluctuations in the segmentation accuracy for low values of s, the results become stable and almost same for s bigger than 8. Therefore, we also use 10 for s parameter. Now, we will find the best window size with s set to 10.



(a) $Ev1$ vs. s (b) $Ev2$ vs. s

Figure 5-9 $Ev1$, $Ev2$ measurements of EPSF pre-processed segmentation results of test images 1-3 for EPSF's ss 1 to 20

In Figure 5-10, we see the effect of EPSF window size on the accuracy and performance of the segmentation results for ws extending from 3 to 15. As seen from Figure 5-10a and b, there is no trend of continuous increase or decrease with increasing window size for both $Ev1$ and $Ev2$ measures. Also, the change of segmentation accuracy measurements between different window sizes is small and could be neglected as in the case of PGF. Therefore, we prefer choosing the optimum window size regarding the segment count. It was observed that $w = 5$ is the optimum selection for test images 1-3 as seen in Figure 5-10c.

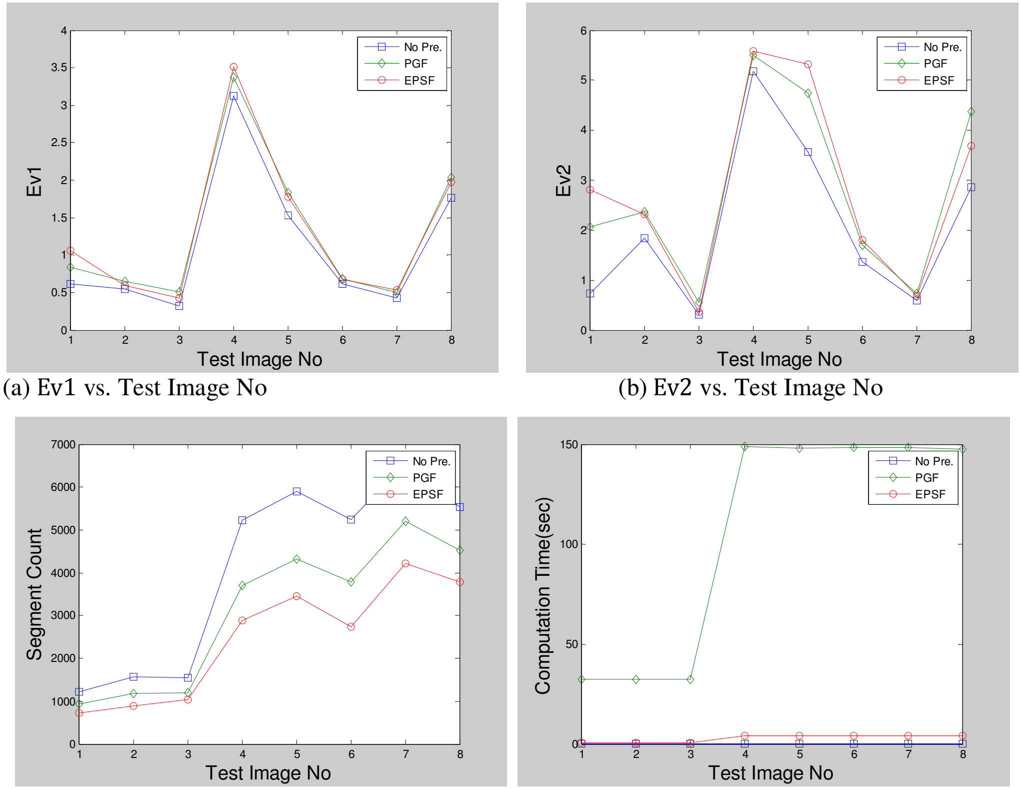


(a) $Ev1$ vs. w (b) $Ev2$ vs. w

(c) Segment Count vs. w (d) Computation Time vs. w

Figure 5-10 $Ev1$, $Ev2$, segment count and computation time measurements of EPSF pre-processed segmentation results of test images 1-3 for EPSF's ws 3 to 15

Finally, we make a comparison of PGF and EPSF pre-processed segmentation results with their corresponding best parameter sets as in Figure 5-11 for test images 1-8. From these measurements, it can be seen that best segmentation accuracy is achieved when no pre-processing applied. However, the reduction of over-segmentation is at minimum for this case. Both PGF and EPSF give very close results in terms of segmentation accuracy for each test image and there is no global dominant in any of measures except for the segment count and computation time. Segment count and the computational complexity of EPSF are much lower than PGF. Therefore, we choose EPSF because of its superior handling of over-segmentation and lower time complexity.



(a) Ev1 vs. Test Image No

(b) Ev2 vs. Test Image No

(c) Segment Count vs. Test Image No

(d) Computation Time vs. Test Image No

Figure 5-11 Quantitative comparison of the results of PGF and EPSF preprocessed segmentations for test images 1-8

5.3 Comparison of Watershed Input Types

Watershed algorithm generally takes gradient magnitude image as input. In this section, Multi-spectral Gradient Magnitude (MSGM) and H-Image methods are compared to find best watershed input that provides the best segmentation accuracy and performance.

Prior to comparison of these two methods, H-Image method is evaluated for best result with different window sizes, w. Then, best H-Image result is compared with MSGM result. The process followed for this step could be seen in Figure 5-12.

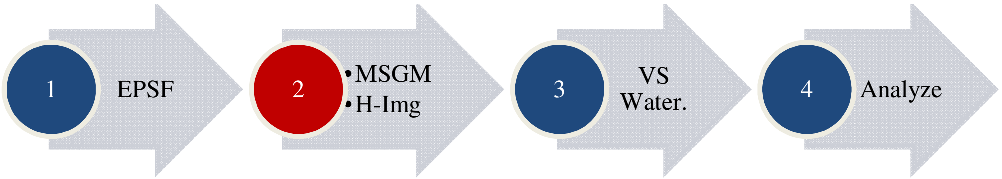


Figure 5-12 The method applied for deciding between MSGM and H-Image algorithms

H-Image method is like a smoothing filter applied to the gradient magnitude image of an image and this smoothing increases with window size. To see the effect of window size on H-Image algorithm, the segmentation results of test images 1-3 for ws extending from 3 to 15 are analyzed. The H-Images for $w = 3$ and $w = 11$ and their corresponding segmented images are presented for test images 1-3 in Figure 5-13. In these results, increased smoothing is obvious with increasing window size on the H-Image. As a result of this, the over-segmentation in the segmented image is extensively reduced.

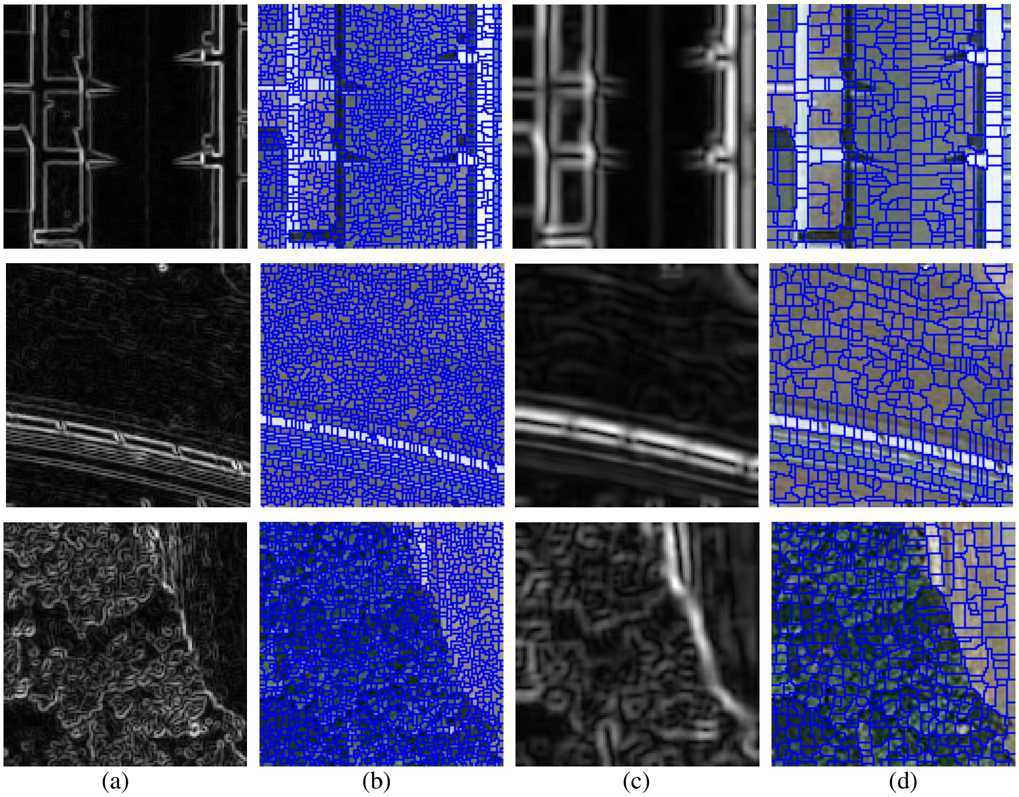
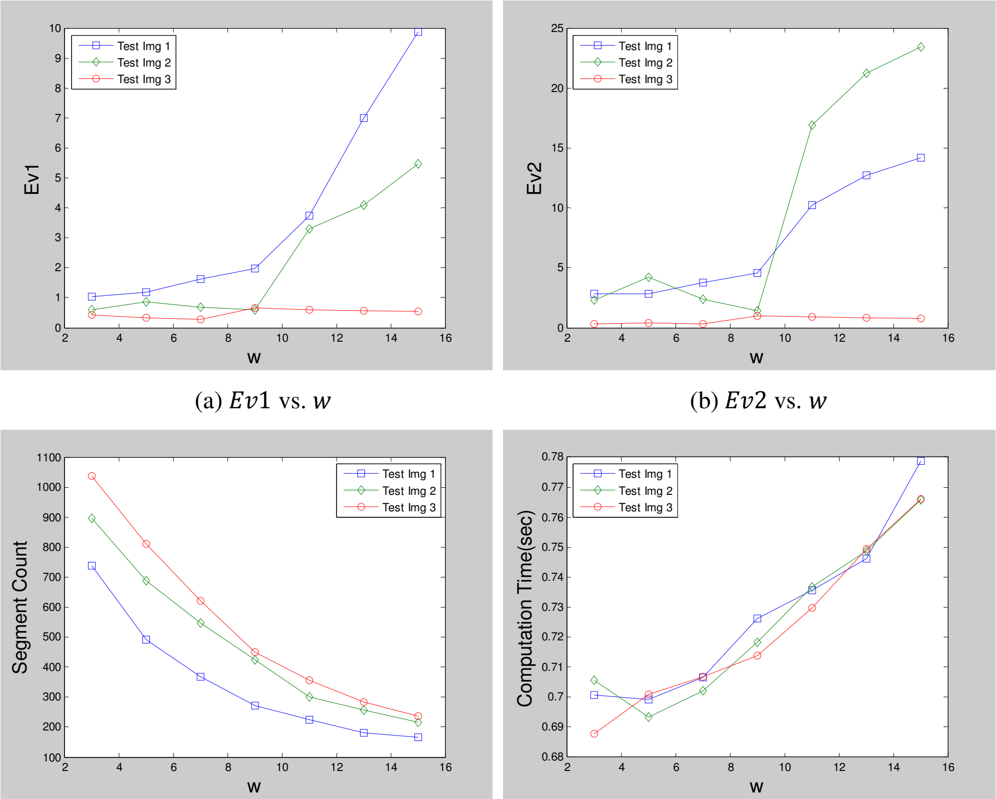


Figure 5-13 H-Images and segmentation results of test images 1-3 obtained for $w = 3$ and $w = 11$ (a) H-Image for $w = 3$, (b) segmentation of (a), (c) H-Image for $w = 11$, (d) segmentation of (b)

In Figure 5-14, we see the accuracy and performance measurements that are obtained with $w = 3$ to $w = 15$ for the H-Image method with the test images 1-3. In these graphics, extensive segmentation accuracy corruption especially for test image 1 and 2 that is not apparent in Figure 5-13 can be observed. For test image 1 and 2, regarding the best $Ev1$ and $Ev2$ measurements, we come up with the optimal w of 3 for test image 1 and optimal w of 9 for test image 2. On the other hand, for test image 3, the variation in $Ev1$ and $Ev2$ measurements is negligible. Therefore, we go for w of 15 that provides best over-segmentation handling.



(a) $Ev1$ vs. w

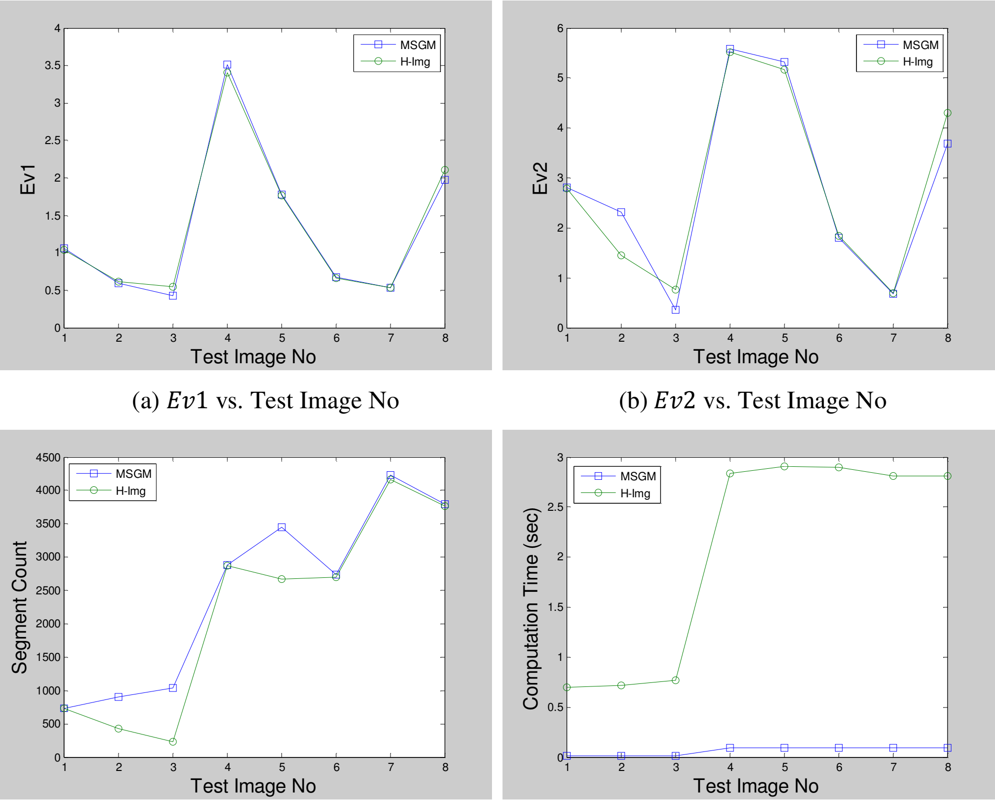
(b) $Ev2$ vs. w

(c) Segment Count vs. w

(d) Computation Time vs. w

Figure 5-14 $Ev1$, $Ev2$, segment count and computation time measurements of H-Image pre-processed segmentation results of test images 1-3 for H-Image's ws 3 to 15

Finally, we make a comparison between MSGM and H-Image methods as in Figure 5-15 for test images 1-8. From this table, it is seen that both methods produce nearly equal results for all test images in terms of segmentation accuracy metrics. However, the over-segmentation handling of H-Image method is superior to MSGM method at the expense of time complexity. By considering these results and adjustable window size property of H-Image, we opt for it rather than MSGM at the expense of computational complexity.



(a) $Ev1$ vs. Test Image No

(b) $Ev2$ vs. Test Image No

(c) Segment Count vs. Test Image No

(d) Computation Time vs. Test Image No

Figure 5-15 Quantitative comparison of the results of MSGM and H-Image preprocessed segmentations for test images 1-8

5.4 Comparison of Wavelet Based MSS Approaches

In this section, we compare two different wavelet based multiscale segmentation algorithms. The process that we follow could be seen in Figure 5-16. In both of the wavelet based algorithms, input image is segmented at a lower resolution (scale). Low resolution version of the input image is gathered by wavelet decomposition (step 2 in Figure 5-16). The two algorithms differ in their projection scheme (step 5 in Figure 5-16). While Kim&Kim's algorithm (Proj1) incorporates watershed segmentation result of every scale during projection to update boundaries, Jung's algorithm (Proj2) uses spectrally nearest neighbor pixels to correct boundaries. These projection schemes are explained in detail in Sections 3.3.1.1 and 3.3.1.2.

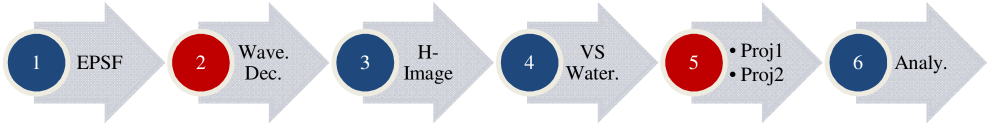


Figure 5-16 The method applied for finding the best wavelet based MSS algorithm

We begin with Jung's MSS algorithm that is explained in Section 3.3.1.2. This algorithm has only one parameter which is the scale parameter, s that shows the resolution level that the original image is decomposed to. In Figure 5-17, we see the supervised accuracy measurements that are obtained with $s = 1$ to $s = 3$ for the test images 1-3. From these measurements, it can be seen that $s = 1$ for test images 1-2 and $s = 2$ for test image 3 provide the best segmentation results amongst others. In these results, significant degradation of segmentation accuracy with increasing s is apparent except for the test image 3. Also, previously found parameters for EPSF and H-Image algorithms are used for each test image.

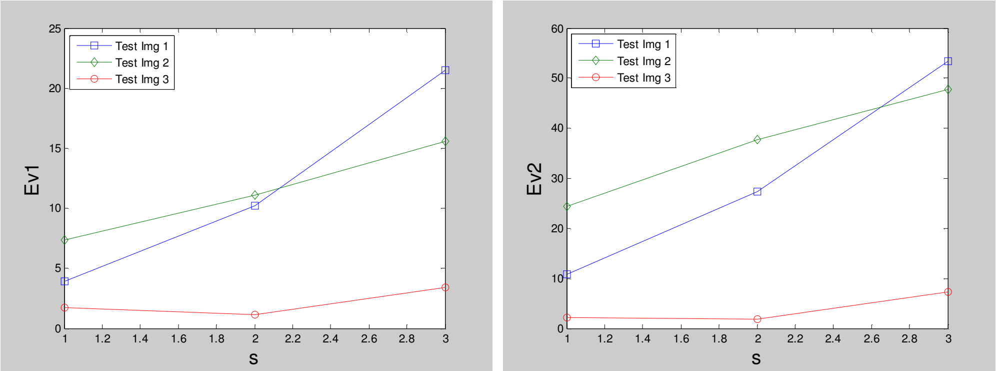


Figure 5-17 $Ev1$ and $Ev2$ measurements of the segmentation results of test images 1-3 with Jung's algorithm for ss 1 to 3

We continue with Kim&Kim's MSS algorithm that is explained in Section 3.3.1.1. This algorithm is similar to Jung’s MSS algorithm except for the projection scheme. This algorithm also has only one parameter which is the scale parameter, s, that shows the resolution level that the original image is decomposed to. In Figure 5-18, we see the supervised accuracy measurements that are obtained with $s = 1$ to $s = 3$. From these measurements, it can be seen that $s = 1$ and $s = 3$ provide the best segmentation results for test images 1-2 and for test image 3, respectively. As in Jung's algorithm, high degradation of segmentation accuracies is present in the results.

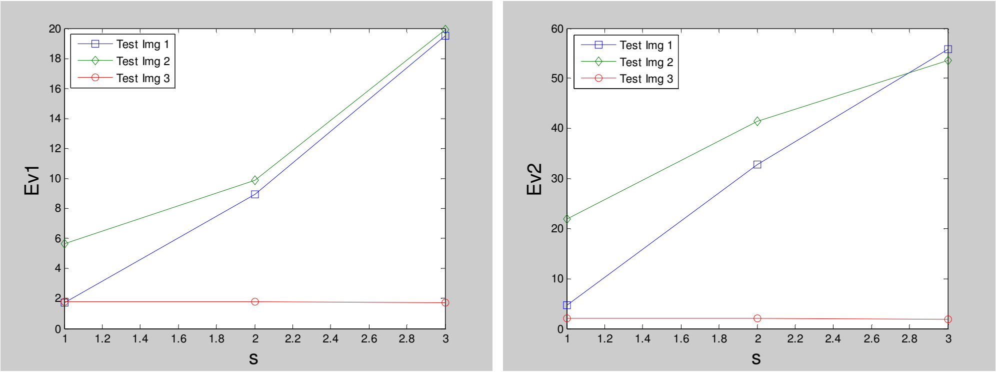
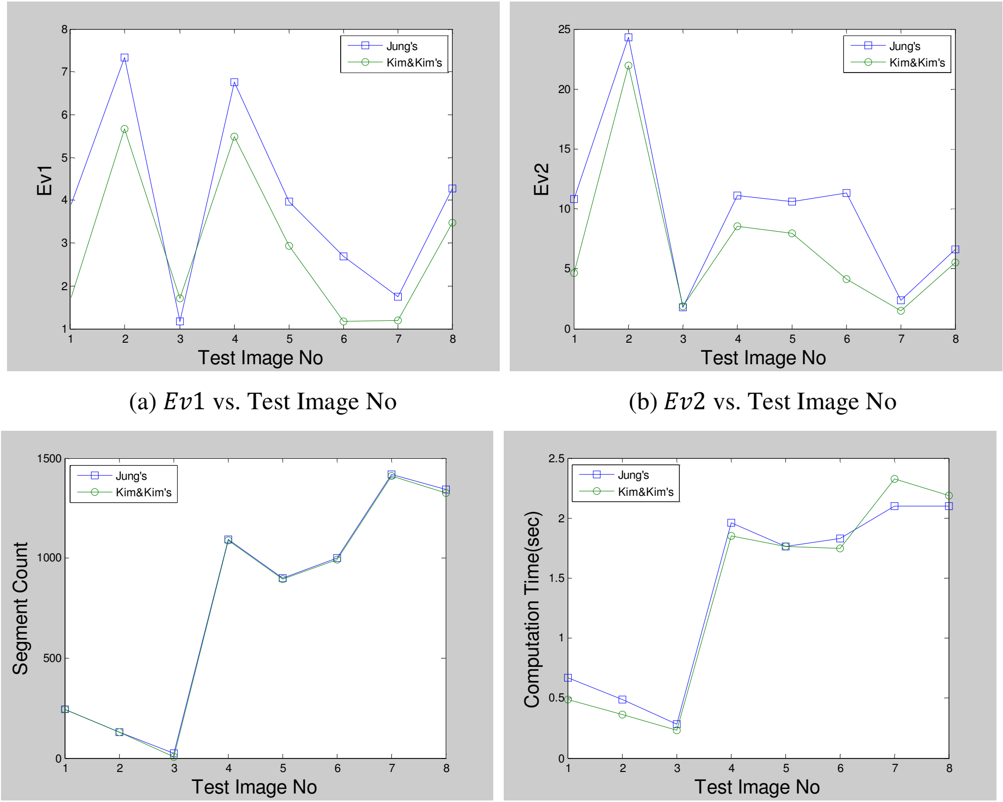


Figure 5-18 $Ev1$ and $Ev2$ measurements of the segmentation results of test images 1-3 with Kim&Kim's algorithm for ss 1 to 3

Finally, we make a comparison between Jung's and Kim&Kim's methods as in Figure 5-19. As shown in this figure, Kim&Kim's method is seen to produce superior results for nearly all test images in terms of the computational complexity and the segmentation accuracy while they produce nearly equal segment count.



(a) $Ev1$ vs. Test Image No

(b) $Ev2$ vs. Test Image No

(c) Segment Count vs. Test Image No

(d) Computation Time vs. Test Image No

Figure 5-19 Quantitative comparison of the results of Jung's and Kim&Kim's algorithms for test images 1-8

5.5 Comparison of Region Merging Based MSS Approaches

In this section, we try to find the best region merging based multi-scale segmentation algorithm. The process that we follow for this purpose could be seen in Figure 5-20.

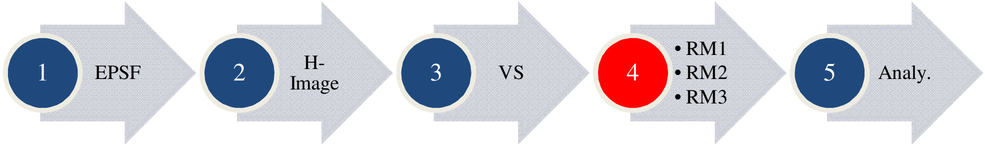
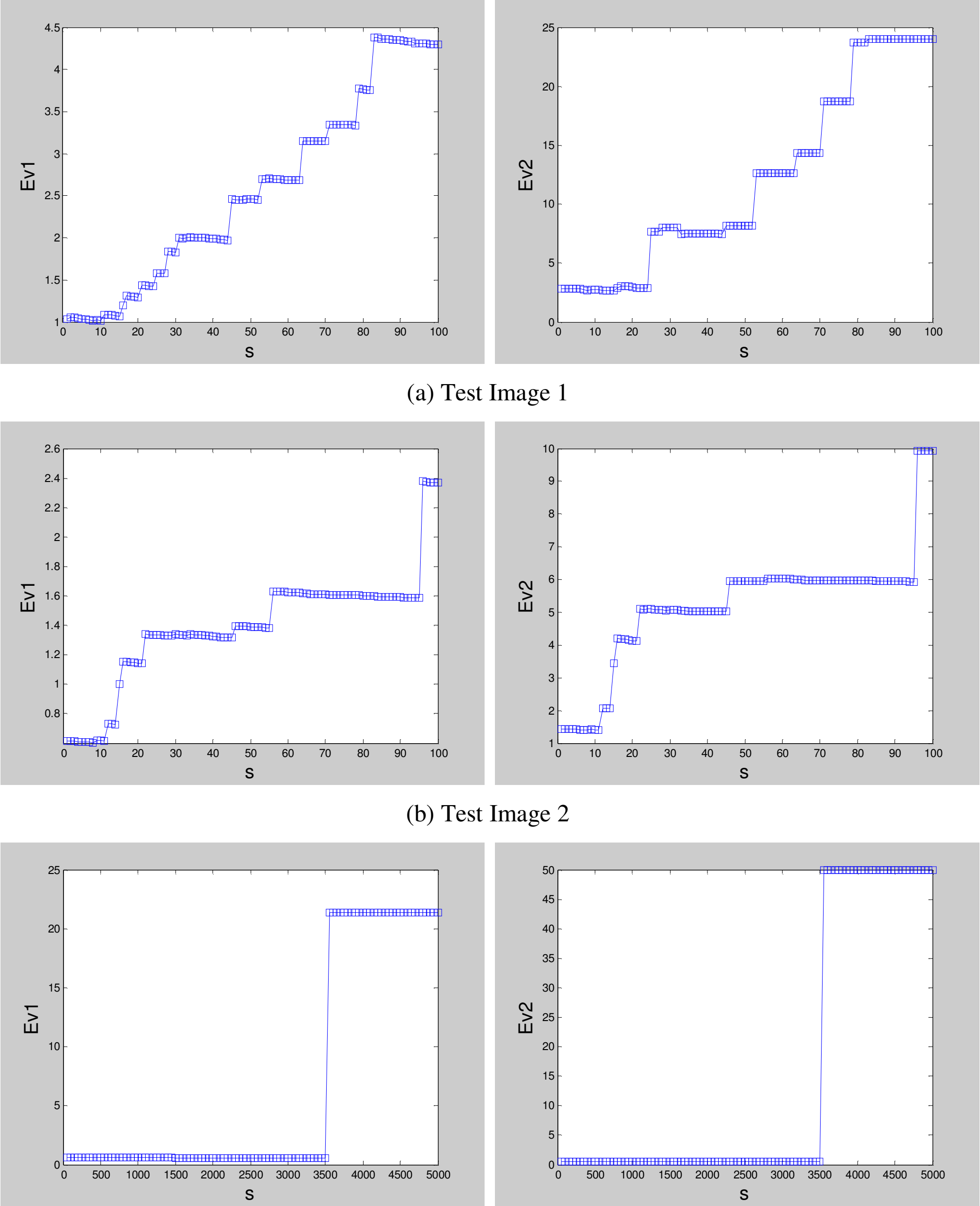


Figure 5-20 The method applied for finding the best region merging based multi-scale segmentation algorithm

We begin with first region merging based MSS algorithm (RM1) that is proposed by [12] and explained in Section 3.3.2.1. The parameters of this algorithm are the scale threshold, s, as the stopping criterion, spectral and smoothness heterogeneity weights ($w_{spectral}, w_{smoothness}$) and the weights of spectral bands (w_i). We opt for using the same weights for each spectral band to benefit equally from all spectral bands. For s, we evaluate 100 different values ranging from $s = 1$ to $s = 100$ with steps of 1 for test images 1-2 and 100 different values ranging from $s = 50$ to $s = 5000$ with steps of 50 for test image 3. The reason for this difference in the resolution of measurements lies in the objects that have different intrinsic scales in different test images. While roads present in the test images 1-2 have lower intrinsic scales, forest and bare ground in the test image 3 have higher intrinsic scales. In Figure 5-21, we can see the segmentation accuracy evaluation measurements for test images 1-3 with changing s and $w_{spectral} = 1$. Considering the results of Figure 5-21, $s = 24$ is selected for test image 1 because $Ev2$ measurements don't change until this threshold while the increase in $Ev1$ is tolerable. $s = 11$ is selected for test image 2. $s = 3500$ which is the optimum segmentation point is selected for test image 3.



(a) Test Image 1

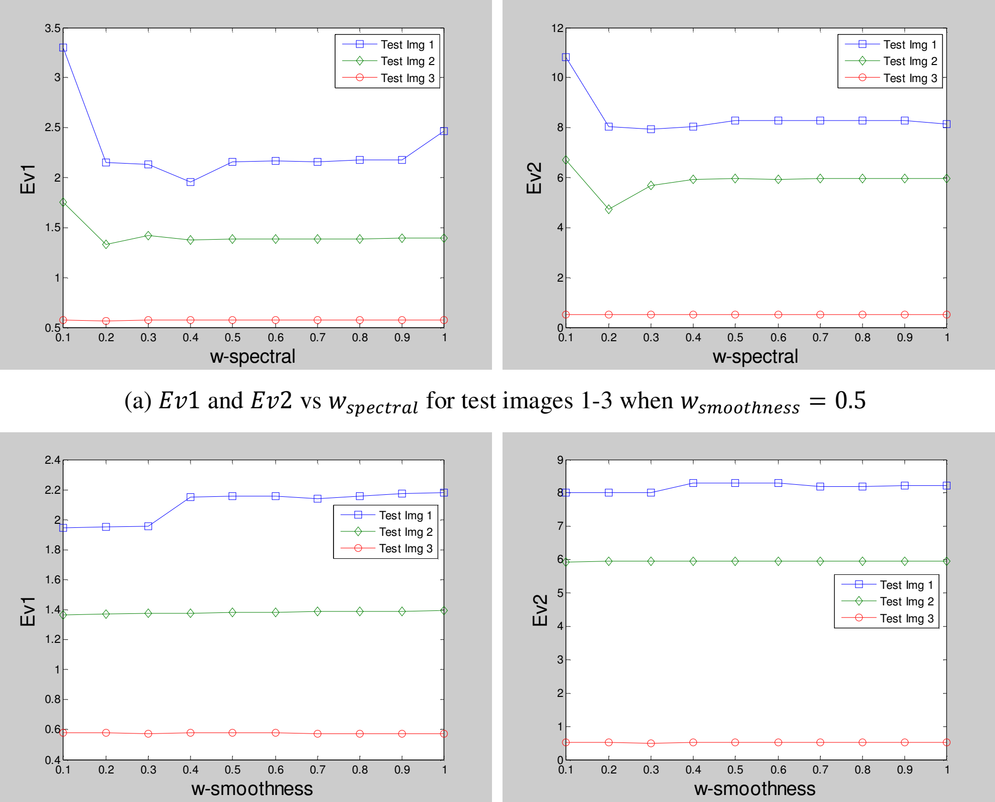
(b) Test Image 2

(c) Test Image 3

Figure 5-21 $Ev1$ and $Ev2$ measurements of first region merging segmentation results of test images 1-3

Now, we try to analyze the effect of $w_{spectral}$ and $w_{smoothness}$ on the accuracy of segmentation results. For this purpose, $Ev1$ and $Ev2$ measurements are taken at $s = 50$ for test images 1-2 and at $s = 2500$ for test image 3 ranging from 0.1 to 1 with the steps of 0.1 for both $w_{spectral}$ and $w_{smoothness}$ as shown in Figure 5-22. From these measurements, it can be seen that $w_{spectral}$ has a very small effect on the segmentation accuracy except for the very small values of it and $w_{smoothness}$

has almost no effect. Considering the results of Figure 5-22, we choose $w_{spectral} = 1$ for all test images.

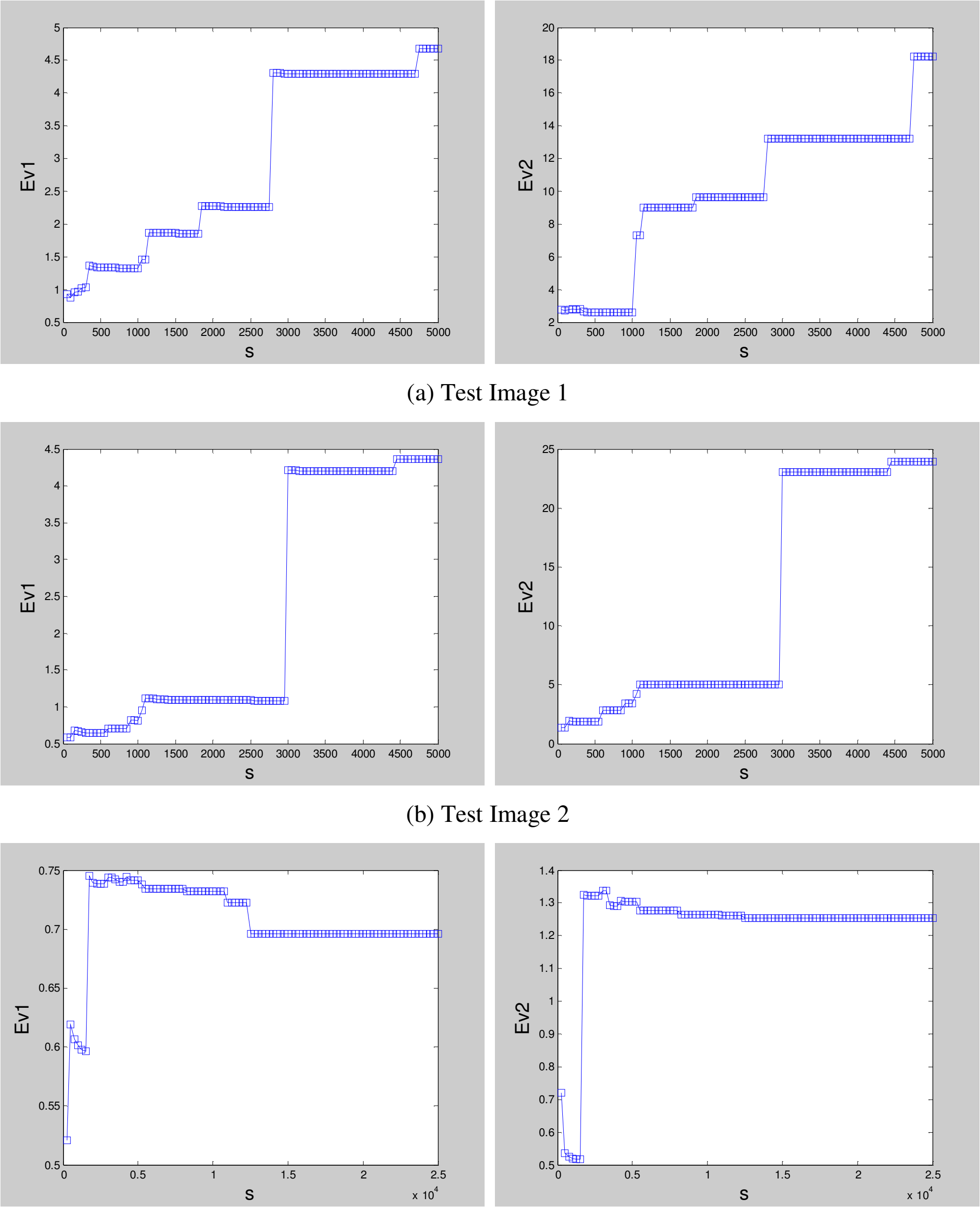


(a) $Ev1$ and $Ev2$ vs $w_{spectral}$ for test images 1-3 when $w_{smoothness} = 0.5$

(b) $Ev1$ and $Ev2$ vs $w_{smoothness}$ for test images 1-3 when $w_{spectral} = 0.5$

Figure 5-22 The effect of $w_{spectral}$ and $w_{smoothness}$ on the accuracy of segmentation results of test images 1-3

We continue with second region merging based MSS algorithm [14] (RM2). This algorithm is more or less the same with the previous algorithm except for the ordering for merging and the stopping criterion. Instead of a size threshold as in RM1, this algorithm incorporates a merging cost threshold as the scale threshold. We use $w_{spectral} = 1$ and equal spectral band weights, w_i, as in RM1 because the merging cost formulation is the same for both algorithms. For the scale threshold, we evaluate 100 different values ranging from $s = 50$ to $s = 5000$ with steps of 50 for test images 1-2 and 100 different values ranging from $s = 250$ to $s = 25000$ with steps of 250. This difference in the range of values has the same reason with the one of RM1 and is explained previously. In Figure 5-23, we can see the segmentation accuracy evaluation measurements for test images 1-3 with changing s. $s = 1000$ and $s = 2950$ are selected for test image 1-2, respectively, because the change in $Ev1$ and $Ev2$ measurements is tolerable up to these points. $s = 12500$ for test image 3 is selected by considering the minimization of over-segmentation because the accuracy reduction is negligible.



(a) Test Image 1

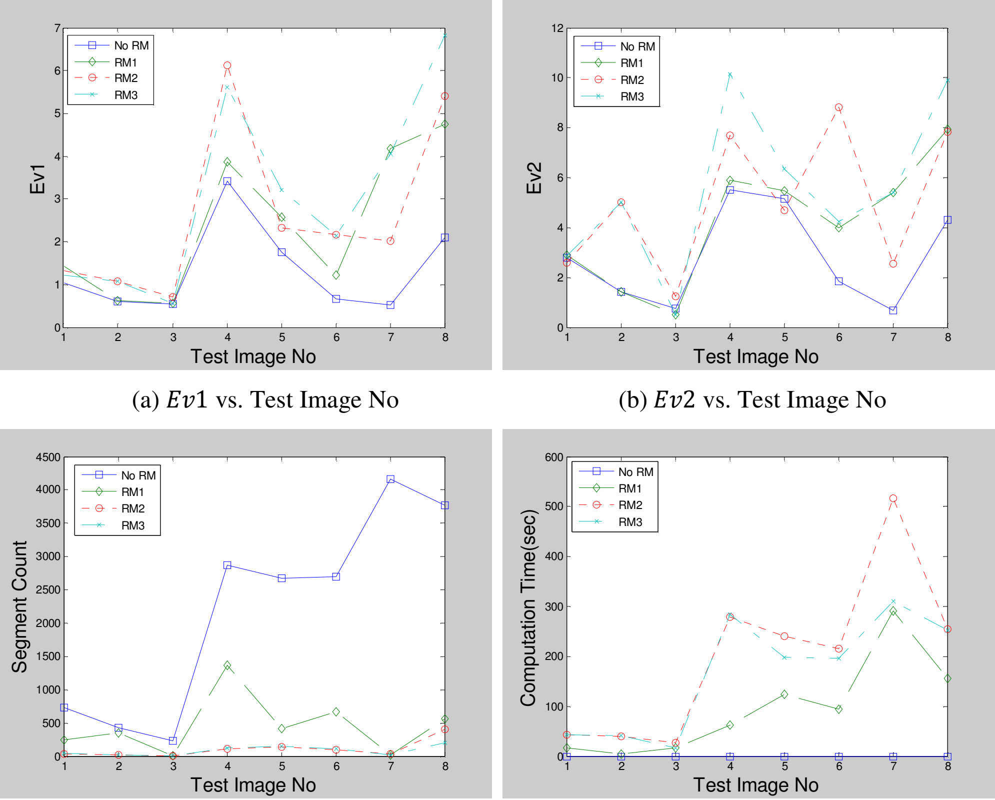
(b) Test Image 2

(c) Test Image 3

Figure 5-23 $Ev1$ and $Ev2$ measurements of second region merging segmentation results of test images 1-3

As the last algorithm, we evaluate third region merging based MSS algorithm proposed in this thesis (RM3) that is explained in detail in Section 4.5. This algorithm is the concatenation of first two region merging methods in other words the application of RM2 after RM1. Therefore, it incorporates the same parameter values found for RM1 but needs new parameter values for RM2. For the scale threshold, we evaluate 100 different values ranging from $s = 50$ to $s = 5000$ with steps of 50 for test images 1-2 and 100 different values ranging from $s = 250$ to $s = 25000$ with steps of 250 as in RM2 experiments. Now, in this case, scale thresholds of RM2 get smaller. Approaching to the best parameter finding process as in other two region merging algorithms, $s = 600$, $s = 3000$ and $s = 0$ are selected for test image 1,2 and 3, respectively.

In Figure 5-24, the accuracy and performance results of all region merging methods are presented for test images 1 to 8. For the comparison purposes, the quantitative results of no region merging applied case are also presented in Figure 5-24. For all test images 1 to 8, the advantage of RM2 over RM1 is apparent in terms of over-segmentation reduction at the expense of computational complexity. If we look at the results of test image 5, even if RM2 reduces the over-segmentation much higher than RM1, it also provides better segmentation accuracy than RM1. RM3, which is the combination of RM1 and RM2, provides segmentation results just similar to the RM2 in terms of segmentation accuracy and segment count. However, because it incorporates RM1 as the initial step, it is like a faster form of RM2 as apparently seen in the results of Figure 5-24d. As a result, we choose RM3 as the most effective region merging based MSS algorithm.



(a) $Ev1$ vs. Test Image No

(b) $Ev2$ vs. Test Image No

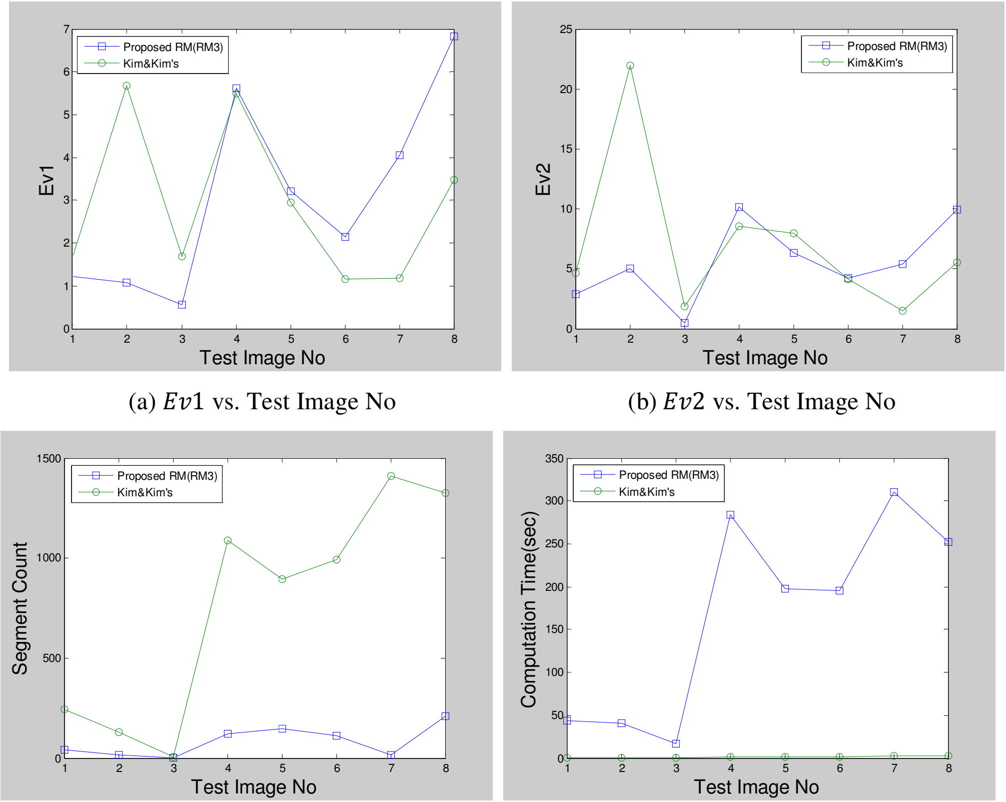
(c) Segment Count vs. Test Image No

(d) Computation Time vs. Test Image No

Figure 5-24 Quantitative comparison of the results of region merging algorithms for test images 1-8

5.6 Wavelet Based vs. Region Merging Based MSS

In Sections 5.4 and 5.5, we compare and evaluate two different algorithms for each category of MSS which are wavelet based and region merging based. We also propose a hybrid region merging based MSS algorithm. We come up with the best algorithms for each category by comparing the algorithms with respect to the segmentation accuracy and performance for test images 1 to 8. Now, in this section, we compare those best algorithms of each category which are Kim&Kim's algorithm and RM3 region merging algorithm. In Figure 5-25, accuracy and performance measurements of these algorithms could be found for test images 1 to 8. In these results, very high computational complexity and a little lower segmentation accuracy of proposed region merging algorithm are the disadvantages. However, the over-segmentation handling capability of Kim&Kim's algorithm is much deficient. The little lower segmentation accuracy of RM3 is probably having lower segment count. Therefore, we opt for region merging based MSS rather than wavelet based MSS.



(a) $Ev1$ vs. Test Image No

(b) $Ev2$ vs. Test Image No

(c) Segment Count vs. Test Image No

(d) Computation Time vs. Test Image No

Figure 5-25 Quantitative comparison of the results of Kim&Kim's and RM3 algorithms for test images 1-8

5.7 Comparison of Unsupervised Goodness Metrics

At this point, the best algorithm for every step of the segmentation approach is defined as can be seen in Figure 5-26. However, for a given remote sensed input image, the parameters of algorithms need to be adjusted manually for best results. In this section, we evaluate the methods to automate this segmentation approach.

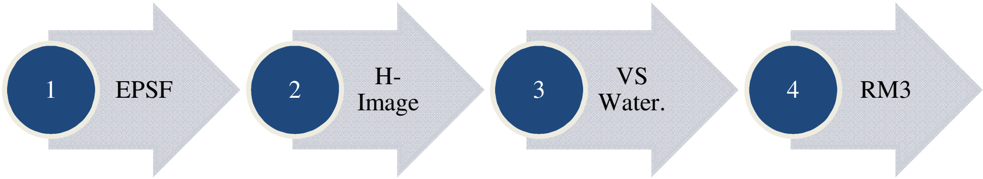
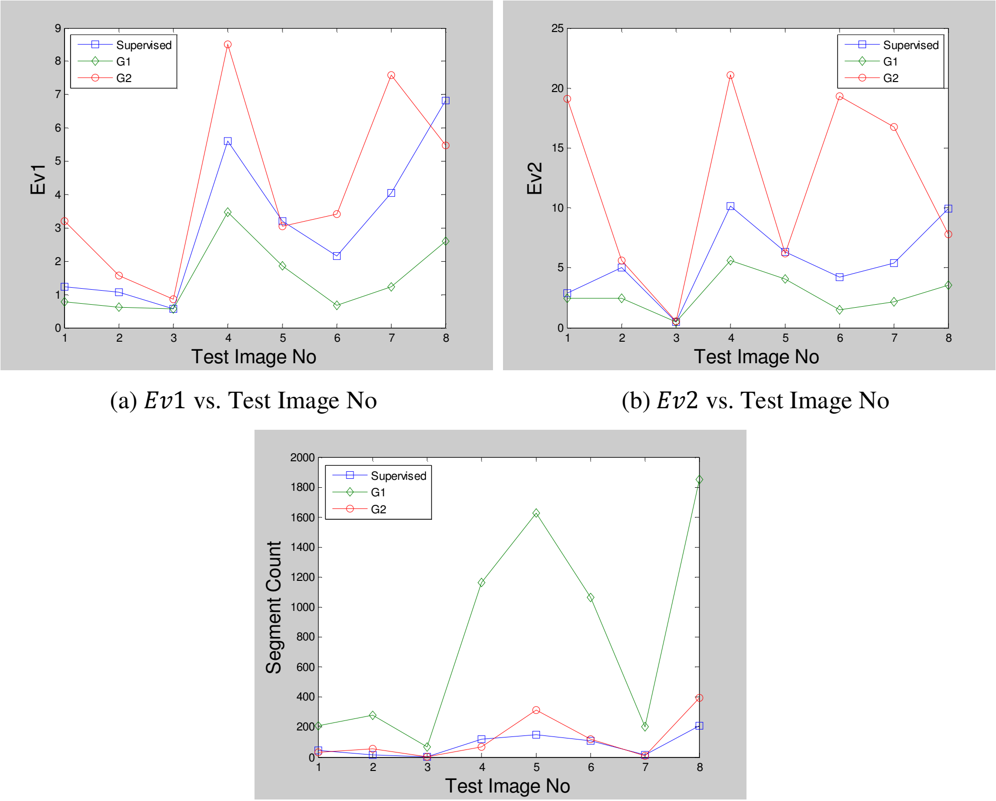


Figure 5-26 The flow of segmentation approach proposed in this thesis

To evaluate the effectiveness of the two unsupervised goodness metrics ($G1$ and $G2$) described in Section 3.4.1, we select the parameters of all algorithms that are previously found in a supervised manner with these metrics and compare the results. From the results presented in Figure 5-27, it is seen that $G1$ measure produces the best results in terms of segmentation accuracy. However, $G1$is highly unsatisfactory in handling over-segmentation problem. To keep segmentation accuracy at maximum, $G1$ applies less smoothing by filters and lower scale thresholds by region merging. On the other hand, $G2$ generates superior segmentation results from the over-segmentation reduction perspective at the expense of decreasing segmentation accuracy. However, it is seen that $G2$ produces more similar visual results to supervised results than $G1$. Therefore, we incorporate $G2$ metric to our proposed segmentation flow and make the approach a fully automatic one.



(a) $Ev1$ vs. Test Image No

(b) $Ev2$ vs. Test Image No

(c) Segment Count vs. Test Image No

Figure 5-27 Comparison of the segmentation results produced by following manually set parameters and parameters set by $G1$ and $G2$

Overall accuracy and performance results of the proposed approach when applied manually and automatically (with $G2$ metric) for the test images 1-8 can be seen in Table 5-3.

Table 5-3 Final results of the proposed approach for the test images 1-8 when applied manually and automatic

Test Image	*Algorithm*	*Seg. Cnt.*	*Ev1*	*Ev2*
1	Manual	42	1.23	2.86
	Automatic	32	3.21	19.08
2	Manual	16	1.08	5.01
	Automatic	55	1.57	5.64
3	Manual	2	0.57	0.50
	Automatic	2	0.86	0.56
4	Manual	121	5.62	10.15
	Automatic	67	8.51	21.05
5	Manual	148	3.21	6.35
	Automatic	311	3.04	6.21
6	Manual	111	2.14	4.25
	Automatic	120	3.42	19.32
7	Manual	16	4.05	5.41
	Automatic	8	7.59	16.75
8	Manual	210	6.83	9.90
	Automatic	396	5.48	7.83

5.8 Comparison with Previous Literature

In this section, we compare the performance of the proposed approach in this thesis with the performances of two similar approaches in the previous literature. We measure the performances of all approaches by using test images 1-8 that have ground truth images and by using test images 9-20 that do not have. During the comparison by using test images 1-8, we compare the results generated by supervised segmentation evaluation metrics, on the other hand, while comparing the algorithms by using test images 9-20, we compare the results generated by unsupervised segmentation evaluation metrics. In this section, first, we present the comparison results for test images 1-8 and then, we present the comparison results for test images 9-20.

One of the approaches to compare [12] is using a segmentation process flow as shown in Figure 5-28. Actually, the gradient magnitude computation algorithm is not clearly specified in the paper but we used MSGM algorithm for that step. The region merging algorithm used in this paper is corresponding to our first region merging algorithm coded as RM1.

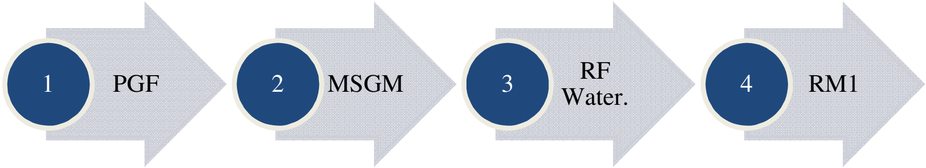


Figure 5-28 The flow of segmentation approach proposed by [12]

The second approach for comparison [14] is using a segmentation process flow as shown in Figure 5-29. No pre-filtering step is incorporated in this paper. The region merging algorithm used in this paper is corresponding to our second region merging algorithm coded as RM2.

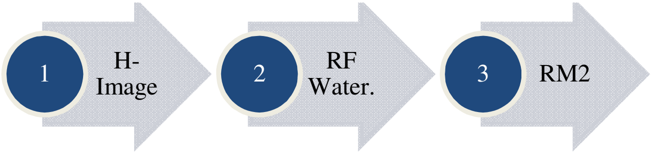
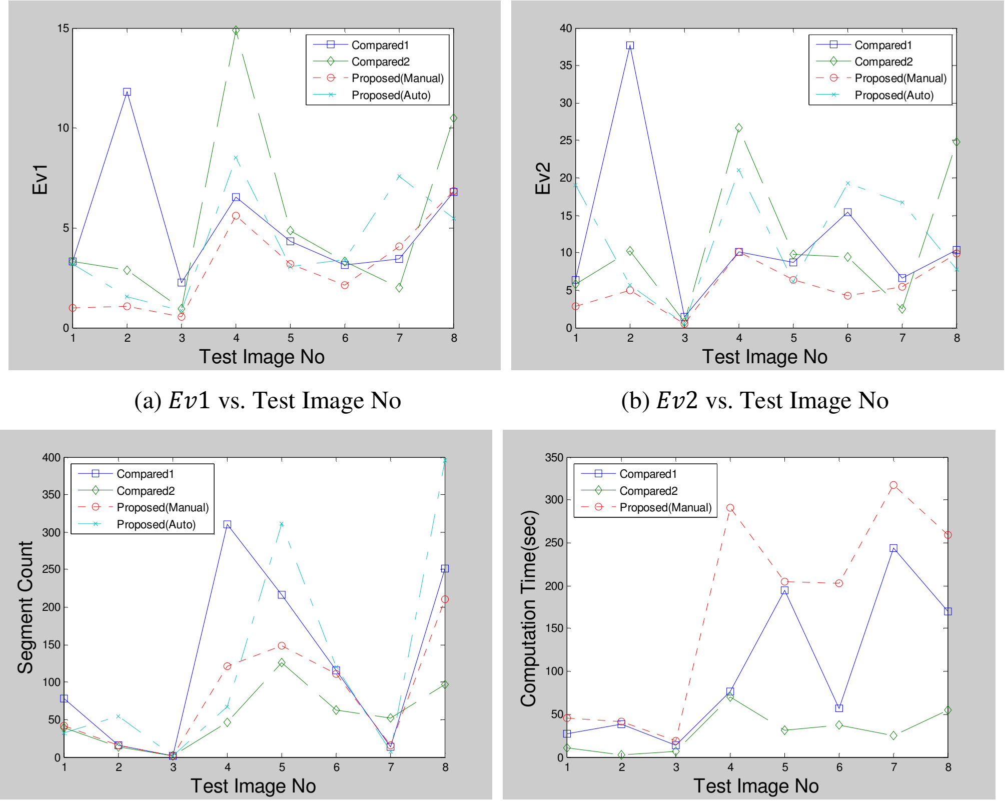


Figure 5-29 The flow of segmentation approach proposed by [14]

In Figure 5-30, accuracy and performance comparison of proposed approach (manual and auto), the method proposed by [12] (called as 'Compared1') and the method proposed by [14] (called as 'Compared2') for test images 1-8 could be seen. As can be seen from these graphics, proposed method with manually adjusted parameters provides the best segmentation results in terms of segmentation accuracy in nearly all test images. Average of segmentation accuracies weighted by image size for all test images could be seen in Table 5-4. Also, over-segmentation handling of proposed approach when used manually is more successful than [12]. However, [14] is the most successful one in reduction of over-segmentation at the expense of lower accuracy. From the computational complexity perspective, proposed method is unsatisfactory by considering opponent methods. Lastly, although proposed method when used by automatic adjustment of parameters produces lower accuracy, it provides visually acceptable results as shown in Figure 5-31 to Figure 5-34. Computation time of proposed approach in automatic mode is not presented because all possible selections of all parameters are evaluated in this mode.



(a) $Ev1$ vs. Test Image No

(b) $Ev2$ vs. Test Image No

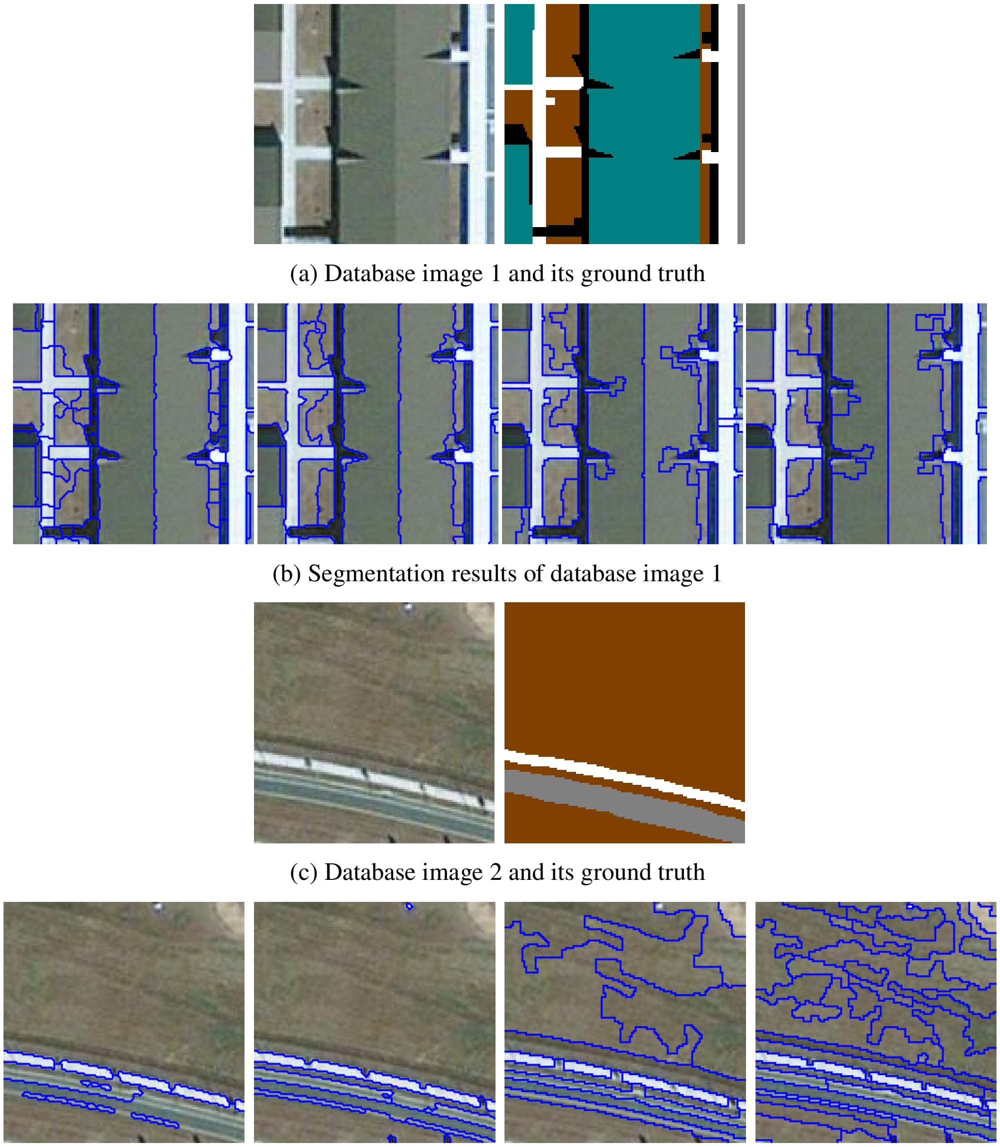
(c) Segment Count vs. Test Image No

(d) Computation Time vs. Test Image No

Figure 5-30 Quantitative comparison of the results of proposed approach and similar approaches in the literature for test images 1-8

Table 5-4 Average segmentation accuracies of the results of proposed approach and similar approaches in the literature for test images 1-8

Algorithm	*Avg Ev1*	*Avg Ev2*
[12] (Compared1)	%4.97	%10.92
[14] (Compared2)	%6.50	%13.49
Proposed(Manual)	**%3.92**	**%6.64**
Proposed(Automatic)	%5.12	%13.47



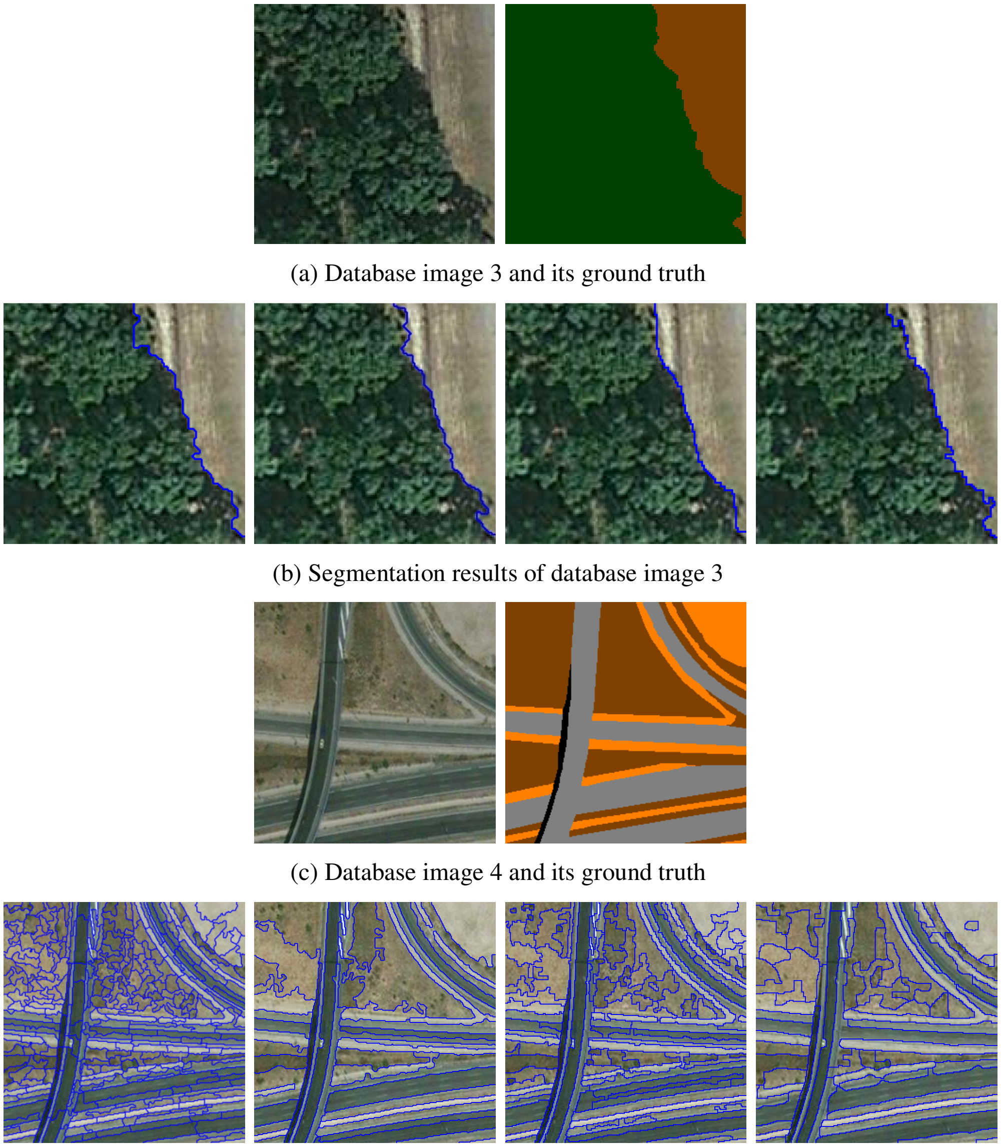
(a) Database image 1 and its ground truth

(b) Segmentation results of database image 1

(c) Database image 2 and its ground truth

(d) Segmentation results of database image 2

Figure 5-31 Test images 1 and 2 of database and their visual segmentation results; 1st column shows the result by [12], 2nd column shows the result by [14], 3rd column shows the result by proposed approach (manual) and 4th column shows the result by proposed approach (automatic) for (b) and (d)



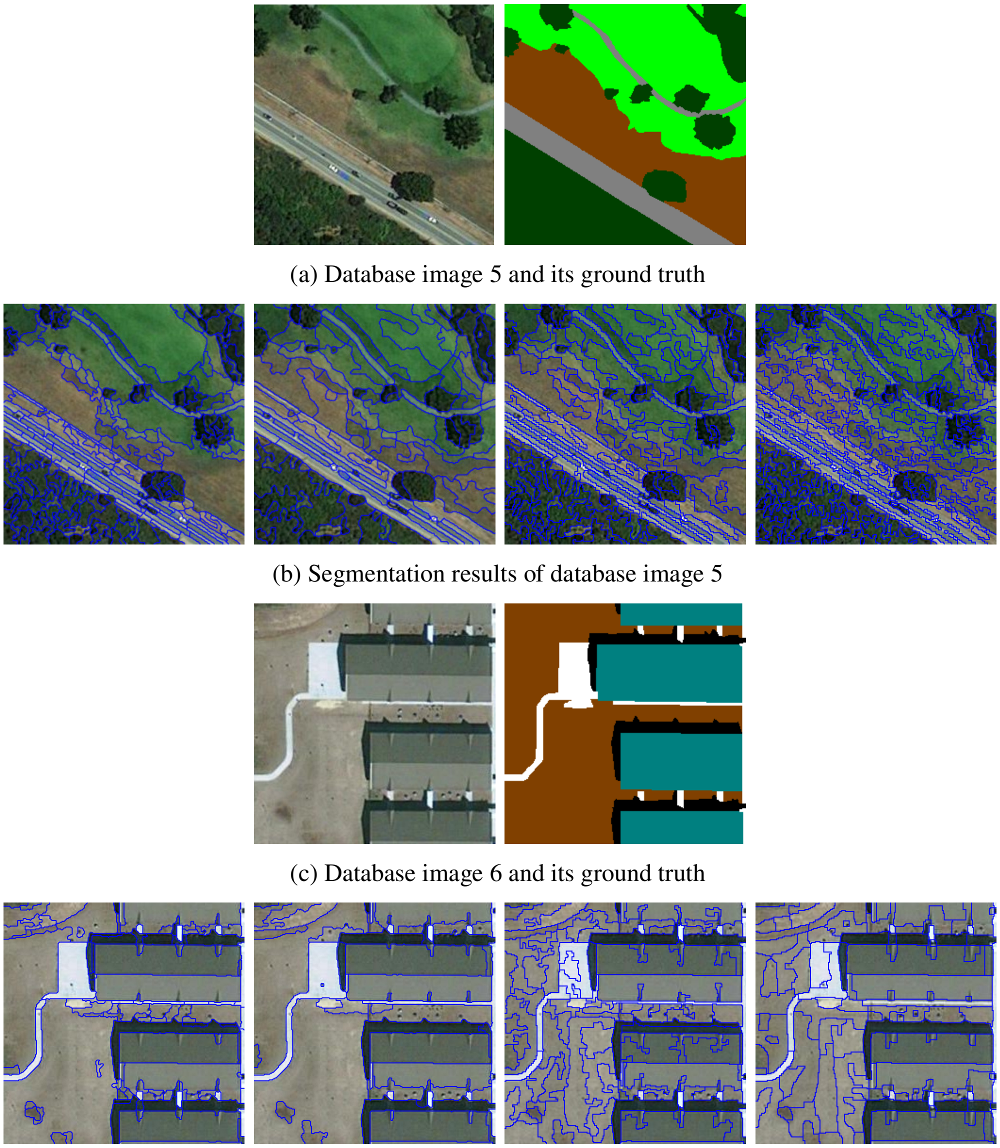
(a) Database image 3 and its ground truth

(b) Segmentation results of database image 3

(c) Database image 4 and its ground truth

(d) Segmentation results of database image 4

Figure 5-32 Test images 3 and 4 of database and their visual segmentation results; 1st column shows the result by [12], 2nd column shows the result by [14], 3rd column shows the result by proposed approach (manual) and 4th column shows the result by proposed approach (automatic) for (b) and (d)



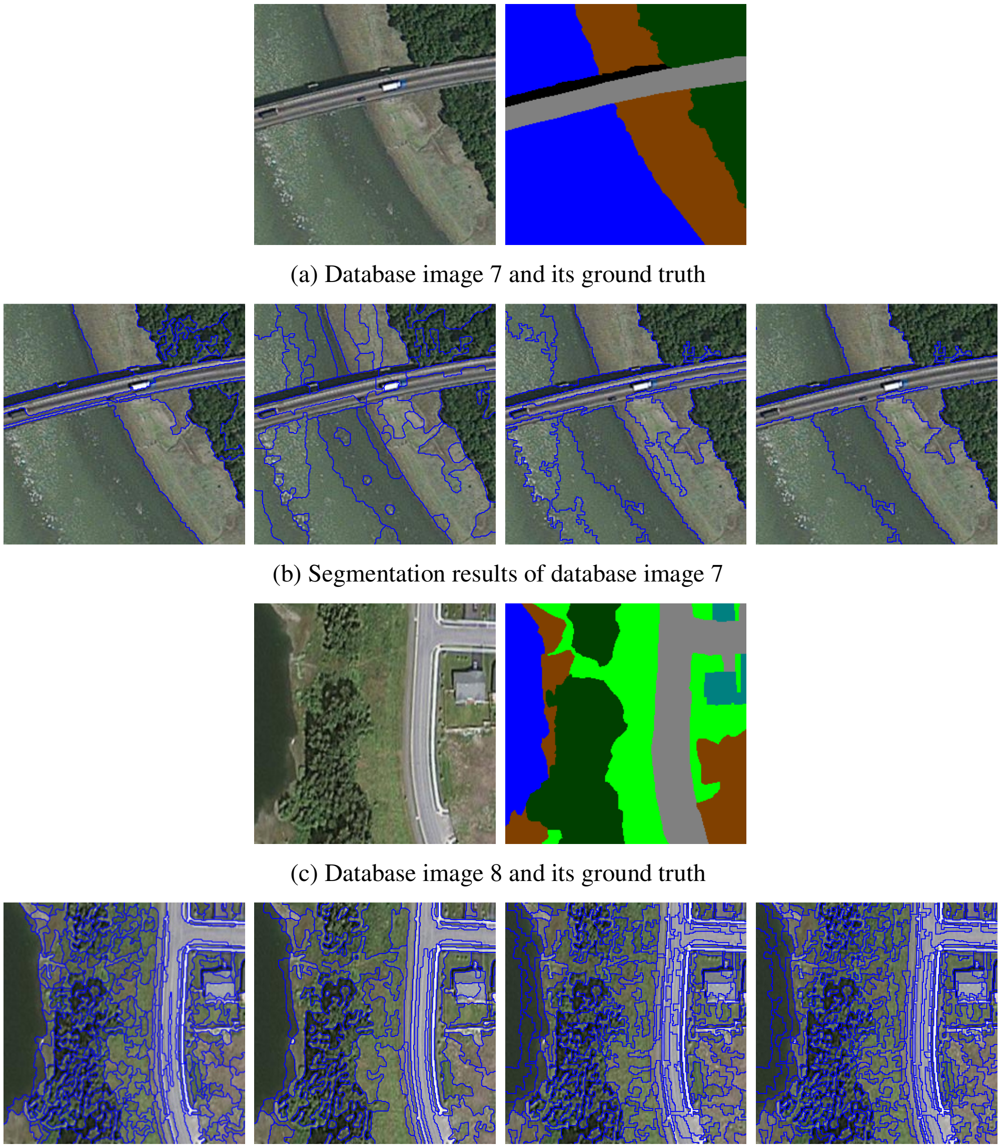
(a) Database image 5 and its ground truth

(b) Segmentation results of database image 5

(c) Database image 6 and its ground truth

(d) Segmentation results of database image 6

Figure 5-33 Test images 5 and 6 of database and their visual segmentation results; 1st column shows the result by [12], 2nd column shows the result by [14], 3rd column shows the result by proposed approach (manual) and 4th column shows the result by proposed approach (automatic) for (b) and (d)



(a) Database image 7 and its ground truth

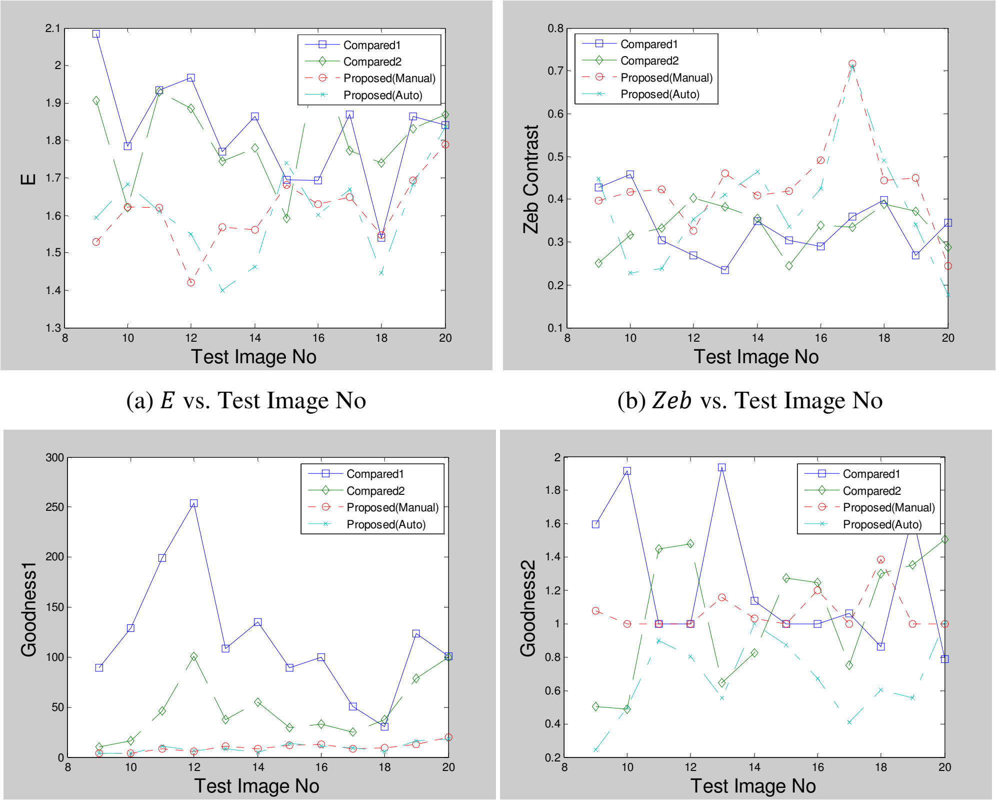
(b) Segmentation results of database image 7

(c) Database image 8 and its ground truth

(d) Segmentation results of database image 8

Figure 5-34 Test images 7 and 8 of database and their visual segmentation results; 1st column shows the result by [12], 2nd column shows the result by [14], 3rd column shows the result by proposed approach (manual) and 4th column shows the result by proposed approach (automatic) for (b) and (d)

In Figure 5-35, accuracy and performance comparison of proposed approach (manual and auto), the method proposed by [12] (called as 'Compared1') and the method proposed by [14] (called as 'Compared2') provided by four unsupervised segmentation evaluation metrics for test images 9-20 could be seen. Average of segmentation accuracies weighted by image size for all test images could be seen in Table 5-5. Except for the Zeb contrast metric, all metrics produce lower values for better segmentations. All metrics try to balance over and under segmentation and try to get an optimum segmentation result. Although the segmentation performance of the algorithms changes from image to image, proposed method with its manual and automatic settings produces better results than the other two opponent algorithms in the literature at the average as seen in Table 5-5.



(a) E vs. Test Image No

(b) Zeb vs. Test Image No

(c) Goodness1 vs. Test Image No

(d) Goodness2 vs. Test Image No

Figure 5-35 Quantitative comparison of the results of proposed approach and similar approaches in the literature for test images 9-20

Table 5-5 Average segmentation accuracies of the results of proposed approach and similar approaches in the literature for test images 9-20

Algorithm	*Avg. E*	*Avg. Zeb*	*Avg. Goodness1*	*Avg. Goodness2*
[12] (Compared1)	1.81	0.32	118.59	1.17
[14] (Compared2)	1.82	0.34	52.29	1.15
Proposed(Manual)	1.61	**0.44**	10.41	1.08
Proposed(Automatic)	**1.60**	0.39	**10.22**	**0.72**

Finally, the segmentation results which are generated by applying our proposed segmentation algorithm with automatic parameter selection to our database images 9-20 are shown in Figure 5-36 to Figure 5-47. The results of images 11-20 are obtained by directly segmenting the corresponding input image as with images 9-10. However, when the results are put in one piece for the images 11-20, it is seen that the segment boundaries look as if they are corrupted. Therefore, the segmentation results of the images 11-20 are divided into two pieces and presented in that way.

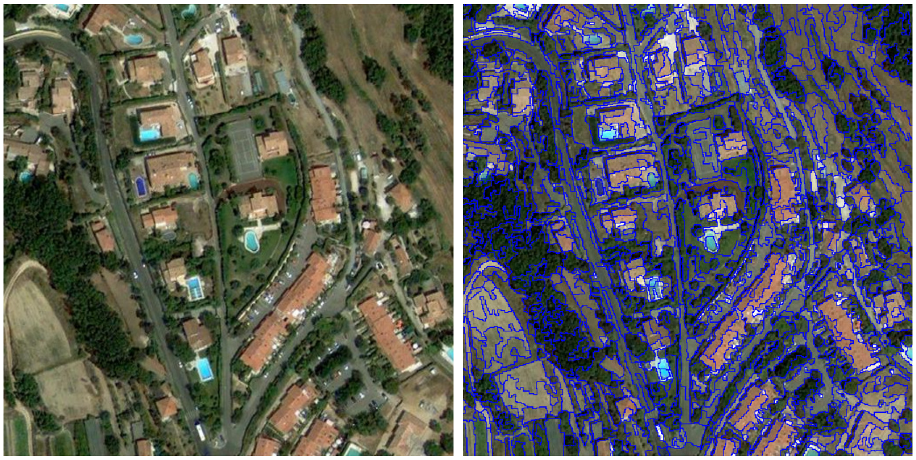


Figure 5-36 Test image 9 and its segmentation result

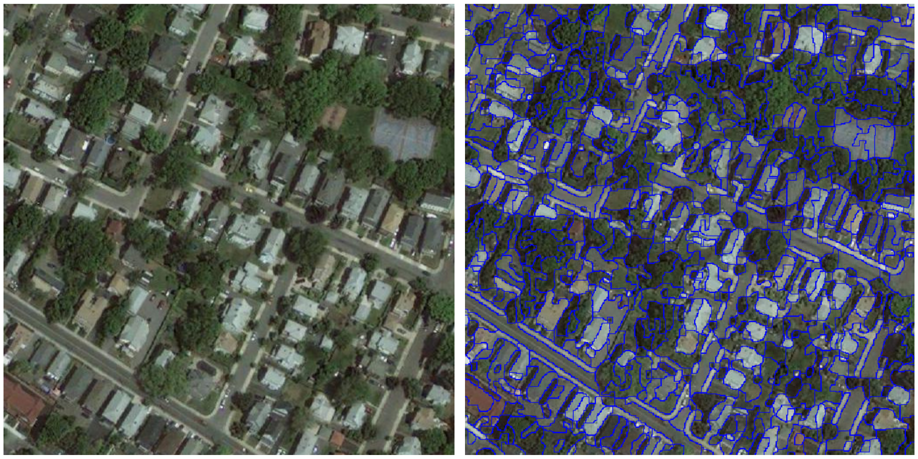


Figure 5-37 Test image 10 and its segmentation result

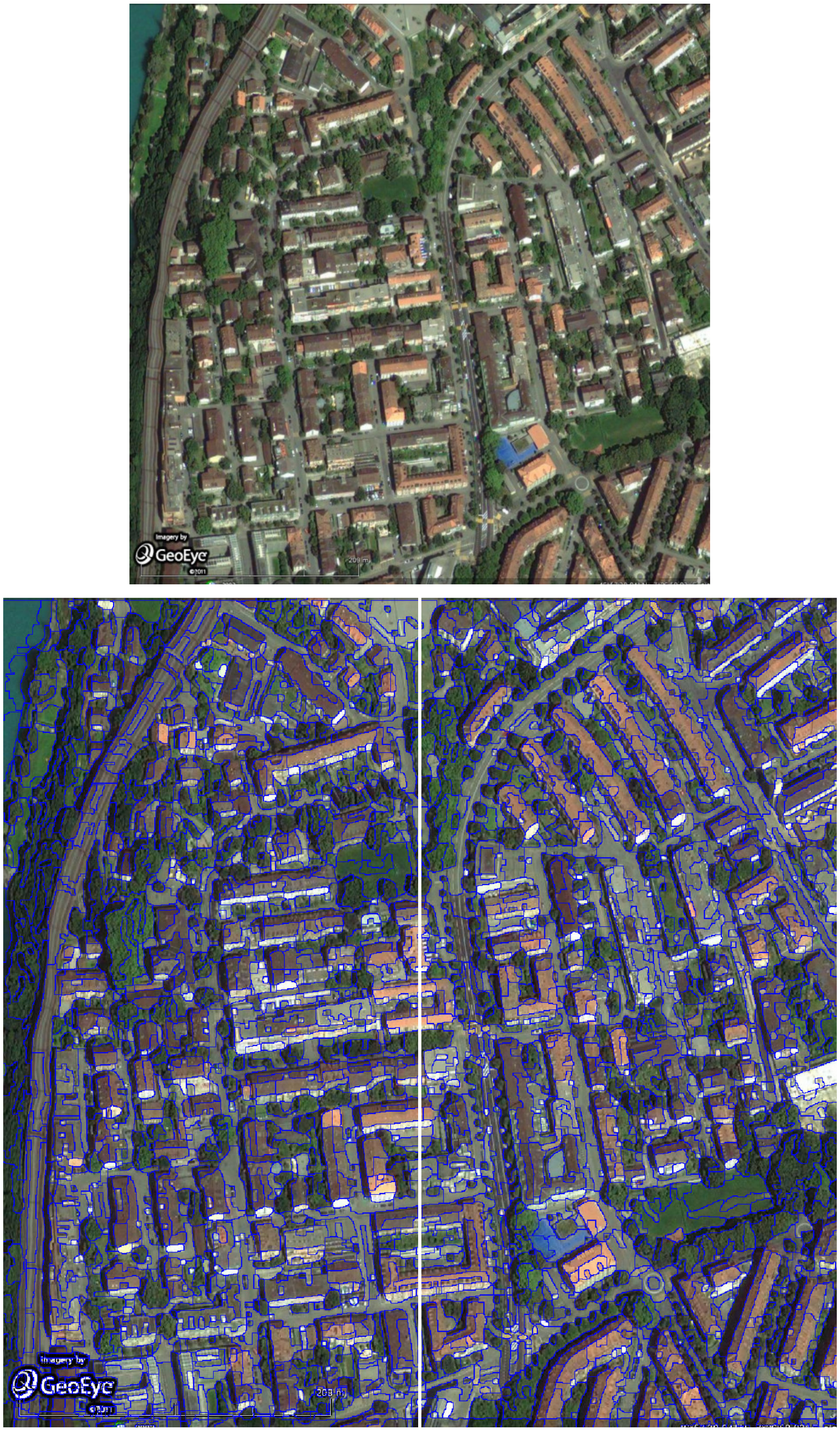


Figure 5-38 Test image 11 and its segmentation result

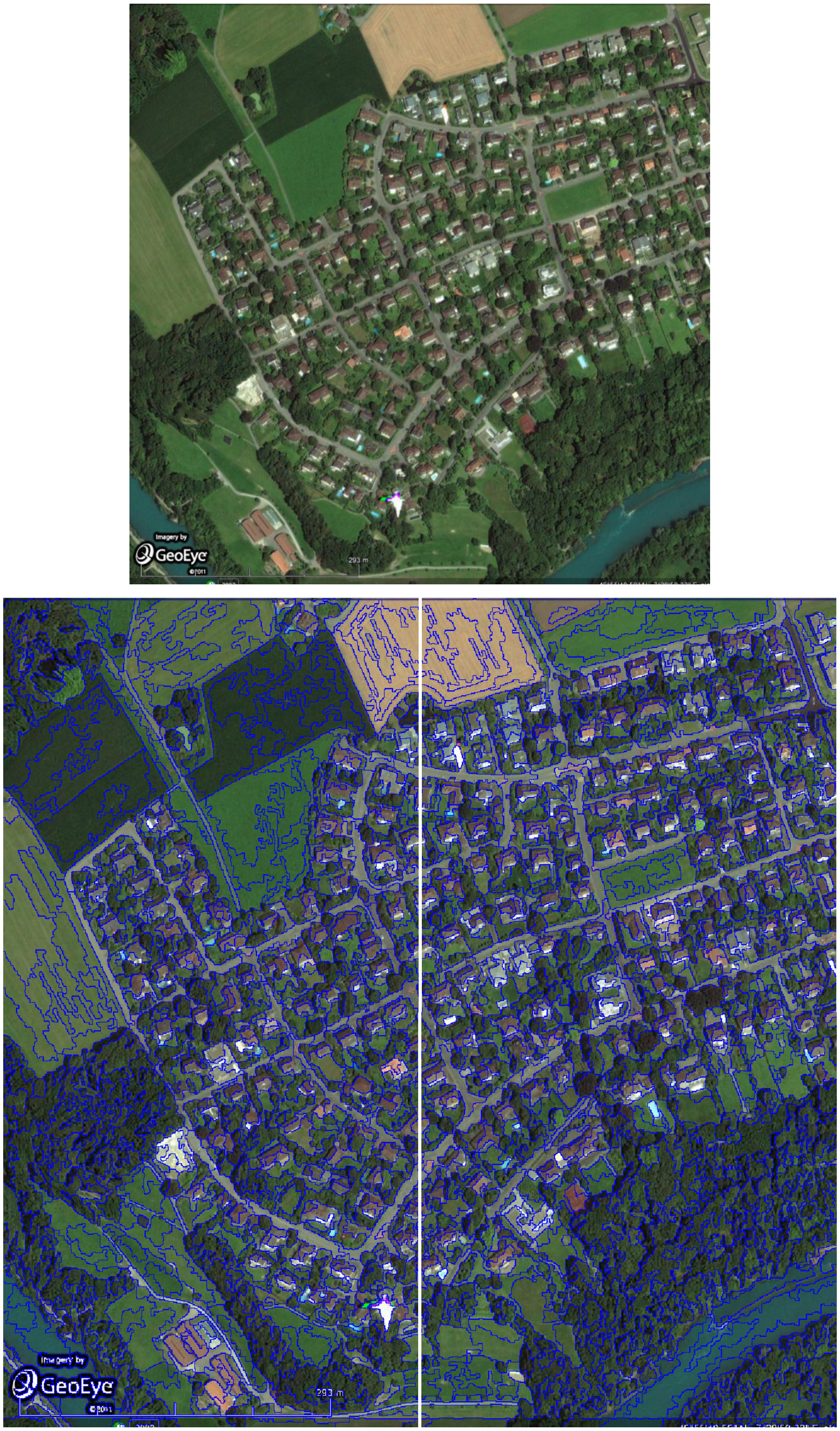


Figure 5-39 Test image 12 and its segmentation result

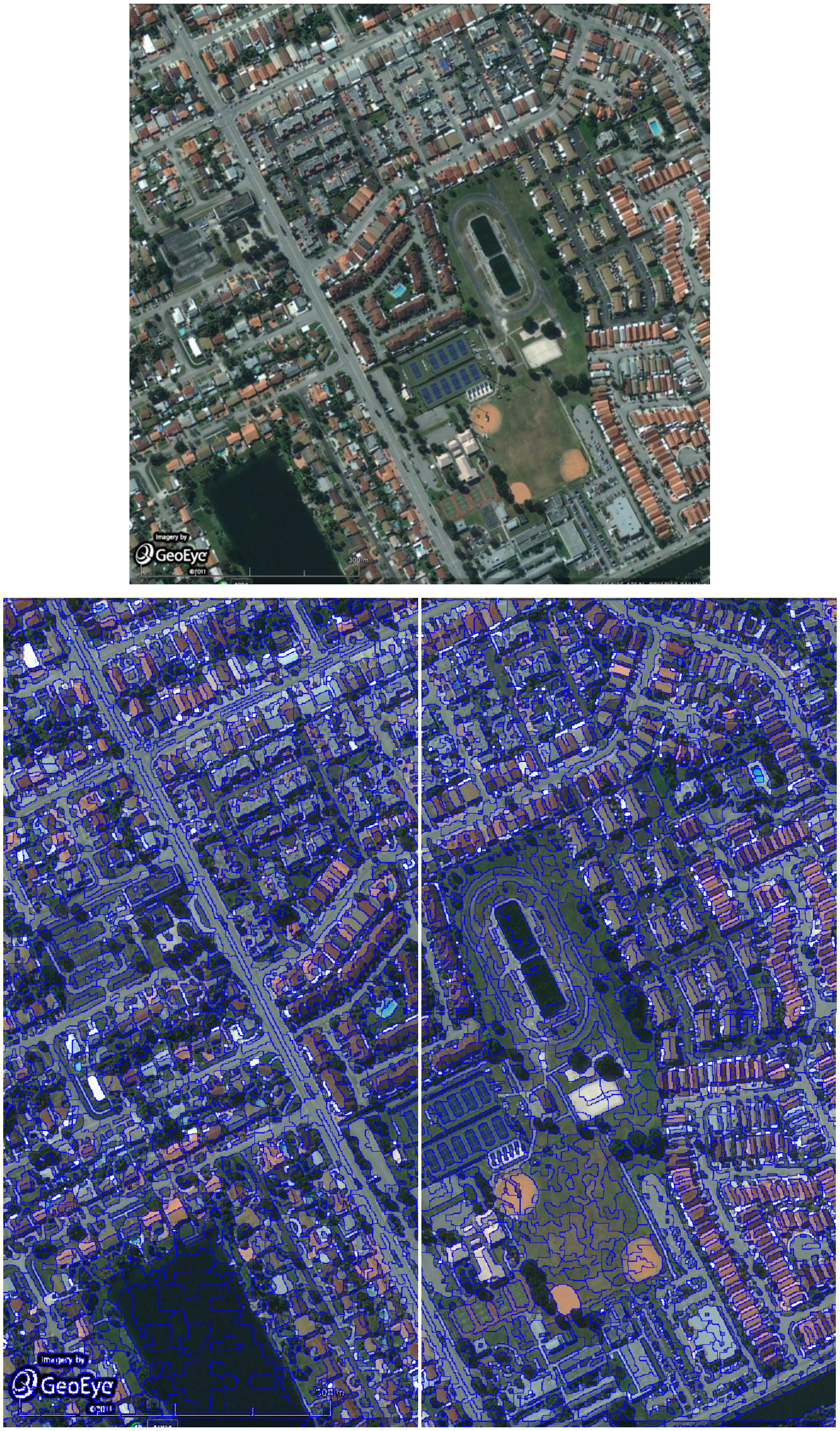


Figure 5-40 Test image 13 and its segmentation result

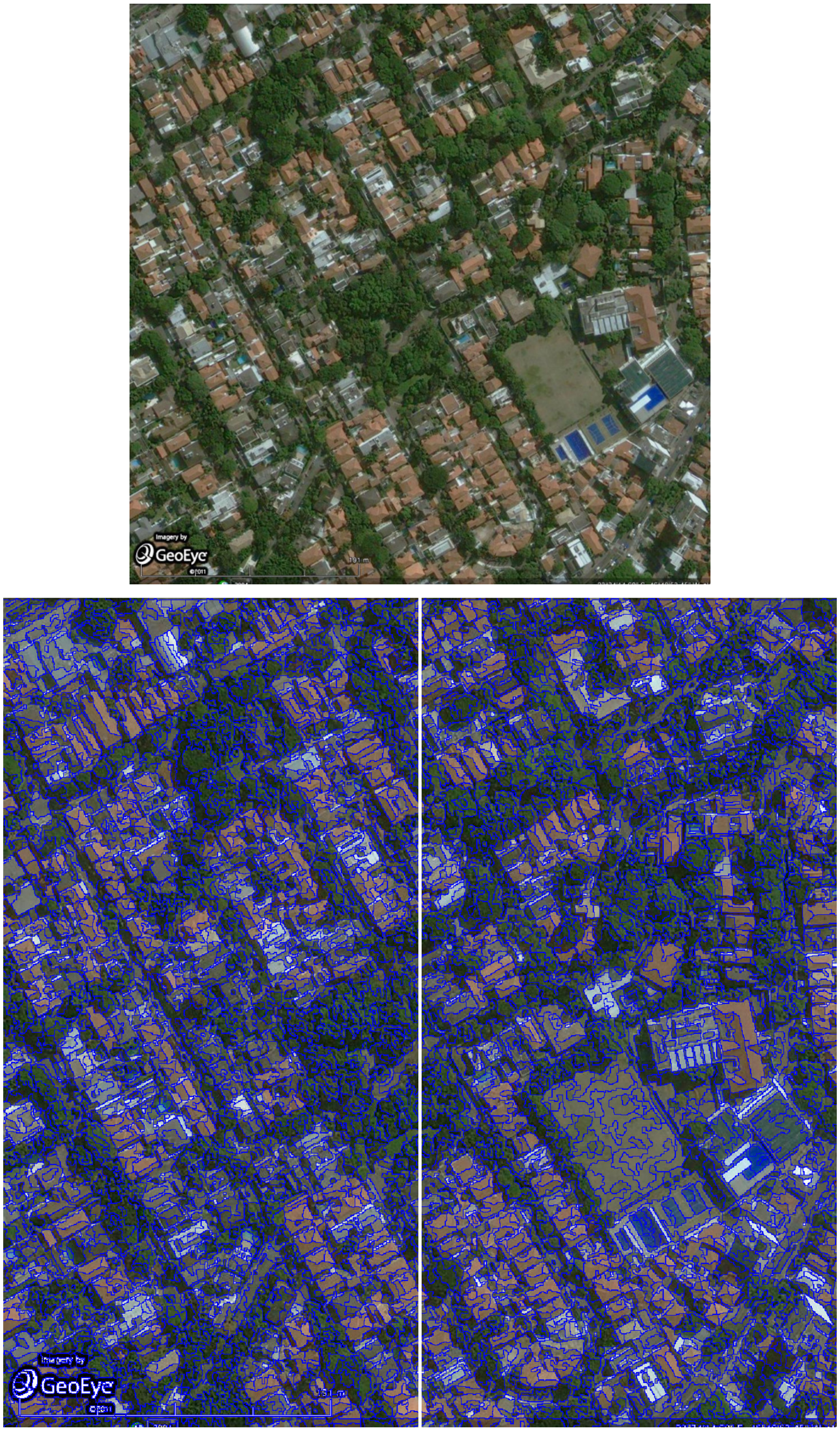


Figure 5-41 Test image 14 and its segmentation result

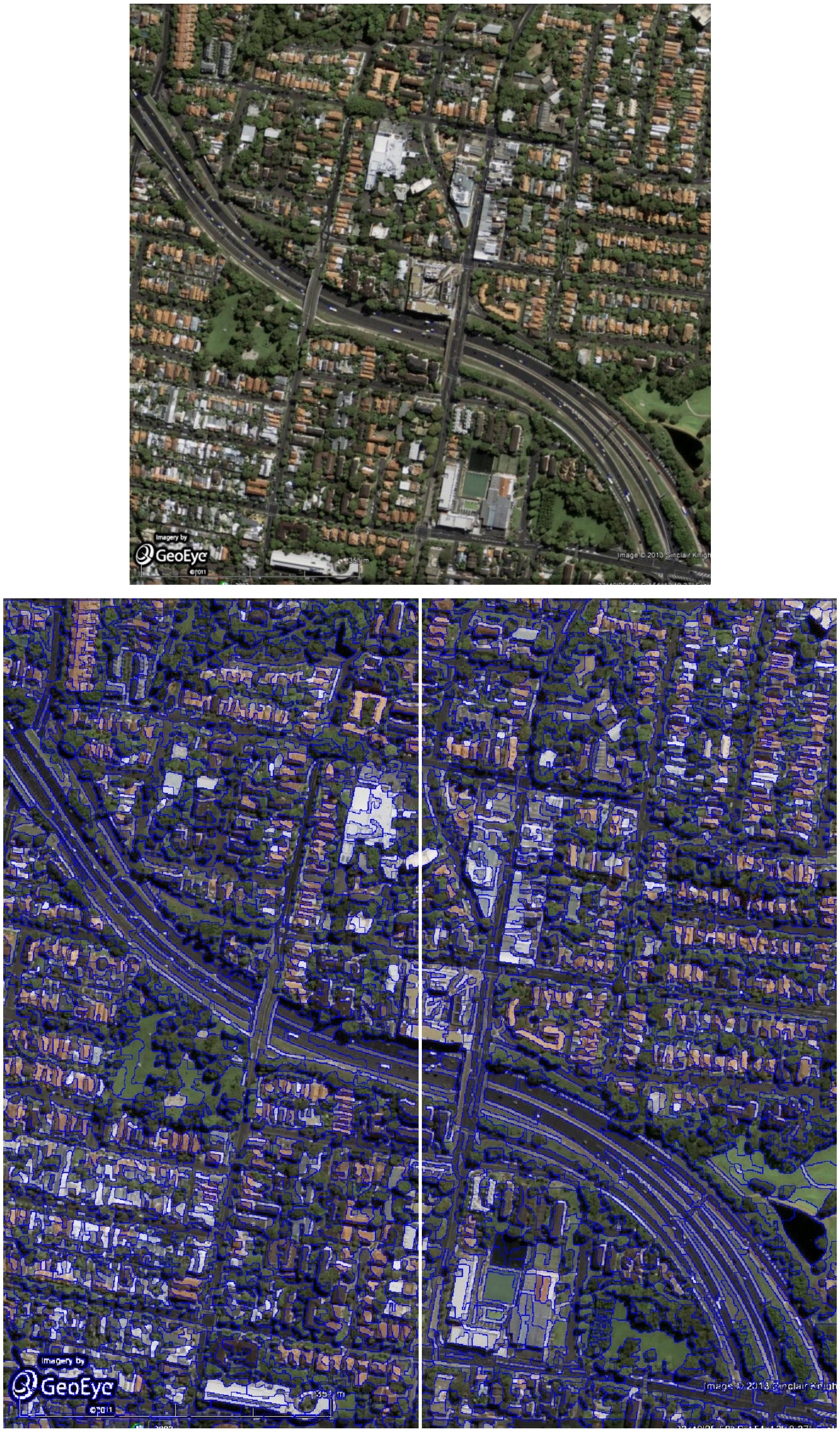


Figure 5-42 Test image 15 and its segmentation result

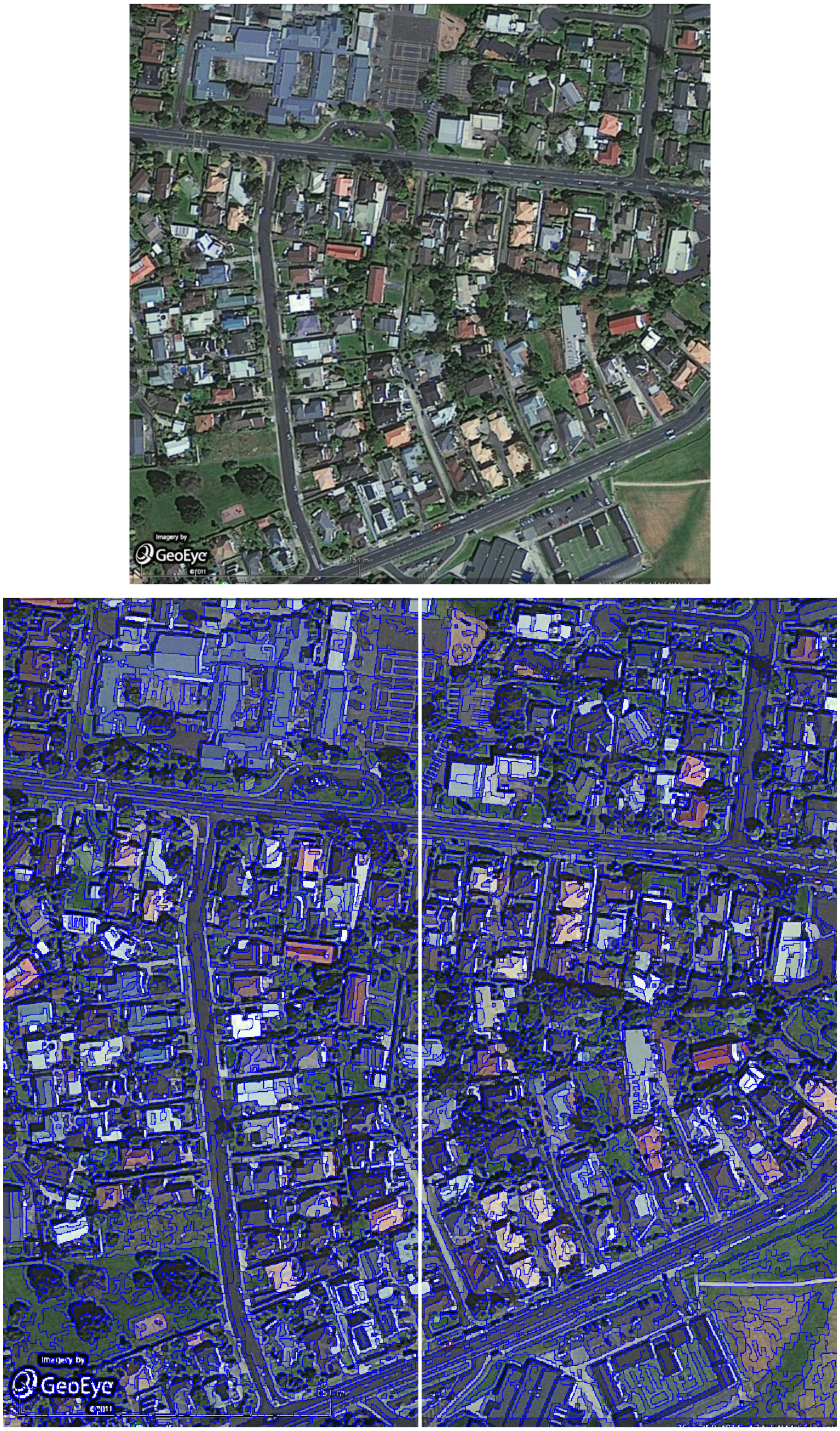


Figure 5-43 Test image 16 and its segmentation result

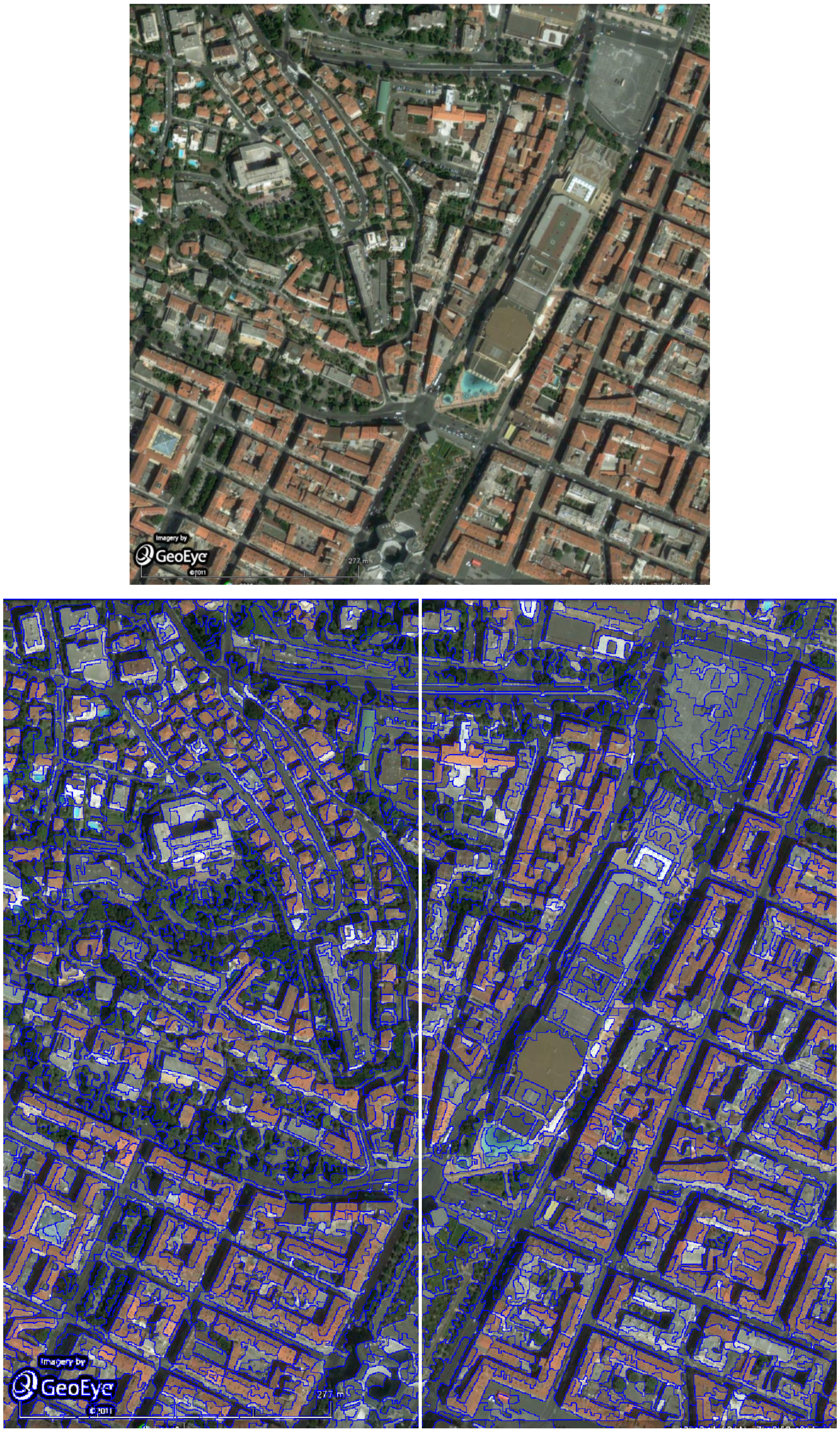


Figure 5-44 Test image 17 and its segmentation result

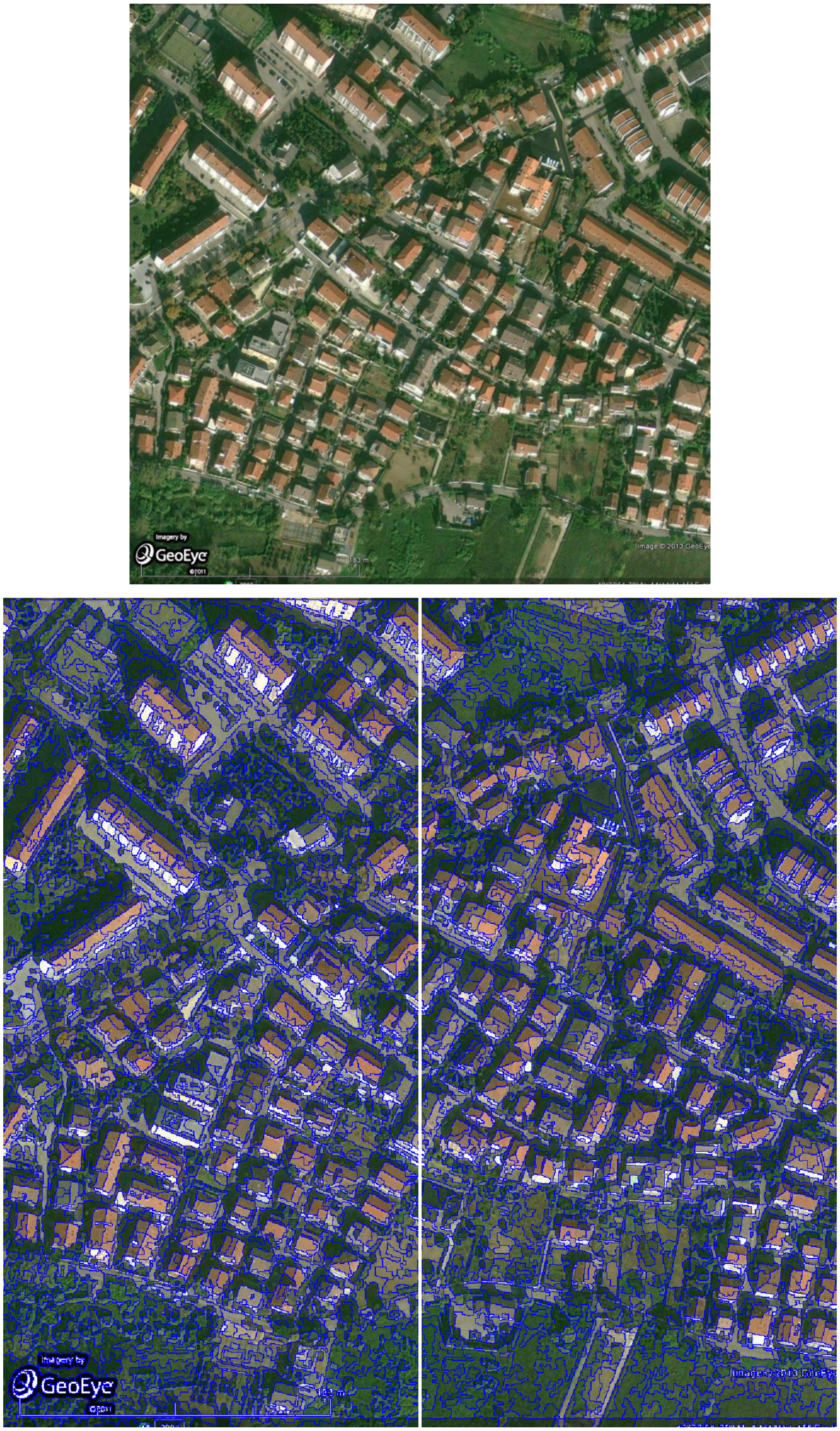


Figure 5-45 Test image 18 and its segmentation result

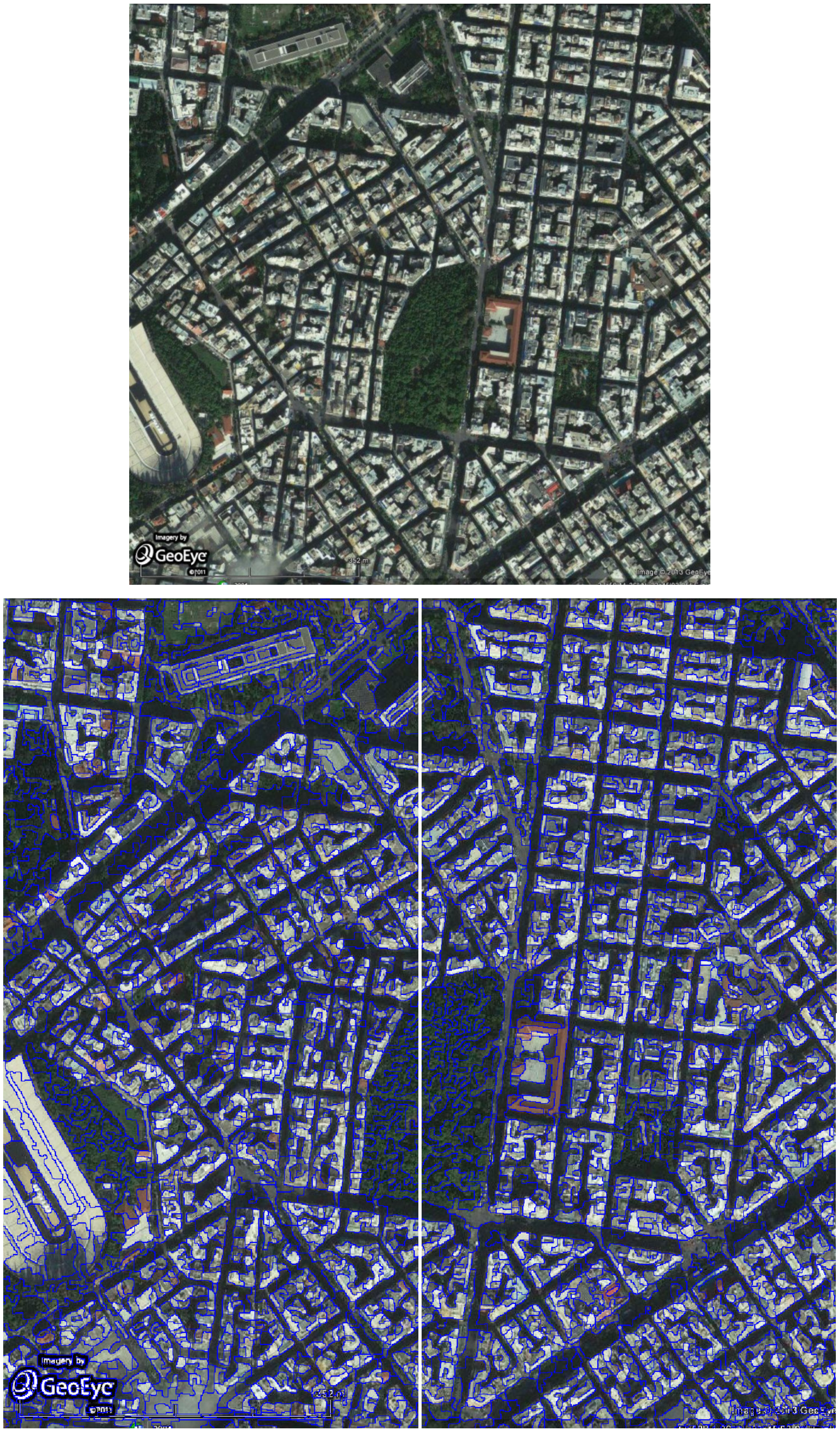


Figure 5-46 Test image 19 and its segmentation result

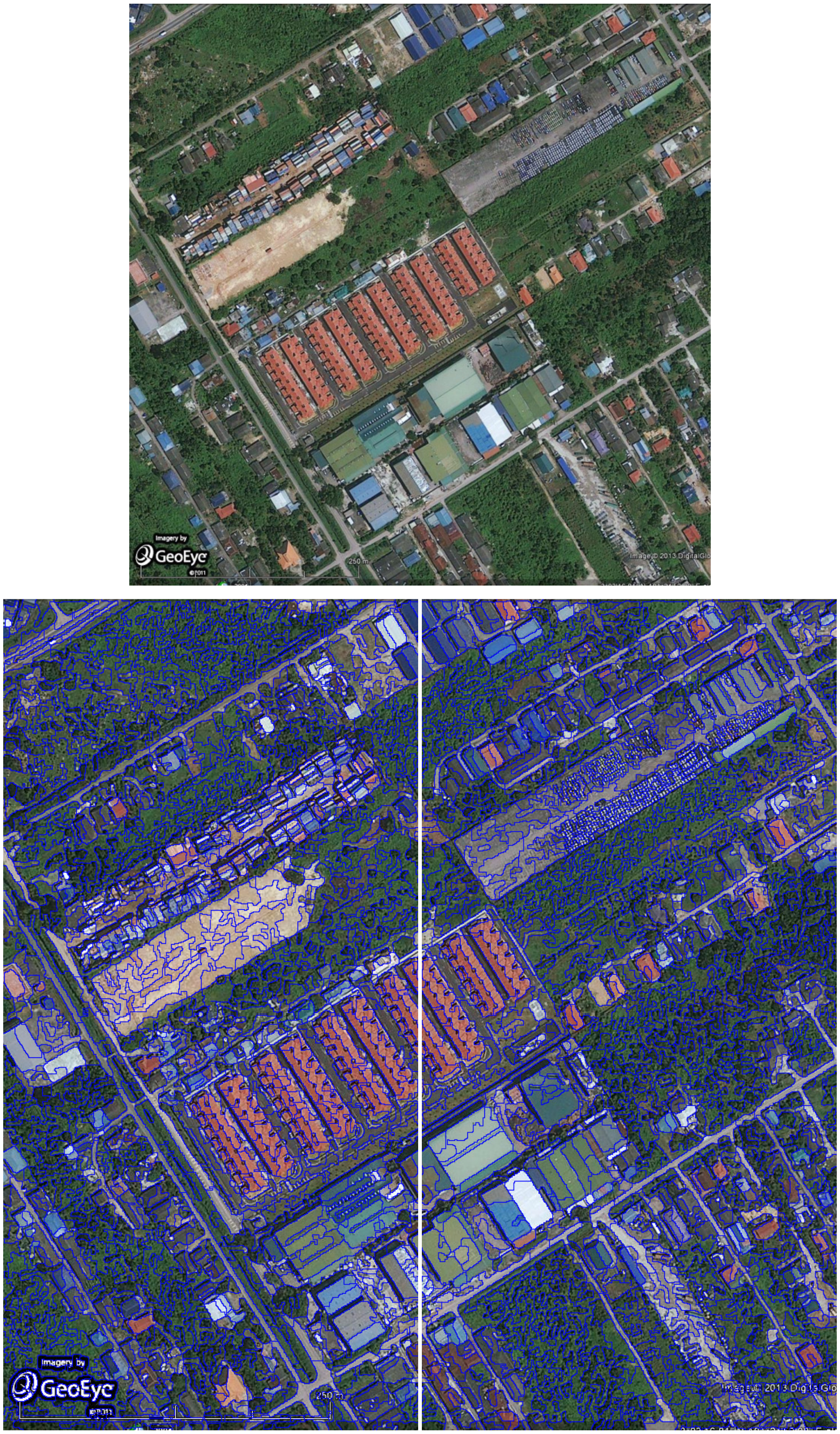


Figure 5-47 Test image 20 and its segmentation result

CHAPTER 6

CONCLUSION

6.1 Summary and Conclusions

In this thesis work, a fully automatic unsupervised segmentation method is proposed for high spatial resolution satellite images.

In the literature, the classification approach to any image classification problem can be divided into two groups as pixel-based and object-based. After observing the deficiency of pixel-based approaches in high spatial resolution satellite images, researchers tend to apply object-based approaches for classification. Object-based approaches contain two sequential steps which are segmentation and classification. The segmentation method which is based on watersheds and multi-scale segmentation in this thesis is a very good candidate to use in an object-based classification system.

We evaluate two different watershed algorithms which are immersion simulation type and rainfalling simulation type. Although, rainfalling type watershed algorithm are more successful with its drowning threshold parameter, it damages to segmentation accuracy by not successfully preserving the segment boundaries. Therefore, Vincent-Soille's watershed segmentation algorithm which is an immersion type watershed algorithm is incorporated in this thesis. Also, to decrease the over-segmentation resulted from watershed segmentation, a pre-processing smoothing filter and an alternative approach to gradient magnitude image which is homogeneity gradient image (H-Image) is used.

We classify multi-scale segmentation algorithms in the literature as wavelet based and region merging based. We evaluate both types and find region merging based approaches more useful for our problem. The possible reason for this is the corruption of the boundaries of segments in the input image that is exposed to a strong smoothing inherent in wavelet decomposition. We evaluate two different algorithms and propose another one that is the hybrid form of the other two algorithms for region merging based multi-scale segmentation algorithms category. After qualitative and quantitative comparisons, we find that this new one overwhelms others.

We realize that the effect of a parameter change of an algorithm on the segmentation results is very hard to understand by examining visual results only because of the excessive over-segmentation in the results. Therefore, when we compare alternative approaches throughout all the experiments, for finding the optimum parameters of an algorithm for a specific segmentation, we benefit from two supervised segmentation accuracy evaluation metrics and some auxiliary performance metrics. While one of the supervised segmentation accuracy metrics measures the global segmentation accuracy, the other one measures the segment based accuracy. These metrics compare the segmentation result with a manually segmented ground truth to measure the segmentation accuracy.

Because of the need for a ground truth of supervised accuracy evaluation metrics, the ground truths of the images 1-8 in the database, which is composed of 20 images, are extracted. The database images 1-8 are smaller in size and less complex than the remaining images. Therefore, ground truth extraction is more reliable in these images. These ground truths are used for performance evaluation and parameter selection of alternative approaches throughout the thesis, and also used for comparing the proposed approach with the two similar approaches in the previous literature.

However, finding a ground truth image may not always be possible for an image segmentation possible. Even if it is possible, the correctness of ground truth may not be certain, especially for images that have complex contents as in the database images 9-20. Because of these reasons, we evaluate two unsupervised segmentation accuracy evaluation metrics for selection of optimum parameters of the algorithms.

Experiments that are done on the database images 1-8 show that second metric produces closer results to manually found results than first metric. Even, by incorporating this metric to our proposed segmentation algorithm, our algorithm gains fully automatic property.

Comparisons with similar approaches in the previous literature and the segmentation results on complex images show the success of proposed automatic method of this thesis.

6.2 Future Work

One possible future work of the proposed approach could be to use the segmentation results for classification purposes. By this way, instead of labeling many pixels, labeling a small amount of segments could increase the speed and reliability of a classification.

Also, region merging algorithms, including the proposed one, are found to be computationally too complex during the experiments. Computational complexity of the region merging step should be improved for real time uses of the proposed approach.

Lastly, automatic extension of the proposed approach needs to scan all the values in a pre-defined interval of values for an optimum parameter selection. Depending on the input image set characteristics, the adjustment of the intervals to reduce the possible number of values to scan may provide excessive speed-ups. However, this process would need an expert and hence it is supervised. The techniques to automate this process could be searched for the future.

REFERENCES

[1] R. Gamanya, P.D. Maeyer and M.D. Dapper, “An Automated Satellite Image Classification Desing Using Object-Oriented Segmentation Algorithms: A Move towards Standardization”, Expert Systems with Applications, Volume 32, Issue 1, Pages 616-624, 2007.

[2] Z.P. Sun, W.M. Shen, B. Wei, X.M. Liu, W. Su, C. Zhang and J.Y. Yang, “Object-Oriented Land Cover Classification Using HJ-1 Remote Sensing Imagery”, Science China Earth Sciences, Volume 53, Issue 1, Pages 34-44, December 2010.

[3] U.C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder and M. Heynen, “Multi-resolution, Object-oriented Fuzzy Analysis of Remote Sensing Data for GIS-ready Information”, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 58, Issues 3-4, Pages 239-258, January 2004.

[4] A.P. Carleer, O. Debeir and E. Wolff, “Assessment of Very High Spatial Resolution Satellite Image Segmentations”, Photogrammetric Engineering and Remote Sensing, Volume 71, Issue 11, Pages 1285-1294, November 2005.

[5] D. Lu and Q. Weng, “A Survey of Image Classification Methods and Techniques for Improving Classification Performance”, International Journal of Remote Sensing, Volume 28, Issue 5, Pages 823-870, March 2007.

[6] V. Dey, Y. Zhang and M.Zhong, “A Review on Image Segmentation Techniques with Remote Sensing Perspective”, ISPRS TC VII Symposium, Volume 38, Pages 31-42, 2010

[7] J. Schiewe, “Segmentation of High-Resolution Remotely Sensed Data – Concepts, Applications and Problems”, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume 34, Part 4, Pages 358-363, 2002

[8] T. Blaschke, “Object based image analysis for remote sensing”, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 65, Issue 1, Pages 2-16, 2010

[9] K. Haris, S.N. Efstratiadis, N. Maglaveras and A.K. Katsaggelos, “Hybrid image segmentation using watersheds and fast region merging”, IEEE Transactions on Image Processing, Volume 7, Issue 12, Pages 1684-1699, December 1998

[10] X. Gigandet, “Satellite Image Segmentation and Classification”, EPFL Diploma Project, Fall 2004

[11] J.B. Kim and H.J. Kim, “Multiresolution-based watersheds for efficient image segmentation”, Pattern Recognition Letters, Volume 24, Pages 473-488, 2003

[12] Z. Chen, Z. Zhao, P. Gong and B. Zeng, “A new process for the segmentation of high resolution remote sensing imagery”, International Journal of Remote Sensing, Volume 27, Issue 22, Pages 4991-5001, 2006

[13] C.R. Jung, “Combining wavelets and watersheds for robust multiscale image segmentation”, Image and Vision Computing, Volume 25, Issue 1, Pages 24-33, 2007

[14] S. Chen, J. Luo, Z. Shen, X. Hu and L. Gao, “Segmentation of Multi-spectral Satellite Images Based on Watershed Algorithm”, 2008 International Symposium on Knowledge Acquisition and Modeling, Pages 684-688, 2008

[15] Y. Zhang and T. Maxwell, "A Fuzzy Logic Approach to Supervised Segmentation for Object-Oriented Classification", ASPRS 2006 Annual Conference, 2006

[16] Q.X. Chen, J.C. Luo, C.H. Zhou and T. Pei, "A Hybrid Multi-Scale Segmentation Approach for Remotely Sensed Imagery", IEEE International Geoscience and Remote Sensing Symposium Proceedings, Volume 6, Pages 3416-3419, 2003

[17] M. Tsaneva, "Texture Features for Segmentation of Satellite Images", Bulgarian Academy of Sciences Cybernetics and Information Technologies, Volume 8, Issue 3, Pages 73-85, 2008

[18] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Kluwer Academic Publishers, Norwell, MA, 1981

[19] B.U. Shankar, "Novel Classification and Segmentation Techniques with Application to Remotely Sensed Images", Lecture Notes on Computer Science Series 4400 Transactions on Rough Sets VII, Pages 295-380, 2007

[20] P.M. Atkinson and A.R.L. Tarnall, "Neural Networks in Remote Sensing", International Journal of Remote Sensing, Volume 18, Issue 4, Pages 699-709, 1997

[21] G. Dong and M. Xie, "Color Clustering and Learning for Image Segmentation Based on Neural Networks", IEEE Transactions on Neural Networks, Volume 16, Issue 4, Pages 925-936, July 2005

[22] G. Kuntimad and H.S. Ranganath, "Perfect Image Segmentation Using Pulse Coupled Neural Network", IEEE Transactions on Neural Network, Volume 10, Issue 3, Pages 591-598, May 1999

[23] L. Li, J. Ma and Q. Wen, "Parallel Fine Spatial Resolution Satellite Sensor Image Segmentation Based on An Improved Pulse-Coupled Neural Network", International Journal of Remote Sensing, Volume 28, Issue 18, Pages 4191-4198, September 2007

[24] D. Patra and J. Mridula, “Feature Based Segmentation of Color Textured Images using GLCM and Markov Random Field Model”, World Academy of Science Engineering and Technology, Volume 77, Pages 108-113, 2011

[25] C. Zhong, Z. Zhongmin, Y.D. Mei and C. Renxi, "Multi-scale Segmentation of the High Resolution Remote Sensing Image", Proceedings of IEEE International Geoscience and Remote Sensing Symposium, Volume 5, Pages 3682-3684, 2005

[26] L. Vincent and P. Soille, “Watersheds in Digital Space: An Efficient Algorithm Based on Immersion Simulation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 13, Issue 6, Pages 583-598, 1991

[27] P. De Smet and Rui Luis V. P. M. Pires, “Implementation and analysis of an optimized rainfalling watershed algorithm”, Proceedings of SPIE, Volume 3974, Pages 759-766, 2000

[28] L.A. Ruiz, A. Fdez-Sarria and J.A. Recio, “Texture Feature Extraction for Classification of Remote Sensing Data Using Wavelet Decomposition: A Comparative Study”, IAPRS, Volume 35, 2002

[29] D. Potere, "Horizontal Positional Accuracy of Google Earth's High Resolution Imagery Archive", Sensors, Volume 8, Issue 12, Pages 7973-7981, December 2008

[30] Y. Deng, C. Kenney, M.S. Moore and B.S. Manjunath, "Peer Group Filtering and Perceptual Color Image Quantization", Proceedings of IEEE International Symposium on Circuits and Systems, Volume 4, Pages 21-24, July 1999

[31] M. Feng, L. Ze, Z. Wensheng and L. Qiang, "Extracting of Urban Features From High Resolution Remote Sensing Data Based on Multiscale Segmentation", Proceedings of Urban Remote Sensing Joint Event, Pages 1-6, May 2009

[32] N. Nikolaou and N. Papamarkos, "Color reduction for complex document images", International Journal of Imaging Systems and Technology, Volume 19, Issue 1, Pages 14-26, March 2009

[33] J. Liu and Y.H. Yang, "Multiresolution Color Image Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 16, Issue 7, Pages 689-700, July 1994

[34] P.A.P. Moran, “Notes on Continuous Stochastic Phenomena”, Biometrika, Volume 37, Issue 1, Pages 17-23, 1950

[35] J.D. Bock, P. De Smet and W. Philips, "A Fast Sequential Rainfalling Watershed Segmentation Algorithm", Advanced Concepts for Intelligent Vision Systems, Lecture Notes in Computer Science, Volume 3708, Pages 476-482, 2005

[36] C. Drewniok, "Multi-Spectral Edge Detection: Some Experiments on Data from Landsat-TM", International Journal of Remote Sensing, Volume 15, Issue 18, Pages 3743-3765, 1994

[37] F. Jing, M. Li, H.-J. Zhang and B. Zhang, "Unsupervised Image Segmentation Using Local Homogeneity Analysis", Proceedings of the 2003 International Symposium on Circuits and Systems, May 2003

[38] G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 11, Issue 7, Pages 674-693, 1989

[39] B. Johnson and Z. Xie, “Unsupervised image segmentation evaluation and refinement using a multi-scale approach”, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 66, Pages 473-483, 2011

[40] Y. Cai, X. Tong and R. Shu, “Multi-scale Segmentation of Remote Sensing Image Based on Watershed Transformation”, 2009 Joint Urban Remote Sensing Joint Event, Pages 1-6, 2009

[41] C. Drewniok, "Multi-Spectral Edge Detection: Some Experiments on Data from Landsat-TM", International Journal of Remote Sensing, Volume 15, Issue 18, Pages 3743-3765, 1994

[42] S. Chabrier, B. Emile, H. Laurent, C. Rosenberger and P. Marche, "Unsupervised Evaluation of Image Segmentation Application to Multi-spectral Images", Proceedings of the 17th International Conference on Pattern Recognition, 2004

[43] H. Zhang, J.E. Fritts and S.A. Goldman, "An Entropy-based Objective Evaluation Method for Image Segmentation", Proceedings of SPIE Storage and Retrieval Methods and Applications for Multimedia, 2004