DEVELOPING AN INTEGRATED SYSTEM FOR SEMI-AUTOMATED SEGMENTATION OF REMOTELY SENSED IMAGERY

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE HAMİT KÖK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

MAY 2005

Approval of the Graduate School of Natural and Applied Sciences

____

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Asst. Prof. Dr. Zuhal Akyürek
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Assoc. Prof. Dr. Mustafa TÜRKER

Supervisor

Examining Committee Members

Prof. Dr. Vedat TOPRAK (Chairperson) (METU, GGIT) ___________________
Assoc. Prof. Dr. Mustafa TÜRKER (METU, GGIT) ___________________
Asst. Prof. Dr. M. Onur KARSLIOĞLU (METU, GGIT) ___________________
Dr. Uğur Murat LELOĞLU (TÜBİTAK) ___________________
Dr. Erhan GÖKÇAY (HAVELSAN A.S.) ___________________

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: Emre Hamit KÖK

Signature:

ABSTRACT

DEVELOPING AN INTEGRATED SYSTEM FOR SEMI-AUTOMATED SEGMENTATION OF REMOTELY SENSED IMAGERY

KÖK, Emre Hamit
MSc., Geodetic and Geographic Information Technologies
Supervisor: Assoc. Prof. Dr. Mustafa TÜRKER

May 2005, 146 pages

Classification of the agricultural fields using remote sensing images is one of the most popular methods used for crop mapping. Most recent classification techniques are based on per-field approach that works as assigning a crop label for each field. Commonly, the spatial vector data is used for the boundaries of the fields for applying the classification within them. However, crop variation within the fields is a very common problem. In this case, the existing field boundaries may be insufficient for performing the field-based classification and therefore, image segmentation is needed to be employed to detect these homogeneous segments within the fields.

This study proposed a field-based approach to segment the crop fields in an image within the integrated environment of Geographic Information System

(GIS) and Remote Sensing. In this method, each field is processed separately and the segments within each field are detected. First, an edge detection is applied to the images, and the detected edges are vectorized to generate the straight line segments. Next, these line segments are correlated with the existing field boundaries using the perceptual grouping techniques to form the closed regions in the image. The closed regions represent the segments each of which contain a distinct crop type. To implement the proposed methodology, a software was developed. The implementation was carried out using the 10 meter spatial resolution SPOT 5 and the 20 meter spatial resolution SPOT 4 satellite images covering a part of Karacabey Plain, Turkey. The evaluations of the obtained results are presented using different band combinations of the images.

Keywords: Field Based, Perceptual Grouping, Image Segmentation, Image Classification

ÖZ

UZAKTAN ALGILANMIŞ GÖRÜNTÜLERİN KESİMLENMESİ İÇİN YARI OTOMATİK BİR SİSTEM GELİŞTİRİMİ

KÖK, Emre Hamit
Yüksek Lisans, Jeodezi ve Coğrafi Bilgi Teknolojileri Ana Bilim Dalı
Tez Yöneticisi: Doç. Dr. Mustafa TÜRKER

Mayıs 2005, 146 sayfa

Uydu görüntüleri kullanılarak tarım alanlarının sınıflandırılması, ekili ürün haritalarını çıkartılmasında kullanılan en popüler metodlardan biridir. En son sınıflandırma teknikleri ekili ürünün alan bazında belirlenmesi yaklaşımına dayanmaktadır. Sınıflandırılma işlemine tabi tutulacak bu alanlar, genellikle sınır bilgilerini içeren vektörel verilerle tarif edilirler. Ancak bir alan içinde ekili farklı ürün bölgelerinin bulunması sıkça karşılaşılan bir durumdur. Bu durumda, sınıflandırma uygulamak için seçilen alanların belirlenmesinde mevcut sınır bilgilerini içeren vektör veriler yetersiz kalmaktadır. Bu probleme çözüm olarak, alan içindeki farklı ürün bölgelerinin saptanmasında görüntü kesimleme yöntemi tercih edilebilir.

Bu çalışma, alan içindeki alt ürün bölgelerinin belirlenmesi için Coğrafi Bilgi Sistemleri ve Uzaktan Algılama bilimlerinin entegre bir şekilde kullanılmasını öngören bir görüntü kesimleme tekniği önermektedir. Önerilen yöntemde, sınır bilgileri tanımlanmış alanlar ayrı ayrı işlenerek, her bir alan içinde alt kesimlerin bulunması yaklaşımı ele alınmaktadır. Bu yaklaşımda öncelikle, bir kenar belirleme yöntemi ile bulunan kontur bilgileri, vektörize edilerek düzgün doğru parçaları haline getirilir. Bulunan bu dogru parçaları mevcut sınır verileri ile algısal gruplama yöntemi aracılığıyla ilişkilendirilerek alan içinde kapalı kesimler oluşturulur. Bu kesimlerin her biri alan içinde farklı bir ekili ürün bölgesini tanımlar. Önerilen metodolojiyi uygulamaya koyan bir yazılım geliştirilmiştir. Karacabey Ovası'na ait 10 metre çözünürlükteki SPOT 5 ve 20 metre çözünürlükteki SPOT 4 uydu görüntüleri üzerinde uygulamalar yapılmış ve farklı bant kombinasyonları kullanılarak elde edilen sonuçlar üzerinde değerlendirmeler ve analizler sunulmuştur.

Anahtar Kelimeler: Alan Bazlı, Algısal Gruplama, Görüntü Kesimleme, Görüntü Sınıflama

To My Parents

ACKNOWLEDGEMENTS

I would like to thank Assoc. Prof. Dr. Mustafa Türker for his valuable supervision and support throughout the development and improvement of this thesis. This thesis would not have been completed without his guidance. Special thanks go to my father Prof. Dr. Recep Kök, who has always been a great motivation and a guide for me to understand the philosophy and the meaning of the academic studies. And I would like to express my gratitude to Ali Özgün Ok and Aslı Özdarıcı for supplying the data to be used for this study. I also would like to thank to my friends for their help, patience and kindness to me, and especially to my dear fiancée Gaye Öncül for always being next and supportive to me during the whole process of this thesis.

TABLE OF CONTENTS

PLAGIARISM....................iii
ABSTRACT....................iv
ÖZ....................vi
ACKNOWLEDGEMENTS....................ix
TABLE OF CONTENTS....................x
LIST OF TABLES....................xii
LIST OF FIGURES....................xiii
CHAPTERS
1. INTRODUCTION....................1
1.1 Objectives and Motivation....................5
1.2 Organization of the Thesis....................5
2. FIELD-BASED IMAGE SEGMENTATION....................7
2.1 Related Work....................7
2.1.1 Review on Image Segmentation....................7
2.1.1.1 Edge Based Segmentation Methods....................8
2.1.1.2 Region Based Segmentation Methods....................10
2.1.1.3 The Combined Methods....................12
2.1.2 Image Segmentation for Crop Mapping....................14
2.1.3 The Use of Spatial Data in Image Segmentation....................16
2.2 The Methodology....................19
2.2.1 Detection of the Edges....................24
2.2.2 Boundary Masking....................26
2.2.3 Contour Detection....................27
2.2.4 Line Simplification....................30
2.2.5 Perceptual Grouping....................34

2.2.5.1 Input Structure ... 35
2.2.5.2 Analysis Structure ... 37
2.2.5.3 Perceptual Grouping Rules ... 42
2.2.6 Forming the Polygons within the Fields ... 56
2.3 Software ... 64
2.3.1 The Capabilities of the Software ... 64
2.3.2 Software Design ... 69
3. THE IMPLEMENTATION ... 71
3.1 Input Data ... 71
3.1.1 Raster Data ... 73
3.1.1.1 The SPOT 5 Image ... 75
3.1.1.2 The SPOT 4 Images ... 80
3.1.2 Vector Data ... 84
3.2 The Methodology for Evaluation of the Results ... 87
3.2.1 Assignment Problem ... 99
3.3 Results and Analysis ... 102
3.3.1 The SPOT_5_PCA Image ... 104
3.3.2 The SPOT_5_Intensity Image ... 113
3.3.3 The SPOT_4_PCA Image ... 116
3.3.4 The SPOT_4_Intensity Image ... 119
3.3.5 Discussion of the Results ... 122
4. CONCLUSION ... 125
REFERENCES ... 130
APPENDIX ... 136
A. Sample Results ... 136

LIST OF TABLES

Table 2.1 – Line To Line Analysis Parameters 37
Table 3.1 – The Spatial and Spectral Properties of the SPOT Satellites............. 74
Table 3.2 – The PCA Output for the SPOT 5 Data 76
Table 3.3 – The PCA Output for the SPOT 4 Data 81
Table 3-4 – The Statistical Information for the Fields in Vector Data................ 86
Table 3.5 – General Information for the Parcels in the Vector Data................. 103
Table 3.6 – The Quantitative Results of the Segmentation for SPOT_5_PCA. 105
Table 3.7 – The Verification Parameters for SPOT_5_PCA 106
Table 3.8 – The Quantitative Results of the Segmentation for SPOT_5_Intensity 113
Table 3.9 – The Verification Parameters for SPOT_5_Intensity 114
Table 3.10 – The Quantitative Results of the Segmentation for SPOT_4_PCA116
Table 3.11 – The Verification Parameters for SPOT_4_PCA 117
Table 3.12– The Quantitative Results of the Segmentation for SPOT_4_Intensity 120
Table 3.13 – The Verification Parameters for SPOT_4_Intensity 120
Table 3.14 – General Overview of the Assessed Accuracies for Each Input Image 123
Table A.1 – The Sample Outputs for the SPOT_5_PCA Image 137
Table A.2 – The Sample Outputs for the SPOT_5_Intensity Image................. 140
Table A.3 – The Sample Outputs for the SPOT_4_PCA Image 143
Table A.4 – The Sample Outputs for the SPOT_5_Intensity Image................. 145

LIST OF FIGURES

Figure 1.1 – The Field-based Segmentation Process Flow 4

Figure 2.1 – The Geo-referencing Process... 20

Figure 2.2 – Sub-Images Extracted Using the Existing Field Boundary Data.... 21

Figure 2.3 – The Representation of a Sample Image 23

Figure 2.4 – The Sample Outputs of the Edge Detection Operation.................. 26

Figure 2.5 – Sample Outputs of the Boundary Masking Operation.................... 27

Figure 2.6 – A Binary Image and Contours Chain Graph.................................. 29

Figure 2.7 – The Illustration of the Douglas-Peucker Algorithm 31

Figure 2.8 – The Illustration of the Contours after the Line Simplification Procedure.. 32

Figure 2.9 – The Sample Outputs of the Line Simplification Operation 33

Figure 2.10 – Sample Input Image for Perceptual Grouping 36

Figure 2.11 – Line To Line Distance Parameters... 39

Figure 2.12 – Line To Line Intersection Parameters.. 40

Figure 2.13 – Line To Line Intersection Parameters Continued 41

Figure 2.14 – The Status of the Contours after Applying Rule-1 of the Perceptual Grouping……………………………………………………………...44

Figure 2.15 – The Status of the Contours after Applying Rule-2 of the Perceptual Grouping……………………………………………………………...45

Figure 2.16 – The Status of the Contours after Applying Rule-3 of the Perceptual Grouping……………………………………………………………...47

Figure 2.17 – The Status of the Contours after Applying Rule-4 of the Perceptual Grouping……………………………………………………………...48

Figure 2.18 – The Status of the Contours after Applying Rule-5 of the Perceptual Grouping……………………………………………………………...49

Figure 2.19 – The Status of the Contours after Applying Rule-6 of the Perceptual Grouping……………………………………………………………...51
Figure 2.20 – The Status of the Contours after Applying Rule-7 of the Perceptual Grouping……………………………………………………………...52
Figure 2.21 – The Status of the Contours after Applying Rule-8 of the Perceptual Grouping……………………………………………………………...53
Figure 2.22 – The Sample Outputs of the Perceptual Grouping Operation 55
Figure 2.23 – The Hash Table for Node Connections....................................... 57
Figure 2.24 – All Cyclic Paths for a Node 58
Figure 2.25 – The Traverse of All Possible Polygons For a Node..................... 59
Figure 2.26 – The Polygons Determined For a Field 60
Figure 2.27 – The Sample Outputs of the Polygon Construction Operation 61
Figure 2.28 – The Sample Outputs after the Merging Operation....................... 63
Figure 2.29 – The Main Window of FBISS 65
Figure 2.30 – The Application Parameters Window of FBISS 66
Figure 2.31- The Image Metadata Window of FBISS 66
Figure 2.32- The Parcel List Window of FBISS 67
Figure 2.33- The Right Click Menu Snapshot of FBISS 68
Figure 2.34- The Analysis of the Results Window of FBISS 69
Figure 2.35- The Class Diagram for FBISS Design... 70
Figure 3.1 – The Location of Study Area in Turkey 72
Figure 3.2 – The First Principle Component of SPOT 5 Image......................... 77
Figure 3.3 – The Intensity Image of SPOT 5 79
Figure 3.4 – The SPOT_4_PCA Image and the Overlaid Vector Data.............. 82
Figure 3.5 – The SPOT_4_Intensity Image and the Overlaid Vector Data 83
Figure 3.6 – The Distribution of the Sizes of the Fields 86
Figure 3.7 – The Identification Convention 87
Figure 3.8 – The Manually Segmented Field Boundaries and the Overlaid Display on SPOT_4_PCA 89
Figure 3.9 – The Manually Segmented Fields with the Overlaid Sub-Boundaries 90

Figure 3.10 – The Calculation of the Matching Percentages for a Field............. 94
Figure 3.11 – The SPOT_5_PCA Image with the Segmentation Results Overlaid 108
Figure 3.12 – The Illustration of the Under-segmentation Errors (A1, A2) 111
Figure 3.13 – The Illustration of the Under-segmentation Errors (A2, A3) 112
Figure 3.14 – The Illustration of the Over-segmentation Errors (B1, B2)........ 112
Figure 3.15 – The SPOT_5_Intensity Image with the Segmentation Results Overlaid 115
Figure 3.16 – The Detected Field Boundaries and the Overlay on SPOT_4_PCA 118
Figure 3.17 –The Effect of the Field Size on the Results of the Segmentation 119
Figure 3.18 – The Detected Field Boundaries and the Overlay on SPOT_4_ Intensity 121

CHAPTER I

INTRODUCTION

Agriculture plays a dominant role in the economies of both developed and undeveloped countries. The production of food is important to everyone and producing food in a cost-effective manner is one of the key points for the politics of every country. Many analyses are being performed to monitor the agricultural status of the fields and to get the correct decisions for future both by the farmers individually and by the governmental foundations. One of the important agricultural analyses is the mapping and identification of the crop fields. This analysis serves the purpose of forecasting grain supplies (yield estimation), collecting crop production statistics, facilitating crop rotation records, mapping soil productivity, identification of factors influencing crop stress, assessment of crop damage due to storms and drought, and monitoring farming activity. Traditional methods of crop field mapping are census and ground surveying. However, these methods are inefficient for the time and cost aspects and may not contain standard measurements. With the availability of high resolution satellite images and the improvements in digital processing and analysis techniques, remote sensing can provide efficient and reliable solutions for the identification of the crop fields.

Image classification is one of the frequently used techniques in detecting the crop fields from remotely sensed data. Most current automatic classification techniques to obtain land cover maps from digital imagery operate on a per-pixel basis in isolation from other pertinent information. Pixel-based classification methods take a pixel as a main element and try to label each pixel individually. However, various errors are involved in the image classification that operates on per-pixel basis. The mixed pixels, simplicity of the basic assumptions in the classification algorithms, the sensor effects, the atmospheric effects and the radiometric overlap of land cover objects lead to wrong detection in the image analysis. To overcome these drawbacks, field-based classification methods are developed that take segments of pixels as the main element and try to label the fields individually. These methods provide significantly accurate results with respect to the pixel-based approaches [2][27][28][32]. The field-based classification approaches generally use the spatial vector data (e.g. cadastral maps) in order to access the boundary data of the fields on which the classification will be performed. Each field described by the spatial vector data is utilized during classification and a single label is assigned to the pixels contained within the field. However, multiple crops may be planted within the stable boundaries of agricultural fields and therefore, the fields may sometimes cover more than one crop types. And the existence of multiple crops within the fields creates an accuracy problem for the field-based approaches. To overcome this accuracy problem, the homogeneity of the fields to be processed by the classification can be satisfied using an image segmentation technique [2][21].

Image segmentation is a process of partitioning an image into some disjoint (non-overlapping) regions such that each region is homogeneous and no adjacent regions can be merged without the violation of the homogeneity [46]. There are several methods used in the segmentation of the remotely sensed images [19][31]. The three main approaches for image segmentation are; (i) edge-based methods, (ii) region-based methods and (iii) combined methods integrating edge and region data. Each approach has advantages and disadvantages, and none of them can guarantee a high accuracy for the homogeneity and the geometries of the segmented fields. For example, Amadasun, M., and King, R. A. demonstrated that the same image can be segmented into three or five types of regions using two different segmentation techniques [24]. The common problems of the general segmentation techniques can be explained briefly as follows [21]:

1) *the fact that that the ground resolution of RS data is not always being appropriate for the size (shape) of the terrain objects,*
2) *the absence of simple 1 : 1 or m: 1 relations between image segments and real word objects,*
3) *the lack of proper theoretical framework to decide when to use which technique.*

In addition, applying the segmentation on the whole image has a basic problem as including all the elements like roads, ditches, hedges and streams into process which should not belong to any crop field segments. These issues show that using a general image segmentation technique without including any priori knowledge about the image content cannot completely replace the use of vector field boundaries for the field-based classification. However, there are some segmentation techniques that use the available semantic knowledge to refine the segmentation and to obtain more accurate results [17]. The integration of Geographic Information Systems (GIS) and Remote Sensing and the use of vector field boundary data within the image segmentation process constitute a crucial point for detecting the boundaries of the homogenous crop fields.

This thesis proposes a field-based image segmentation technique to segment the crop fields in an agricultural area using the existing field boundary data. In this approach, the segmentation process is applied within the fields whose boundary data are available from the cadastral maps. An edge based methodology was chosen to find the missing boundaries of the different homogeneous regions in the fields. The missing field boundaries are extracted through the output of the edge detected image. The detected edges are vectorized and the constructed line segments are modified in order to form the closed regions through a rule based perceptual grouping procedure. Figure 1.1 describes the main steps for the field-based image segmentation process.

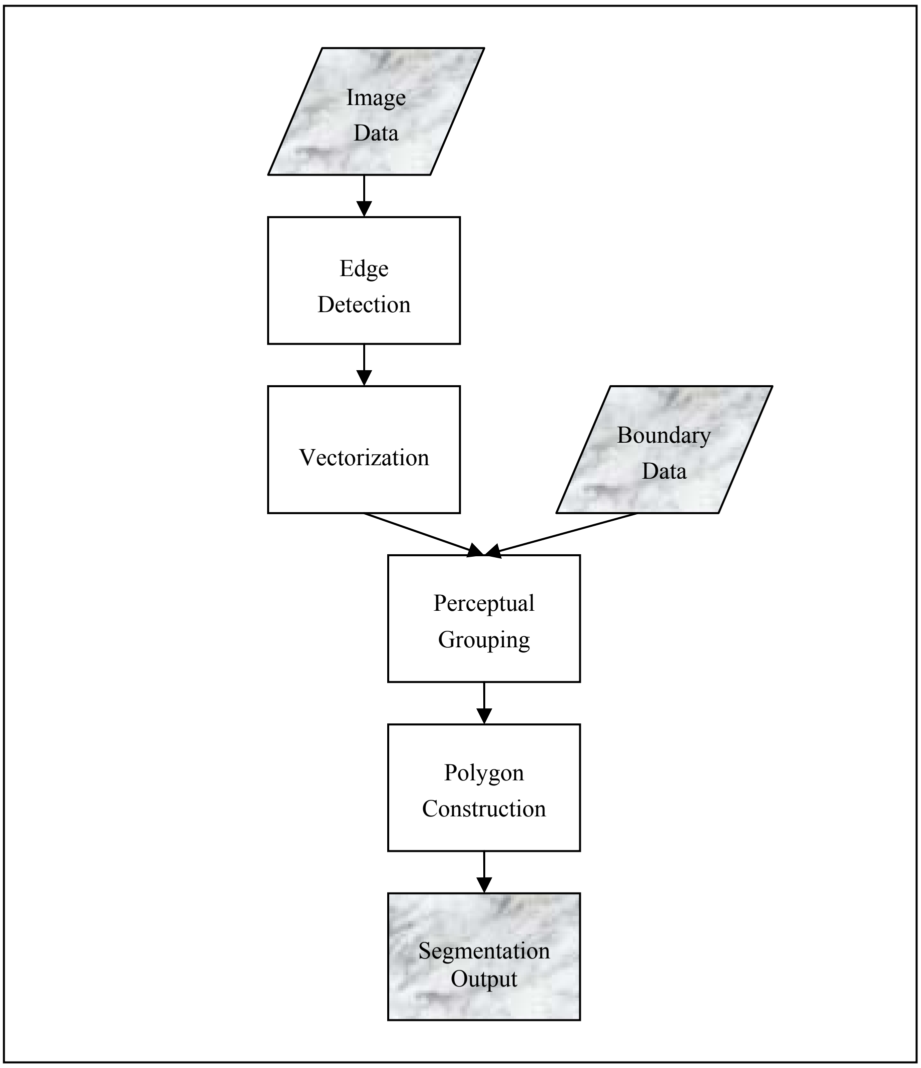


Figure 1.1 – The Field-based Segmentation Process Flow

To implement the concept, Field-based Image Segmentation Software (FBISS) was developed using Visual C++ 6.0 and Open Computer Vision (OpenCV, Version 4 Beta) library on Windows platform [48]. FBISS provides the full capability of using the raster and vector data together and performing the proposed segmentation procedure within the fields and the display of the input and the outputs. In addition, the analysis functionality for evaluating the results was developed in the software.

The proposed field-based image segmentation methodology was implemented using 10 meter spatial resolution SPOT 5 and 20 meter spatial resolution SPOT 4 images covering a part of the Karacabey Plain. The cadastral maps of Karacabey Plain were used as the existing vector boundary data of the fields. Different band combinations of the SPOT 5 and SPOT 4 images were used and the obtained results were evaluated through accuracy assessment.

1.1 Objectives and Motivation

The main objectives of this thesis are as follows;

- To develop a field-based image segmentation approach for detecting the homogenous sub-fields within existing agriculture fields.
- To implement the proposed approach by developing a software and applying it in a study area.
- To analyze the advantages and the drawbacks of the proposed methodology on the obtained results.
- To supply an error free input to the field-based classification approaches that have accuracy problems caused by the within field crop variations.

1.2 Organization of the Thesis

The organization of the thesis is as follows;

Chapter 1 presents an introduction for the problem and the methodology proposed in this thesis.

In Chapter 2, first, the related works on the image segmentation are discussed and then the field-based image segmentation methodology is explained in detail.

In Chapter 3, the implementation of the concept is presented, the description of the input data and the evaluation approach for the methodology are given, and the analysis results are discussed.

Finally, the conclusions and the future works are provided in Chapter 4.

CHAPTER II

FIELD-BASED IMAGE SEGMENTATION

2.1 Related Work

This section covers the descriptions of the related work reviewed in the literature concerning the subject studied in this thesis. First, a review is given for the basic image segmentation techniques in order to highlight the advantages and the limitations. Second several studies that have focused on field-based image analysis for crop mapping and the employment of the image segmentation for these analyses are discussed. Finally, several studies that have used a knowledge base in the segmentation of the images in order to improve the accuracy of crop mapping are explained.

2.1.1 Review on Image Segmentation

Image segmentation is one of the fundamental concepts in image processing. Therefore, over the years a number of different approaches have been developed.. Excellent reviews of this subject can be found in [19] and [31]. Although there are various classification schemas used for different segmentation techniques, the following categorization is the most common in the literature.

- Edge Based Methods
- Region Based Methods
- Combined Methods, Integration of Edge and Region Based techniques

2.1.1.1 Edge Based Segmentation Methods

Image segmentation techniques based on edge detection have long been in used since the early work of Roberts in 1965 [22]. Edge based methods basically rely on the pixel-wise difference to guide the process of segmentation. Thus, they try to locate points of abrupt changes in the grey tone images. Edge detectors are model-free local boundary finders where for any image the edge strength is calculated by using a local differential operator. There are a variety of edge operators such as Robert's, Sobel, Prewitt etc. [36]. One of the most influential methods uses the zero-crossings of the Laplacian of the Gaussian of the image. It produces continuous closed boundaries (being the zero-set of an implicit function), but since it is a second derivative operator, it is extremely noise sensitive. Also, it is difficult to distinguish between more and less probable edges and the only way of discarding false edges would be to do so interactively by a human operator.

The Canny edge detector seeks to optimize by making the best trade-of between detection and localization of the edges, the two main yet opposing goals of edge detection, in addition to producing one and only one response for a single edge point [14]. He showed that the ideal edge detector can be approximated by first taking the gradient of the image convolved with the Gaussian kernel and then choosing the maxima of the gradient image in the direction of the gradient However, except under ideal circumstances, it results in broken edges and thus identifying objects would require grouping those edges either interactively or by using some other algorithm [4].

Hough transform may be considered to be an alternative to boundary analysis. It is essentially a mapping from the image to a parameter space. It was originally used for parameterizable shapes like curves, but was extended later to include the general shapes [37]. Likely boundary points are identified first in the image, which are then mapped to the parameter space. Maxima in the Hough space correspond to possible shape instances. The advantage of this method is that it is relatively insensitive to gaps and noise. The drawback however is the computational requirement that goes up geometrically with an increase in the number of parameters. And for deformable boundaries, it becomes even more unsuitable.

A specific approach is presented by Hoffman and Boehner [44]. In this study, the proposed image segmentation strategy is based on the representativeness values of each pixel. First these values are calculated by a harmonic analysis of the values for each spectral channel. The minima in the matrix of representativeness – typically arranged in pixel-lineaments - represent spatial unsteadiness in the digital numbers. For image segmentation, the vectorized minima of the representativeness delimit areas consisting of pixels with similar spectral properties (spatial segments). A convergence index is combined with a single-flow algorithm for the vectorization of the representativeness minima. And a standardization is performed through the calculation of a convergence index for every pixel in a 3×3 window.

The common problem for all edge based methods is the lack of knowledge to determine a detected edge as a segment boundary or not. The secondary common problem is the need for the post processing of the detected edges. Since detected edges in an image cannot directly form the segments as closed curves, the detected edge features should be enhanced using the line smoothing algorithms, and perceptual grouping techniques such as defining the thresholds for linking the missing connections, and for removing the wrongly detected lines or areas forming very small closed curves.

2.1.1.2 Region Based Segmentation Methods

The region based approaches try to isolate the areas of the images that are homogeneous according to a given set of characteristics. Most of the common studies on region based approaches can be grouped in two classical methods;
(i) region growing and (ii) split and merge.

Region growing is one of the most simple and popular region-based segmentation algorithms [35][41]. It starts by choosing a starting point or seed pixel. The most habitual way is to select these seeds by randomly choosing a set of pixels in the image or by following a priori set direction of the scan of the image. The region grows by successively adding neighboring pixels that are similar according to a certain homogeneity criterion, increasing step by step the size of the region. This criterion can be, for example, to require the variance of a feature inside the region does not exceed a threshold, or that the difference between the pixel and the average of the region does not exceed a predefined value. The growing process is continued until a pixel not sufficiently similar to be aggregated is found, which means that there is not any neighboring pixel which is similar to the region and the growing of the region is ended. After that, another seed point is selected from the unassigned pixels and growing iterations are continuously applied until the boundaries of the regions correspond with the edges of the image and not any pixel is left included in one of the regions

One of the basic properties of the segmentation is the existence of a predicate which measures the homogeneity. If this predicate is not satisfied for a region, It means that, that region is inhomogeneous should be split into sub-regions. On the other hand if the predicate is satisfied for the union of two adjacent regions, then these regions are collectively homogeneous and should be merged. This method is known as the Split and Merge Algorithm. The algorithm works as firstly splitting the image until all the regions satisfy the homogeneity criterion, and secondly reassembling the adjacent regions which resultant combinations of the regions still verify the homogeneity criterion. The split-and-merge techniques, developed originally by Chen and Pavlidis, use a linked pyramid structure and statistical decision criteria to combine global and local region information [33]. These ideas were adopted and developed with variations in the further studies [34][42][43]. But like their predecessors, they required prior knowledge of the size and number of the regions present. The quadtree based method of Spann and Wilson [30] made use of the non-parametric classification [38] at a low spatial resolution, followed by downward boundary estimation, to regain the spatial resolution. It was notable because of its efficiency and that did not require prior knowledge of the image regions. This method was also shown to be readily extensible to the problem of texture segmentation. Some of the successful ideas from the quadtree method have been used by Spann and Horn [29], to overcome the problems in the linked pyramid based split-and-merge method.

The following issues can be listed for the common problems of the region based methods:

- The Results are sensitive to the merging order - even if a merge results in a homogeneous region, it may not be the optimal.
- There is no mechanism for seeking the optimal merges. The semantic region growing approach to segmentation and interpretation starts with an oversegmented image in which some merges were not best possible.
- The semantic process is then trying to locate the maximum of some objective function by grouping regions which may already be incorrect and is therefore trying to obtain an optimal image interpretation from partially processed data where some significant information has already been lost.

- Conventional semantic region growing merges regions in an interpretation level only and does not evaluate properties of newly merged regions. It also very often ends in a local optimum of region labeling; the global optimum is not found because of the character of the optimization.
- Unreliability of image segmentation and interpretation of complex images results.

2.1.1.3 The Combined Methods

The combination of the edge based and the region based methods yields far better results than those of a single method and there is a certain tendency on the recent works seeking to integrate either different features or different modules to improve the segmentation process. Simultaneous consideration of the smoothness and the contrast was first used in which applied dynamic programming for minimizing the weighted sum of a contrast measure and smoothness measure [45].

The recent studies including the integration of the edge based and region based segmentation techniques have the signature of Cufi, Munoz, Freixent, and Marti [46][47]. In these studies, they proposed a method to segment the natural images based on the obtention of a set of frontiers of the perceptively most significant regions, named circumscribed contours of the image. This method is based on the two most relevant properties presented by the most significant frontiers of a determined scene:

- The edges must have an important length with the global frame of the image
- These edges separate sufficiently differentiated regions from the scene, considering basically chromatic, intensity, and textural characteristics.

The extraction of the fragments which have an important length from the information of the local edges obtained in the hue, intensity and saturation components, and a posterior phase of restoration which concatenates the contours.

Additionally several different approaches integrating region and based techniques are examined in the Ph.D thesis of A. Bhalerao in 1991 [1]. This thesis proposed several region growing methods using the edge operators to prevent the merging across the boundaries. The results of a split-and-merge using a linked pyramid approach is additionally processed with a boundary relaxation step to refine and improve the region errors and enhance the resultant contours. By incorporating the global shape and the shadow information they have also shown good results in the restricted applications, such as extracting the boundaries of the buildings from aerial images.

A very recent study was conducted by Tilton who proposed a new approach "hierarchical image segmentation" [15]. It is a hybrid method of region growing and spectral clustering that produces a hierarchical set of image segmentations based on the detected natural convergence points. A hierarchical set of image segmentations is a set of several image segmentations at different levels of the segmentation detail in which the segmentations at coarser levels of detail can be produced from simple merges of regions from the segmentations at finer levels of detail.

Although the use of integrated methods have improved the accuracy of the segmentation results, these techniques still cannot satisfy generic solutions for all the cases due to the heterogeneity of the input domain.

2.1.2 Image Segmentation for Crop Mapping

Image segmentation is basically the process of determining the homogeneous regions in the image and is highly used for extracting the field boundaries of the agricultural fields in the remotely sensed imagery. Segmenting the image by defining the fields having distinct crop types is commonly known as crop mapping. There are a number of studies and publications carried out concerning the image segmentation for agricultural areas. However, due to its concept of labeling the content of the fields, crop mapping is studied as an image classification problem, and image segmentation can be used as a pre-process to supply these homogenous fields to be labeled by the classification process. Especially field-based (object-based) classification techniques, which are concerned with the classification of the fields, generally need either a segmentation process to obtain the homogenous field boundaries, or a spatial reference data such as topographic maps.

There are several studies covering the field-based classification techniques and comparative analyses between these methods and pixel-based methods. G. Smith and Fuller created a land cover map using a field-based approach in a study area [12]. They used a digital cartography for the source of land parcel boundary information and applied a two stage process. In the first step a field-based classification is applied using a maximum likelihood algorithm and in the second step a knowledge-based correction is performed using the contextual information to refine the results. Smith has performed a study for comparing the field-based classification and pixel-based classification techniques with A. M. Dean [3]. They used a maximum likelihood approach for the classification and concluded that field-based classification was shown to be most appropriate for mapping agricultural land cover in comparison to semi-natural areas, However, they stated that, when the land cover is heterogeneous a pixel-based classification would appear to be more appropriate due to the crop variation within the fields and suggested a hybrid classifier which can switch between the field-based and pixel-based classification as a powerful land cover mapping tool. Another comparative study has been performed by Tso and Mather between pixel -based classification and field-based classification approaches [7]. They used maximum likelihood and Kohonen's self organized map (SOM) algorithms to apply classification on SAR imagery. They showed that field-based classification has a greater accuracy (75%) with respect to pixel-based classification (60%). K. Johnsson applied the field-based classification approach for the built-up lands using multispectral SPOT data and showed that this approach can be used also in land-use areas [18]. The classification procedure involved spectral classification, determination of size and neighbor relations for the segments in the spectrally classified image, and rule-based classification of the image segments into land-use categories. A digital land-use map is used for the boundaries of the segments.

Field-based classification techniques generally uses spatial vector data for the existing field boundaries as stated in the above examples. However, there are some studies which employed an image segmentation technique to extract the fields in the image without using any vector data. An automated segmentation procedure for Landsat TM images of farmland into field units was presented by Evans, Jones, Svalbe and Berman [8]. The segmentation procedure, named the canonically-guided region growing (CGRG) procedure assumed that each field contains only one ground cover type and that the width of the minimum field of interest is known. This method segments the images using a seeded region growing algorithm, by generating the internal field markers used as "seeds". These internal field markers are obtained from a multi-band local canonical eigen value image. This study can be summarized as an improvement of a region growing based segmentation strategy, which automates the selection of the seed pixels, and growing criteria using the obtained partial field boundaries with the estimation of the within-field and between-field variation parameters.

Rydberg, A. and Gunilla B., proposes a multispectral segmentation method for the automated delineation of agriculture field boundaries in remotely sensed imagery [5]. In this study, edge information detected by a gradient operator is integrated with a segmentation algorithm based on an ISODATA algorithm. A multispectral edge detector is used which includes all the available multispectral information by adding the magnitudes and the directions derived from the edge detection on single bands. In ISODATA algorithm, the initial centroids are decided by using the distances of them to the detected edges. Using this procedure the number of regions mostly exceeds the actual number of the fields in the image, therefore a further merging operation is applied to overcome the over-segmentation problem. Merging decisions are based on a likelihood-ratio test by calculating the mean and the covariance matrix for neighboring regions.

The study of Boundary-Constrained Agglomerative Segmentation proposed a segmentation technique for the remotely sensed imagery by formulating the segmentation as the source channel coding with the side information [20]. A cost function is developed that approximates the expected code length for a hypothetical two-part coding scheme. The cost function combines region and edge based considerations and supports the utilization of reference data to enhance the segmentation results. The reference data contains a priori information which indicates the potential boundary locations. The technique they propose does not require that these potential locations have to be correct, therefore an edge detector may be employed to create this reference data or GIS data may be used. The approach is based on the hypothetical image coding scheme that composes a given image into homogeneous segments and encode these segments independently of each other, where the encoder and the decoder share some reference information of the image.

2.1.3 The Use of Spatial Data in Image Segmentation

Although per-field classification techniques provide better results for crop mapping, they have a common accuracy problem caused by the within field variations. Commonly field-based (object-based) classification methods use the GIS databases for the boundaries of the fields to be processed by the classification. These databases generally consist of the cadastral maps, in which the boundaries of the agricultural fields owned by the farmers are defined. And the crop variation in a field is a defect for the accuracy of the field-based classification, which will label the field object only with one crop type. The recent studies employs the image segmentation cooperated with the GIS data such as the cadastral maps in order to improve the accuracy of the crop mapping.

Atkinson and Aplin used a new technique for predicting the missing field boundaries which is based on a comparison of the within-field modal land cover proportion and the local variance [32]. Such a comparison provided a simple characterization of the variation in the land cover within each field and therefore, the indication of the likelihood of the missing boundaries. In that study, both the pixel-based and the field-based classification techniques were implemented and mainly per-field classification outputs were revised by calculating a local variance and the modal land cover proportion in the field by using the per-pixel classification outputs. When in a field the proportion of the modal land cover was small, and local variance was large, the field is likely to be mixed. If the proportion was high and the local variance was small, then the field was likely to be homogeneous. On the other hand if both the local variance and proportion were small, then the field is likely to comprise the homogeneous patches of different land cover classes, indicating that the field boundaries may be missing. The technique examined by this study, does not identify the missing field boundaries. It only identifies the fields which may have missing boundaries. And the fields which are detected as candidate fields having missing boundaries are manually checked and divided into sub-fields if necessary. The manual boundary extraction is the drawback of this technique.

A similar study was conducted by De Wit, A. J. W., and Clevers, J.G.P.W. which integrates multi-temporal and multi-sensor satellite imagery, statistical data on crop area and parcel boundaries from a topographical map [2]. First, they created a crop field database by extracting the field boundaries from the digital topographical map. Then they used a manual on-screen digitizing method to add dynamic crop boundaries within fields. They discussed the use of image segmentation instead of using the topographical maps and they concluded that using image segmentation only cannot replace the use of the digitized vector maps due to the unpredictable results that may occur in the image segmentation without using any contextual knowledge for the fields. But the use of image segmentation integrated with the topographic maps to detect the dynamic field boundaries within the fields is suggested in this study.

Janssen and Molenaar discussed the terrain object hierarchies and the need for segmentation in the fields whose boundaries were defined as cadastral maps [21]. The field, parcel, and farm district hierarchies were described with the demonstrations of crop variations within the fields. The discrepancies and disadvantages of the standard image segmentation techniques were described and the need for a knowledge base for the segmentation was clarified. A three phased methodology was developed to keep GIS database up-to-date including the crop field boundaries and the contents. In the first phase, image segmentation was employed for detecting the crop boundaries within the fields. Then an object-based classification method was applied to detect the crop types, and a conditional merging is performed to solve the over-segmentation problems. For the image segmentation process, an edge detection was applied and a rule-based grouping algorithm was described to correlate the lines detected as edges to form the closed polygons using the existing boundary data. In the described rules, the lines were extended towards the boundaries using the predefined distance and angle thresholds, and the smaller edges were cleaned. In addition, an evaluation methodology was presented that is based on the matching percentage between manually segmented geometries and their resultant geometries. 87% accuracy was achieved with the result geometry constructed by the additional boundaries using the proposed segmentation technique. The successful results achieved are mainly due to the large and rectangular shaped fields that have sub-fields with simple geometry, which is interpreted as a deficiency of this technique in several studies. [2]

There are several studies using additional data in order to improve the accuracy of the segmentation. A knowledge-based segmentation approach of Landsat TM images was presented by Ton, J., Sticklen, J., and Jain A. K. who applied different type of knowledge [17]. Another study used boundaries contained in a GIS in an updated strategy that departs form a pixel based classification [26]. The proposed method works under the assumption that only further fragmentation of objects takes place. Another study presented a method which departs from topographical map and in which both fragmentation and merging of terrain objects are possible [9].

2.2 The Methodology

Field-based image segmentation is the process of performing the segmentation within each field separately to determine the homogenous regions by means of detecting the sub-field boundaries in the field.

As a first step of the procedure, the agricultural field to be analyzed is selected from the input image using the existing boundary information. The geographic locations of the vertices belonging to the field, which describes a distinct field, are available as a formatted text file to be used in the processing. The input raster image is geographically referenced data. Therefore, the geographic locations of the upper-left and lower-right corners of the image and the spatial resolution are known as a priory knowledge for the algorithm. The ellipsoidal geographic coordinates of the corners of the image are read and any geographic location falling within the boundary vertex database can be registered to a pixel location in the image. The spatial resolution of the image defines the area value that can be represented by one pixel. Figure 2.1 illustrates the geo-referencing process of the raster image with the existing field boundary vector data.

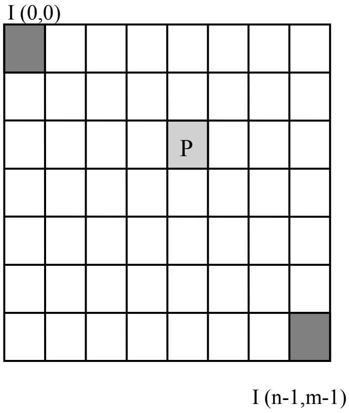


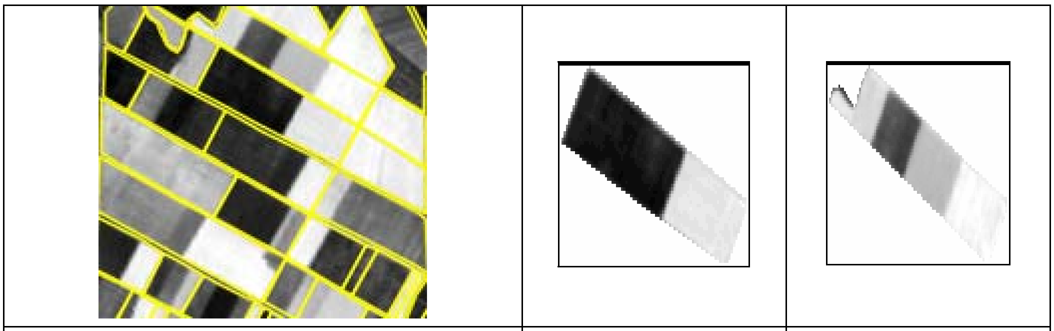
Image I has the size of width=n height=m
The geographic locations for pixels
I(0,0) and **I(n-1, m-1)** are given as;
Easting$_{\mathbf{I(0,0)}}$ = ULEasting (Upper Left Easting)
Northing$_{\mathbf{I(0,0)}}$ = ULNorthing (Upper Left Northing)
Easting$_{\mathbf{I(n-1,m-1)}}$ = LREasting (Lower Right Easting)
Easting$_{\mathbf{I(n-1,m-1)}}$ = LRNorthing (Lower Right Northing)
And resolution for a pixel in X, and Y axes are calculated as;
EastingResolution = (LREasting-ULEasting) / n
NorthingResolution = (ULNorthing-LRNorthing) / m
{Equation 2.1}

For a boundary vertex V, the geographic location (**Easting$_V$**,**Northing$_V$**)
is mapped to a pixel location **P=I(x,y)** using the following formula:
x = [Easting$_V$-Easting$_{\mathbf{I(0,0)}}$] / EastingResolution
y = [Northing$_{\mathbf{I(0,0)}}$-Northing$_V$] / NorthingResolution ***{Equation 2.2}***

Figure 2.1 – The Geo-referencing Process

After registering the raster image to the existing field boundaries, the fields that satisfy the predefined conditions are selected and processed to be segmented. The small and thin fields are excluded from the processing. The determination of the small and thin fields is carried out using the area and shape factor thresholds. Therefore, the shape factors and the areas of those fields falling below the predefined threshold values are not included in the segmentation process. The shape factor (Equation 2.3) is an area vs. perimeter proportion and gives an idea about the robustness of the polygon. Figure 2.2 demonstrates the process of extracting a field as a new sub-image from the input image.

$$SH = \frac{\sqrt{4\pi \times Area}}{Perimeter} \qquad \textit{Equation 2.3}$$



a) Part of the original image displaying the vector boundaries overlaid.

b) Field-1 Image

c) Field-2 Image

Figure 2.2 – Sub-Images Extracted Using the Existing Field Boundary Data

After capturing the image of a field, the segmentation is carried out using the following steps;

i. Edge detection (Canny),

ii. Boundary Masking,

iii. Contour Detection (Raster To Vector Conversion – Suzuki Algorithm),

iv. Line Simplification (Douglas-Peucker Algorithm),

v. Perceptual Grouping (Based On Rules), and

vi. Forming Polygons

The whole process can be summarized as the detection of the sub-field boundaries that form the homogeneous segments in a field and constructing the polygons within a field considered. First, an edge detection is applied to image in order to find those pixels that form a continuous line in the image. After a boundary masking operation, the contours of the edge image are detected through a raster to vector conversion algorithm. The contours consist of the line segments whose vertices can be represented by two coordinate pairs in vector domain. The detected contour set and the existing field boundary coordinates therefore become the input for the next processing stage. In turn, all the other operations are carried out in the vector domain. The remaining processings mainly consist of the association of the edges of these lines with the existing field boundaries to form the closed polygons within the field.

After the vectorization process, a line smoothing algorithm is applied to generate the straight lines and to eliminate the overlapping and the noisy line segments. Then, a rule based procedure is performed to the contours in order to perceptually group the line segments to form closed regions. The perceptual grouping process is basically removing of the noisy lines, and extending the remaining lines to each other to satisfy connections for the closed regions. The deviations are corrected, the dangling arcs are removed and a set of the connected lines are constructed.

The above extracted set becomes the input for the polygon construction procedure. All the polygons that can be constructed using these connected line segments are detected. The detected distinct polygons are processed by a rule based algorithm that merges the small ones with its adjacent polygons.

The following illustration and the definitions are given in order to clarify the terminology used in the proposed field-based segmentation process. Figure 2.3 represents a field with the existing boundary information and the detected inner-lines that are the candidates for the missing boundaries of the sub-fields.

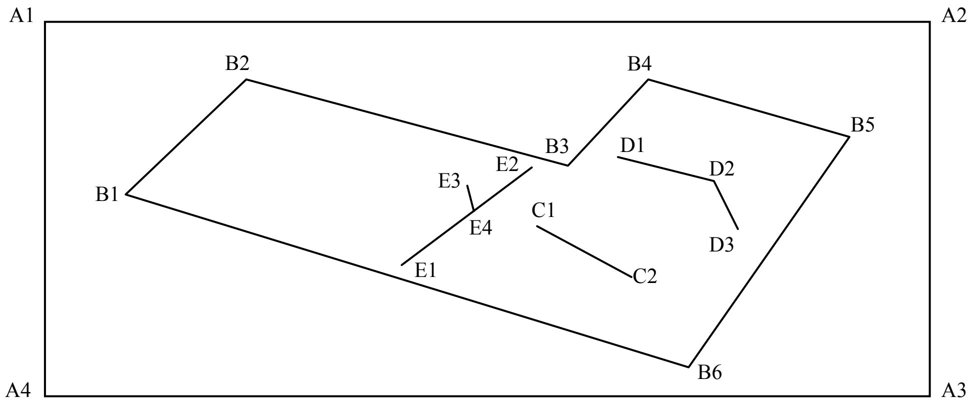


Figure 2.3 – The Representation of a Sample Image

A1, A2, A3 and A4 are the boundary coordinates of the minimal bounding rectangle in the raster domain. These boundary coordinates are used for the geo-referencing step of the segmentation process.

Existing Field Boundary: The known boundary coordinates of the existing field are B1, B2, B3, B4, B5 and B6.

Line Segment: A straight line which is defined by two coordinate pairs. For example, [C1-C2], [D1-D2], [D2-D3], [E1-E2], [E3-E4] are the distinct line segments.

Vertex: The start or end points of the line segments. For example, C1, C2, D1, D2, D3, E1, E2, E3, E4 are the vertices.

Contour: The set of connected line segments. The line segments having adjacency with each other are grouped and called contour. The followings are the contours of the field illustrated in Figure 2.3

Contour-1: ([C1-C2]),

Contour-2: ([D1-D2], [D2-D3])

Contour-3: ([E1-E2], [E3-E4])

The existing field boundaries can also be represented as a contour, such as;

Contour-4: ([B1-B2], [B2-B3] [B3-B4], [B4-B5] [B5-B6], [B6-B7], [B7-B1])

2.2.1 Detection of the Edges

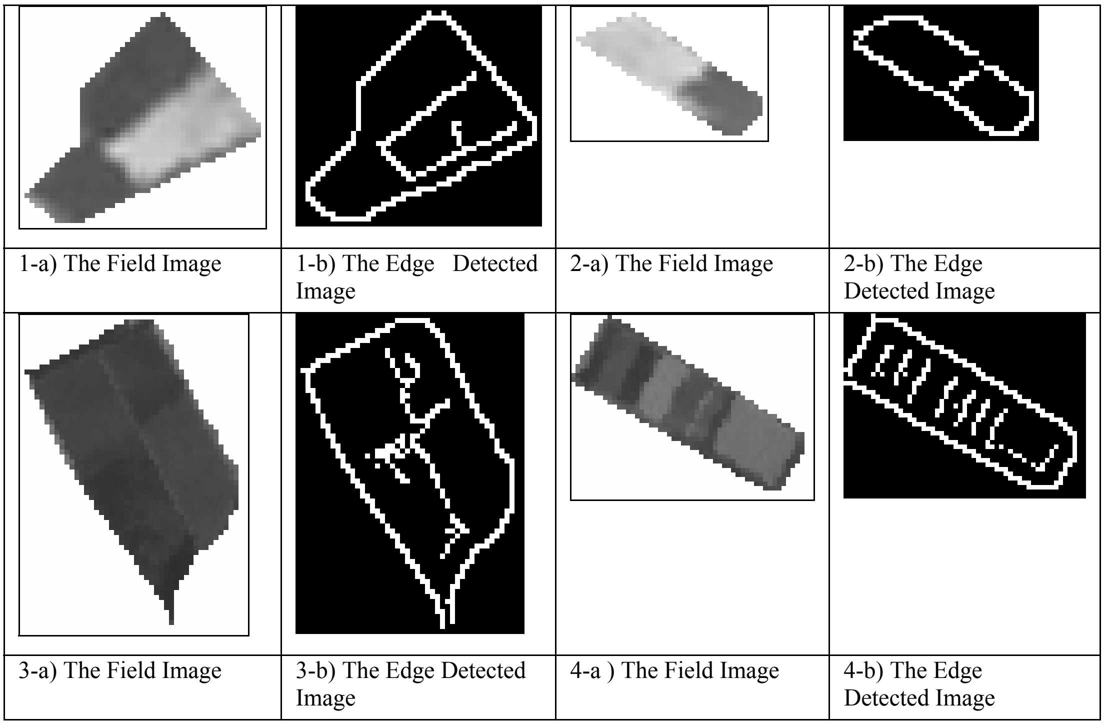
Edge detection is a common image processing operation for detecting the rapid color variations in an image. There are several edge detection methods applied on the images. The Canny Edge Detector is considered to be a popular and successful algorithm among these methods. Due to a good performance of the Canny Edge Detector with respect to the other gradient operators, and the ease of implementation, this technique is chosen to be used for detecting the edges in the proposed segmentation process.

In order to eliminate the noise, the Canny Edge Detector applies a smoothing operation through a Gaussian convolution of the image. The image gradient is found to highlight the regions with high spatial derivatives. Then, the algorithm tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds. If the magnitude of a pixel is below the first threshold, it is set to zero (decided as non-edge). On the other hand, if the magnitude of a pixel is above the high threshold, it is then set to white (decided as edge). And if the magnitude is between the two thresholds, it is set to zero unless there is a path from this pixel to a pixel with a gradient above the high threshold [14].

The effect of the Canny operator can be determined by three parameters; (i) the width of the Gaussian mask used in the smoothing phase, and (ii) the upper threshold, and (iii) lower threshold used by the tracker. Increasing the width of the Gaussian mask reduces the detector's sensitivity to noise at the expense of losing some of the finer detail in the image. The localization error in the detected edges also increases slightly as the Gaussian width is increased. Usually, the upper tracking threshold is set quite high, and the lower threshold is kept quite low to achieve good results. Setting the lower threshold too high would cause noisy edges to break up. Setting the upper threshold too low would increase the number of spurious and undesirable edge fragments appearing in the output.

In the proposed segmentation process, the lower threshold is selected as a very low value and the higher threshold is selected as a value which is not very high. This choice costs to detection of noisy edges. Otherwise the smooth transitions between different crop sub-fields would not be detected. The noise is recoverable and can be cleared by further processings of the segmentation. However, the undetected edges for the sub-field boundaries can never be realized and recovered through the further processings. As a general approach, over-segmentation is preferred to under-segmentation in the field-based segmentation procedure. These threshold values are adapted for the fields according to their sizes. While higher thresholds are used for larger fields, lower thresholds are preferred for smaller fields. In addition, too noisy outputs that have many number of edges detected more than a predefined threshold are re-processed by the edge detection with the higher thresholds.

After performing the edge detection process, a binary image is obtained that consists of white pixels forming the edge lines and black pixels representing the others. Several outputs of the edge detection process of the selected fields are shown in Figure 2.4.



1-a) The Field Image

1-b) The Edge Detected Image

2-a) The Field Image

2-b) The Edge Detected Image

3-a) The Field Image

3-b) The Edge Detected Image

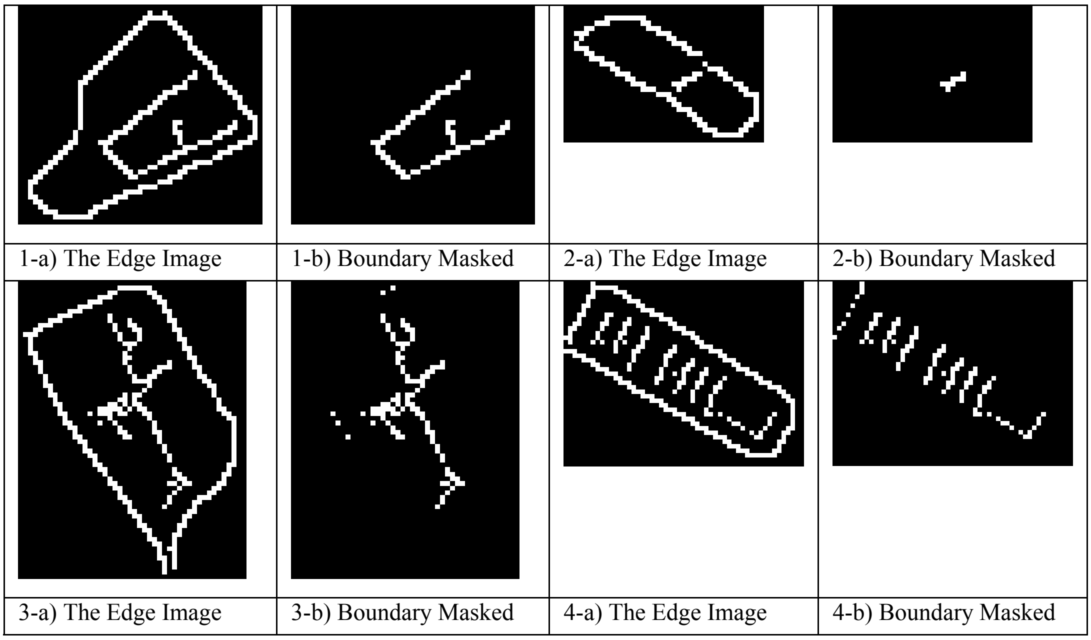
4-a) The Field Image

4-b) The Edge Detected Image

Figure 2.4 – The Sample Outputs of the Edge Detection Operation

2.2.2 Boundary Masking

Many pixels close to the existing field boundary lines will also be detected as the edge pixels through the edge detection process. This is because of the brightness difference between the boundary pixels and the pixels falling outside the boundary that are set as white pixels when the distinct field image is constructed. Here, the main concern is to generate the lines which may be candidates to form the sub-field boundaries within the field. Hence, the white pixels in the edge detected image, which are close to the original boundary lines, should be masked out. This operation is applied in raster domain. To do that, a buffer zone is determined using a threshold close to the original boundary lines and the white pixels residing inside this buffer are set to black. With this operation, only the edge pixels inside the field remain and these pixels may form the missing boundaries within the field. Several outputs of the boundary masking operation applied to the edge images are shown in Figure 2.5.



1-a) The Edge Image
1-b) Boundary Masked
2-a) The Edge Image
2-b) Boundary Masked
3-a) The Edge Image
3-b) Boundary Masked
4-a) The Edge Image
4-b) Boundary Masked

Figure 2.5 – Sample Outputs of the Boundary Masking Operation

2.2.3 Contour Detection

The contour detection is the process of detecting the coordinates of the line segments that may form the missing sub-field boundaries from the white pixels in the boundary masked binary image. It is basically a conversion process from raster form to vector form. There are several known methods and algorithms to perform such a conversion.

Most of the vectorization methods divide the conversion process into two steps; (i) crude vectorization and (ii) post-processing. They extract as many line segments as possible without junctions from raster image during the crude vectorization step. Then the line segments are extended and combined into exact graphic entities, such as straight lines, arcs, and curves, during the post-processing operation. The main reason for dividing the vectorization operation into two steps is that the vectorization methods cannot extract a graphic entity in one step. This is particularly true, if there are some complicated junctions on the processed image.

Typical examples of two-step vectorization are the skeletonization based methods, such as thinning based algorithm, contour based algorithm and Run Length Encoding (RLE) based algorithm [40]. Since the skeletonization is always based on morphology analysis, the noise and the junctions present in the image cause the distortion of segments, which would lead to failure or lower accuracy in the post-processing step. Another weakness is that when the number of segments is large, analyzing the junctions and searching for the collinear segments of an original entity is time-consuming. These methods are inefficient for vectorizing the real-life scanned raster image of engineering drawings with mixed text/graphics contents. The Sparse Pixel Vectorization (SPV) algorithm – a pixel tracking algorithm – is another example of two-step vectorization [10]. SPV uses a proper scan line interval and a tracking step to enhance the performance of vectorization. However, when the size of the junction is larger than the tracking step, an original entity will be broken up into several pieces. Therefore, SPV still need post-processing.

In a recent study, a different algorithm was presented, which does not require post-processing [16]. First, the algorithm extracts a seed segment of a graphic entity from a raster image to obtain its direction and width. Then, it tracks the pixels under the guidance of the direction so that the tracking can track through junctions and is not affected by the noise and the degradation of image quality. Thus, an entity is vectorized in one step without post-processing. The relations among lines are also used to realize the continuous vectorization of a line network.

In the present case, the Suzuki algorithm is used for the vectorization process [39]. First, the thinning of the binary image is performed and then a chain graph is constructed from the white pixels, which have another white pixel in its 8 neighborhood. The neighboring pixels are associated with each other and a chain graph is constructed form the raster data. All the lines that exist in the raster data can be extracted from these graphs. Figure 2.6 demonstrates an edge detected binary image and the extracted chain graph.

Image (I)

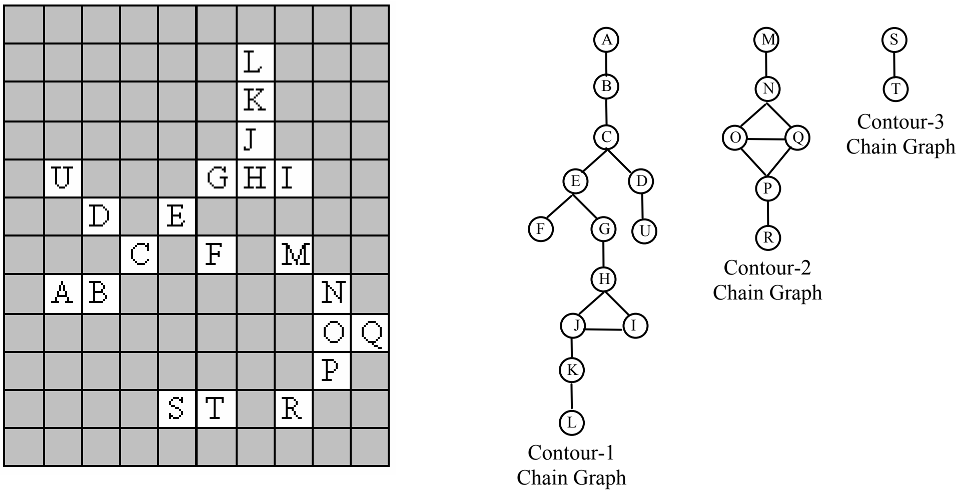


Figure 2.6 – A Binary Image and Contours Chain Graph

In Figure 2.6, the pixels that have letters in them stand for the white pixels, while all the empty pixels stand for the black pixels in the image. All the connections between the white pixels are represented as the chain graphs.

In the first step, all the paths are traced and those paths whose slopes are consistent are determined as separate line segments. But the connectivity relations between the line segments are preserved and the connected line segments are grouped to be used for the next step in which a line smoothing process will be applied. Each chain graph represents a set of connected line segments. This is called as "contour". There is no connection between the different contours. Considering the sample image and the constructed chain graphs given in Figure 2.6, the following contours are extracted from each graph.

Contour-1 = ([A-B], [B-J], [C-U], [H-L], [G-I], [E-F])
Contour-2 = ([M-Q], [O,Q], [N-P], [Q,R])
Contour-3 = ([S-T])

{Expresion. 2.1}

The output of this process is the set of contours where each contour is a group of connected line segments. With this vectorization process, the input is converted to vector form, and all the other processings of the segmentation will be applied in vector domain.

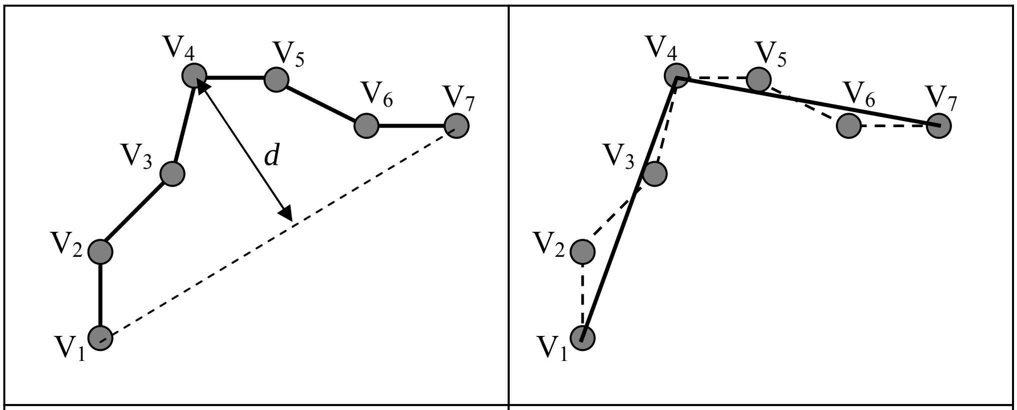
2.2.4 Line Simplification

The output of the contour detection process is the set of contours consisting of the connected line segments, each of which has a constant slope. In order to determine the closed regions within a field, it is necessary to make some associations between the line segments and the existing field boundaries or between the line segments. However, the set of line segments obtained through the contour detection operation becomes very large and may be cumbersome when used without any simplification. Thus, a line simplification procedure is needed to be applied to reduce the number of line segments, and to obtain bigger straight lines.

The Douglas–Peucker algorithm is known to be one of the most popular methods for line simplification [13]. The algorithm iteratively selects new points for inclusion in the thinned output polyline based on their deviation from a baseline connecting two neighboring points already chosen for the inclusion. The method can be best described recursively: To approximate the chain from V_i to V_j, start with the line segment V_iV_j. If the farthest vertex from this segment has distance at most threshold ε , then accept this approximation. Otherwise, split the chain at this vertex and recursively approximate the two pieces. The following pseudo code represents the algorithm more precise. *DouglasPeucker (V; i; j)* simplifies the chain from V_i to V_j.

```
DouglasPeucker(V; i; j)
1. Find the vertex Vf farthest from the line $V_iV_j$ .
        Let dist be its distance.
2. if dist > $\varepsilon$ then
        3a. DouglasPeucker (V; i; f)  /*Split at Vf and approximate recursively */
        3b. DouglasPeucker (V; f; j)  /* call for other split recursively */
  else
4. Output($V_iV_j$)                        /* Use ViVj as the approximated output */
```

Expression 2.2



a) The contour consisting of the connected line segments [V_1-V_2], [V_2-V_3]…[V_6-V_7]

b) The appearance of the contour after applying the Douglas-Peucker algorithm

Figure 2.7 – The Illustration of the Douglas-Peucker Algorithm

Figure 2.7 demonstrates the line simplification for the contour starting from point V_1 and ending with V_7. In the first iteration of the algorithm, the farthest vertex (V_4) of the contour to the straight line segment V_1V_7 is found. Since the distance (*d*) is bigger than the specified threshold, the straight line segment V_1V_7 is divided into two line segments V_1V_4 and V_4V_7. After applying the algorithm recursively to these line segments, no other splitting occurs and the whole contour having vertices $V_{1..7}$ is reduced to two line segments consisting of V_1V_4 and V_4V_7.

The following contours are obtained from the example contour set (Expression 2.1) extracted from the sample image given in Figure 2.6. The contours with the reduced line segments include:

Contour-1 = ([A-I], [H-L], [C-U])

Contour-2 = ([M-Q], [Q-R])

Contour-3 = ([S-T])

Expression. 2.3

Figure 2.8 is the display of the contours after the line simplification from the binary image given in Figure 2.6. Since the contours are in vector format, they are drawn on the image as straight lines and a pixel based notation is not used.

Image (I)

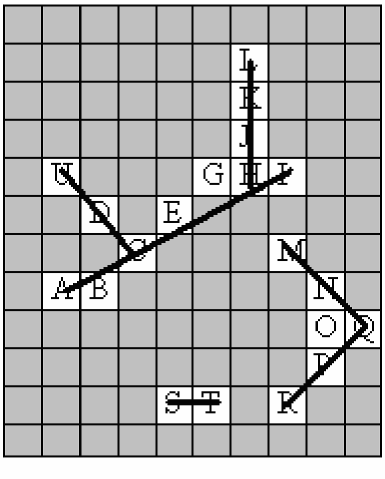
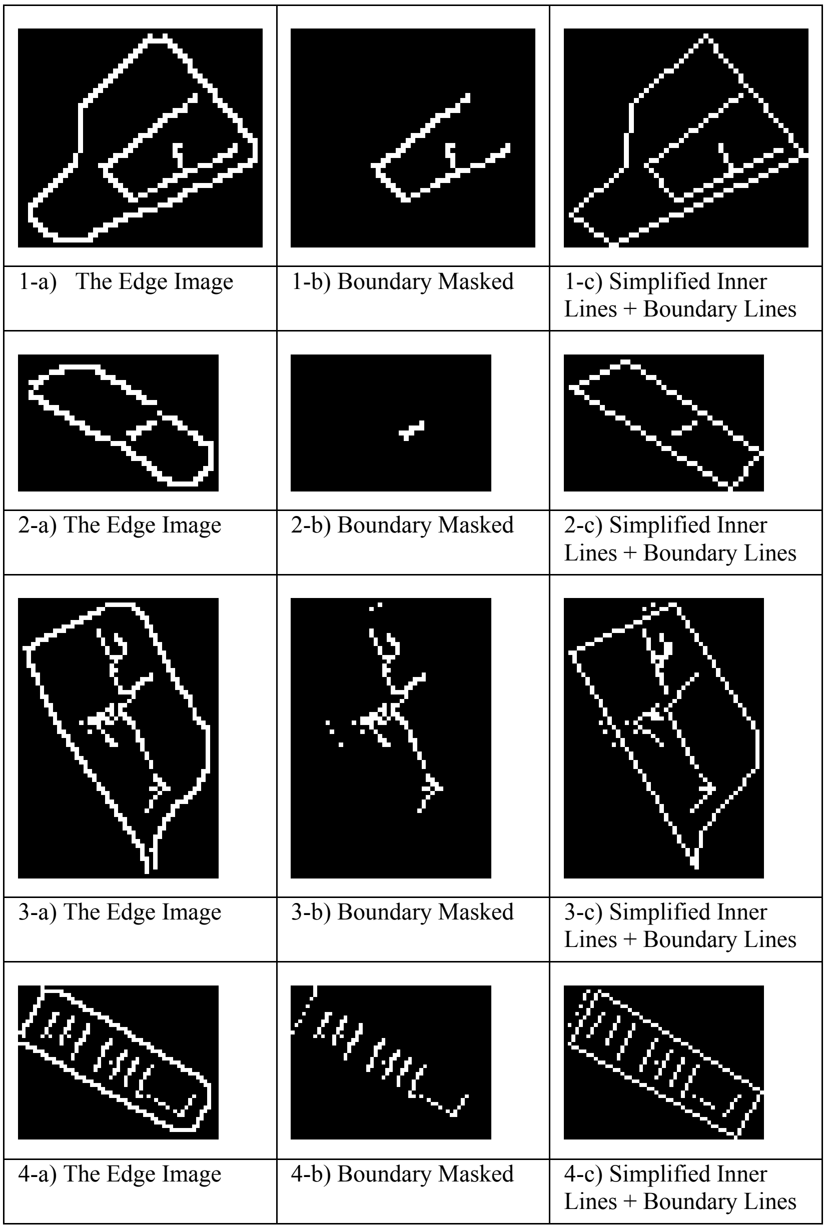


Figure 2.8 – The Illustration of the Contours after the Line Simplification Procedure

Figure 2.9 provides the example outputs of the line simplification process applied to the detected contours within the boundary masked image. The simplified contours are displayed with the existing field boundaries on images.



1-a) The Edge Image

1-b) Boundary Masked

1-c) Simplified Inner Lines + Boundary Lines

2-a) The Edge Image

2-b) Boundary Masked

2-c) Simplified Inner Lines + Boundary Lines

3-a) The Edge Image

3-b) Boundary Masked

3-c) Simplified Inner Lines + Boundary Lines

4-a) The Edge Image

4-b) Boundary Masked

4-c) Simplified Inner Lines + Boundary Lines

Figure 2.9 – The Sample Outputs of the Line Simplification Operation

2.2.5 Perceptual Grouping

An edge detection operation was applied within the field to find the candidate pixels that may form the boundaries of the sub-fields. Next a boundary masking operation was performed on the edge image to prevent generation of the redundant lines which will be close to the existing field boundaries. These two operations were applied in the raster domain. Then, the contours were extracted from the raster data through a raster to vector conversion operation. The contours are the connected line segments in the image and they are passed to the line simplification procedure. Now the input dataset consists of a set of coordinates of the simplified line segments and the coordinates of the vertices of the existing boundary lines of the field. The goal of the perceptual grouping is to obtain the closed polygons, which will form the sub-fields having a different type of crop.

In order to obtain the closed polygons, the vertices of the line segments must be associated with the existing field boundaries or with the other line segments. To do that, a rule based mechanism is designed. Basically, the process consists of two main steps;

- removing the noisy line segments, and
- modifying the vertices of the remaining line segments.

As mentioned earlier, the extracted contours are simplified using the Douglas-Peucker algorithm. However, due to the use of small threshold for Douglas-Peucker algorithm, there may still exist several redundant and noisy line segments in the input set. Initially these redundant lines are removed from the input set by evaluating the slope, and the distance values of the vertices of the line segments to other lines. Then, the vertices of the remaining line segments are analyzed, and the candidate connections are determined between the lines.

2.2.5.1 Input Structure

All the detected contours are stored in an input set. The existing field boundaries consisting of the connected lines are also included in this set as another contour. But this contour containing the existing field boundary is a special element for the whole set. This means that, the algorithm must not process boundary line segments just like other line segments. No modifications should be applied to the vertices of the boundary line segments. Therefore, a flag is included to the contour set structure indicating whether the line segments belong to the boundary or not. Considering the above descriptions, the following structure is decided to be used as the input set.

MS: Main Set
CS: Contour Set
LS: Line Segment
BI: Boundary Indicator
(x,y) = Coordinate of a Pixel in the Image I
MS_I *= { CS-1,CS-2,…}*
where
CS = { LS-1, LS-2, …, BI}
LS = $[(x_0,y_0), (x_1,y_1)]$ *{Expression 2.4}*

To clarify the input structure, Figure 2.10 illustrates a sample field that contains the contour sets extracted from the image. Each point having a letter and a numbered text indicates a pixel coordinate in the image. The vertices of the existing field boundaries have the labels starting with letter “B”. In turn, Figure 2.10 and the constructed contours will be used as the input for the further demonstration of the perceptual grouping rules.

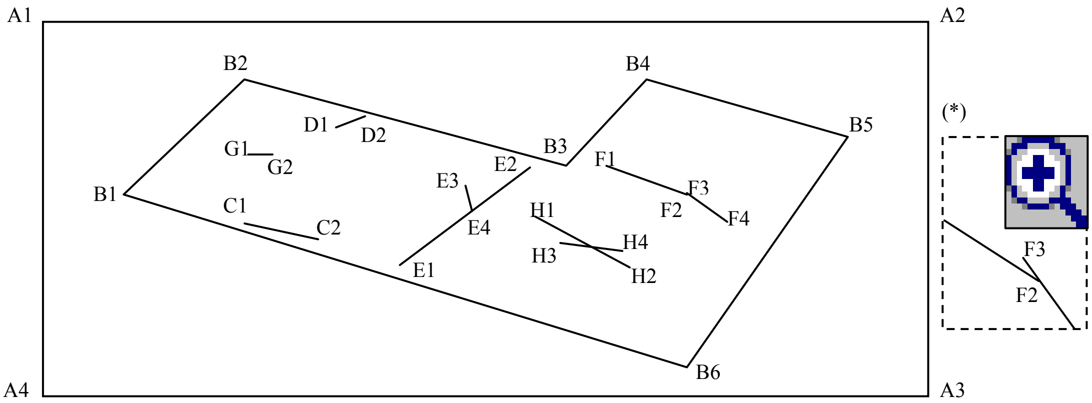


Figure 2.10 – Sample Input Image for Perceptual Grouping

(*) The zoomed in display of the F2 and F3 vertices.

MS= { ([B1-B2], [B2-B3], [B3-B4], [B4-B5], [B5-B6], [B6-B1], TRUE),
([C1-C2], FALSE),
([D1-D2], FALSE),
([E1-E2], [E3-E4], FALSE),
([F1-F2], [F3-F4], FALSE),
([G1-G2], FALSE),
([H1-H2],[H3-H4],FALSE) }; *{Expression 2.5}*

MS is the main set of contours extracted from the image of the field. The minimal bounding rectangle of the field (*A1, A2, A3, A4*) has no meaning, because the segmentation process is applied for each field, and within the field. Therefore, the region residing outside the boundary of the field will not be included in the processing. The first element of the set consists of the line segments of the existing field boundary. This contour element has the boundary indicator flag as *TRUE* and will be called boundary line segments or boundary contour in the following steps. The other contour elements consist of the line segments determined through the edge detection and the line smoothing process. These contour elements have the boundary indicator flag of *FALSE*.

2.2.5.2 Analysis Structure

For applying the rules of the perceptual grouping, an analysis is made between all the line segment pairs. The detected line segments are processed together with the existing field boundaries, and between each other. The analysis structure used for a line segment pair contains the following parameters (Table 2.1);

Table 2.1 – Line To Line Analysis Parameters

V_1L_1	One vertex of the first line segment
V_2L_1	Other vertex of the first line segment
V_1L_2	One vertex of the second line segment
V_2L_2	Other vertex of the second line segment
$\lvert L_1 \rvert$	Length of the first line segment
$\mu(L_1)$	Slope of the first line segment
$\lvert L_2 \rvert$	Length of the second line segment
$\mu(L_2)$	Slope of the second line segment
$\Delta\mu$	Slope difference between the line segments
$d\left(V_1L_1 - L_2\right)$	Distance from V_1L_1 to the line segment L2
$\theta\left(V_1L_1 - L_2\right)$	Angle of the line from V_1L_1 to the line segment L2
$p\left(V_1L_1 - L_2\right)$	Closest point to V_1L_1 on the line segment L2
$d\left(V_2L_1 - L_2\right)$	Distance from V_2L_1 to the line segment L2
$\theta\left(V_2L_1 - L_2\right)$	Angle of the line from V_2L_1 to the line segment L2
$p\left(V_2L_1 - L_2\right)$	Closest point to V_2L_1 on the line segment L2
$d\left(V_1L_2 - L_1\right)$	Distance from V_1L_2 to the line segment L1
$\theta\left(V_1L_2 - L_1\right)$	Angle of the line from V_1L_2 to the line segment L1
$p\left(V_1L_2 - L_1\right)$	Closest point to VL_2 on the line segment L1
$d\left(V_2L_2 - L_1\right)$	Distance from V_2L_2 to the line segment L1
$\theta\left(V_2L_2 - L_1\right)$	Angle of the line from V_2L_2 to the line segment L1
$p\left(V_2L_2 - L_1\right)$	Closest point to V_2L_2 on the line segment L1
$Int\left(L_1 - L_2\right)$	Intersection status of the two line segments
$IntP\left(L_1 - L_2\right)$	Intersection point of the two line segments
$IntD\left(L_1 - L_2\right)$	Shortest distance to the intersection point
$IntVI\left(L_1 - L_2\right)$	Extended vertex index to satisfy the intersection

Therefore, the analysis parameters between each of the line segment pairs consist of the above structure, and all the perceptual grouping rules are applied based on the values of these parameters.

The first four parameters (V_1L_1, V_1L_2, V_2L_1 and V_2L_2) indicate the vertex locations of the line segments. The parameters $|L_1|$ and $|L_2|$ are the lengths of the line segments in pixel units. The slope of each line segment is stored using the parameters $\mu(L_1)$ and $\mu(L_2)$, where $\Delta\mu$ is used for the smaller angle values between the analyzed line segments.

The other parameters stand for the analysis results between the vertices of one line segment and the other line segment. The distance of each vertex of one line segment to other, and the angle between the line segment and the shortest line that can be drawn from that point to that line segment is calculated and stored in the corresponding parameter. For example, $d(V_1L_1 - L_2)$ is used for the distance between the first vertex of the first line segment and the second line segment. $\theta(V_1L_1 - L_2)$ is the angle between the second line segment and the shortest line that can be drawn from that vertex to the second line segment. Similarly, $p(V_1L_1 - L_2)$ is calculated for each vertex and is the closest point on one line segment to the vertex of the other line segment. Figure 2.11 demonstrates the meaning of these parameters.

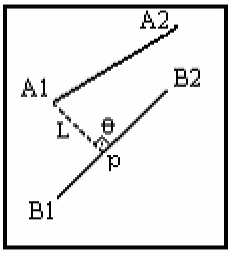
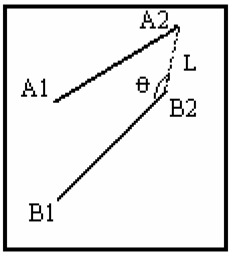
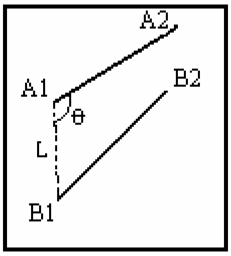
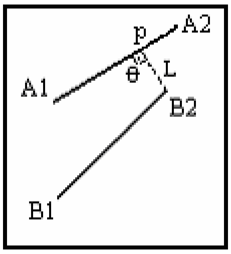
			
(a) L=Shortest line that can be drawn from A1 to segment B Θ=Angle between line L and segment B (90°) $d\left(\mathrm{V_1L_1 - L_2}\right)$ = *length(L);* $\theta\left(\mathrm{V_1L_1 - L_2}\right)$ = Θ; $p\left(\mathrm{V_1L_1 - L_2}\right)$=p	(b) L=Shortest line that can be drawn from A2 to segment B Θ=Angle between line L and segment B $d\left(\mathrm{V_2L_1 - L_2}\right)$ = *length(L);* $\theta\left(\mathrm{V_2L_1 - L_2}\right)$ = Θ; $p\left(\mathrm{V_2L_1 - L_2}\right)$ = B2	(c) L=Shortest line that can be drawn from B1 to segment A Θ=Angle between line L and segment A $d\left(\mathrm{V_1L_2 - L_1}\right)$ = *length(L);* $\theta\left(\mathrm{V_1L_2 - L_1}\right)$ = Θ; $p\left(\mathrm{V_1L_2 - L_1}\right)$ = A1	(d) L=Shortest line that can be drawn from B2 to segment A Θ=Angle between line L and segment A (90°) $d\left(\mathrm{V_2L_2 - L_1}\right)$ = *length(L);* $\theta\left(\mathrm{V_2L_2 - L_1}\right)$ = Θ; $p\left(\mathrm{V_2L_2 - L_1}\right)$=p

Figure 2.11 – Line To Line Distance Parameters

There is also a group of analysis parameters related to the intersection status of the two line segments. The parameter $Int\left(\mathrm{L_1 - L_2}\right)$ is used as an enumerated type indicating whether the line segments can intersect or not, in case of extending the vertices of the line segments. The following five different possible situations about the intersection of the two line segments can be considered;

- Never Intersects (line segments are parallel)
- Possible Intersection On First (the possible intersection point will be on the first line segment, in case of extending the second line segment)
- Possible Intersection On Second (the intersection point will be on the second line segment, in case of extending the first line segment)
- Possible Intersection Out (the intersection point will neither be on the first line segment nor on the second. Both segments should be extended to satisfy the intersection)
- Possible Intersection On Both (already intersected line segments, no extensions are needed)

The attribute $IntP(L_1 - L_2)$ for storing the coordinate of the possible intersection point is included in the analysis structure. In addition, the attribute $IntD(L_1 - L_2)$ is used for storing the length of the line extension needed to intersect the line segments. The attribute $IntVI(L_1 - L_2)$ indicates the index of the vertex that should be modified to provide the intersection.

Figure 2.12 is given to clarify the meanings and the use of these attributes related to intersection. The figure demonstrates the possible intersection types between the first line segment [A1-A2] and the second line segment [B1-B2];

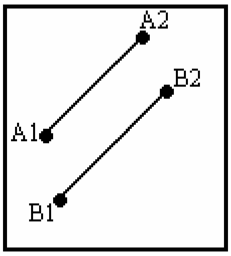
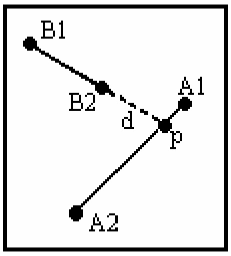
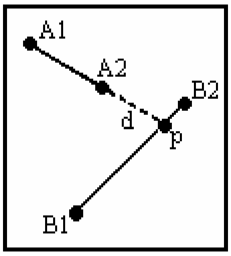
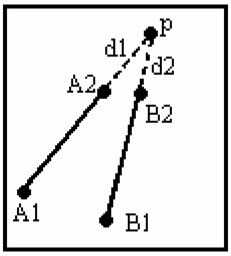
			
(a) Two line segments parallel. $Int(L_1 - L_2) =$ **Never** $IntP(L_1 - L_2) = NA$ $IntD(L_1 - L_2) = NA$ $IntVI(L_1 - L_2) = NA$ *NA: Not Applicable*	(b) Intersection possible. $Int(L_1 - L_2) =$ **OnFirst** If one vertex of the second line segment is extended to the first line segment, preserving the slope, an intersection occurs on first line segment $IntP(L_1 - L_2) = p$ $IntD(L_1 - L_2) =$ length(d) $IntVI(L_1 - L_2) = B2$	(c) Intersection possible. $Int(L_1 - L_2) =$ **OnSecond** If one vertex of the first line segment is extended to the second line segment, preserving the slope, an intersection occurs on second line segment $IntP(L_1 - L_2) = p$ $IntD(L_1 - L_2) =$ length(d) $IntVI(L_1 - L_2) = A2$	(d) Intersection possible. $Int(L_1 - L_2) =$ **Out** If one vertex of the each line segments are extended preserving the slopes, an intersection occurs at point p outside the line segments. $IntP(L_1 - L_2) = p$ $IntD(L_1 - L_2) =$ min(d1,d2) $IntVI(L_1 - L_2) = NA$

Figure 2.12 – Line To Line Intersection Parameters

Above given possibilities are the normal situations for the intersections of two line segments, that are not already intersected. Figure 2.13 represents the situations for the already intersected segment pairs. In addition, an extreme case is presented in Figure 2.13-d, where the line segments are parallel but an intersection is still possible.

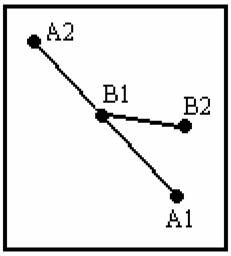
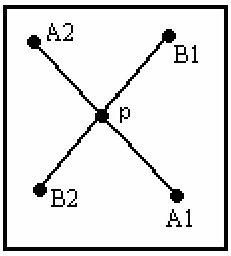
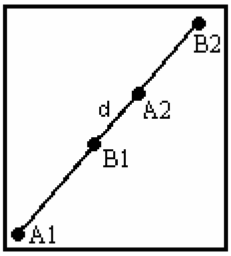
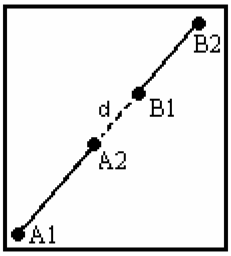
			
(a) Two line segments already have an intersection. No extension needed. $Int(\mathrm{L}_1 - \mathrm{L}_2) =$ **OnBoth** $IntP(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{B1}$ $IntD(\mathrm{L}_1 - \mathrm{L}_2) = 0$ $IntVI(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{B1}$	(b) Two line segments already have an intersection. No extension needed. $Int(\mathrm{L}_1 - \mathrm{L}_2) =$ **OnBoth** $IntP(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{p}$ $IntD(\mathrm{L}_1 - \mathrm{L}_2) = 0$ $IntVI(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{NA}$ (Not Applicable)	(c) Two parallel line segments already have an intersection. $Int(\mathrm{L}_1 - \mathrm{L}_2) =$ **OnBoth** $IntP(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{NA}$ $IntD(\mathrm{L}_1 - \mathrm{L}_2) = 0$ $IntVI(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{NA}$ (Not Applicable)	(d) Intersection possible. $Int(\mathrm{L}_1 - \mathrm{L}_2) =$ **OnFirst.** $IntP(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{A2}$ $IntD(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{d}$ $IntVI(\mathrm{L}_1 - \mathrm{L}_2) = \mathrm{B1}$ There is no difference of selecting the status as **OnSecond** and the intersectable vertex index as A2 with the above selection. The output becomes the same for both line extensions, if the connection is decided to be made.

Figure 2.13 – Line To Line Intersection Parameters Continued

2.2.5.3 Perceptual Grouping Rules

The perceptual grouping rules are based on the above described analysis parameters. For each pair of the two line segment, these analysis are performed and an iterative decision mechanism is used to modify the line segments in order to extract the closed polygons.

The input field provided in Figure 2.10 and the constructed contour sets given in Expression. 2.5 will be used to demonstrate the line segment modification rules step by step. Also the pseudo codes of the rules will be given as the mathematical expressions to describe the algorithms. The main modification steps applied for perceptual grouping of the line segments can be listed as follows;

i. removing the overlapping and the intersecting line segments in each contour set,
ii. removing the close line segments that are in different contour sets,
iii. extending the line segments that can intersect with the existing field boundaries,
iv. extending the line segments that can intersect with each other,
v. removing the line segments that are not extended and shorter than the pre-defined threshold,
vi. modifying the vertices of the line segments that have open ends by moving the vertex to intersect with the closest line segment,
vii. removing the dangling arcs, and
viii. removing the overlapping line segments and resolving the deviations

For each field, the perceptual grouping rules are applied to the contours extracted from the image for that field. These contours are expressed as the elements of the main input set (MS). The line segments (LS), which belong to the same contour, are expressed with a contour set (CS). The mathematical expressions of the rules will be given using these set notations. The line segments of the existing field boundaries are included in the main set as a contour with the boundary indicator flag set to TRUE. As a common principle for all the rules, the boundary line segment set is never modified by checking the boundary indicator flag. Each rule is described below in detail.

i. The first rule is applied to each contour of the main set except the one which consists of the real boundary line segments (boundary indicator is equal to TRUE). The line segments belonging to the same contour set (CS) are processed between each other. And those which are very close to another line segment are removed from the contour sets. The mathematical expression of the algorithm of Rule-1 is as follows;

```
$\forall CS_i \in MS \; where \; BI(CS_i) = FALSE$
        $\forall LS_m, LS_n \in CSi$
        $analyze(LS_m, LS_n)$
        $if \; (\Delta\mu \leq ConnectedLineSlopeThreshold)$
                $if \; (d\left(\mathrm{V_1L_1 - L_2}\right) \leq ConnLineDistThreshold \; and$
                    $d\left(\mathrm{V_2L_1 - L_2}\right) \leq ConnLineDistThreshold)$
                        $remove(LS_m)$
                $else \, if \; (d\left(\mathrm{V_1L_2 - L_1}\right) \leq ConnLineDistThreshold \; \mathrm{and}$
                        $d\left(\mathrm{V_2L_2 - L_1}\right) \leq ConnLineDistThreshold)$
                        $remove(LS_n)$
```

$Rule1$

All the line segment pairs belonging to same contour set are processed with each other and the redundant line segments are removed. The line segments having the slope difference ($\Delta\mu$) smaller than a defined threshold value are kept and used for further processing. The line segments whose both vertices are very close to an other line segment (distance from the vertex to other line segment is smaller than a defined threshold) are removed. After applying this rule to input set, the output given in Figure 2.14 is obtained. For example, the line segment [H3-H4] is removed from the set since this segment satisfies the removal condition based on the result of the analysis with the line segment [H1-H2]. The line segment [E3-E4] is not removed because of the slope difference angle with the line segment [E1-E2] although its vertices are very close to that line segment.

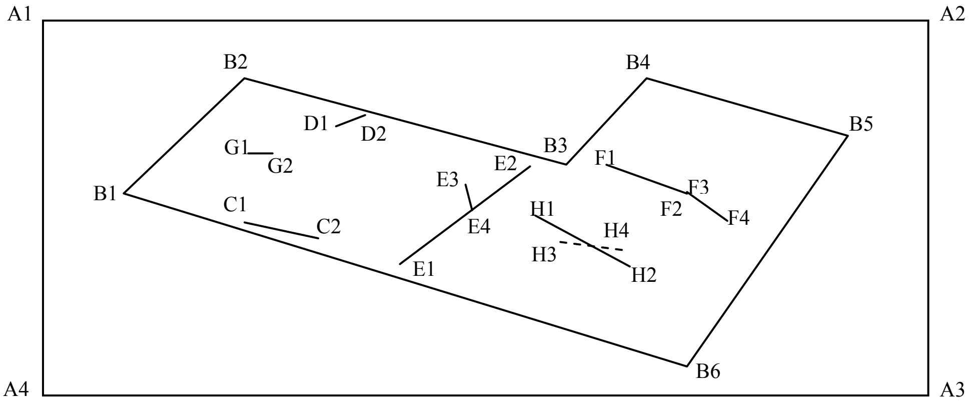


Figure 2.14 – The Status of the Contours after ApplyingRule-1 of the Perceptual Grouping

ii. The second rule is similar to the first rule. The difference is that this rule is applied to the line segment pairs that are in different contour set. In addition, a different threshold value is used from the value used for Rule-1. Note that the existing field boundaries are not totally exempt this time. They are involved in the analysis in order to find the line segments that are close to the boundaries and delete those that satisfy the removal condition described in Rule-1. The mathematical expression of the algorithm used of Rule-2 is as follows;

```
$\forall CS_i, CS_j \in MS$
        $\forall LS_m, \in CS_i, \forall LS_n \in CS_j$
        $analyze(LS_m, LS_n)$
        $if\ (\Delta\mu \leq SlopeThreshold)$
                $if\ (BI(CS_i) = FALSE\ and$
                    $d\left(\mathrm{V_1L_1 - L_2}\right) \leq DistanceThreshold\ \ and$
                    $d\left(\mathrm{V_2L_1 - L_2}\right) \leq DistanceThreshold)$
                        $remove(LS_m)$
                $if\ (BI(CS_j) = FALSE\ and$
                    $d\left(\mathrm{V_1L_2 - L_1}\right) \leq DistanceThreshold\ \ and$
                    $d\left(\mathrm{V_2L_2 - L_1}\right) \leq DistanceThreshold)$
                        $remove(LS_n)$                                    $Rule\,2$
```

Upon applying Rule-2 to the line set modified using Rule-1, the output illustrated in Figure 2.15 is obtained. The line segment [C1-C2] is removed from the set, since this segment satisfies the removal condition according to the analysis result with the boundary line segment [B6-B1]. The line segment [D1-D2] is not removed because of the slope angle difference with the line segment [B2-B3], although it is very close to that line segment.

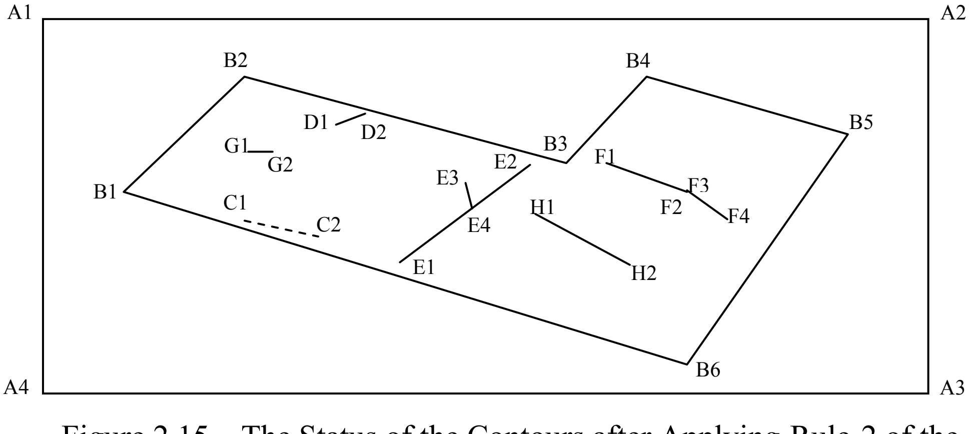


Figure 2.15 – The Status of the Contours after Applying Rule-2 of the Perceptual Grouping

iii. The third rule is based on extending the line segments to check if there is an intersection possibility with the existing field boundaries. The line segments of the contours are processed with the existing boundary line segments. If there is an intersection possibility when the line segment is extended towards the boundary line by a value smaller than a predefined threshold, then the corresponding vertex of the line segment is modified so that it is connected to the boundary. The mathematical expression of the algorithm of Rule-3 is as follows;

```
$\forall CS_i \in MS, \exists CS_j \in MS\ where\ BI(CS_i) = FALSE\ and\ BI(CS_j) = TRUE$
    $\forall LS_m, \in CS_i, \forall LS_n \in CS_j$
    $analyze(LS_m, LS_n)$
    $if\ (Int(\mathrm{L}_1 - \mathrm{L}_2) == \mathrm{ON\_SECOND}$ and
      $IntD(\mathrm{L}_1 - \mathrm{L}_2) \leq boundary\_intersection\_thershold)$
        $if\ (IntVI(\mathrm{L}_1 - \mathrm{L}_2) == 1)\quad [x_0, y_0]\ of\ \mathrm{LS}_\mathrm{m} = IntP(\mathrm{L}_1 - \mathrm{L}_2)$
        $\mathrm{if}\ (IntVI(\mathrm{L}_1 - \mathrm{L}_2) == 2)\quad [x_1, y_1]\ of\ \mathrm{LS}_\mathrm{m} = IntP(\mathrm{L}_1 - \mathrm{L}_2)$
```

Rule 3

After applying Rule-3 to the line set modified through the previous step, the output given in Figure 2.16 is obtained. The vertices of the line segments are modified so that they could intersect with the boundary lines. For example, both vertices of the line segment [E1-E2] are extended towards the boundary line segments [B6-B1] and [B2-B3]. Similarly, vertex D2 of the line segment [D1-D2] is extended towards the line segment [B2-B3], and vertex F1 of the line segment [F1-F2] is extended towards the boundary line segment [B3-B4]. Note that, although vertex F4 of the line segment [F3-F4] can be extended to intersect with the boundary line segment [B5-B6], it is not modified because of the distance required for the needed extension which exceeds the predefined threshold value.

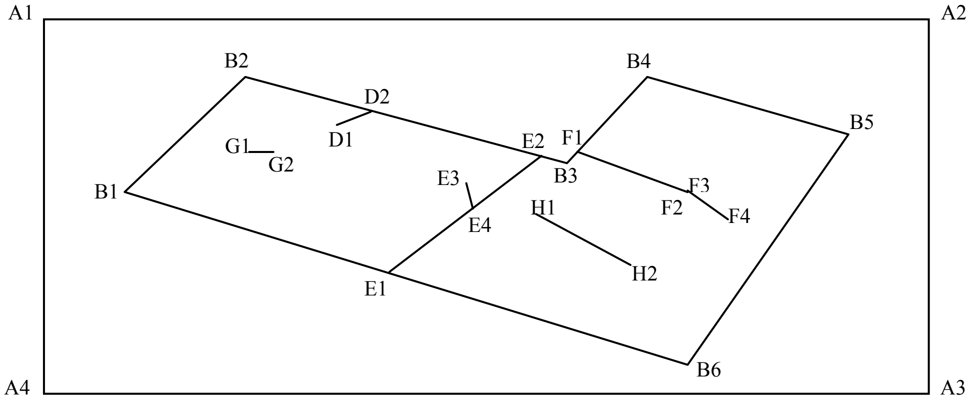


Figure 2.16 – The Status of the Contours after Applying Rule-3 of the Perceptual Grouping

iv. The fourth rule is similar to third rule. In Rule-3 all the line segments are processed with the boundary line segments only. In this rule, the line segments are processed with each other so that they can be modified to have the connections between each other. Note that, a different threshold value is used from the one used in Rule-3. The set of line segments which have the existing field boundary is exempt from this step. The mathematical expression of the algorithm of Rule-4 is as follows;

```
$\forall CS_i \text{ and } \forall CS_j \in MS \text{ where } BI(CS_i) = FALSE \text{ and } BI(CS_j) = FALSE$
    $\forall LS_m, \in CS_i, \forall LS_n \in CS_j$
    $analyze(LS_m, LS_n)$
    $if\,(Int(\mathrm{L}_1 - \mathrm{L}_2) == \mathrm{ON\_SECOND}$ and
       $IntD(\mathrm{L}_1 - \mathrm{L}_2) \le intersection\_threshold)$
          $if\,(IntVI(\mathrm{L}_1 - \mathrm{L}_2) == 1)\quad [x_0, y_0]\,of\ \mathrm{LS_m} = IntP(\mathrm{L}_1 - \mathrm{L}_2)$
          $\mathrm{if}\,(IntVI(\mathrm{L}_1 - \mathrm{L}_2) == 2)\quad [x_1, y_1]\,of\ \mathrm{LS_m} = IntP(\mathrm{L}_1 - \mathrm{L}_2)$
    $if\,(Int(\mathrm{L}_1 - \mathrm{L}_2) == \mathrm{ON\_FIRST}$ and
       $IntD(\mathrm{L}_1 - \mathrm{L}_2) \le intersection\_threshold)$
          $if\,(IntVI(\mathrm{L}_1 - \mathrm{L}_2) == 3)\quad [x_0, y_0]\,of\ \mathrm{LS_n} = IntP(\mathrm{L}_1 - \mathrm{L}_2)$
          $\mathrm{if}\,(IntVI(\mathrm{L}_1 - \mathrm{L}_2) == 4)\quad [x_1, y_1]\,of\ \mathrm{LS_n} = IntP(\mathrm{L}_1 - \mathrm{L}_2)$
```

Rule 4

Upon applying Rule-4 to the line set obtained after Rule-3, the output given in Figure 2.17 is obtained. The vertices of the line segments are modified by extending them in the direction of the possible intersections. For example, vertex H1 of the line segment [H1-H2] is extended towards to the line segment [E1-E2]. No other modification is applied using Rule-4.

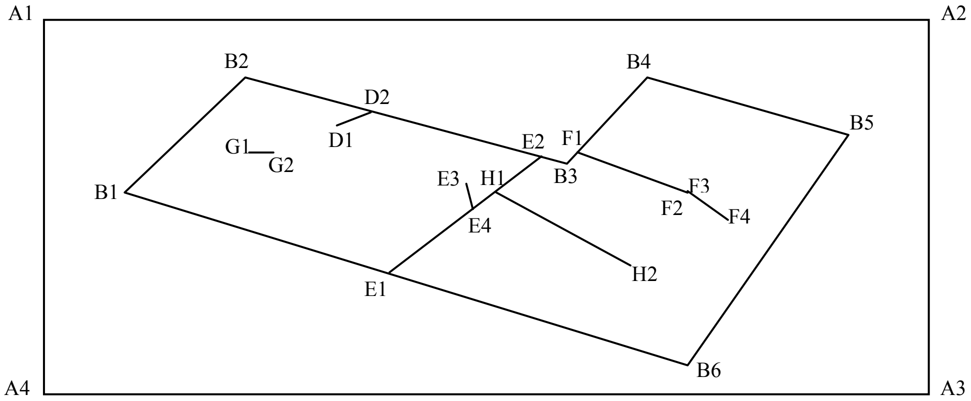


Figure 2.17 – The Status of the Contours after Applying Rule-4 of the Perceptual Grouping

v. In this rule, the length of all the line segments in a contour is computed and those contours shorter than the predefined threshold are removed. The contour set consisting of the existing field boundary is exempt from this step. The mathematical expression of the algorithm of Rule-5 is as follows;

```
$\forall CS_i \in MS \text{ where } BI(CS_i) = FALSE$
        $\forall LS_m \in CS_i$ where $LS_m$ is not extended by the previous rules
                $totalLength = totalLength + |LS_m|);$
        if $totalLength \le length\_threshold$
        $remove(CS_i)$                                        Rule 5
```

Upon applying Rule-5 to the line set obtained after Rule-4, the output given in Figure 2.18 is obtained. The small line segments having no connections with the other lines are deleted from the set. For example, the contour consisting of the line segment [G1-G2] is removed from the main input set. Although the contour consisting of the line segment [D1-D2] has a very small length, it is not removed. This is because this line segment was extended using Rule-3 to have a connection with the existing field boundary. Such small segments having connections with the other segments can be in a bridge position for two other line segments. Therefore, they should not be removed. Although the line segment [D1-D2] is not in a position like that it is not removed due to the constraint in the definition of this rule.

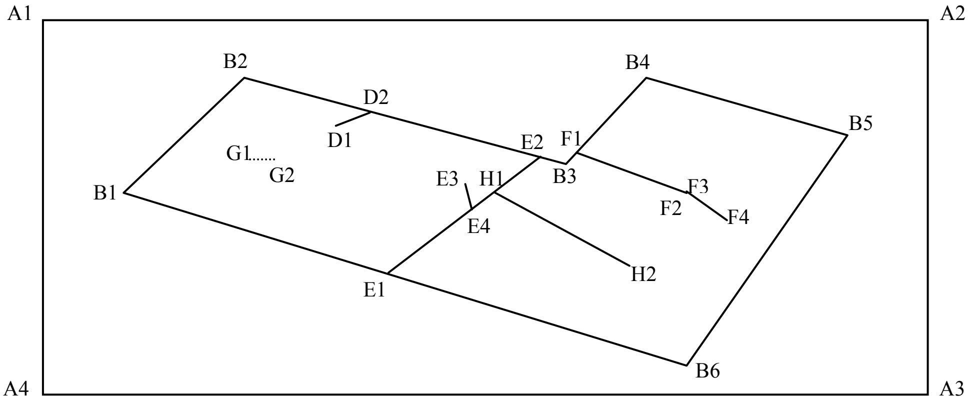


Figure 2.18 – The Status of the Contours after Applying Rule-5 of the Perceptual Grouping

vi. In this rule, the vertices of the line segments which have no connections with the other line segments are modified. The open ends of the line segments are moved to the closest points of the other line segments. A threshold is defined to limit the distance of these movements for the vertices. In the previous steps, the line segments were modified by extending them only and the extensions were applied to preserve the slope of the line segments. In this rule, since the closest points will be chosen as the new location of the vertices, the modification may apparently affect the slope of the line segment. The analysis parameter $p\left(\mathrm{V}_x\mathrm{L}_\mathrm{m} - \mathrm{L}_n\right)$ described in Figure 2.12 and Figure 2.13 is used for selecting the closest point locations. The mathematical expression of the algorithm of Rule-6 is as follows;

```
$\forall CS_i, CS_j \in MS$
    $\forall LS_m, \in CS_i, \forall LS_n \in CS_j \; where \; BI(CS_i) = FALSE$
    $analyze(LS_m, LS_n)$
     $for \; the \; open \; end \; vertices \; of \; LS_m$
       $find \; movementForV_1 = \min(d\left(\mathrm{V}_1\mathrm{L}_1 - \mathrm{L}_n\right) \text{for } \forall \mathrm{LS}_\mathrm{n})$
       $find \; movementForV_2 = \min(d\left(\mathrm{V}_2\mathrm{L}_1 - \mathrm{L}_n\right) \text{for } \forall \mathrm{LS}_\mathrm{n})$
    $if \; (movementForV_1 \leq distance\_threshold)$
          $[x_0, y_0] \, of \; \mathrm{LS}_\mathrm{m} = p\left(\mathrm{V}_1\mathrm{L}_1 - \mathrm{L}_n\right)$
    $if \; (movementForV_2 \leq distance\_threshold)$
          $[x_1, y_1] \, of \; \mathrm{LS}_\mathrm{m} = p\left(\mathrm{V}_2\mathrm{L}_1 - \mathrm{L}_n\right)$
                                                                    $Rule\,6$
```

Upon applying this rule to the line set obtained after Rule-5, the output given in Figure 2.19 is obtained. The line segments that have open end vertices are tried to be modified by moving the vertex to the closest point of another segment, unless the distance of movement exceeds the predefined threshold. The line segment [D1-D2] whose original location is shown as dashed line in Figure 2.19 is modified by moving the vertex D1 to the closest point on the line segment [B2-B3]. Similarly, another open end vertex E3 of the line segment [E3-E4] is moved to a point on the line segment [E1-E2]. The vertex F4 of the line segment [F3-F4] is moved to the closest point on the line segment [B5-B6]. The vertex F3 is connected with the vertex F2, which is the closest point of the all line segments. By connecting the vertices F2 and F3, a continuous contour “F” is obtained. The enlarged display of the F2 and F3 vertices is given in

Figure 2.19. Note that the open end H2 cannot be modified as the distance to closest point exceeds the predefined threshold.

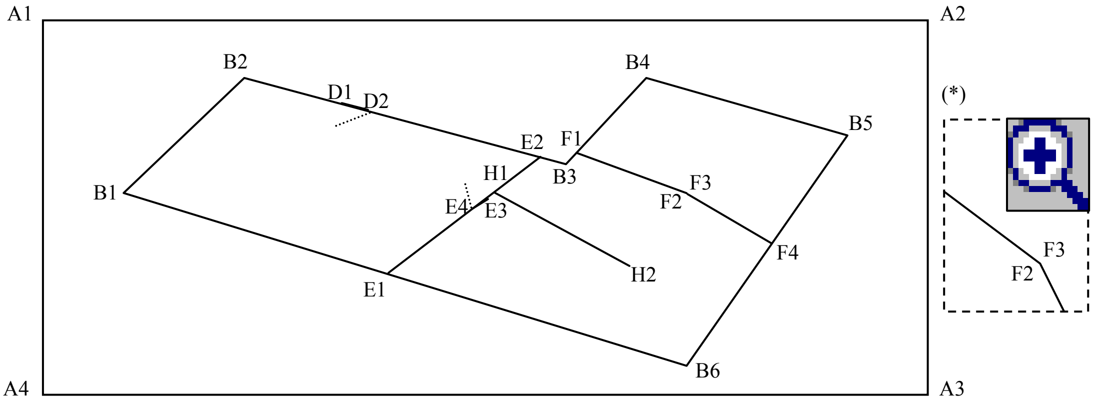


Figure 2.19 – The Status of the Contours after Applying Rule-6 of the Perceptual Grouping

(*) The zoomed in display of the F2 and F3 vertices.

vii. In the previous step, all the open end line segments were tried to be modified by moving the vertices to the closest line segments in order to make the connections between the line segments. However, if the minimum distance between a vertex of line segment and the closest line segment exceeds the predefined threshold, then the modification was not applied. This means that, there can still be line segments with no connections to other line segments. They have no meanings to form the closed regions and therefore, the dangling arcs with the open ends are deleted. The mathematical expression of the algorithm of Rule-6 is as follows;

```
$\forall CS_i \in MS \text{ where } BI(CS_i) = FALSE$
    $\forall LS_m \in CS_i \text{ where } LS_m \text{ still has an open end}$
        $remove(LS_m)$
```

Rule 7

After applying this rule, the output given in Figure 2.20 is obtained. The dangling arc [H1-H2] shown as a dashed line in the figure is removed from the line set.

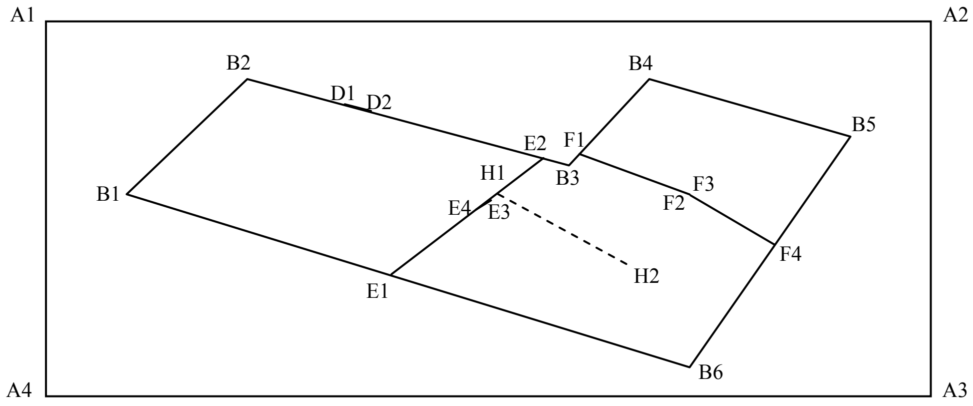


Figure 2.20 – The Status of the Contours after Applying Rule-7 of the Perceptual Grouping

viii. In this step, Rule-2 is applied with an additional functionality. The overlapping line segments are removed in order to find the distinct connected lines that form the closed polygons within the field. In addition, the vertices that are very close to each other are also merged in order to reduce the number of vertices. This is an important issue for reducing the computational complexity in the following steps.

Upon applying this rule the final output of the perceptual grouping is obtained (Figure 2.21). The line segments [D1-D2], and [E3-E4] are removed because they were almost on the line segments [B2-B3] and [E1-E2] respectively. A further modification is applied on the vertex F1 of the line segment [F1-F2]. The location of F1 is shifted so as it could coincide with B3. The line segment [F3-F4] is converted to the line segment [F2-F4] because F2 and F3 were exactly the same points.

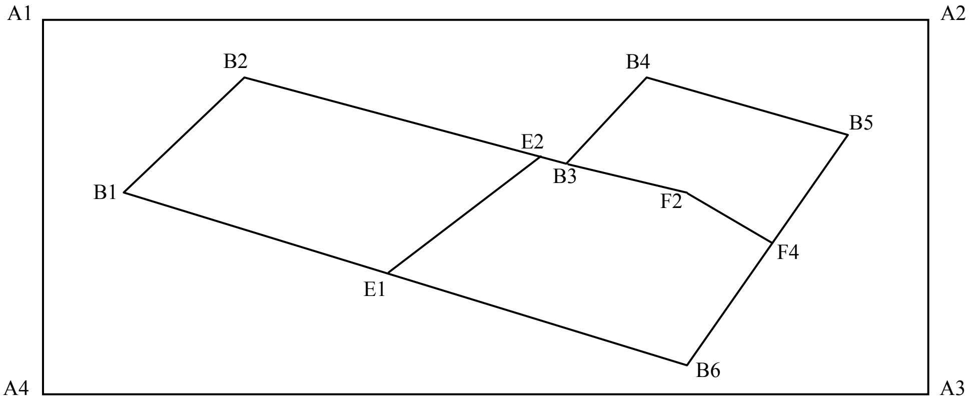


Figure 2.21 – The Status of the Contours after Applying Rule-8 of the Perceptual Grouping

After applying the above given perceptual grouping rules, the following set of contours (MS) is obtained;

MS = { ([B1-B2], [B2-B3], [B3-B4], [B4-B5], [B5-B6],[B6-B1], TRUE),
([E1-E2], FALSE),
([B3-F2], [F2-F4], FALSE) }; *{Expression 2.6}*

As a final step, this output set is modified in order to define every line segment considering the applied connections. For example, the line segment [B6-B1] must be re-expressed using two line segments because of the vertex E1. The line segment [B6-B1] must be converted to the line segments [B1-E1] and [E1-B6]. Thus, all the line segments are grouped in the same contour set. And for the the contour set the boundary indicator flag is set to TRUE, because all the detected line segments in the set are the boundaries of the sub-fields, and are not temporary lines any more. The final output set becomes as follows:

MS = { ([B1-B2], [B2-E2], [E2-B3], [B3-B4], [B4-B5], [B5-F4], [F4-B6],[B6-E1], [E1-B1], ([E1-E2],([B3-F2], [F2-F4], TRUE) }

{Expression 2.7}

The sample outputs obtained after applying the perceptual grouping rules given in Figure 2.22.

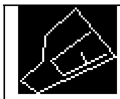
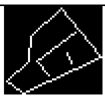
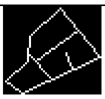
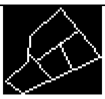
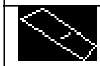
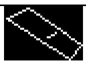
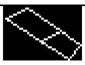
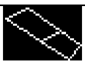
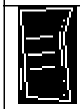
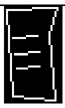
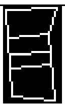
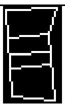
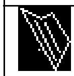
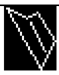
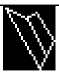
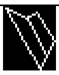
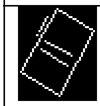
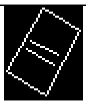
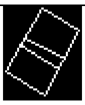
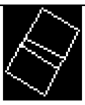
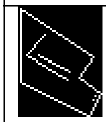
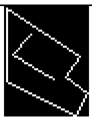
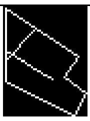
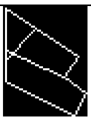
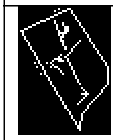
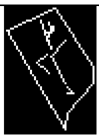
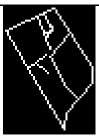
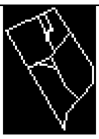
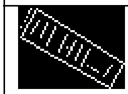
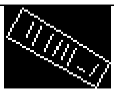
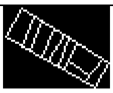
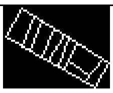
			
1-a) Smoothed Inner Lines + Boundary Lines	1-b) Rule 1&2 applied	1-c) Rule 3&4&5 applied	1-d) Rule 6&7 applied (Final output)
			
2-a) Smoothed Inner Lines + Boundary Lines	2-b) Rule 1&2 applied	2-c) Rule 3&4&5 applied	2-d) Rule 6&7 applied (Final output)
			
3-a) Smoothed Inner Lines + Boundary Lines	3-b) Rule 1&2 applied	3-c) Rule 3&4&5 applied	3-d) Rule 6&7 applied (Final output)
			
4-a) Smoothed Inner Lines + Boundary Lines	4-b) Rule 1&2 applied	4-c) Rule 3&4&5 applied	4-d) Rule 6&7 applied (Final output)
			
5-a) Smoothed Inner Lines + Boundary Lines	5-b) Rule 1&2 applied	5-c) Rule 3&4&5 applied	5-d) Rule 6&7 applied (Final output)
			
6-a) Smoothed Inner Lines + Boundary Lines	6-b) Rule 1&2 applied	6-c) Rule 3&4&5 applied	6-d) Rule 6&7 applied (Final output)
			
7-a) Smoothed Inner Lines + Boundary Lines	7-b) Rule 1&2 applied	7-c) Rule 3&4&5 applied	7-d) Rule 6&7 applied (Final output)
			
8-a) Smoothed Inner Lines + Boundary Lines	8-b) Rule 1&2 applied	8-c) Rule 3&4&5 applied	8-d) Rule 6&7 applied (Final output)

Figure 2.22 – The Sample Outputs of the Perceptual Grouping Operation

2.2.6 Forming the Polygons within the Fields

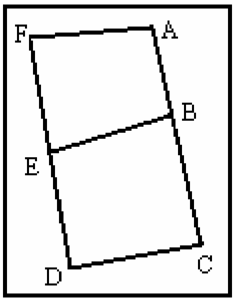
After applying the perceptual grouping rules, the connected line segments are found within the field. The vertex coordinates of each line segment and the connectivity relations between the line segments are determined. However, this information is not sufficient for determining the closed polygons within the fields. The connected line segments must be grouped together such that each group defines a disjoint polygon. This problem is known as “line to polygon conversion” and there are several different algorithms to create polygons from the connected lines.

Region growing technique can be used to identify the disjoint polygons in the image using a seed point and growing the pixels iteratively unless a line segment is reached. The image pixels falling in each polygon in the image can be identified using this method and the coordinates of the vertices of the polygons can be found by a further process. However, this is a raster based operation and therefore it must be applied on pixel domain.

One other technique for constructing the polygons is to find the shortest paths from a selected point of a line segment to itself. A cyclic path from a selected point to itself that consists of the connected line segments represents a closed area. If this operation is applied for each vertex of the line segments, then all the polygons within the field can be detected.

Another technique for constructing the polygons is using a chain tree of the line segments through the connectivity relations and trying to find the cyclic paths from a point back to itself in this tree. A cyclic path from a point to itself represents a closed region. In this tree structure, each node is a vertex of a line segment and this node has child nodes which can be directly reached from that vertex. Finding all the possible cyclic paths for a point means that, all the possible polygons having that point as a vertex are constructed. During the construction of the tree for one point as the root node, the cyclic paths for other points are also determined because all the possible paths are being traced. If there is no node which is not included in the tree, all the possible polygons are constructed in one tree at a time. If there exist some points, which have no possible connections with the root node, they will not exist in that tree. Then one of them is selected and another tree is constructed using that point as the root node. This process is applied until all the points of the line segments exist in these trees.

First, a hash table is built containing the direct connection points for each node. This is actually a different representation of the line segments. A set of connected line segment set is obtained as the final output of the perceptual grouping process. For a sample field, the set and the corresponding hash table representations of the line segments are illustrated in the Figure 2.23.



a) Perceptual Grouping Result

MS = { ([A-B], [B-C],
[C-D], [D-E],
[B-E], [E-F],
[F-A]) };

b) Set of the Connected Line Segments as Perceptual Grouping Output.

Points	Connections
A	B,F
B	A,E,C
C	B,D
D	E,C
E	B,D,F
F	A,E

c) Hash Table Representation Constructed From the Set.

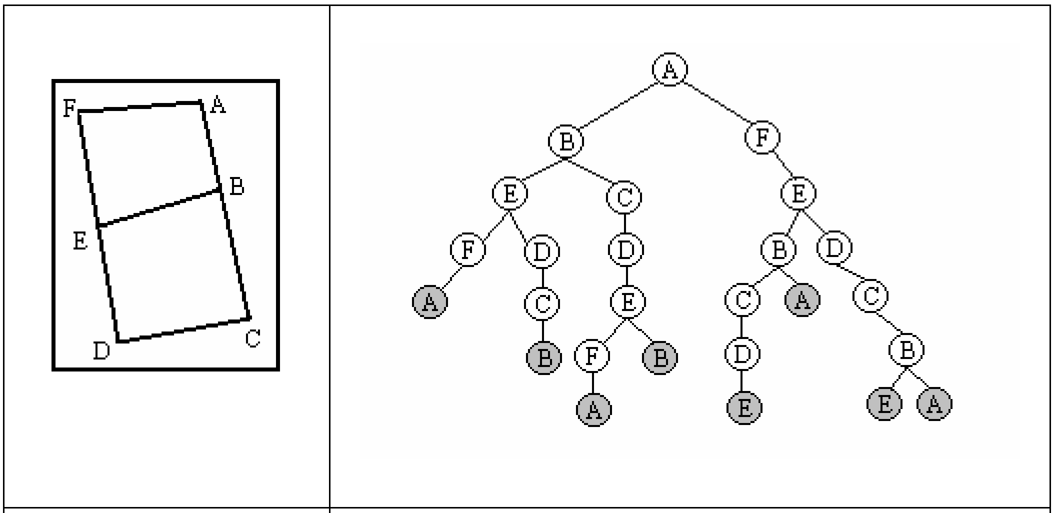
Figure 2.23 – The Hash Table for Node Connections

All the possible cyclic paths can be found by constructing a tree using a recursive algorithm for making the nodes and its children. The algorithm starts by selecting a point randomly from the hash table. Then, this point is passed to *MakeTree* function and all the nodes are recursively constructed from this root node. The pseudo code of the *MakeTree* function is as follows;

```
MakeTree(x) =
        Node=MakeNode(x)
        If Node Exists On Ancestors Stop.
        Else
        For Each y in [(Connections(x)-Parent(x)]
        MakeTree(y)
```

{Expression 2.8}

Figure 2.24 represents the tree constructed using the point "A" as the root node. *MakeTree* function is called with vertex and the tree representation is obtained.



a) Connected Line Segments

b) Tree Representation.

Figure 2.24 – All Cyclic Paths for a Node

As can be seen in Figure 2.24, the algorithm always stops to grow the tree deeper, whenever a node is reached which already exists in its ancestor nodes. The leaf nodes are the ones which have no children. And a cyclic path should exist when traversing from these leaves to the upper levels. For example, the bottom left leaf node is the point "A", which is also the root node. Traversing from this leaf node "A" to the upper node "A", a path is formed which is actually a polygon in the field. If we traverse to the upper levels from each leaf, all the possible sub-polygons are determined in the field.

In Figure 2.25, traversing from the leaf nodes to its ancestor nodes to find the cyclic path is illustrated. The arrows showing the traverse paths were drawn for three nodes only. But the same process is applied for each leaf node and all the paths can be traversed. Some of the paths may be the same having the same vertices, when all leaf nodes are traversed. Next, the redundant paths should be eliminated.

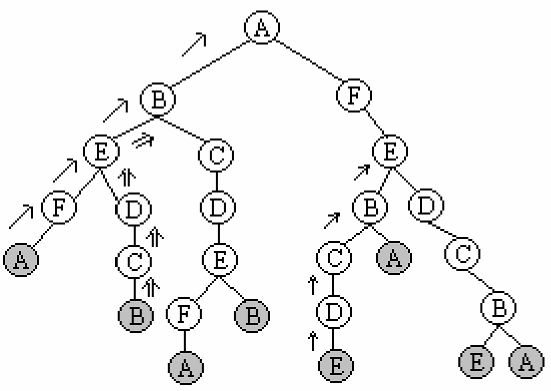
 A B F E C E F D D B D A C E C A C B F B D B A E E A	1) A-F-E-B-A 2) B-C-D-E-B 3) A-F-E-D-C-B-A 4) B-E-D-C-B (Same with 2) 5) E-D-C-B-E (Same with 2) 6) A-B-E-F-A (Same with 1) 7) E-B-C-D-E-F-A (Same With 3) 8) A-B-C-D-E-F-A (Same with 3)
a) Traverse From Leafs To Upper Nodes	b) All Cyclic Paths Found

Figure 2.25 – The Traverse of All Possible Polygons For a Node

After constructing the tree, all the leaves of the tree must be traversed to the upper nodes until the point in the leaf node is reached again in the ancestors. Figure 2.25-b has the list of all the paths traversed from each leaf node. As can be seen in the figure, only the first three cyclic paths represent the distinct polygons. The other cyclic paths are the same with one of these three polygons. The start and the end points or the order of the vertices may differ. However, they represent the same polygons. Figure 2.26 shows all the polygons after removing the redundant ones.

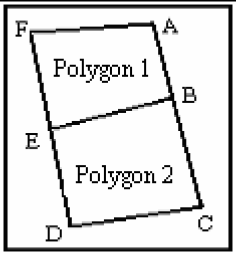
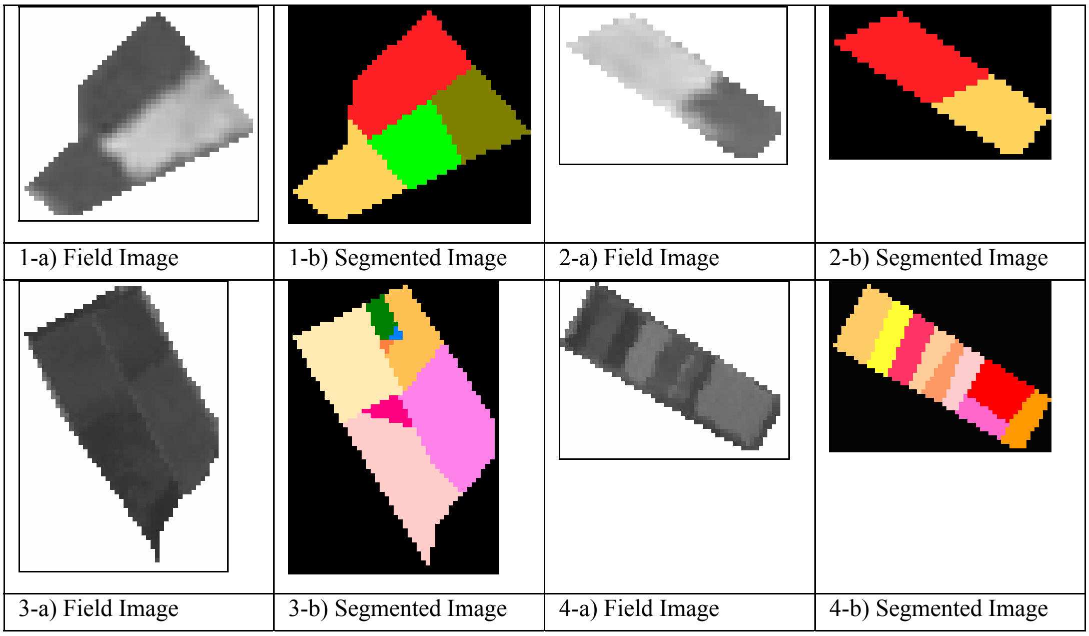
	Polygon 1: [A-B-E-F-A] Polygon 2: [B-C-D-E-B] Polygon 3: [A-F-E-D-C-B-A] *where Polygon 3 = (Polygon 1 + Polygon 2)*
a) Polygons In The Field	b) Vertices Of The Polygons

Figure 2.26 – The Polygons Determined For a Field

Since all the possible polygons present within the field are determined, there can be several polygons detected, that may contain two or more sub-polygons in it. Since it is necessary to detect distinct and non-overlapping polygons as the final segmentation output, the redundant polygons are needed to be deleted. In the above example given in Figure 2.26, Polygon-3 is the sum of the other two polygons. Therefore, this polygon should be removed from our output.

A simple algorithm can be used to remove the overlapping polygons. The algorithm has one simple rule: If a polygon contains a sub polygon, then it is removed. In the example given in Figure 2.26, Polygon-3 contains Polygon-1 and Polygon-2. Therefore Polygon-3 is removed from the output. Thus, the distinct Polygon-1 and Polygon-2 are taken as the final output of this process. Figure 2.27 illustrates the segmentation results of the fields after applying the polygon construction process.



1-a) Field Image
1-b) Segmented Image
2-a) Field Image
2-b) Segmented Image
3-a) Field Image
3-b) Segmented Image
4-a) Field Image
4-b) Segmented Image

Figure 2.27 – The Sample Outputs of the Polygon Construction Operation

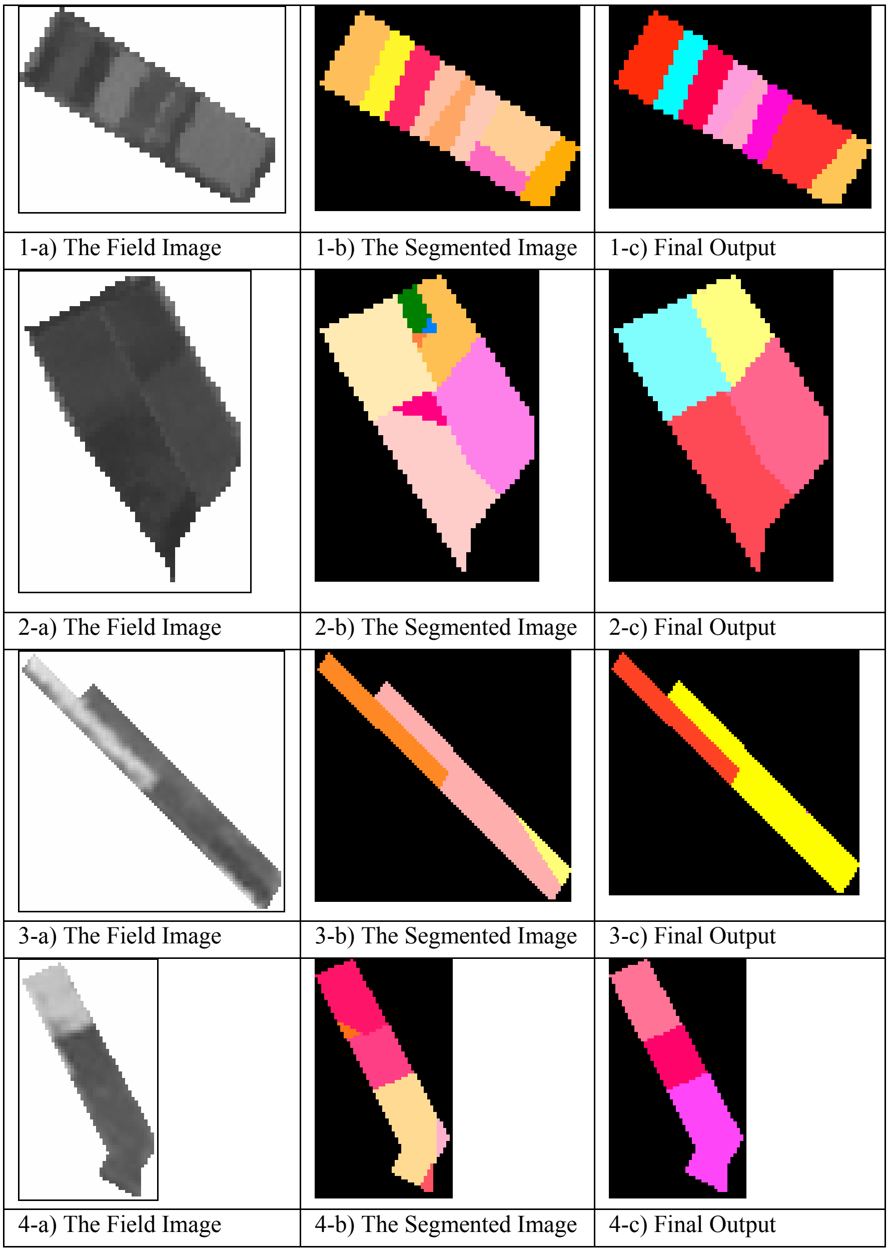
As can be seen in Figure 2.27, several small polygons may be generated as a result of over-segmentation. This is because of the noisy lines formed through the edge detection. These polygons are generally the noise in the output and are not actually the distinct segments containing a different land cover. Therefore, the small polygons must be removed through a further process. The two parameters that can be used as hints for detecting whether the polygon is a regular segment or not are; (i) the shape and (ii) the area of the polygon.

If a polygon does not satisfy the specified thresholds defined for the area and the shape, it is merged with one if its adjacent polygons. The detected segments actually represent homogenous areas that have distinct crop types. Therefore, these fields must not be too small and must not have extra-ordinary shapes. Small and triangular shaped fields should be merged with the adjacent polygons. Here there are two problems that are; (i) the determination of the threshold and (ii) the determination of the adequate adjacent polygon if a merge is decided to be performed. The choice of the threshold value is a trade off and it must be chosen as an optimum value. For example, the area threshold value should be big enough to enable the removal of the small noisy areas and should be small enough to prevent the removal of the real distinct crop fields.

To determine the shape of the field, the number of the vertices of a polygon and the angles between the vertices can be utilized. The triangular shaped fields may be merged to the adjacent polygons.

The selection of the adjacent polygons to which the small or irregular shaped polygons to be merged is another important issue. Different approaches exist for resolving the over-segmentation problem. In one approach, the spectral properties of the segments can be used to decide if the adjacent polygons should be merged or not. Depending on the decision, the two adjacent polygons that have similar reflectance characteristics can be merged. In another approach, the adjacent polygon that has the longest boundary with the considered polygon can be selected and the merging can be performed between these two polygons. This second approach is preferred in the present case. The first method seems more related with the classification of the fields and it is decided to be used in the future, if a classification methodology is integrated with this proposed segmentation process.

Figure 2.28 illustrates several outputs of the polygon construction phase. Each output contains the original field data, all the constructed polygons and the status of the polygons after the merging operation is performed.



1-a) The Field Image

1-b) The Segmented Image

1-c) Final Output

2-a) The Field Image

2-b) The Segmented Image

2-c) Final Output

3-a) The Field Image

3-b) The Segmented Image

3-c) Final Output

4-a) The Field Image

4-b) The Segmented Image

4-c) Final Output

Figure 2.28 – The Sample Outputs after the Merging Operation

The polygon construction was the last step of the field-based segmentation process. Thus, the disjoint fields representing the homogeneous sub-fields within the existing field boundaries were detected.

2.3 Software

The proposed field-based image segmentation process was automated by developing a software. The software is named Field-Based Image Segmentation Software (FBISS). A brief description of FBISS is given in this section with the snapshots of the user interfaces.

FBISS was developed using ANSI C++, except for the user interfaces that are implemented using Microsoft Foundation Classes (MFC) of Visual C++ 6.0. OpenCV Library [48] which is a powerful library for C++ was used for the basic image processing operations such as reading images from files and writing them back, edge detection, contour detection, etc. Visual SourceSafe was used as the source control tool. All the source code and the documentation were developed under configuration control.

2.3.1 The Capabilities of the Software

The software implements the whole segmentation process and includes several analysis functionalities. The following operations can be performed using the developed software:

- Open/Save/Save As/Print an Image (bmp, jpg, gif, tiff formats)
- Zoom In/Out, Fit to Window, Full Screen Display Capabilities
- Load Vector File (Formatted Text File)
- Determine Application and Segmentation Parameters
- Perform Segmentation
- Display the Results, and Internal Outputs for each Step
- Comparison Between Truth Segments and Results
- Generate Reports of Results (Formatted Text File)
- Merge of Segments or Parcels

Figure 2.29 is the snapshot of the main window of the FBISS. An image and the existing field boundaries of the agricultural fields are shown on the screen.

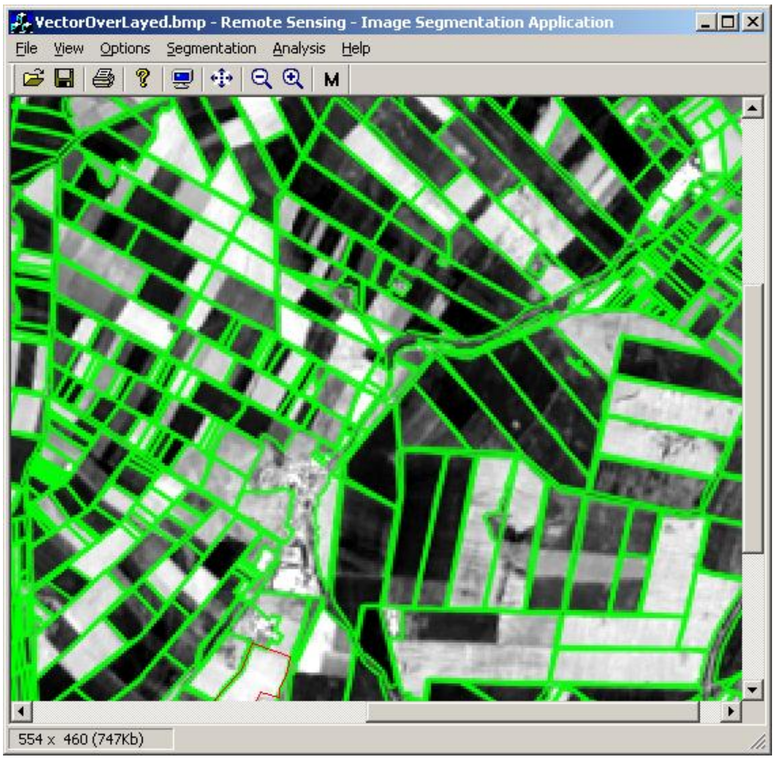
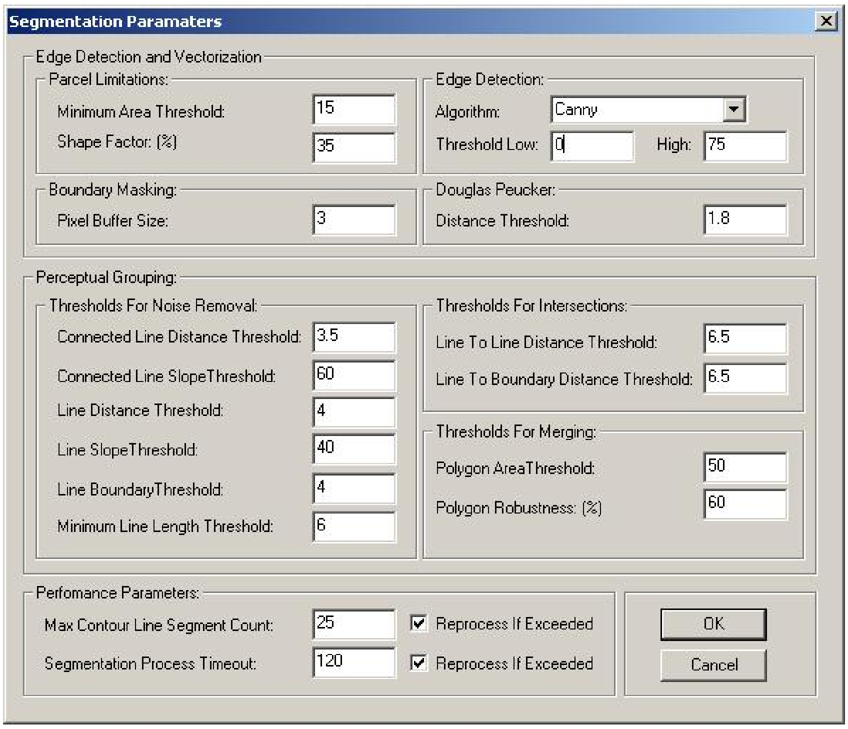

VectorOverLayed.bmp - Remote Sensing - Image Segmentation Application
File View Options Segmentation Analysis Help
M
554 x 460 (747Kb)

Figure 2.29 – The Main Window of FBISS

Figure 2.30 is the snapshot of the window, on which the user can set the application parameters to be used in the segmentation. The threshold values used in each step is adaptable and can be changed by the user using this window. In addition, the user can define several constraints such as the process timeout limits and the maximum segment count limits in order to utilize the performance of the segmentation.



Segmentation Paramaters

Edge Detection and Vectorization

Parcel Limitations:

Minimum Area Threshold: 15

Shape Factor: (%) 35

Edge Detection:

Algorithm: Canny

Threshold Low: 0 High: 75

Boundary Masking:

Pixel Buffer Size: 3

Douglas Peucker:

Distance Threshold: 1.8

Perceptual Grouping:

Thresholds For Noise Removal:

Connected Line Distance Threshold: 3.5

Connected Line SlopeThreshold: 60

Line Distance Threshold: 4

Line SlopeThreshold: 40

Line BoundaryThreshold: 4

Minimum Line Length Threshold: 6

Thresholds For Intersections:

Line To Line Distance Threshold: 6.5

Line To Boundary Distance Threshold: 6.5

Thresholds For Merging:

Polygon AreaThreshold: 50

Polygon Robustness: (%) 60

Perfomance Parameters:

Max Contour Line Segment Count: 25 ☑ Reprocess If Exceeded

Segmentation Process Timeout: 120 ☑ Reprocess If Exceeded

Figure 2.30 – The Application Parameters Window of FBISS

In order to geographically reference the bounded area, the user must enter the corner coordinates of the image. The ellipsoidal geographic coordinates are entered as easting and northing on a selected UTM zone using the window in Figure 2.31

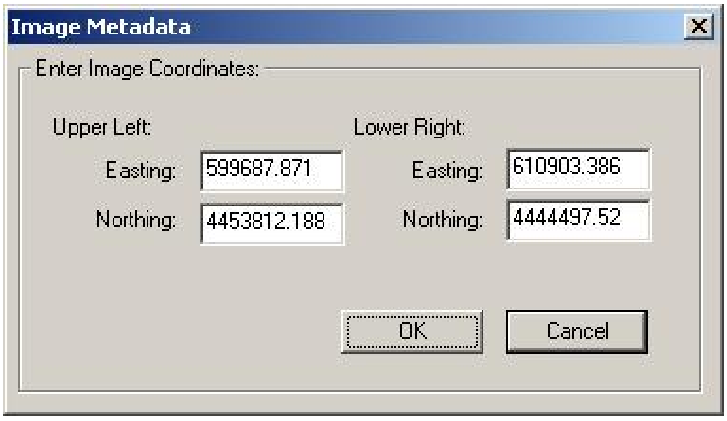


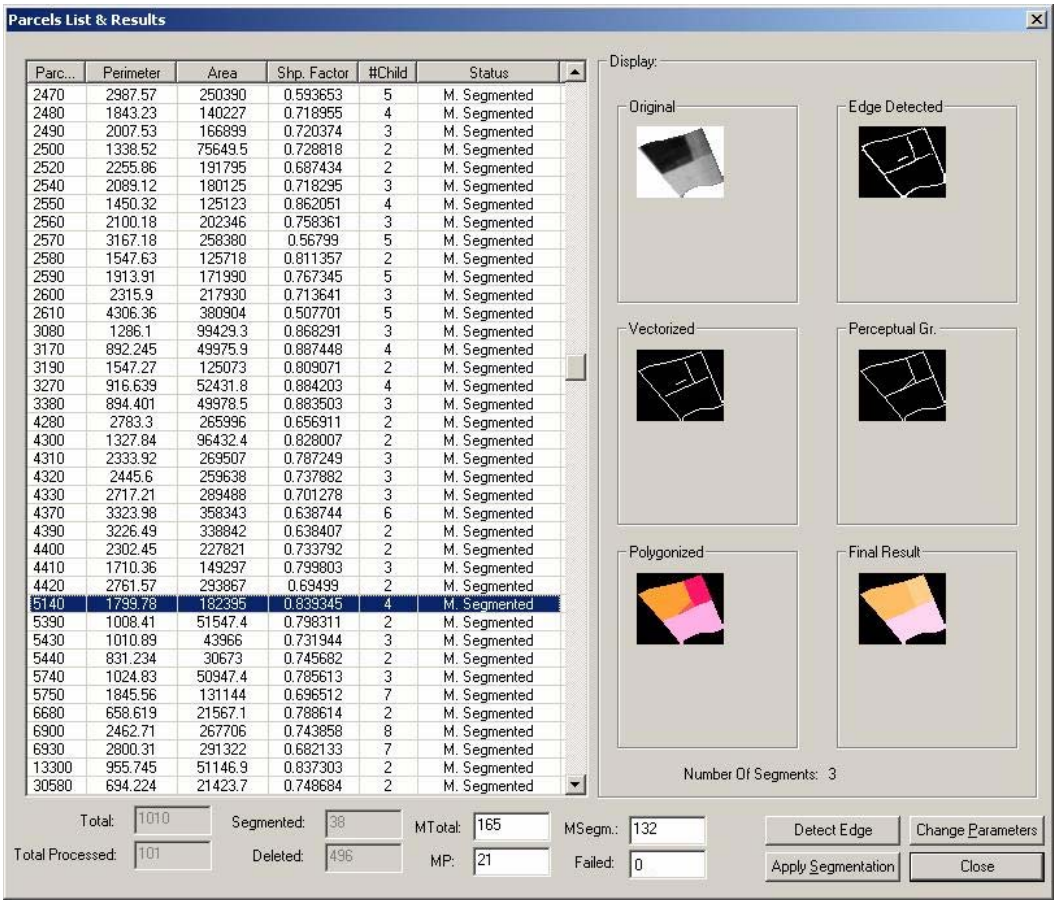
Image Metadata

Enter Image Coordinates:

Upper Left:		Lower Right:	
Easting:	599687.871	Easting:	610903.386
Northing:	4453812.188	Northing:	4444497.52

Figure 2.31- The Image Metadata Window of FBISS

All the parcels, the results and the intermediate outputs of the segmentation procedure can be displayed in a window as sub-images (Figure 2.32). In addition, for each parcel the area, the perimeter, the shape factor and number of sub-fields is displayed as a list in the same window. Further, a statistical information is provided such as the total number of the parcels, the number of the total processed parcels, the number of the parcels under the area threshold, etc. The user can also adapt the segmentation parameters and re-apply the segmentation process with the new parameters for that field considered using the related buttons.



Parc...	Perimeter	Area	Shp. Factor	#Child	Status
2470	2987.57	250390	0.593653	5	M. Segmented
2480	1843.23	140227	0.718955	4	M. Segmented
2490	2007.53	166899	0.720374	3	M. Segmented
2500	1338.52	75649.5	0.728818	2	M. Segmented
2520	2255.86	191795	0.687434	2	M. Segmented
2540	2089.12	180125	0.718295	3	M. Segmented
2550	1450.32	125123	0.862051	4	M. Segmented
2560	2100.18	202346	0.758361	3	M. Segmented
2570	3167.18	258380	0.56799	5	M. Segmented
2580	1547.63	125718	0.811357	2	M. Segmented
2590	1913.91	171990	0.767345	5	M. Segmented
2600	2315.9	217930	0.713641	3	M. Segmented
2610	4306.36	380904	0.507701	5	M. Segmented
3080	1286.1	99429.3	0.868291	3	M. Segmented
3170	892.245	49975.9	0.887448	4	M. Segmented
3190	1547.27	125073	0.809071	2	M. Segmented
3270	916.639	52431.8	0.884203	4	M. Segmented
3380	894.401	49978.5	0.883503	3	M. Segmented
4280	2783.3	265996	0.656911	2	M. Segmented
4300	1327.84	96432.4	0.828007	2	M. Segmented
4310	2333.92	269507	0.787249	3	M. Segmented
4320	2445.6	259638	0.737882	3	M. Segmented
4330	2717.21	289488	0.701278	3	M. Segmented
4370	3323.98	358343	0.638744	6	M. Segmented
4390	3226.49	338842	0.638407	2	M. Segmented
4400	2302.45	227821	0.733792	2	M. Segmented
4410	1710.36	149297	0.799803	3	M. Segmented
4420	2761.57	293867	0.69499	2	M. Segmented
5140	1799.78	182395	0.839345	4	M. Segmented
5390	1008.41	51547.4	0.798311	2	M. Segmented
5430	1010.89	43966	0.731944	3	M. Segmented
5440	831.234	30673	0.745682	2	M. Segmented
5740	1024.83	50947.4	0.785613	3	M. Segmented
5750	1845.56	131144	0.696512	7	M. Segmented
6680	658.619	21567.1	0.788614	2	M. Segmented
6900	2462.71	267706	0.743858	8	M. Segmented
6930	2800.31	291322	0.682133	7	M. Segmented
13300	955.745	51146.9	0.837303	2	M. Segmented
30580	694.224	21423.7	0.748684	2	M. Segmented

Number Of Segments: 3

Total: 1010
Segmented: 38
MTotal: 165
MSegm.: 132
Total Processed: 101
Deleted: 496
MP: 21
Failed: 0

Figure 2.32- The Parcel List Window of FBISS

Figure 2.33 illustrates the right click menu functionality of FBISS. If the user selects a parcel by just clicking on a region inside the field, he/she can either see the detailed information of the parcel or perform the segmentation on the parcel. If the user selects two fields, these fields can be merged using the merge option of the menu.

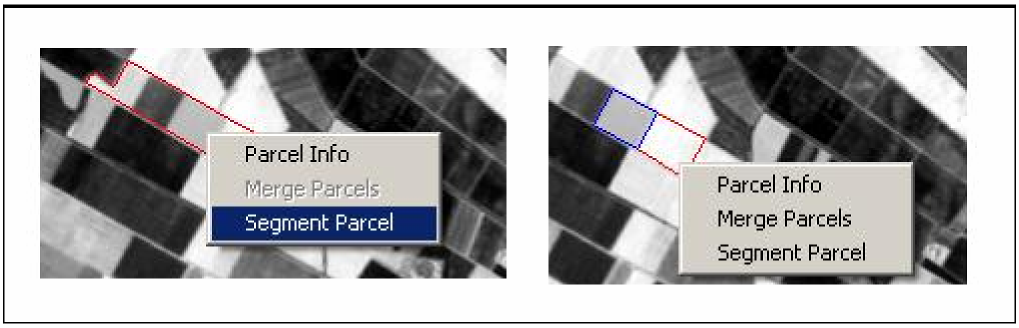
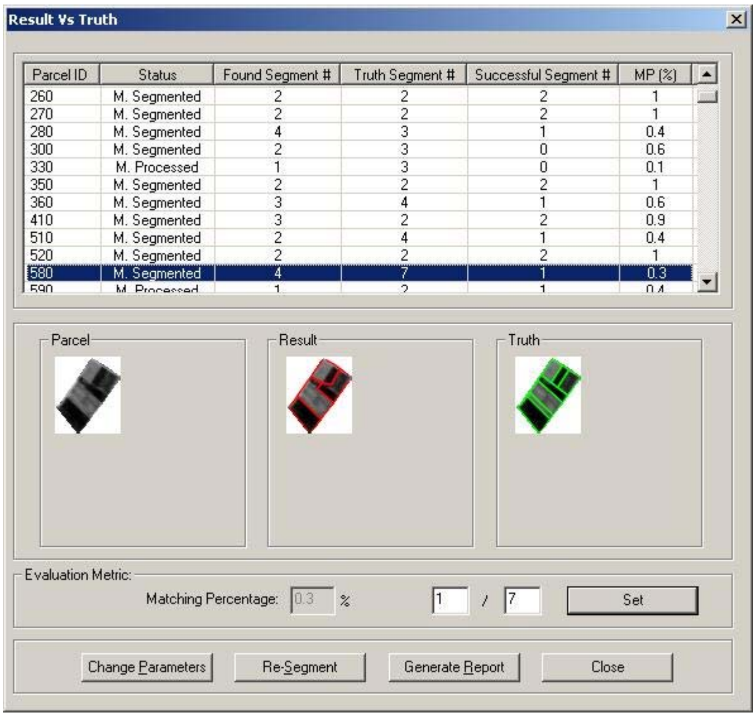


Figure 2.33- The Right Click Menu Snapshot of FBISS

FBISS has the capability of displaying both the results of the automated segmentation and the truth segments on the same window (Figure 2.33). As well, the success of the segmentation on each field is calculated and displayed. In addition, the number of successfully detected sub-field numbers can be calculated and displayed in the user-interface. The computation of the accuracy and the successfully detected sub-fields are explained in the following section. In this window, the user can access to the application parameters and apply the segmentation again on a parcel using the adapted parameters.



Parcel ID	Status	Found Segment #	Truth Segment #	Successful Segment #	MP (%)
260	M. Segmented	2	2	2	1
270	M. Segmented	2	2	2	1
280	M. Segmented	4	3	1	0.4
300	M. Segmented	2	3	0	0.6
330	M. Processed	1	3	0	0.1
350	M. Segmented	2	2	2	1
360	M. Segmented	3	4	1	0.6
410	M. Segmented	3	2	2	0.9
510	M. Segmented	2	4	1	0.4
520	M. Segmented	2	2	2	1
580	M. Segmented	4	7	1	0.3
590	M. Processed	1	2	1	0.4

Evaluation Metric:
Matching Percentage: 0.3 %   1 / 7

Figure 2.34- The Analysis of the Results Window of FBISS

2.3.2 Software Design

The software implementation was based on an object-oriented design, which is a common approach for the C++ applications. The different functionalities were implemented in the separate classes, considering the coupling and complex interface issues. Figure 2.35 represents the class diagram of the software. As can be seen in the figure, only one class *cImageProcessor* has an interface with OpenCV library, and all the raster based functions are implemented in this class. The lines and polygons obtained during the segmentation process are handled by *cContor* and *cPolygonManager* classes respectively. *cVectorReader* is the class responsible for reading and storing the vector data, as well as implementing the search/merge/split operations between the parcels. *cMathHelper* class is the mathematics functionality toolbox for the software that uses a *cMatrix* class which implements the matrix operations.

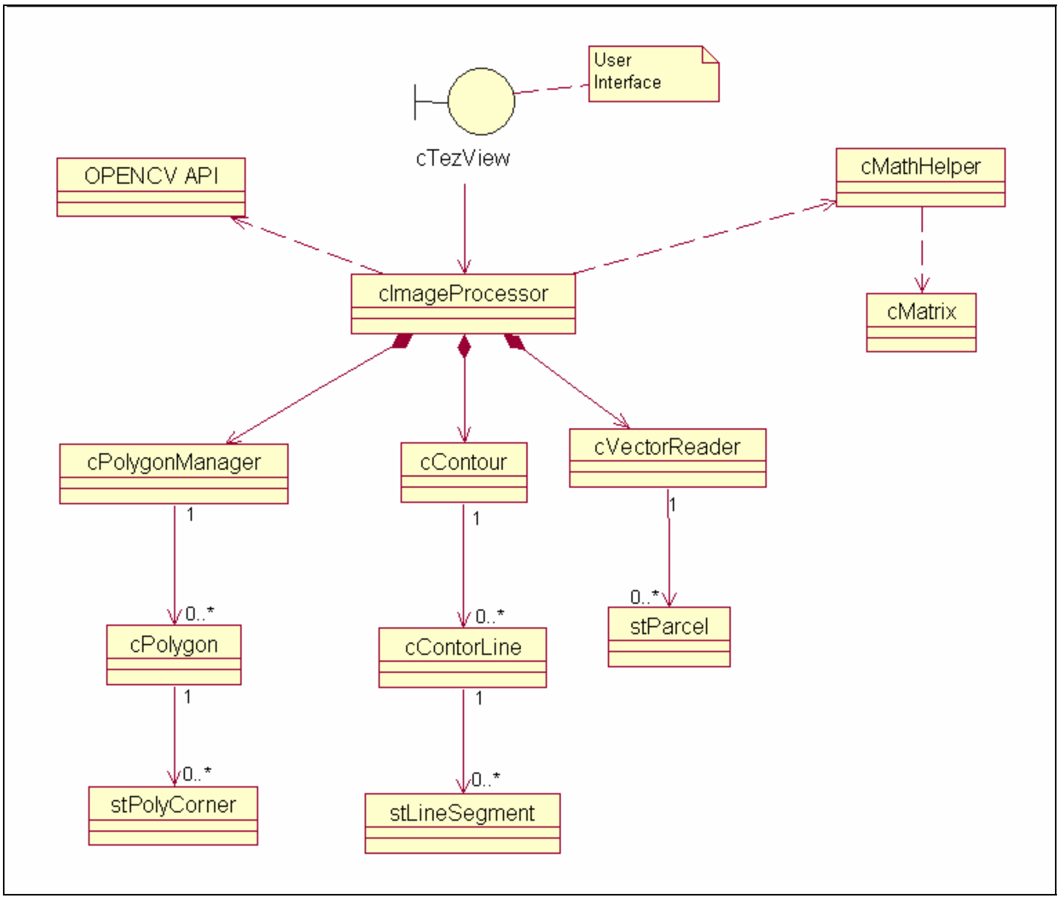


Figure 2.35- The Class Diagram for FBISS Design

CHAPTER III

THE IMPLEMENTATION

In this chapter, the implementation of the proposed segmentation procedure on a selected area is described. The SPOT 4 and SPOT 5 satellite images covering the agriculture fields and the existing field boundary information were processed and the results were obtained using the developed image segmentation software. The details of the input raster and the vector data, the evaluation methodology and the results of the analysis are presented.

3.1 Input Data

The Karacabey Plain, which consists of the agricultural fields of various sizes, was selected as the study area to implement the segmentation process. The Karacabey Plain is situated in Marmara Region of Turkey, near Bursa (Figure 3.1). It is one of the most important agricultural areas in Turkey. The area is characterized by rich, loamy soils which, in addition to the excellent weather conditions, make agriculture the main land use in the region. The agricultural land is predominantly used for the cultivation of arable crops that are wheat, corn, tomato, sugar beet, rice, pepper, and watermelon as well as other crops of secondary importance that are pea, onion and cauliflower. In addition, there are several pasture and clover fields and small townships are scattered throughout the area. In the region, the majority of the fields have regular shapes due to a land consolidation project performed between 1988 and 1992. The elevation difference across the study area is very small within 10 m. [28]. A number of small fields exist in the study area. However, since the small fields fall below the predefined threshold, they were not included in the segmentation process.

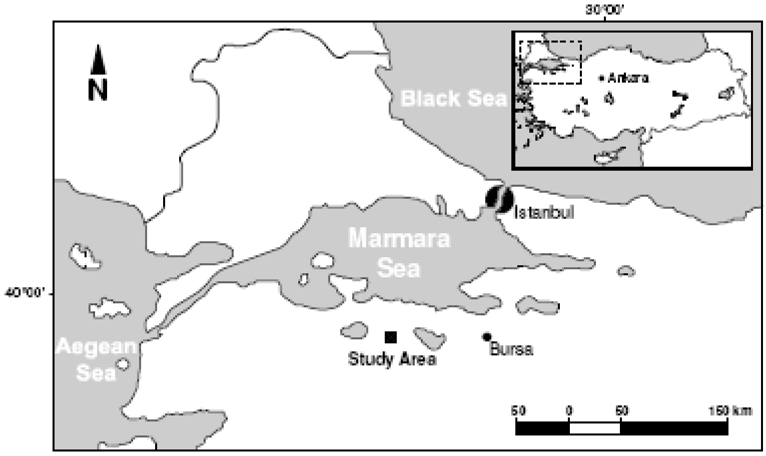


Figure 3.1 – The Location of Study Area in Turkey [28]

An area of interest having the size of 4600×7200 m. was determined from the Karacabey Plain. The ellipsoidal geographic coordinates of the selected area are as follows (UTM Zone-35): Upper-Left Corner (Easting: 605,297, Northing: 4,452,014) and Lower-Right Corner (Easting: 609,903 Northing: 4,444,780)

The proposed segmentation procedure is based on the integrated analysis of the raster image and existing field boundary information. Therefore, the implementation requires two input data sets;

- raster image, and
- vector data for existing field boundaries

3.1.1 Raster Data

The SPOT 5 and SPOT 4 satellite images covering the study area were used as the raster data in this study. The use of SPOT images was due to the following reasons;

- the availability,
- having two different spatial resolution images,
- the ease of use and
- the quality of the images.

Both images were purchased to be used for an ongoing research study conducted by Ozdarici [6] in the department of Geodetic and Geographic Information Technologies of METU. The availability of the two different resolution SPOT images of the same area was an advantage to better evaluate the proposed segmentation process. The technical characteristics and the spectral and geographic quality of the SPOT images were also additional factors for choosing the SPOT data as the raster input.

The images acquired by the SPOT earth observation satellites are an unparalleled source of information for studying, monitoring, forecasting and managing natural resources and human activities. The SPOT images can be directly integrated in image viewing, image processing, and geographic information or map-making systems. SPOT data also stand as one of the key components for large-scale agricultural projects. The global coverage and the revisit capability of the SPOT satellites give agronomists, decision-makers and other stakeholders in the farming sector the current information they need to support the decisions. The resolution of SPOT imagery enables very precise identification of the crops and the other land use classes at the parcel level [49].

The SPOT satellites operate from a sun-synchronous, near polar orbit 832 km above the earth. The satellites are inclined 98.7 degrees, cross the equator and have an orbital period of 26 days. The ground imaging swath is 60 km per High Resolution Visible (HRV) sensor. With both HRVs scanning in the twin vertical viewing mode the cross-track swath is 117 km. Each HRV sensor has the capability to scan 27 degrees off-nadir allowing for repeat coverage of an area up to every three or four days depending upon the latitude of the area. This cross-track pointing capability allows the same point on the earth to be viewed from several different orbits and enables the acquisition of stereoscopic imagery [49].

For SPOT 1, 2 and 3, each satellite carries two HRVs with the capability of scanning in either a multispectral mode or a panchromatic mode. For SPOT 4 and SPOT 5, the multispectral mode was improved to provide 4 spectral bands.

Table 3.1 – The Spatial and Spectral Properties of the SPOT Satellites

SPOT satellite	**Spectral Bands**	**Ground Pixel Size**	**Spectral Range**
SPOT 5	Panchromatic	2.5 or 5 metres	0.48 - 0.71 µm
	B1: green	10 meters	0.50 - 0.59 µm
	B2: red	10 meters	0.61 - 0.68 µm
	B3: near infrared	10 meters	0.78 - 0.89 µm
	B4: short-wave infrared (SWIR)	20 meters	1.58 - 1.75 µm
SPOT 4	Monospectral – Panchromatic	10 meters	0.61 - 0.68 µm
	B1: green	10 meters	0.50 - 0.59 µm
	B2: red	10 meters	0.61 - 0.68 µm
	B3: near infrared	10 meters	0.78 - 0.89 µm
	B4: short-wave infrared (SWIR)	20 meters (*)	1.58 - 1.75 µm
SPOT1 SPOT2 SPOT3	Panchromatic	10 meters	0.50 – 0.73 µm
	B1: green	20 meters	0.50 - 0.59 µm
	B2: red	20 meters	0.61 - 0.68 µm
	B3: near infrared	20 meters	0.78 - 0.89 µm

(*) SPOT 5 SWIR band 20 meters resolution images are resampled to 10 meters resolution by a further process.

3.1.1.1 The SPOT 5 Image

The selected study area is covered by the 460×720 pixel sized SPOT 5 satellite image. The SPOT 5 data contains four spectral bands. However, the edge detection algorithm (Canny) can only be applied on one band at a time. Therefore, single band images were generated using the combinations of the four spectral bands of SPOT 5. The following two types of images of the spectral bands were used;

- the first component of the Principle Component Analysis (PCA) output,
- the intensity image of the first three bands ([Band1+Band2+Band3] / 3)

The principal component analysis involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The covariance of the first principal component accounts for the highest variability in the data, and each succeeding component accounts for the less variability. Mainly the objectives of PCA are;

- to reduce the dimensionality of the data set without much loss of information and
- to identify new meaningful underlying variables.

Table 3.2 represents the spectral variability of each band of the SPOT 5 data and the PCA output obtained from these bands. The spectral variability of each band is given with the mean and standard deviation values.

Table 3.2 – The PCA Output for the SPOT 5 Data

<table>
<tr><th colspan="4">SPOT 5 Bands</th></tr>
<tr><td>Channel</td><td>Mean</td><td colspan="2">Std. Dev.</td></tr>
<tr><td>1</td><td>130.034</td><td colspan="2">39.060</td></tr>
<tr><td>2</td><td>67.835</td><td colspan="2">21.657</td></tr>
<tr><td>3</td><td>70.146</td><td colspan="2">11.441</td></tr>
<tr><td>4</td><td>82.072</td><td colspan="2">18.518</td></tr>
<tr><th colspan="4">PCA Output</th></tr>
<tr><td>EigenChannel</td><td>EigenValue</td><td>Std. Dev.</td><td>%Variance</td></tr>
<tr><td>1</td><td>1836.711</td><td>42.857</td><td>74.41%</td></tr>
<tr><td>2</td><td>592.737</td><td>24.346</td><td>24.01%</td></tr>
<tr><td>3</td><td>34.531</td><td>5.876</td><td>1.40%</td></tr>
<tr><td>4</td><td>4.488</td><td>2.119</td><td>0.18%</td></tr>
</table>

The first component of the PCA output contains most of the original data of the all spectral bands (74.41%). Therefore, it is adequate to use the first component for the segmentation process. The first component of the PCA generated from SPOT 5 bands with the existing field boundaries overlaid is illustrated in Figure 3.2. Before starting the segmentation process a linear contrast enhancement was applied to the images.

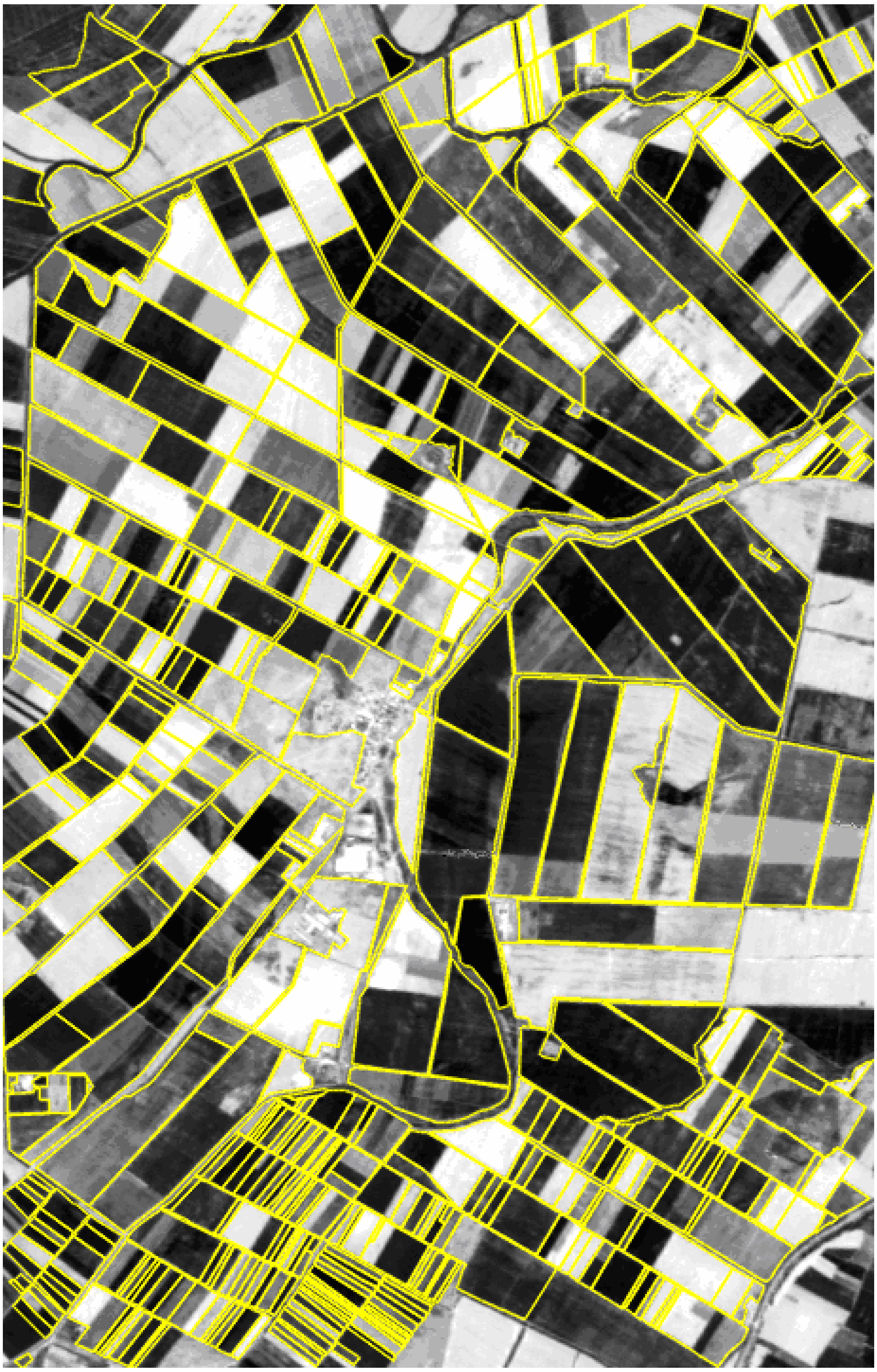


Figure 3.2 – The First Principle Component of SPOT 5 Image

This image was named as **SPOT_5_PCA** and it is referenced with this name in the following sections of the thesis. The general spectral properties of SPOT_5_PCA are as follows:

- the mean value: 121.6,
- the standard deviation: 78.9, and
- the median value: 106.0.

The other image generated from the SPOT 5 spectral bands was the intensity image computed using Band 1 (green), Band 2 (red) and Band 3 (near infrared). Since a single band image is needed to be used in the segmentation process, the arithmetic average of the DN values of these three bands were calculated and an output image was constructed after applying a linear contrast enhancement. The intensity image generated from the bands 1, 2, and 3 with the existing field boundaries overlaid is illustrated in Figure 3.3.

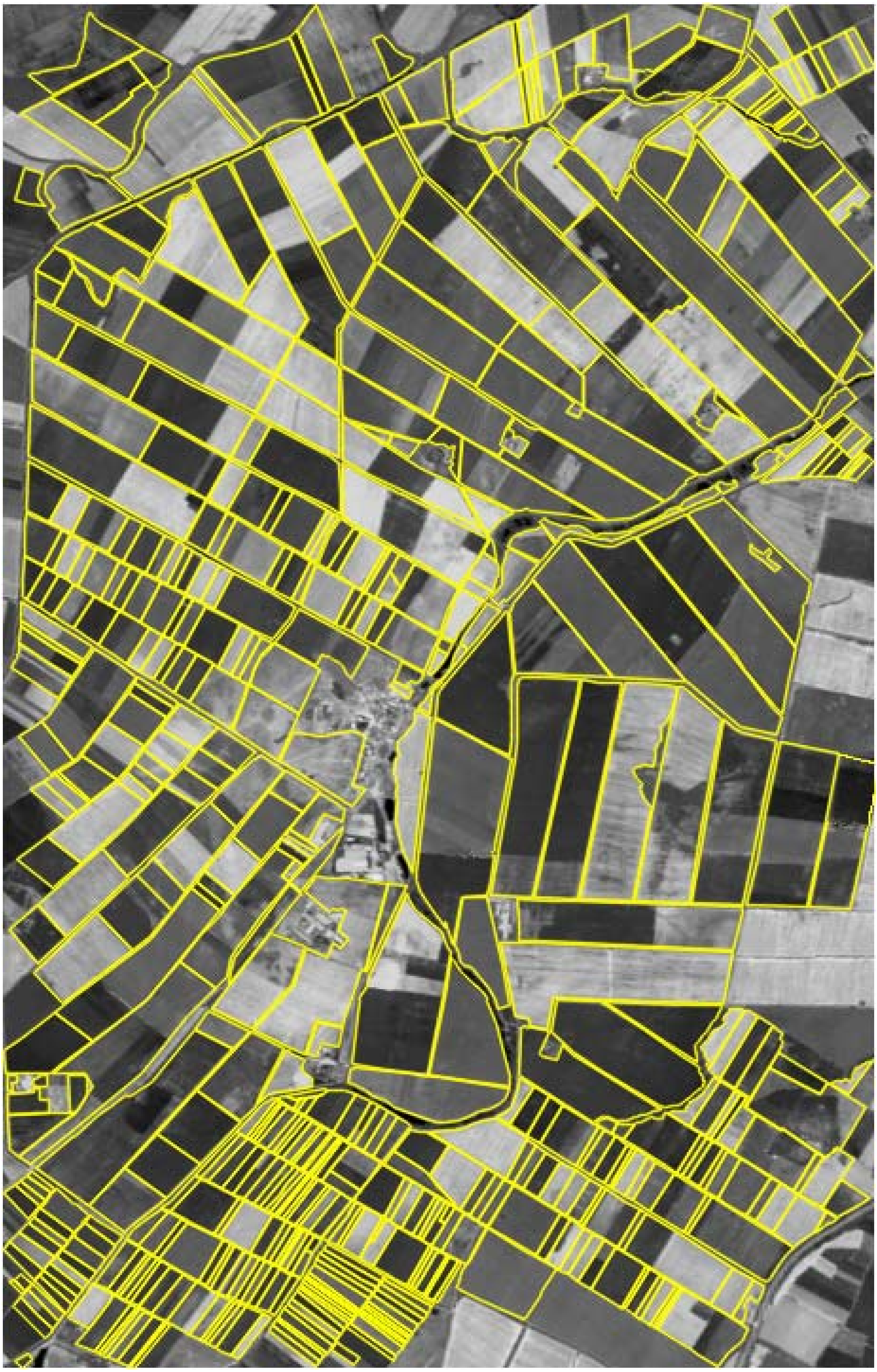


Figure 3.3 – The Intensity Image of SPOT 5

This image was named as **SPOT_5_Intensity** and it is referenced with this name in the following sections of the thesis. The general spectral properties of SPOT_5_Intensity are as follows:

- the mean vaue: 101.7,
- the standard deviation: 58.8, and
- the median value: 85.0.

The SPOT_5_Intensity image appears to be darker when compared to the SPOT_5_PCA image. This difference can also be seen by comparing the mean and the median values (PCA vs. Intensity - Mean: 121.6/101.7 and median: 106.0/85.0). On the other hand, the SPOT_5_Intensity image appears to be smoother than the SPOT_5_PCA image. The comparison of the standard deviations supports the fact that there is less spectral variability within the SPOT_5_Intensity image (PCA vs. Intensity - Std. Dev: 78.9/58.8). These spectral differences affect the segmentation results. These effects will be discussed in the results and analysis section.

3.1.1.2 The SPOT 4 Images

SPOT4 was the other image used to implement the proposed image segmentation procedure. The study area is covered by 230×360 pixel sized SPOT 4 satellite image. This image size is one fourth of the SPOT 5 image and contains less detail due to its lower spatial resolution (20 m). The SPOT 4 data also contain four bands just like the SPOT 5 images. Similarly, two types of single band images were generated using the same band combinations and the enhancement techniques as performed for the SPOT 5 image. The derived images are namely:

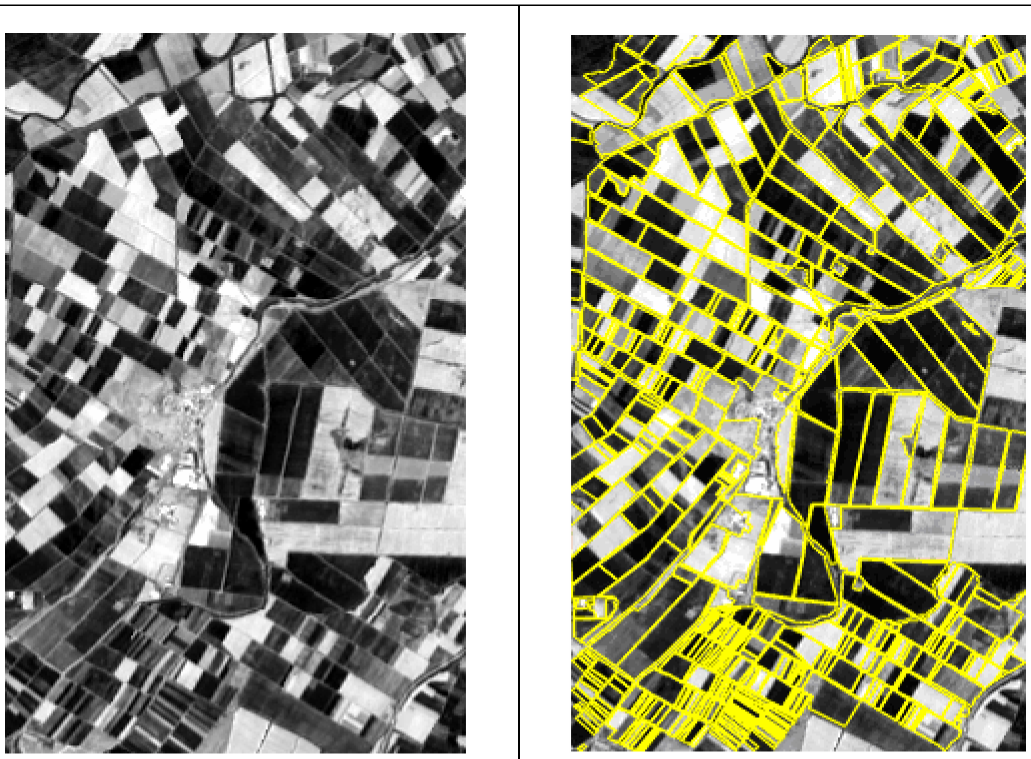
- the first component of the PCA output, and
- the intensity image of the first three bands ([Band1+Band2+Band3] / 3)

The principle component analysis was therefore applied to the SPOT 4 bands, and the output given in Table 3.3 was obtained. Table 3.3 also contains the statistical values (mean, and standard deviation) for each band of the SPOT 4 image.

Table 3.3 – The PCA Output for the SPOT 4 Data

<table>
<tr><th colspan="4">SPOT 4 Bands</th></tr>
<tr><td>Channel</td><td>Mean</td><td colspan="2">Std. Dev.</td></tr>
<tr><td>1</td><td>78.906</td><td colspan="2">18.152</td></tr>
<tr><td>2</td><td>86.937</td><td colspan="2">39.722</td></tr>
<tr><td>3</td><td>61.135</td><td colspan="2">27.539</td></tr>
<tr><td>4</td><td>56.874</td><td colspan="2">15.256</td></tr>
<tr><th colspan="4">PCA Output</th></tr>
<tr><td>EigenChannel</td><td>EigenValue</td><td>Std. Dev.</td><td>%Variance</td></tr>
<tr><td>1</td><td>23277.318</td><td>48.242</td><td>80.29%</td></tr>
<tr><td>2</td><td>529.876</td><td>23.010</td><td>18.28%</td></tr>
<tr><td>3</td><td>28.143</td><td>5.305</td><td>0.97%</td></tr>
<tr><td>4</td><td>13.1678</td><td>3.6287</td><td>0.46%</td></tr>
</table>

For the SPOT 4 PCA output, the first principle component contains 80.29% variability of the four spectral bands. Thus, the first component was used as the single band image to be processed for the segmentation after applying a linear contrast enhancement. The generated image from the first component of PCA of the SPOT 4 data is illustrated in Figure 3.4-a, and the overlay of the existing field boundaries is illustrated in Figure 3.4-b.



a) SPOT_4_PCA Image

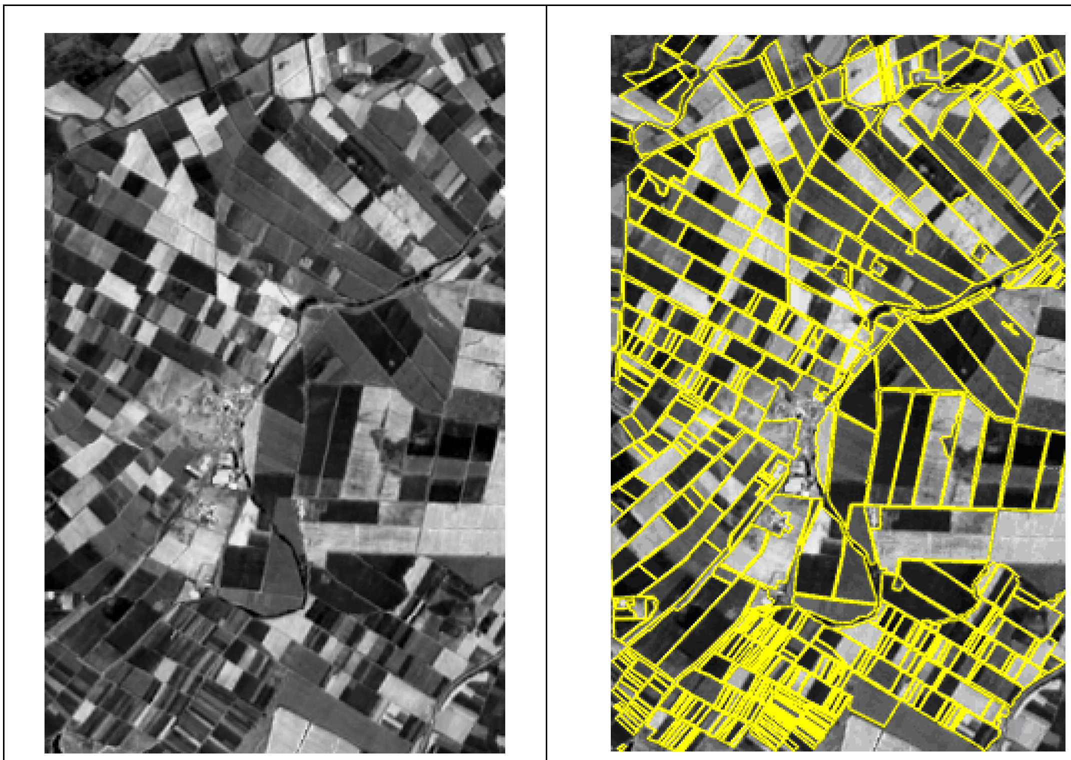
b) Existing Field Boundaries Overlaid

Figure 3.4 – The SPOT_4_PCA Image and the Overlaid Vector Data

This image was named **SPOT_4_PCA** and it is referenced with this name in the following sections of the thesis. The general spectral characteristics of the SPOT_4_PCA image are similar to the SPOT_5_PCA image. The statistical properties for SPOT_4_PCA can be listed as follows:

- the mean value: 111.3,
- the standard deviation: 77.2, and
- the median value: 92.0.

The other image generated from the spectral bands of SPOT 4 was the intensity image computed using Band 1 (green), Band 2 (red) and Band 3 (near infrared). Same as the SPOT_5_Intensity image, the arithmetic average of the DN values of the three bands were calculated and an output image was generated after applying a linear contrast enhancement. The generated image (Figure 3.5-a) and the overlay of the existing field boundaries on the generated image (Figure 3.5-b) is illustrated in the following figure.



a) SPOT_4_Intensity Image

b) Existing Field Boundaries Overlaid

Figure 3.5 – The SPOT_4_Intensity Image and the Overlaid Vector Data

This image was named **SPOT_4_Intensity** and it is referenced with this name in the following sections of the thesis. The general spectral characteristics of the SPOT_4_Intensity image that are similar to the SPOT_5_Intensity image can be listed as follows;

- the mean value: 95.5,
- the standard deviation: 57.3, and
- the median value: 81.0.

Similar to the SPOT 5 outputs, the intensity image of SPOT 4 appears to be darker compared to the PCA image. This difference can also be seen by comparing the mean and the median values of both images (PCA vs. Intensity - Mean: 111.3/95.5 and median: 92.0/81.0). The SPOT_4_Intensity image was smoother than the SPOT_4_PCA image. The comparison of the standard deviation indicates that the SPOT_4_Intensity image has less spectral variability than the SPOT_4_PCA image (PCA vs. Intensity - Std. Dev.: 78.9/58.8). These spectral differences affect the segmentation results, and these effects will be discussed in the results and the analysis section.

Since the resolution of the SPOT 4 images is coarser than the resolution of the SPOT_5 images, therefore SPOT 4 images contain less detail. This fact may become an important factor and would have a significant effect on the results of the segmentation. The effects of the resolution difference between SPOT 4 and SPOT 5 images and the spectral differences between the PCA and the Intensity images on the segmentation outputs are discussed in the results and analysis section.

3.1.2 Vector Data

The existing field boundaries were available in digital form. The field boundaries were prepared by digitizing them from the cadastral map sheets [28]. The field boundaries were in UTM projection system. A formatted text file that includes the ID, area, perimeter and the boundary coordinates information for each field was generated from the GIS database. Each field was uniquely described with an identification code (ID) in the attributes table. The developed software uses this formatted text file to access the boundary data, by parsing it at run time. The following is the format of the vector data for one field:

```
Shape Id:11     Vertices:7
        entity:   197
        area:   821916
   perimete:   40034.9
            id:   1415
… //Some additional attributes, not used for segmentation

(     603022.632 E,     4452487.930 N,          0.000 Z)
(     602858.572 E,     4452512.014 N,          0.000 Z)
(     602856.799 E,     4452715.201 N,          0.000 Z)
(     603019.848 E,     4452769.027 N,          0.000 Z)
(     603023.246 E,     4452623.595 N,          0.000 Z)
(     603024.605 E,     4452525.055 N,          0.000 Z)
(     603022.632 E,     4452487.930 N,          0.000 Z)
```

Expression 3.1

In the above expression, a field (ID 1451) is described by the area, perimeter, and other attributes as well as easting and northing coordinates of the vertices of the boundary.

There were a total of 514 fields in the vector data for the selected study area. Table 3-4 represents several descriptive details for the field geometries. The mean area of the fields is 52,570 m^2, and the standard deviation value is 71.94 which is rather large. The large standard deviation means that the areas of the fields vary greatly and there exist many small and big sized fields in the study area. The distribution of the field sizes in the study area is given as a graph in Figure 3.6.

Table 3-4 – The Statistical Information for the Fields in Vector Data

# Fields		514
Area Mean (m^2)		52,570
Area Std. Dev		71.94
Perimeter Mean (m)		493.96
Perimeter Std. Dev		487.14
# Fields Having	Area <15,000 m^2	222
	15< Area <100,000 m^2	206
	Area > 100,000 m^2	86

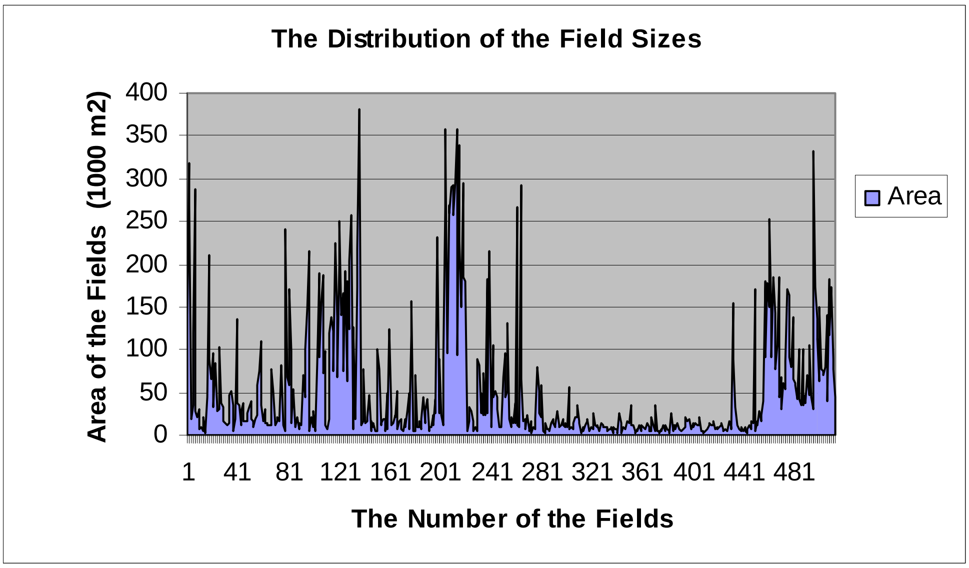


Figure 3.6 – The Distribution of the Sizes of the Fields

The display of the field boundaries overlaid on raster images to be used in the segmentation procedure are illustrated in Figure 3.2, 3.3, 3.4, 3.5.

3.2 The Methodology for Evaluation of the Results

The results were assessed by comparing the outputs with the truth segmentation information. The truth information contains the boundaries that include the boundaries of the manually segmented sub-fields. The missing boundaries of the sub-fields were manually delineated using the PCI Geomatica software for another study currently being conducted in the department [6]. The coordinates of the manually segmented sub-fields were exported to a text file in the same format (Expression 3.1) of the original vector data.

An identification convention was used for the manually segmented fields, which preserves the parent field identification relation for the segmented sub-fields. For each field, the ID information for each field was modified by multiplying it by 10. And if any segmentation was present, the segments belonging to the field was identified with the number Parent_Id+1, Parent_Id+2, Parent_Id+3, etc., and the parent field was removed from the vector data. Figure 3.7 illustrates the identification procedure for a segmented field.

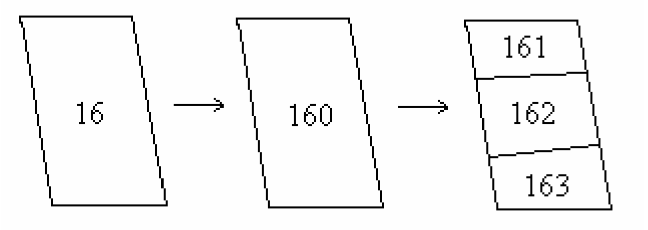
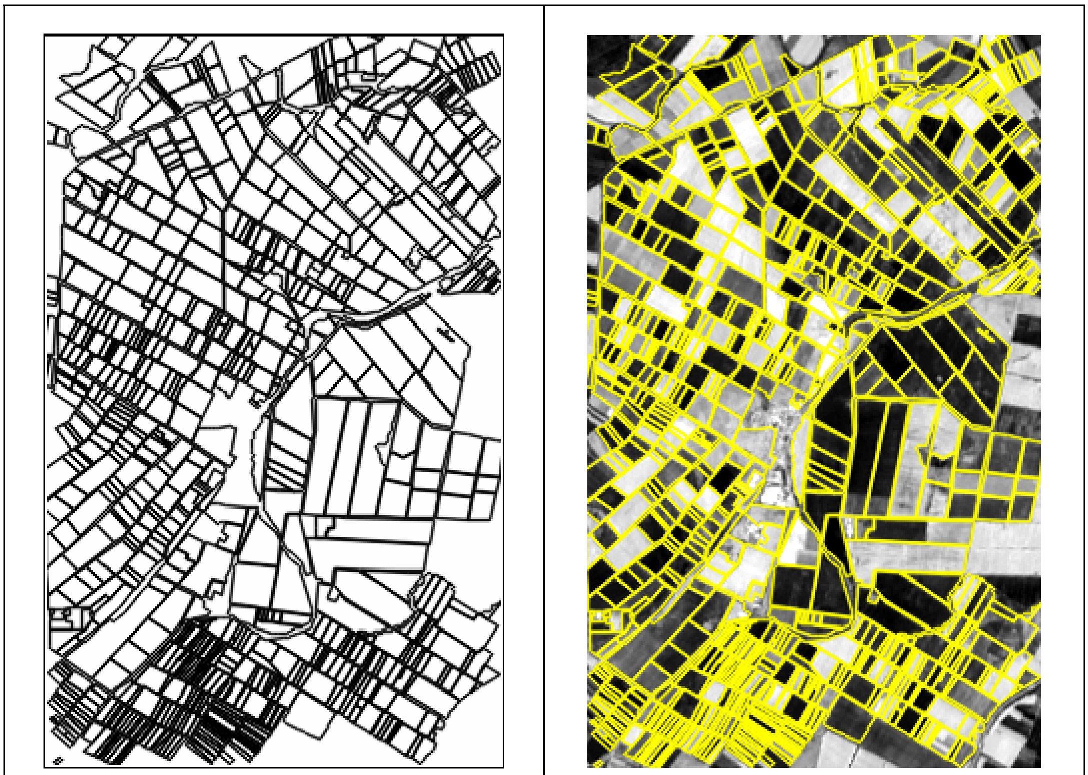


Figure 3.7 – The Identification Convention

For example, the field with an ID of 16 is re-identified as 160 (16*10) and the child segments of the field are identified as 161, 162, and 163. This convention provides the association of the vector data containing the manually segmented field boundaries, with the original vector data. Using this procedure, the following statement can be made; for the field #16 in the original vector data, the output of the truth segmentation contains the sub-fields 161, 162, and 163 of the vector data prepared through the manually segmentation. The verification of the segmentation outputs of the proposed segmentation process is performed using this truth knowledge.

However, there are several restrictions for this identification procedure. There should not be any fields containing more than 9 child segments. Otherwise, the ID numbers may conflict with the other fields. For example, in the above case, if the field #16 had 11 segments, the ID numbers for the 10th and 11th segments would be 170 (160+10), and 171 (160+11), which would cause a conflict with the field #17 in the original vector data. Therefore, the fields 170 and 171 may be misinterpreted as the 1st and the 2nd child segments of the field #17. In addition, if a field was not divided into sub-fields by manual segmentation, the ID is multiplied with 10, and should be included in the new vector data with the new ID (for example, the field #18 without any sub-fields exists in the new vector data with ID 180). This means that the fields having ID numbers ending with "0" in the new vector data are those that are not divided into sub-fields through the manual segmentation operation and the others will be known as the segmented sub-fields.

The erroneous fields and the identifications contradicting the above restrictions are corrected in the vector data. The manually segmented fields are illustrated in Figure 3.8-a. Figure 3.8-b demonstrates the SPOT_4_PCA image with the segmented fields overlaid.

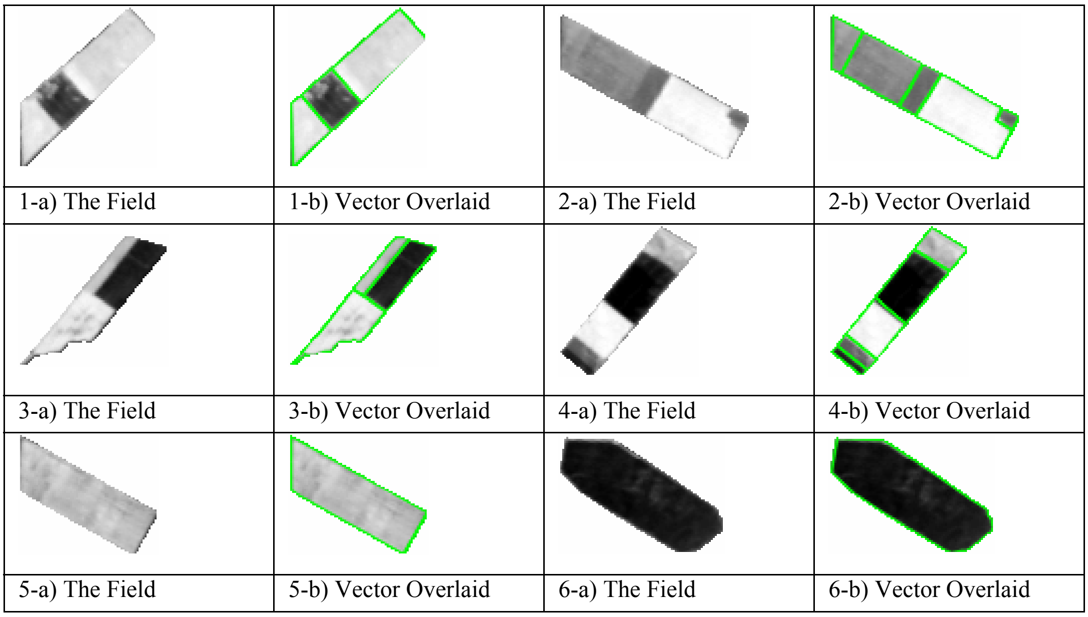


a) Manually Segmented Field Boundaries

b) Overlaid on SPOT_4_PCA

Figure 3.8 – The Manually Segmented Field Boundaries and the Overlaid Display on SPOT_4_PCA

Figure 3.9 illustrate the selected several fields for which the manually segmented sub-boundaries are overlaid on the SPOT_5_PCA image. As can be seen in the Figure 3.9-5b and Figure 3.9-6b, there also exist the fields, which do not contain any sub-fields.



1-a) The Field
1-b) Vector Overlaid
2-a) The Field
2-b) Vector Overlaid
3-a) The Field
3-b) Vector Overlaid
4-a) The Field
4-b) Vector Overlaid
5-a) The Field
5-b) Vector Overlaid
6-a) The Field
6-b) Vector Overlaid

Figure 3.9 – The Manually Segmented Fields with the Overlaid Sub-Boundaries

In order to clarify the terminology, the terms used for the fields and the segments are explained as follows:

Parcel: The un-processed (un-segmented) field defined in the original vector data (Parent Field).

Truth Segment: The manually extracted sub-fields defined as the segments of the parcel.

Result Segment: The sub-fields detected within the parcels through the automated field-based segmentation process.

The verification of the segmentation was carried out using the geometries of the manually segmented fields supplied. The methodology is mainly based on the overlaying of the field geometries derived from the automated segmentation process with the geometries of the manually segmented field geometries. The match between the two objects M_{ij} can be expressed as a geometrical mean of the two conditional probabilities of M_i and M_j [21].

$$
\begin{aligned}
M_{ij} &= \sqrt{(M_i \bullet M_j)} \\
M_i &= Area(i \cap j)/Area(i) \\
M_j &= Area(i \cap j)/Area(j) \qquad \textit{Equation 3.1}
\end{aligned}
$$

M_{ij} gets a value between 0 and 1, where 0 means no matching at all, and 1 indicates a complete match. The matching percentages of all the manually segmented fields with the related output segments are calculated. These values are then used for the accuracy assessment of the segmentation outputs. Two different methodologies are used for evaluating the matching percentage values;

(i) for each parcel, calculating a mean percentage by selecting the overlapping pairs between its sub-fields (truth segments) and the result segments, and

(ii) for each truth segment, defining a success criteria using a threshold for the matching percentage.

In the first method, for each parcel, a matrix is constructed, which contains the values of the matching percentage for each of the truth segment with each of the result segment. Since the number of the detected segments between the truth and the segmentation output may not be equal or there may be more than one match between the segments, this matrix becomes an $m \times n$ matrix which may have more than one non-zero value in its columns and rows.

Each truth segment must be associated with a distinct result segment and the matching percentage between the pair is taken into consideration. Here, the keyword is “distinct” because the pairs can only be selected by satisfying the one-to-one relation. This means that if a truth segment is associated with a result segment then, no other truth segment can also be associated with that result segment. The problem is the selection of the most appropriate pairs in order to satisfy the maximum sum of the matching percentages. This problem is known as the Assignment Problem in the literature [25]. There are several algorithms to solve the one-to-one correlation problem that provides the maximum gain. In the present case, Auction Algorithm is used for selecting the best pairs of the truth and the result segments with one-to-one relation for maximizing the matching percentage sum [11].

When constructing the one-to-one correlations, there may exist some unassigned truth segments. This may be the case when the number of the truth segments is more than the number of the result segments. In such a case, the matching percentage of the unassigned truth segment is set to 0. For each parcel, the mean value is calculated by dividing the sum of the matching percentages by the number of all the truth segments of that parcel. The unassigned truth segments are accepted as the not detected segments and this kind of failures are included in the accuracy measurement. The mean percentage values calculated using all the parcels provides the **overall accuracy**.

In the second interpretation method, a success criterion is determined by defining a threshold value for the matching percentage. The truth segments that have a matching percentage with the result segments higher than the predefined threshold are accepted as the successfully detected segments. The outputs for the other truth segments are considered as unsuccessful. Since one truth segment may match with more than one result segment there may be more than one matching percentage calculated for a truth segment. The maximum of all the calculated matching percentages is chosen to be compared with the success criteria threshold. This case occurs when a truth segment has the intersection areas with more than one result segment.

In the present case, the threshold for the matching percentage was defined as 75% in order to satisfy the success criteria. The similar method and the threshold value were used for the verification of the segmentation outputs by Janssen and Molenaar [21]. In their study, the following parameters and the values were used.

$$\begin{aligned}&Succcess=1 \text{ where } M_{ij} \geq 0{,}75\\&Success=0 \text{ where } M_{ij} < 0.75\end{aligned}$$

Equation 3.2

The ratio of the successfully detected truth segments to all the truth segments is calculated and used as an another verification parameter. In addition, the average matching percentages are calculated for the successfully detected truth segments and the unsuccessful segments. These values are used as other verification parameters. For a parcel, the calculation of the matching percentages of the truth segments is illustrated in Figure 3.10.

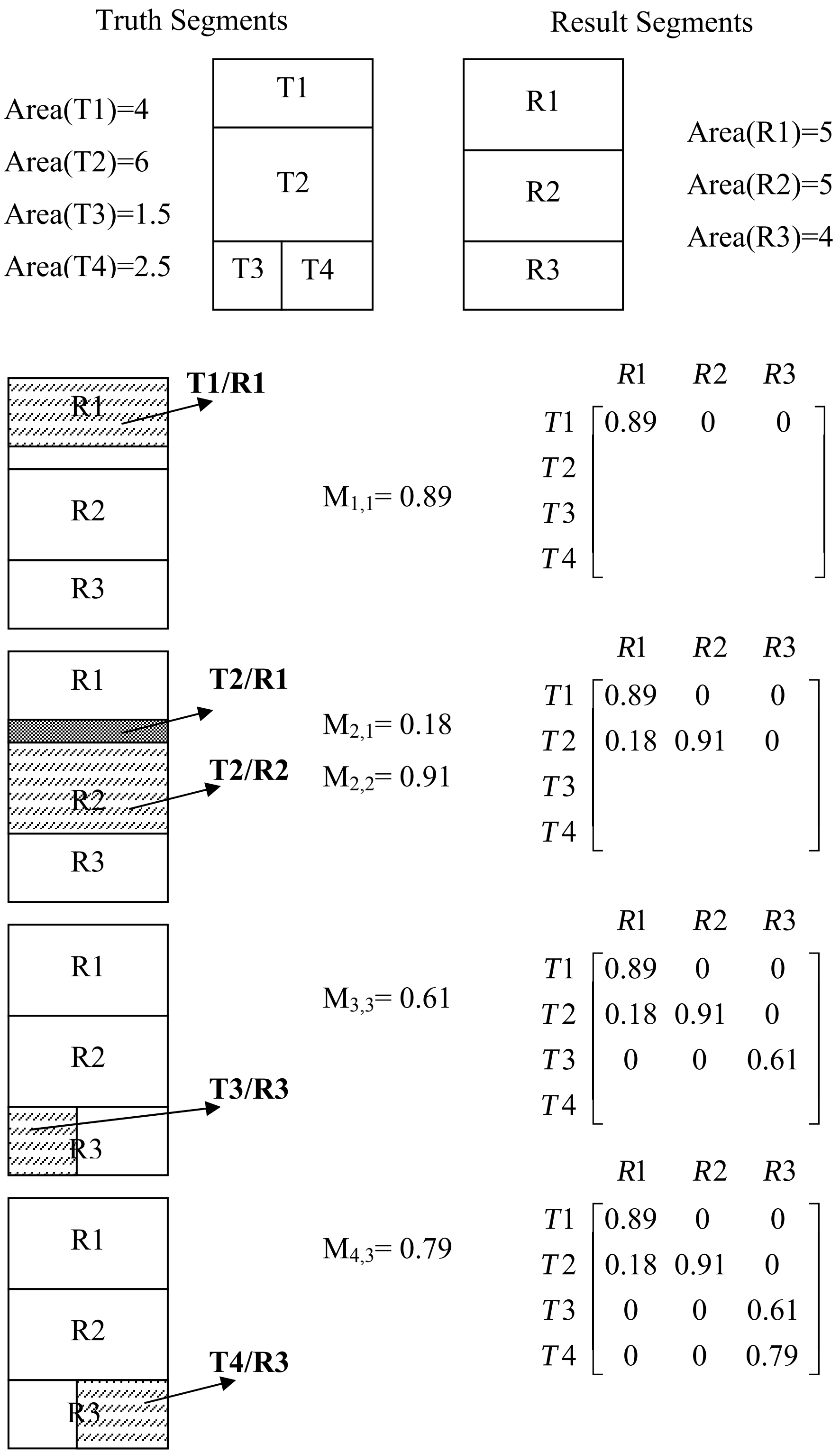


Figure 3.10 – The Calculation of the Matching Percentages for a Field

In Figure 3.10, the output of the segmentation and the truth segments are compared for a sample parcel. There are four segments in the truth and three sub-fields were detected as the segmentation output. For the truth segment T1, since there is no overlaying area between R2 and R3 fields, the $M_{i,j}$ matching percentage between these fields becomes 0. The matching percentage $M_{1,1}$ between T1 and R1 is calculated and written to a matrix. For the truth segment T2, there are two possible overlays that are between T2 and R2 and between T2 and R1. And the matching percentages for the items are calculated and written as the related items in the matrix. For the truth segments T3 and T4, there is an overlay with the result segment R3 only and therefore, $M_{3,3}$ and $M_{4,3}$ values are calculated and written as the related items in the matrix.

After populating the matching percentage matrix, the accuracy assessment was carried out using these values. Using the first interpretation methodology of the matching percentage values, the overall accuracy was calculated as follows:

- Calculate the assignment matrix for the matching percentage matrix that provides one-to-one correlation between the truth segments and the result segments using the Auction algorithm. The detailed explanation of the Auction algorithm is given later in this section.

$$Assignment\ Matrix = \begin{array}{c c} & \begin{array}{ccc} R1 & R2 & R3 \end{array} \\ \begin{array}{c} T1 \\ T2 \\ T3 \\ T4 \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

- Select the matching percentage values from the matching percentage matrix for those pairs whose values are set to 1 in the assignment matrix (T1-R1: 0.89, T2-R2: 0.91, T4-R3:0.79).
- Set the matching percentages of unassigned truth segments to 0 (T3:0).
- Calculate a mean matching accuracy by dividing the sum of the matching percentage values to the number of truth segments ((0.89+0.91+0+0.79)/4 = **65%**).

In order to calculate the overall accuracy of the all segmentation process, this procedure is applied for each parcel. The average value for the mean matching accuracies of the parcels is calculated as the first verification parameter. This parameter indicates the overall accuracy of the proposed segmentation process.

$$1^{st}\ Verification\ Param. = \frac{\sum (Mean\,Matching\ Accuracy)\ \ for\ All\ Parcels}{\#of\ \ Parcels}$$

Equation 3.3

Using the second interpretation methodology of the matching percentage values, the corresponding verification parameters are calculated as follows:

- For each truth segment select the maximum matching percentages (T1: 89%, T2: 91%, T3: 61%, T4:79%).
- Accept those whose matching percentage values are bigger than the predefined threshold value of 75% as successful (T1, T2, T4).
- Calculate the ratio between the successfully detected truth segments and the number of all the truth segments (3/4=**75%**).

$$2^{nd}\ Verification\ Param. = \frac{\#of\ \ Truth\ Segments\ Having\ M_{ij} \geq 0.75}{\#of\ \ Truth\ Segments}$$

Equation 3.4

- In addition, calculate the matching percentage average for the successfully detected segments. (For T1, T2, and T4: [89+91+79]/3=86%)

$$3^{rd}\ Verification\ Param. = \frac{\sum_{i=1}^{n} M_{ij}\ Truth\ Segments\ Having\ M_{ij} \geq 0.75}{n}$$

Equation 3.5

where;

$n = \#\ of\ Truth\ Segments\ Having\ M_{ij} \geq\ 0.75$

- Further, calculate the matching percentage average for the unsuccessfully detected segments (For T3: 61/1=61%).

$$4^{th}\ Verification\ Param. = \frac{\sum_{i=1}^{k} M_{ij}\ Truth\ Segments\ Having\ M_{ij} < 0.75}{k}$$

Equation 3.6

where;

n = # of Truth Segments Having M_{ij} *< 0.75*

As a last evaluation approach, a quantitative analysis is applied between the outputs of the segmentation and the truth segments. This analysis is performed, for each parcel, by comparing the number of the truth segments and the number of the result segments detected through the field-based segmentation process. The parcels are grouped according to this comparison as follows:

- Over-segmented parcel: the number of the result segments is more than the number of the truth segments.
- Under-segmented parcel: the number of the result segments is less than the number of the truth segments.
- Equally-segmented parcel: the number of the result segments is equal to the number of the truth segments.

The number of the parcels belonging to each group are computed and used for analyzing the results. For each group, the total number of the result segments and the total number of the truth segment are calculated. In addition, the mean of the differences between the number of the result segments and the truth segments detected, for each parcel, should be calculated to better evaluate the over-segmentation and the under-segmentation problem. The mean of the differences (MD) can be calculated for each group as follows:

$$MD = \frac{\sum_{i=1}^{n} |R_p - T_p|}{n}$$

Equation 3.7

where;

Rp = # of the Result Segments detected for the parcel p

Tp = # of the Truth Segments for the parcel p

n = # of the parcels belonging to that group

In addition, the ratio between the number of the total result segments obtained from all the parcels and the number of the total truth segments can be given as an another verification parameter.

$$5^{th}\ Verification\ Param = \frac{\#of\ \ Total\ \ Result\ \ Segments}{\#of\ \ Total\ Truth\ Segments}$$

Equation 3.8

However, these values do not reflect the accuracy of the segmentation results completely, because they do not comprise any measurements for the matching of the field geometries. The quantities of the result segments and the truth segments may be similar, but this does not mean that the segmentation results are very accurate. The detected result segments and the truth segments can be completely different and have irrelevant geometries. As described above, the first four verification parameters are used for the qualitative analysis of the segmentation process considering the matching of the geometries between the detected sub-fields and the truth segments.

In the first verification parameter, the calculation of the assignment matrix, to select the most appropriate pairs is realized using the Auction algorithm. Auction is one of the most powerful algorithms to solve the assignment problems. A brief description of the assignment problem and the Auction algorithm is presented in the following.

3.2.1 Assignment Problem

The assignment problem was originally considered for the problems in the economy theory such as assigning personnel to jobs and delivery trucks to locations. The objective of this problem is to minimize the cost (or maximize profit) using the available resources. As the assignment problem has become relevant to other fields such as tracking, the new improved techniques have evolved [25].

The assignment problem can be represented as follows:

Given the scalar elements a_{ij} of the matrix, find $X = \{x_{ij}\}$ so that the gain C is maximized.

$$C = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij}$$

is minimized subject to:

$$\sum_{i} x_{ij} = 1 \forall j$$

$$\sum_{j} x_{ij} = 1 \forall i$$

An optimal solution to this problem allows the elements x_{ij} to be either 0 or 1. The a_{ij} is the value that is gained when assigning element i of the matrix to element j. And x_{ij} is the value of the assignment which is 1 or 0.

In the present case, two sets of segments (the truth segments and the result segments for a parcel) are assigned to each other. A more general form of the assignment problem is encountered due to the following factors:

- The number of the truth segments and the result segments need not be equal all the time.

- There may be some truth segments which are remaining unassigned at end of the assignment operation in the case the number of the truth segments is bigger than the number of the result segments.
- There may also be some result segments that are remaining unassigned at the end of the assignment operation in the case the number of truth segments is smaller than the number of the result segments.

The Auction Algorithm

As in a real auction, the Auction Algorithm seeks to maximize the gain (the total cost) [11]. The algorithm is composed of two phases: (i) the bidding phase and (ii) the assignment phase.

The bidding phase consists of finding the "best" result segment for each unassociated truth segment and bidding for it. A bid is computed to raise the "price", *P*, of the result segment so that the truth segment would be "satisfied "with its second best result segment if another truth segment "buys" the best result segment.

The assignment phase assigns a result segment to a truth segment and removes the previous assignment if necessary. This iterative process continues untill all the truth segments are "almost happy" and, thus the "almost optimum assignment" or the optimal assignment has been found. As in real auctions, the algorithm converges quicker with larger building steps but larger steps may not achieve the optimal assignment.

A truth segment is "almost happy" if the truth segment's assignment to result segment i_j has a value that is within ε of being maximal (ε determines whether the assignment will be optimal), that is,

$$(\max_i (a_{ij} - P_i)) - (a_{i_j j} - P_{i_j}) = \varepsilon$$

Equation 3.9

where

a_{i_j} = *gain from assigning the truth segment j to the result segment* i

and

P_{i_j} = *price of the segment* ij.

If at least one feasible assignment exists, the auction algorithm terminates with a feasible assignment that is within $n\varepsilon$ of being optimal (where *n is* the min[number of truth segments, number of result segments]), that is, the total benefit of the final assignment is within $n\varepsilon$ of the total benefit optimal assignment (and is optimal if the benefits a_{ij} are integer and $\varepsilon < 1/n$).

A gain matrix **G** (which is $m\times n$) is constructed containing the matching percentages for each truth segment and the result segment.

$$\mathbf{G} = \begin{bmatrix} M_{11} & M_{12} & . & . & M_{1n} \\ M_{21} & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ M_{m1} & . & . & . & M_{mn} \end{bmatrix}$$

The Auction algorithm continues the process until all the truth segments are assigned to result segments. In order to prevent the infinite loop, the candidate result segments corresponding to all the truth segments should be generated and the distance matrix should be augmented in order to obtain **AG** (which is $m\times(n+m)$).

$$\mathbf{AG} = \begin{bmatrix} M_{11} & M_{12} & . & . & M_{1n} & t & \times & \times & \times & \times \\ M_{21} & . & . & . & . & \times & t & \times & \times & \times \\ . & . & . & . & . & \times & \times & t & \times & \times \\ . & . & . & . & . & \times & \times & \times & t & \times \\ M_{m1} & . & . & . & M_{mn} & \times & \times & \times & \times & t \end{bmatrix}.$$

Again, *t* is chosen to be equal to the threshold, and × is a value that prevents the association. The above operation has the following physical meaning. A truth segment is automatically assigned to the corresponding tentative result segment if there is no other result segment to be assigned.

Steps

a. Initialize all truth segments as unassigned. Initialize the segment prices P_i (usually small or zero).

b. Select a truth segment, *j*, that is unassigned. If none exists, done.

c. Find the best result segment i_j for each truth segment *j*. Find i_j such that

$$(a_{i_j j} - P_{i_j}) = \max_{i=1,\ldots,n}(a_{ij} - P_i).$$

Equation 3.10

d. Unassign the truth segment previously assigned to i_j (if any) and assign the result segment i_j to truth segment *j*.

e. Set the price of result segment i_j to the level at which the truth segment *j* is almost happy.

$$P_{i_j} = P_{i_j} + y_j + \varepsilon.$$

Equation 3.11

where;

y_j = *difference between the best and second best assignment values for the truth segment j.*

f. return to step b.

After the assignment matrix is obtained, the augmented part is thrown and the remaining part is returned as the result of the assignment operation.

3.3 Results and Analysis

As described in the input data section, two types of images acquired by the SPOT 5 and SPOT 4 sensors were processed using different band combinations. The segmentation was performed on each data set using the same vector data. For each image, the results are provided and a comparative analysis is presented for all the results obtained. The results of the automated segmentation include the verification parameters and the displays of several sample output fields.

The general statistical information for the parcels in the existing field boundary data, which is valid for each input image, is given in Table 3.5. For the segmentation results, the additional analysis and the accuracy assessment parameters are presented separately for each input dataset in the following sections. Table 3.5 consists of the total number of the existing parcels and the number of parcels processed through the proposed automated segmentation procedure. As stated previously, small fields were excluded from segmentation process.

Table 3.5 – General Information for the Parcels in the Vector Data

Item	**Value**
# of the Total Parcels in the Vector Data	514
# of the Parcels Falling Below the Area Threshold	222
# of the Processed Parcels by Automated Segmentation (514-222)	292

There are 222 parcels falling below the predefined threshold of 15000m^2. These fields generally (210/222) consist of one land cover. However, 12 out of the 222 parcels were divided into sub-fields through the manual segmentation. Since these 12 parcels were not processed by the automated segmentation, their sub-fields were not detected. Therefore, this common error occurred for each of the input data set, due to using the same area threshold used for all.

For each input image, the quantitative measurements are given in Table 3.6, Table 3.8 and Table 3.10 to provide the over-segmentation and the under-segmentation analysis of the results. These tables include the number of the over-segmented, under-segmented and equally segmented parcels with the means of the differences (Equation 3.7). In addition, the total number of the sub-fields obtained through the automated and the manual segmentations are given in these tables.

Tables 3.7, 3.9 and 3.11 contain the verification parameters. The methodology behind the calculations for these verification parameters were explained in the evaluation methodology section. The accuracy assessment was performed using the values contained within these tables obtained for each image.

For each image, the common errors realized in the outputs of the automated segmentation are categorized into the groups and discussed with their reasons on the sample fields.

The images containing the new sub-boundaries obtained through the automated segmentation are displayed for each dataset. For each input image, several samples of the successfully and unsuccessfully detected outputs are provided in the Appendix-A.

3.3.1 The SPOT_5_PCA Image

First, the automated segmentation procedure was implemented on the SPOT_5_PCA image using the developed software. A total of 292 parcels falling above the specified area threshold were processed by the FBISS software. Table 3.6 represents the quantitative results for the SPOT_5_PCA image through the over-segmentation and the under-segmentation perspectives.

Table 3.6 – The Quantitative Results of the Segmentation for SPOT_5_PCA

	# of The Parcels	# of the Result Segments	# of the Truth Segments	MD(*)
Equally-segmented Parcels	180	311	311	0
Over-segmented Parcels	60	197	98	1.65
Under-segmented Parcels	52	122	206	1.61
Total	292	630	615	-

(*) MD: Mean of the Differences (Equation 3.7)

As can be seen in Table 3.6, there are 60 over-segmented and 52 under-segmented parcels. However, the results for these 112 (60+52) parcels cannot be declared as totally unsuccessful outputs. Similarly, the outputs for equally-segmented 180 parcels cannot be declared as totally successful. This is because, as explained in the evaluation methodology section, the geometries of the detected segments are not included in these measurements. On the other hand, the outputs of the 112 parcels that are either over-segmented or under-segmented parcels can be considered as known erroneous outputs. And for these fields, the percentages of the errors are calculated considering the geometries in the verification parameters.

Table 3.7 contains the verification parameters obtained through the automated segmentation results. The first verification parameter (Equation 3.3) stands for the overall accuracy. The second verification parameter (Equation 3.4) is the measurement of the verification computed as the ratio between the number of successfully detected truth segments and the total number of truth segments. The third verification parameter (Equation 3.5) is the average matching percentage values for the successfully detected truth segments. The fourth verification parameter (Equation 3.6) is the average matching percentage values for the unsuccessfully detected truth segments. The last verification parameter (Equation 3.8) is the ratio between the number of the truth segments and the result segments.

Table 3.7 – The Verification Parameters for SPOT_5_PCA

Item	**Value**
Verification Parameter-I:(*Overall Accuracy)* (Matching Percentage Average For All Parcels)	**83.8%**
Verification Parameter-II : (Successful / Total Truth: 434/615)	70.6%
Verification Parameter-III (Matching Percentage Average For Successful)	94.6%
Verification Parameter-IV (Matching Percentage Average For Unsuccessful)	54.8%
Verification Parameter-V (Number of Results/ Number Of Truth: 630/615)	1.02 2% difference

As described in the evaluation methodology section, the first parameter (83.8%) is accepted as the overall accuracy of the results. The second verification parameter (70.6%) can also be used for the accuracy assessment [21]. However it must be considered together with the third and the fourth verification parameters. As can be seen in Table 3.7, the matching percentage average for the successfully detected truth segments is very high (94.6%), and the matching percentage average for the unsuccessfully detected truth segments is considerably low (54.8%). This means that the automated segmentation has detected 70 of the 100 truth segments successfully with an average of 94% geometry matching and the has wrongly detected the 30 of 100 truth segments with an average error of 45.2% (100-54.8) geometry matching. If these average values are considered with the second verification parameter (70.6%), the difference between the calculated overall accuracy (83.8%) seems reasonable. Because the calculation of the overall accuracy is based on the matching of the geometries of all parcels and 94%

geometry matching for the successfully detected truth segments is rather a high value which increases the overall accuracy value.

The fifth verification parameter shows the difference between the number of truth segments and the number of result segments which is only 1.02%. This means that the outputs contain neither a significant over-segmentation nor a significant under-segmentation. Also, as can be seen in Table 3.6, there is not a great difference between the numbers of the over-segmented and the under-segmented parcels (60/52). For the over-segmented and the under-segmented parcels the difference between the result segments and truth segments quantities are not very high having the mean of differences of 1.65 and 1.61 respectively. Figure 3.11 illustrates the SPOT_5_PCA image with the new sub-field boundaries extracted through the field-based segmentation process overlaid.

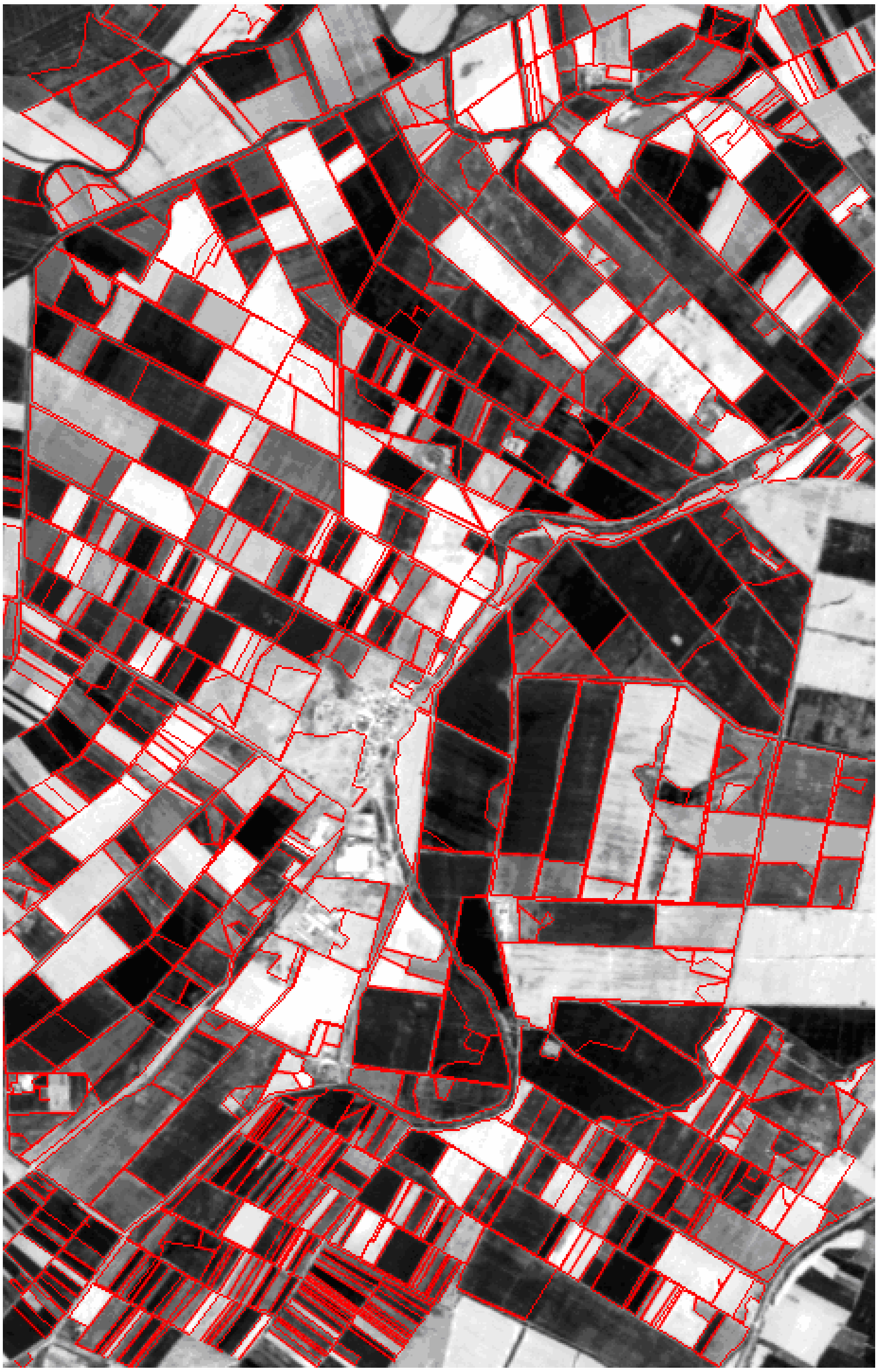


Figure 3.11 – The SPOT_5_PCA Image with the Segmentation Results Overlaid

Several outputs for the sample fields of SPOT_5_PCA containing the field image, the output of Canny Edge Detector, and the results of the automated and the manual segmentation processes are given in Table A.1 in Appendix-A.

The common errors encountered in the outputs of the SPOT_5_PCA image are discussed below on the sample fields. The erroneous outputs can be grouped according to their error types and the reasons that cause errors.

A. Over-segmented Fields (60 Parcels)
    1. The Noisy Edges after the Canny Edge Detector: This is a very common problem for the edge-based image segmentation techniques. The edges detected due to noise may form boundaries of the sub-fields and cause an over-segmentation problem.
    2. The Erroneously Extended Lines by the Perceptual Grouping Rules: The Perceptual Grouping defines rules for extending and modifying the line segments in order to form the closed regions. However, these rules cannot handle all the cases and may apply wrong modifications on the line segments and therefore result in over-segmentation.

B. Under-segmented Fields (52 parcels)
    1. The Missing Edges after the Canny Edge Detector: The edge detection operator may miss some of the edges due to spectral similarities between different land cover types existing within a field. Higher and lower thresholds used in Canny may not be the most appropriate values for all the fields.
    2. The Erroneously Deleted Lines by the Perceptual Grouping Rules: The Perceptual Grouping removes the detected noisy line segments within the image. However, sometimes the line-segments that are necessary to form sub-field boundaries are deleted erroneously during the perceptual grouping process if they satisfy the removal condition defined by the rules.

3. The Erroneously Merged Sub-Fields: In the polygon construction process, a rule based merging mechanism is used to resolve the over-segmentation problem. However, some of the polygons that actually contain distinct crop fields can be erroneously merged to other polygons if they satisfy the pre-defined merge condition and causing an under-segmentation problem.

C. The Field Geometry Errors on Equally-Segmented Fields (3.5% average matching percentage error on 180 parcels): Since the accuracy of the geometries is defined based on the matching percentages, the errors are also expressed as a matching percentage error for the equally-segmented fields. The mean of the matching percentages of 180 parcels is equal to 96.5%. Therefore, the average field geometry error for the equally segmented parcels is 3.5 % (100-96.5). The reasons of the geometry errors are generally the type B2. The erroneously extended line segments forms the erroneous geometries. Also all the other errors defined both for the over-segmentation and the under-segmentation can be the reasons of the field geometry errors.

Figure 3.12 illustrates the reasons of the types A1 and A2 of errors (under-segmentation) on a sample field.

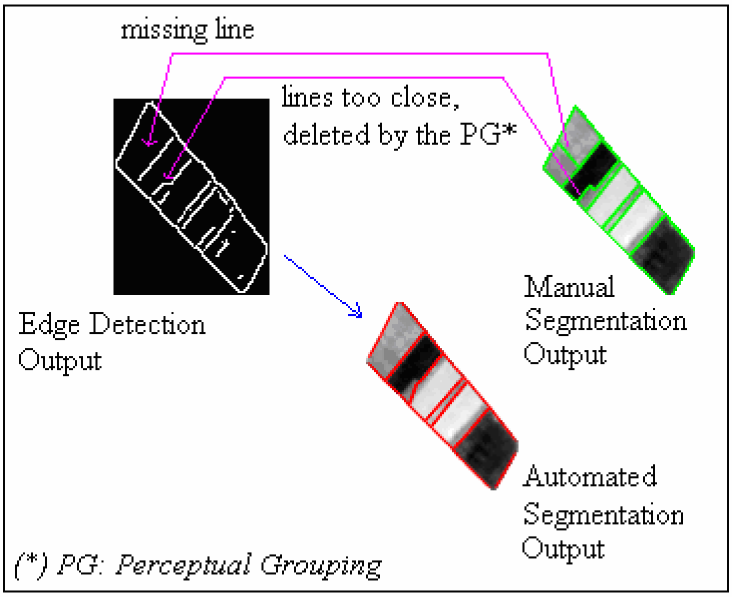


Figure 3.12 – The Illustration of the Under-segmentation Errors (A1, A2)

As can be seen in Figure 3.12 some of the sub-fields cannot be detected due to the missing lines in the output of edge detection. Also one of the necessary line segments is deleted by the Perceptual Grouping Rules due to being very close to another line segment.

Figure 3.13 illustrates the reasons of the A2 and A3 types of errors (under-segmentation) on a sample field. In the sample field, some of the sub-fields cannot be detected because of the missing lines in the edge detection output. However, one of the sub-fields is detected and remains as a close region in the output of the perceptual grouping. But it is erroneously merged with the adjacent polygon due to the size and the shape constraints applied in the polygon construction process.

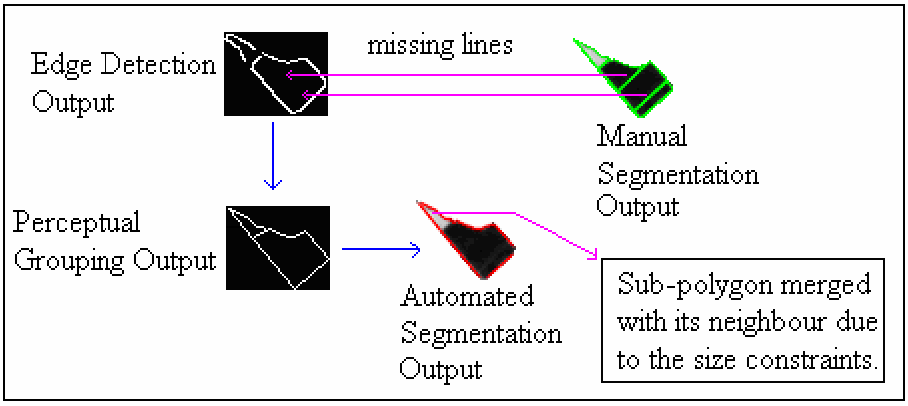

Edge Detection Output
missing lines
Manual Segmentation Output
Perceptual Grouping Output
Automated Segmentation Output
Sub-polygon merged with its neighbour due to the size constraints.

Figure 3.13 – The Illustration of the Under-segmentation Errors (A2, A3)

Figure 3.14 illustrates the over-segmentation errors caused by the noisy lines detected as the edges through the edge detection process. As can be seen in the figure, there are over segmented sub-fields detected which were wrongly constructed using the noisy lines. One of the correct line segments, which is a part of the boundary of a sub-field, is extended to the wrong direction, and an erroneous geometry is constructed.

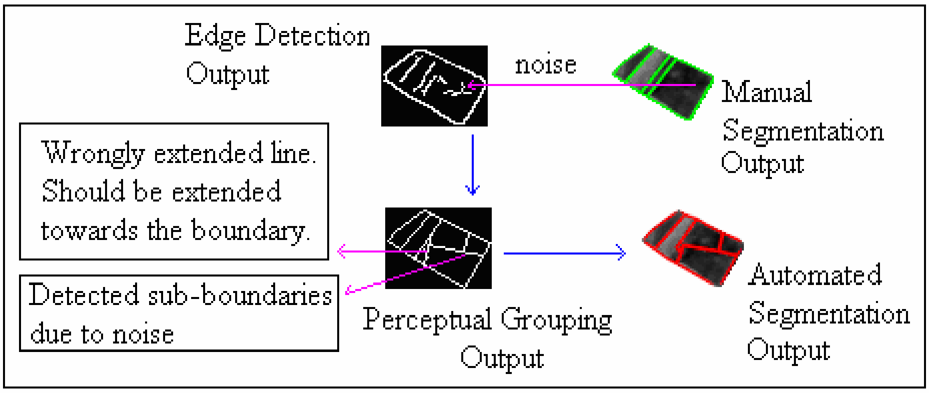

Edge Detection Output
noise
Manual Segmentation Output
Wrongly extended line. Should be extended towards the boundary.
Detected sub-boundaries due to noise
Perceptual Grouping Output
Automated Segmentation Output

Figure 3.14 – The Illustration of the Over-segmentation Errors (B1, B2)

3.3.2 The SPOT_5_Intensity Image

The SPOT_5_Intensity image was the second image processed through the automated segmentation using the developed software. Table 3.8 represents the quantitative results obtained from the segmentation of the SPOT_5_Intensity image through the over-segmentation and the under-segmentation perspectives.

Table 3.8 – The Quantitative Results of the Segmentation for SPOT_5_Intensity

	# of The Parcels	# of the Result Segments	# of the Truth Segments	MD(*)
Equally-segmented Parcels	177	283	283	0
Over-segmented Parcels	53	191	87	1.96
Under-segmented Parcels	62	142	245	1.66
Total	292	616	615	-

(*) MD: Mean of the Differences (Equation 3.7)

There are 53 over-segmented and 62 under-segmented parcels in the segmentation outputs. Compared to the results of the SPOT_5_PCA image less over-segmentation and higher under-segmentation is evident. The smoothness of the SPOT_5_Intensity image and the lower contrast with respect to the SPOT_5_PCA image may be the basic reasons for these results.

Table 3.9 – The Verification Parameters for SPOT_5_Intensity

Item	**Value**
Verification Parameter-I:(*Overall Accuracy)* (Matching Percentage Average For All Parcels)	**82.6%**
Verification Parameter-II : (Successful / Total Truth: 415/615)	67.5%
Verification Parameter-III (Matching Percentage Average For Successful)	94.6%
Verification Parameter-IV (Matching Percentage Average For Unsuccessful)	54.1%
Verification Parameter-V (Number of Results/ Number Of Truth: 616/615)	1.01 1% difference

The verification parameters for the SPOT_5_Intensity image are given in table 3.9. The overall accuracy for the segmentation of the SPOT_5_Intensity image was 82.6%. This is slightly lower accuracy when compared to the accuracy of the SPOT_5_PCA image segmentation. The number of successfully detected truth segments (415) that affect the second accuracy assessment parameter is also lower than the SPOT_5_PCA image. The discussion about the accuracy differences between the SPOT_5_PCA and the SPOT_5_Intensity images is given in 3.3.5 "Discussion of the Results" section.

The verification parameter-II has a value of 67.5%. However the average matching percentages of the successfully found truth segments is rather high (94.6).

The verification parameter-V (1.01%) indicates that the number of the result segments and the number of the truth segments are almost the same. This means that the results are neither significantly over-segmented nor significantly under-segmented. The numbers of the over-segmented and the under-segmented parcels are also similar and the mean of the differences are not very high. Figure 3.15 illustrates the SPOT_5_Intensity image with the overlay of the new sub-field boundaries segmented through the proposed segmentation process.

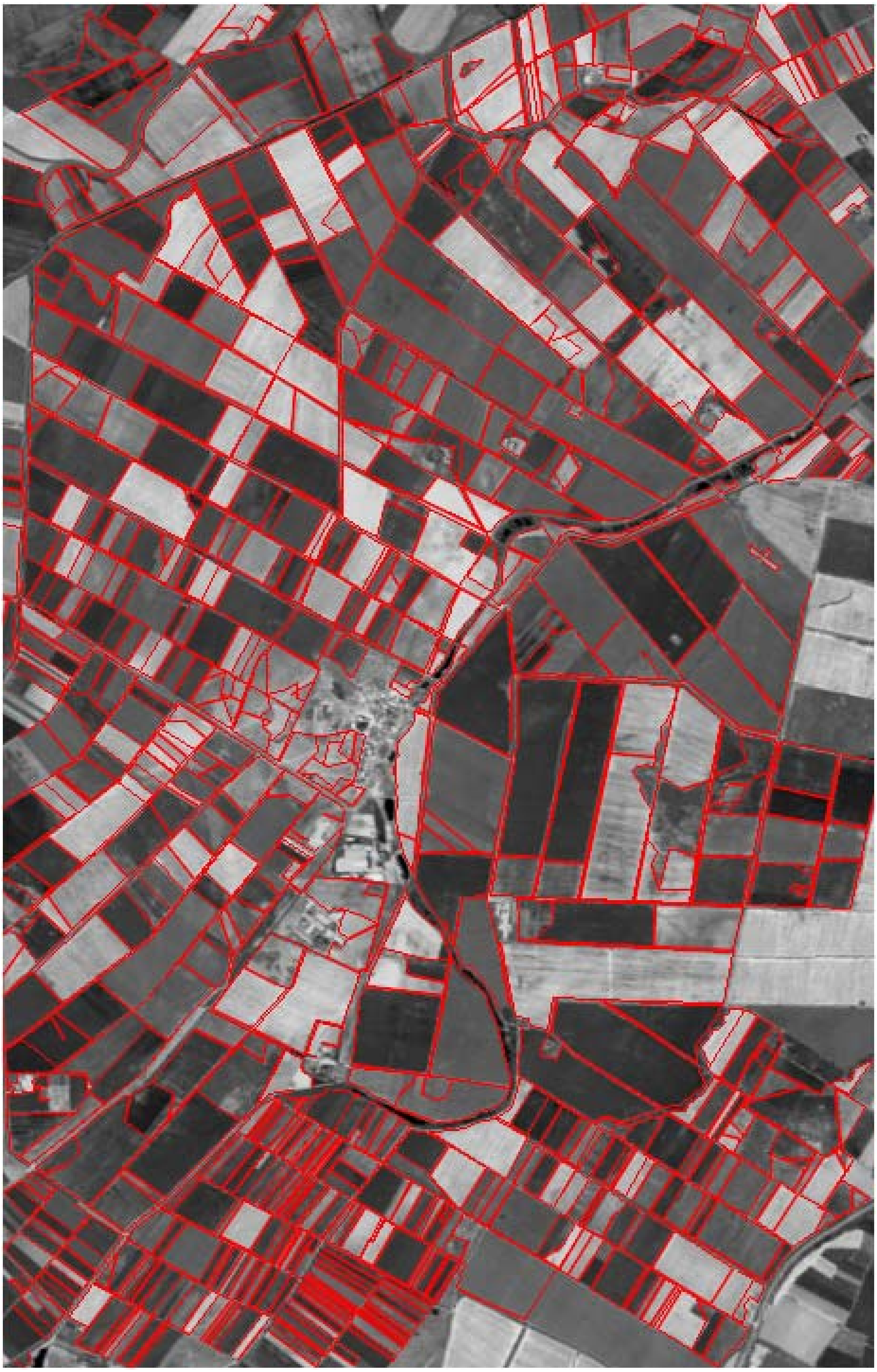


Figure 3.15 – The SPOT_5_Intensity Image with the Segmentation Results Overlaid

Several sample outputs from the SPOT_5_Intensity image including the field image, the output of the Canny Edge Detector, and the results of the automated and the manual segmentation processes are given in Table A.2 in Appendix-A.

According to the common errors mentioned earlier, the outputs of the parcels can be categorized as follows

A. Over-segmented Fields (53 Parcels)
B. Under-segmented Fields (62 Parcels)
C. The Field Geometry Errors on Equally-Segmented Fields (2.8 % average geometry matching error on 177 parcels)

The common reasons for these errors described for the SPOT_5_PCA image are the same for the SPOT_5_Intensity image.

3.3.3 The SPOT_4_PCA Image

The SPOT_4_PCA image was the third image processed to implement the automated segmentation procedure using the developed software. The quantitative results are given in Table 3.10.

Table 3.10 – The Quantitative Results of the Segmentation for SPOT_4_PCA

	# of The Parcels	# of the Result Segments	# of the Truth Segments	MD(*)
Equally-segmented Parcels	165	232	232	0
Over-segmented Parcels	46	178	83	2.06
Under-segmented Parcels	81	175	300	1.56
Total	292	585	615	-

(*) MD: Mean of the Differences (Equation 3.7)

The number of the under-segmented parcels (81) is higher than the number of the over-segmented parcels (46). The crop variations within the fields were rather less detected and the segmentation provided the under-segmented outputs. Table 3.11 presents the verification parameters calculated for the SPOT_4_PCA image.

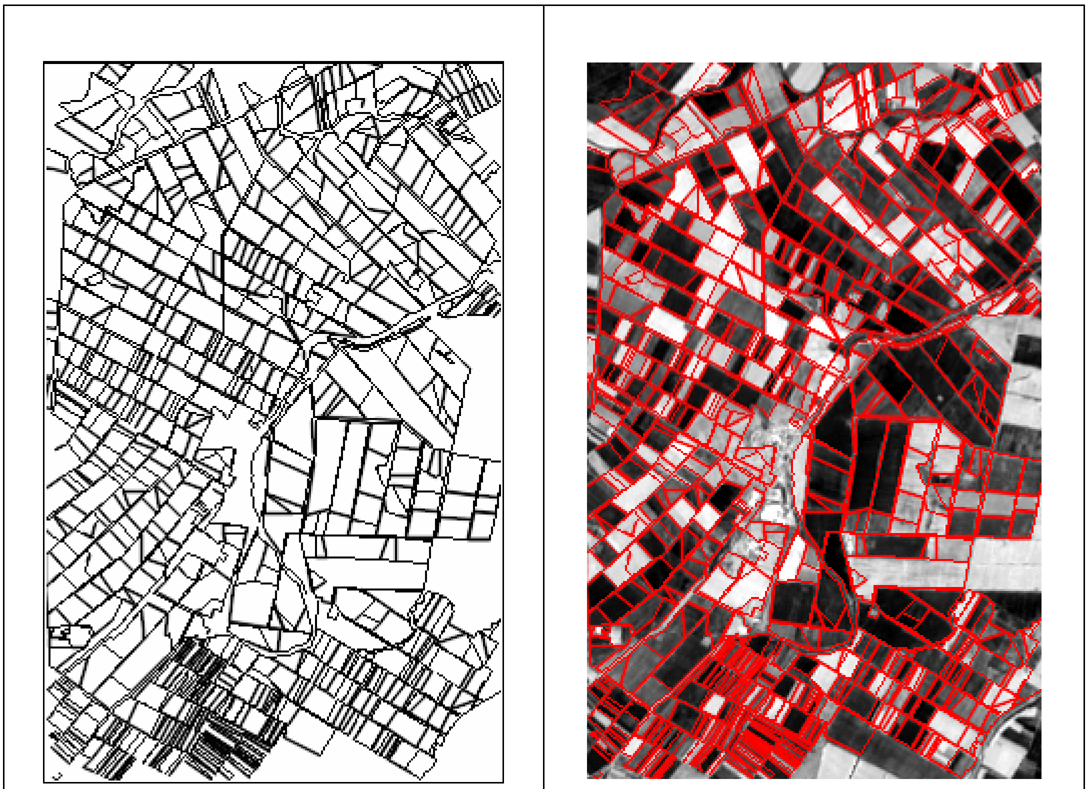
Table 3.11 – The Verification Parameters for SPOT_4_PCA

Item	**Value**
Verification Parameter-I:*(Overall Accuracy)* (Matching Percentage Average For All Parcels)	**78.8%**
Verification Parameter-II : (Successful / Total Truth: 378/615)	61.5%
Verification Parameter-III (Matching Percentage Average For Successful)	94.2%
Verification Parameter-IV (Matching Percentage Average For Unsuccessful)	52.1%
Verification Parameter-V (Number of Results/ Number Of Truth: 585/615)	0.95 5% difference

The overall accuracy calculated for the SPOT_4_PCA image was 78.8%. The number of successfully detected truth segments is 378 out of 615. These values are rather low when compared to the segmentation outputs of the SPOT 5 images. As described in the input section, the SPOT 4 multispectral image has a spatial resolution of 20 meters. Therefore, less number of SPOT 4 image pixels fall within the fields. Obviously, the sizes have affected both the edge detection procedure and the efficiency of the perceptual grouping rules. The accuracy differences between the outputs of the SPOT 4 and SPOT 5 images are discussed in detail in section 3.3.5.

The differences between the total number of the result segments obtained through the automated segmentation and the total number of the truth segments is given using the Verification Parameter-V. The ratio value of 0.95 means that the outputs are said to be under-segmented.

Figure 3.16-a illustrates the sub-field boundaries detected through the automated segmentation procedure. The SPOT_4_PCA image with the new sub-field boundaries overlaid is given in Figure 3.16-b.



a) Found Boundaries

b) Overlaid On SPOT_4_PCA

Figure 3.16 – The Detected Field Boundaries and the Overlay on SPOT_4_PCA

Several sample outputs from the SPOT_4_PCA image including the field image, the output of the Canny Edge Detector, and the results of the automated and the manual segmentation processes are given in Table A.3 in Appendix-A. The parcels according to the above mentioned common errors can be categorized as follows:

A. Over-segmented Fields (46 Parcels)
B. Under-segmented Fields (81 Parcels)

C. The Field Geometry Errors on Equally-Segmented Fields (2.6 % average geometry matching error on 165 parcels)

The main reasons for these errors described above for the SPOT 5 images are also the cases for the SPOT_4_PCA image. In addition, the size of the fields had effects on the outputs of the edge detection and perceptual grouping procedures. These effects are demonstrated in Figure 3.17.

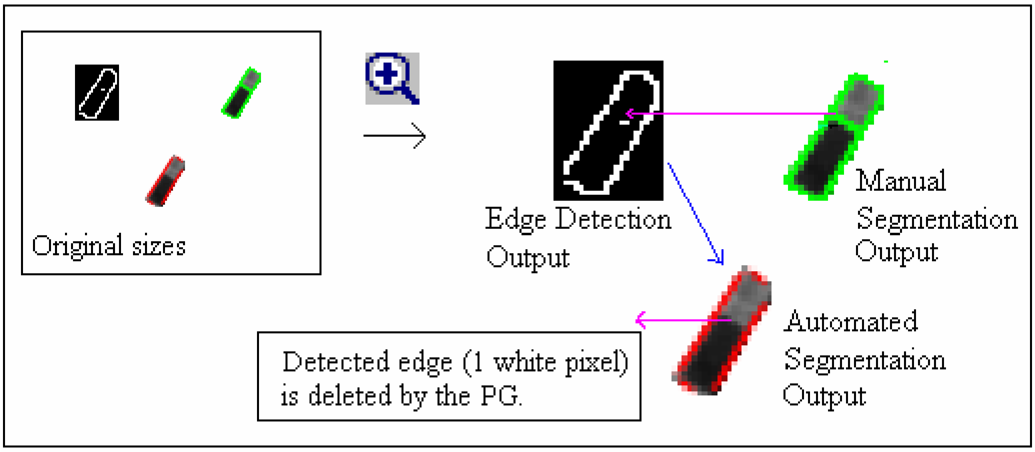

Original sizes
Edge Detection Output
Manual Segmentation Output
Automated Segmentation Output
Detected edge (1 white pixel) is deleted by the PG.

Figure 3.17 –The Effect of the Field Size on the Results of the Segmentation

As can be seen in the figure, the size of the field is very small, only one pixel was able to be detected as the edge. Since there is only one pixel detected as the edge, therefore it cannot be converted into the line segment through the Perceptual Grouping even though using the adapted thresholds for the SPOT 4 images.

3.3.4 The SPOT_4_Intensity Image

SPOT_4_Intensity was the last image processed the automated segmentation procedure using the developed software. The results are given in Table 3.12.

Table 3.12– The Quantitative Results of the Segmentation for SPOT_4_Intensity

	# of The Parcels	# of the Result Segments	# of the Truth Segments	MD(*)
Equally-segmented Parcels	154	205	205	0
Over-segmented Parcels	43	148	74	1.72
Under-segmented Parcels	95	171	336	1.73
Total	292	524	615	-

(*) MD: Mean of the Differences (Equation 3.7)

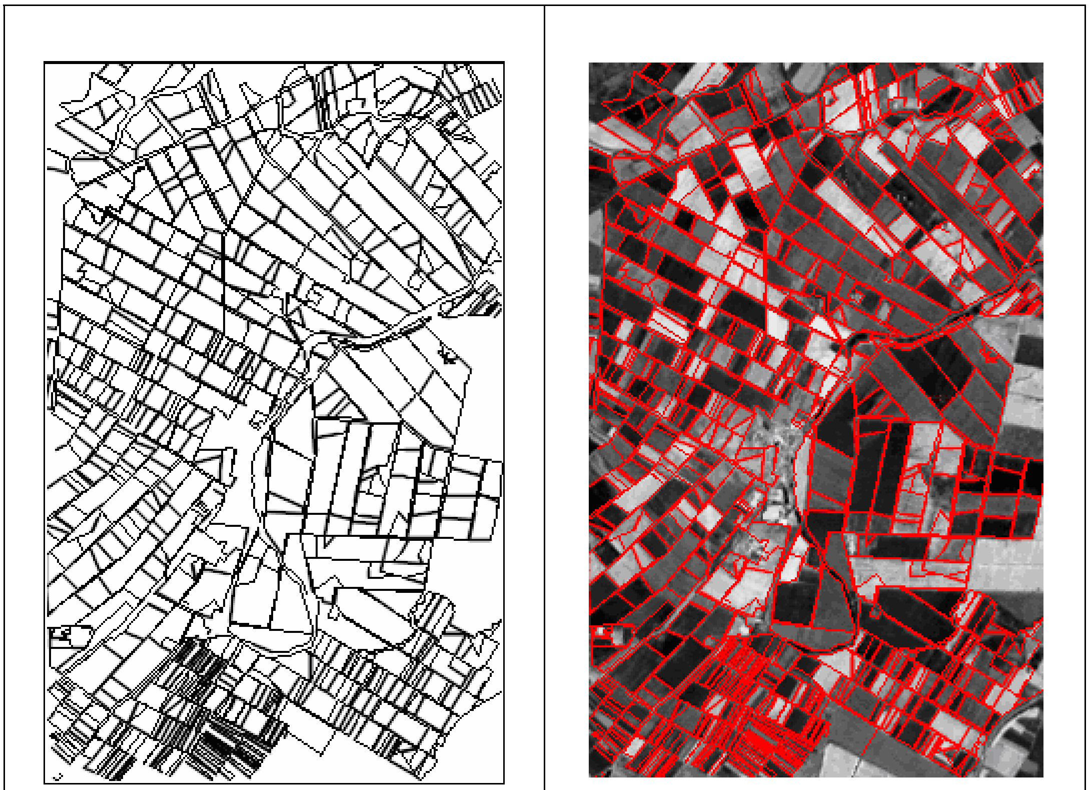
As can be seen in Table 3.12, the under-segmentation is evident. The number of the under-segmented parcels (95) is higher than the number of over-segmented parcels (43). Also, the difference between the total number of the truth and the result segments (615-524=91) is rather high. The low contrast, lower variations within the fields and the small sized fields may be the main reasons caused for the under-segmentation.

Table 3.13 – The Verification Parameters for SPOT_4_Intensity

Item	**Value**
Verification Parameter-I:(*Overall Accuracy)* (Matching Percentage Average For All Parcels)	**76.2%**
Verification Parameter-II : (Successful / Total Truth: 354/615)	57.6%
Verification Parameter-III (Matching Percentage Average For Successful)	93.9%
Verification Parameter-IV (Matching Percentage Average For Unsuccessful)	49.3%
Verification Parameter-V (Number of Results/ Number Of Truth: 524/615)	0.85 15% difference

For the SPOT_4_Intensity image, the overall accuracy was calculated as 76.2%. Among the four input images, this is the lowest accuracy obtained. The verification parameter-II (57.6%) is also low. However, the average value of the geometry matching percentages of the successfully detected truth segments is still remarkably high 94%. The verification parameter-V (0.85) indicates that the under-segmentation exists in the segmentation procedure.

Figure 3.18-a illustrates the sub-field boundaries detected through the automated segmentation procedure. The SPOT_4_Intensity image with the new sub-field boundaries overlaid are shown in Figure 3.18-b.



a) Found Boundaries

b) Overlaid On SPOT_4_Intensity

Figure 3.18 – The Detected Field Boundaries and the Overlay on SPOT_4_ Intensity

Several sample outputs from the SPOT_4_Intensity image including the field image, the output of the Canny Edge Detector, and the results of the automated and the manual segmentation processes are given in Table A.4 in Appendix-A.

The numbers of the parcels according to the above mentioned errors types can be categorized as follows

A. Over-segmented Fields (43 Parcels)
B. Under-segmented Fields (95 Parcels)
C. The Field Geometry Errors on Equally-Segmented Fields (2% average geometry matching error on 154 parcels)

The main reasons for these errors as described for SPOT 5 images and SPOT_4_PCA image are also the same for the SPOT_4_Intensity image. The most common reason was the missing lines which were not detected by the edge detection.

3.3.5 Discussion of the Results

Four different images were processed using the proposed segmentation technique. The results and the accuracies were presented in the previous section. In this section, a comparative analysis between the results is given from the accuracy assessment perspective.

As a common fact for all the input datasets, it is seen that the accuracy of the automated segmentation is strongly related to the performance of the edge detection. Better results are obtained if the Canny edge detector appropriately detects the edges. The unsuccessful results cannot be avoided if the outputs of Canny operator either contains a high amount of noisy lines or does not contain the proper lines which may form the missing boundaries.

Table 3.14 provides a general overview for the accuracies of the four images by means of the verification parameters (VP). To compare the overall accuracies, it can be easily seen that the most successful outputs were obtained from the SPOT_5_PCA image. All the other accuracy parameters are also better for SPOT_5_PCA image than the others.

Table 3.14 – General Overview of the Assessed Accuracies for Each Input Image

	VP –I (Accuracy)	VP-II	V.P -III	V.P –IV	V.P -V
SPOT_5_PCA	**83.8%**	70.6%	94.6%	54.8%	1.02
SPOT_5_Intensity	**82.6%**	67.5%	94.6%	54.1%	1.01
SPOT_4_PCA	**78.8%**	61.5%	94.2%	52.1%	0.95
SPOT_4_Intensity	**76.2%**	57.6%	93.9%	49.3%	0.88

As a general comparison between the outputs of the SPOT 5 and the SPOT 4 images, it is very evident that better results were obtained from the SPOT 5 images. Since the SPOT_4 images have coarser spatial resolution and contain less information for the study area, the segmentation accuracies were not as high as SPOT_5. Due to the spatial resolution factor and the small sized fields the some of the perceptual grouping rules failed.

In both the SPOT_5 and SPOT_4 images, the PCA band combinations provided better results. This may be because all the bands were included in the applied principle component analysis, and therefore the spectral variability in all bands was reflected in the generated PCA images. For PCA images, the Canny edge detection algorithm provided better outputs for the missing boundary lines. However, the inclusion of the variability in all the bands brought more noise to the image. Therefore, this had a side affect as the over-segmentation for several fields in the PCA images. This effect can be easily observed by comparing the total number of the result segments for the outputs of the PCA and the Intensity images.

The Intensity images for both SPOT 5 and SPOT 4 had lower contrast making the images to be darker than the PCA images. Especially in darker fields, the segmentation did not give good results for the Intensity images. Thus, the edge detection could not supply the lines successfully for the missing boundaries. Considering the whole picture, it can be stated that the PCA band combination images provided slightly better outputs than the Intensity images.

If the overall outputs are analyzed considering the verification parameter-V, it can be stated that neither a significant over-segmentation nor a significant under-segmentation occurred in the outputs of the segmentation. However, only in in the SPOT_4_Intensity image, there was a noticeable under-segmentation (%15).

CHAPTER IV

CONCLUSION

Image segmentation has always been one of the most important steps leading the analysis of the image data. The pixel-based image analysis methods require significant overheads especially in the high resolution remote sensing data. In order to deal with each pixel distinctly, the grouping of the homogenous neighboring pixels as the segments and applying analyses on these segments (e.g. field-based classification) reduces the parameters and increase the efficiency of the analysis. However, the field-based analysis techniques need the boundary data of the homogeneous fields on which the analysis is performed. Image segmentation can be employed to detect these field boundaries.

Most segmentation techniques without any additional spatial knowledge provide unpredictable results. In many cases, they fail to collect all the boundaries that can be discriminated by the human eye. Also, the regions that a segmentation algorithm creates do not necessarily relate to meaningful entities, because the algorithms cannot take contextual information (region shape, occupation pattern) into account [2]. Hence, without including any additional topographic knowledge, the segmentation has to deal with so many parameters and specific cases and generally cannot provide accurate outputs.

This thesis proposed an automated segmentation process using the existing field boundaries of the fields and detecting the missing boundaries between the crops within these fields. FBISS (Field-Based Image Segmentation Software) was developed for implementing the proposed segmentation process as well as several analysis functionalities. The segmentation methodology was applied on a study area using the SPOT 5 (10 meter spatial resolution) and SPOT 4 (20 meter spatial resolution) satellite images. A part of Karacabey Plain, Turkey was selected as the study area and single band images were generated using the different band combinations of SPOT 5 and SPOT 4 data. For both satellite images, the first components of PCA outputs were converted to single band images after applying a linear enhancement operation (SPOT_5_PCA and SPOT_4_PCA images). Secondly the unweighed combination of Band 1, 2 and 3 of the images were converted to single band intensity images. (SPOT_5_Intensity and SPOT_4_Intensity images)

The segmentation process was carried out on these four images separately using the developed software. An evaluation methodology was defined which involves the matching percentages of the field geometries between the obtained result segments and the manually delineated segments. The accuracy assessment of the automated segmentation was performed using these parameters.

After evaluating the segmentation results, it was seen that the performance of the proposed segmentation technique is strongly correlate with the performance of the edge detection. Better results are obtained where Canny provided appropriate lines as detected edges. On the other hand, unsuccessful results are obtained where Canny output either contained a lot of noisy edges or did not contain the proper edges which may form the missing boundaries.

After applying a comparative analysis, it is evident that a better performance is achieved for the segmentation outputs of the SPOT 5 images due to the higher resolution with respect to SPOT 4 images. In addition, the outputs for the PCA images that contain the spectral variability information for all bands resulted with better accuracy assessment values through the applied analyses. The PCA images contained higher contrasts and sharper transitions between the crop fields which caused better results by the edge detection. The most accurate segmentation results were obtained from the SPOT_5_PCA image.

Considering the whole results, it can be stated that the proposed segmentation strategy gives better outputs for the higher resolution images when compared to the low resolution images. Since SPOT 4 images have 20 meters resolution, less number of pixels fall within the fields. And even though the thresholds and edge detection parameters are adapted considering the sizes of the fields, is evident that the perceptual grouping rules did not perform for the SPOT 4 images as well as they worked for SPOT 5 images. The size issue also affects the Canny outputs, which is an important factor for the accuracy of the segmentation. In fact, a coarse resolution means that the image contains less information. Thus, obtaining better results for the SPOT 5 images when compared to the SPOT 4 images are reasonable.

As a general performance evaluation, the proposed segmentation process provided almost $80\% \pm 5$ percent accurate outputs for each of the SPOT 4 and SPOT 5 images, which is an acceptable performance. In image processing studies, a complete performance is almost impossible to acquire since the input domain is too heterogeneous due to the noise and illumination effects. The noise, the erroneously detected edges within the fields, and the erroneously modified line segments through the perceptual grouping rules can be said to be the common reasons for the over-segmented outputs. The missing lines that could not be detected by the Canny edge detector, the erroneously deleted line segments through the perceptual grouping rules, and the erroneously merged sub-fields were the common reasons for the under-segmented outputs. In addition, the conversion of the multi-spectral satellite images to a single band image means loss of some information. And this loss had significant affects on the accuracy of the segmentation.

Recommendations and Future Work

Applying an image segmentation operation before the analysis of an image is getting very popular in the current studies for image processing and remote sensing. As an example, object based classification which deals with the classification of the fields, not with the pixels is one of the very hot topics. It is proven with many studies and experiments that the field-based classification approaches have significantly better outputs with respect to the pixel-based approaches [3][7][12]. Determining the homogenous fields to apply classification is actually the problem of image segmentation. The outputs of this study can be used in classification, and the proposed segmentation process can be integrated with a classification procedure in order to realize an automated object based classification system. The software implemented in the concept of this thesis is based on a design considering such expandability issues and this kind of classification functionalities. Actually a strong infrastructure is implemented using OpenCV libraries, so that the software can be improved to a more capable image processing toolbox.

Also the proposed segmentation process based on the edge detection and the perceptual grouping rules may be improved considering the multispectral properties of the images. Since it is an edge based segmentation process, the accuracy of the segmentation is directly related with the edge detection outputs. The used edge detection algorithm Canny can only perform on single band images, therefore single band input images are prepared by combining different bands. Although Canny is one of the most powerful edge detection algorithms, a multispectral approach can be used and information in all bands can be used to detect edges without any information loss [44].

Additionally the perceptual grouping rules may be improved considering the context of the fields. In the proposed method, the spectral domain is left with vectorization process after the detection of edges. And all the additional operations (perceptual grouping, polygon construction and merging) are applied on the vector domain without using any spectral content of the fields. Including the spectral properties of the fields to these operations may significantly improve the accuracy of the segmentation. The decision mechanisms for determining a detected line whether it is a missing boundary or not, can be supported with the spectral content in the neighboring pixels of the line. If a major spectral difference exists, the line can be decided as a boundary candidate, and vice versa. Also, in the merging procedure for the constructed polygons, the decision for merging can be supported again with the spectral contents of the fields. The spectral content of the segmented polygons can be compared with their adjacent polygons and those, having similar spectral characteristics may be merged. This operation is generally performed in the classification process in similar studies [21]. No merging is applied in the segmentation phase and the over-segmented areas that are assigned to the same classes are merged after the classification. No classification capability was considered in this study. Therefore, merging is needed to be applied in the segmentation phase.

As the last words for the conclusion, it can be stated that including the topographical maps in the image segmentation procedure and performing a field-based process seems to be the most promising way to determine the homogenous crop fields accurately. And the proposed segmentation strategy is actually a good start point for the development of high performance field based image analysis process including both the segmentation and classification procedures.

REFERENCES

[1] A. H. Bhalerao, "Multiresolution Image Segmentation", PhD Thesis, The University of Warwick, November 1991.

[2] A. J. W. De Wit and J. G. P. W. Clevers, "Efficiency and Accuracy of Per-Field Classification for Operational Crop Mapping", International Journal of Remote Sensing, 25(20): 4091-4112, October 2004.

[3] A. M. Dean and G. M. Smith, "An Evaluation of Per-parcel Land Cover Mapping Using Maximum Likelihood Class Probabilities", International Journal of Remote Sensing, 24(14):2905-2920, 2003.

[4] A. Rosenfeld and A. Kak, "Digital Picture Processing", Text Book, 2nd edition, Academic Press, New York, 1982.

[5] A. Rydeberg and G. Borgefors, "Integrated Method for Boundary Delineation of Agriculture Fields In Multispectral Satellite Images", IEEE Transactions on Geoscience and Remote Sensing, 39(11):2514-2520, May 2002.

[6] A. Özdarıcı, "Accuracy Assessment of the Field-Based Classification Using the Different Spatial Resolution Images", On-going Thesis Study, GGIT, METU, May 2005.

[7] B. Tso and P. M. Mather, "Crop Discrimination Using Multi-temporal SAR Imagery", International Journal of Remote Sensing, 20(12):2443-2460, 1999.

[8] C. Evans, R. Jones, I. Svalbe and M. Berman, "Segmenting Multispectral Landsat TM Images Into Field Units", IEEE Transactions on Geoscience and Remote Sensing, 40(5):1054-1064, May 2002.

[9] D.C. Mason, D.G. Orr, A. Cross, D.C. Hogg and D.H. Lawrence, “The Use of Digital Map Data in the Segmentation and Classification of Remotely Sensed Images”, International Journal of Geographical Information Systems, 2(3):195-215, 1988.

[10] D. Dori, and W. Liu, “Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(3): 202-215, 1999.

[11] D. P. Bertsekas, “The Auction Algorithm for Assignment and Other Network Problems: A Tutorial”, Interfaces, 20(4):33-149, 1990.

[12] G. M. Smith and R. M. Fuller “An Integrated Approach To Land Cover Classification: an Island of Jersey”, International Journal of Geographical Information Systems, 22(16):3123-3142, 2001.

[13] J. Hershberger and J. Snoeyink, “Speeding Up the Douglas-Peucker Line-Simplification Algorithm”, Technical Report, In Proc. 5th Intl. Symp. Spatial Data Handling, 134-143, 1992.

[14] J. Canny, “A Computational Approach to Edge Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986.

[15] J. C. Tilton. “Method for recursive hierarchical segmentation by region growing and spectral clustering with a natural convergence criterion”, Disclosure of Invention and New Technology: NASA Case No. GSC 14,328-1, 2004.

[16] J. Song, F. Su, J. Chen, C. Tai, and S. Cai, “Line Net Global Vectorization: an Algorithm and Its Performance Evaluation”, Computer Vision and Pattern Recognition, IEEE Conference, Vol.1, 383-388, June 2000.

[17] J. Ton, J. Sticklen and A.K. Jain, “Knowledge-Based Segmentation of Landsat Images”, IEEE Transactions on Geoscience and Remote Sensing, 29(2):222-232, March 1991.

[18] K. Johnsson, “Segment-Based Land-Use Classification from SPOT Satellite Data”, Photogrammetric Engineering and Remote Sensing, 60(1):47-53, 1994.

[19] K. S. Fu, and J. K. Mui, “A survey of image segmentation”, Pattern Recognition, 1981.

[20] L. Hermes and J. M. Buhmann, “Boundary-Constrained Agglomerative Segmentation”, IEEE Transactions on Geoscience and Remote Sensing, 42(9):1984-1995, September 2004.

[21] L. L. F. Janssen and M. Molenaar, “Terrain Objects, Their Dynamics, and Their Monitoring by The Integration of GIS and Remote Sensing”, IEEE Transactions On Geoscience and Remote Sensing, 33(3): 749- 758, May 1995.

[22] L. Roberts, “Machine Perception of 3-Dimensional Solids”, Optical and Electrooptical Information Processing. MIT Press, Cambridge MA, 1965.

[23] L. Smith, “A tutorial on Principal Component Analysis”, Student Tutorial, University of Otago, 2002.

[24] M. Amadasun and R.A. King, “Low-level Segmentation of Multispectral Images via Agglomerative Clustering of Uniform Neighborhoods”, Pattern Recognition, 21(3):261-268, 1988.

[25] M. I. Skolnik, “Radar Handbook”, Text Book, Mc-Graw Hill Publishing Company, 2nd Edition, 1990.

[26] M. J. P. M. Lemmens and C. S. L. A. Han, “Automized Thematic Interface of Digital Spatial Imagery and Geographical Information Systems”, Int. Arch. Photogrammetry Remote Sensing, 28:395-404, 1990.

[27] M. Turker and E. Derenyi, “GIS assisted change detection using remote sensing. Geocarto International”, 15(1):49-54, 2000.

[28] M. Turker and M. Arikan, “Field-Based Crop Mapping Through Sequential Masking Classification of Multi-Temporal Landsat-7-ETM Images in Karacabey, Turkey”, Conference Paper, ISPRS Istanbul, 2004.

[29] M. Spann and C. Horne. “Image Segmentation Using a Dynamic Thresholding Pyramid”, Pattern Recognition, 22(6):729-732, 1989.

[30] M. Spann and R. G. Wilson. “A Quad-Tree Approach to Image Segmentation Which Combines Statistical and Spatial Information”, Pattern Recognition, 18(3/4):257-269, 1985.

[31] N. R. Pal and S. K. Pal, “A Review on Image Segmentation Techniques”, Pattern Recognition, 1993.

[32] P. Aplin and P. M. Atkinson, “Predicting Missing Field Boundaries to Increase Per-Field Classification Accuracy”, Photogrammetric Engineering and Remote Sensing, 70(1):141-149, January 2004.

[33] P.C. Chen and T. Pavlidis, “Image Segmentation as an Estimation Problem”, Computer Graphics and Image Processing, 12:153-172, 1980.

[34] P. J. Burt, T. H. Hong, and A. Rosenfeld. “Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation”, IEEE Transactions on. System, Man and Cybernetics, 11(12):802-809, 1981.

[35] R. Adams and L. Bischof, “Seeded Region Growing”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6):641-647, June 1994.

[36] R. C. Gonzalez, R. E. Woods, “Digital Image Processing”, Text Book, Prentice Hall, 2nd edition, January 2002.

[37] R. O. Duda, P. E. Hart and D. G. Stork, “Pattern Classification”, Text Book, Wiley-Interscience, 2nd edition, October 2000.

[38] R. Wilson and M. Spann. "A New Approach To Clustering", Pattern Recognition, 23(12):1413-1425, 1990.

[39] S. Suzuki, "Graph-based Vectorization Method for Line Patterns", Computer Vision and Pattern Recognition, Computer Society Conference, 616-621, 1988.

[40] S. D. Zenzo, L. Cinque, and S, Levialdi. "Run-Based Algorithms for Binary Image Analysis and Processing". IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(1):83-89, 1996.

[41] S. W. Zucker. "Region Growing: Childhood and Adolescene", Computer Graphic and Image Processing, 5:382-399, 1976.

[42] T. H. Hong and A. Rosenfeld, "Compact Region Extraction Using Weighted Pixel Linking in a Pyramid", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2(12):222-229, 1984.

[43] T. H. Hong and M. Shneier, "Extracting Compact Objects Using Linked Pyramids", IEEE Transactions on Pattern Analysis and Machine Intelligence, 6:229-237, 1984.

[44] T. Hoffman and J. Boehner, "Spatial Recognition by Means of Represent-ativeness Measures", Conference Paper, IEEE 6/99, 1999.

[45] T. Pavlidis and Y. Liow, "Integrating Region Growing and Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990.

[46] X. Cufi, X. Munoz, J. Freixent and J.A. Marti, "A Review on Image Segmentation Techniques Integrating Region and Boundary Information", Advances in Imaging and Electron Physics, Academic Press, 2002.

[47] X. Cufi, X. Munoz, J. Freixent and J.A. Marti. "Strategies for Image Segmentation Combining Region and Boundary Information", Pattern Recognition Letters, January 2003.

Online References:

[48] Open Computer Vison Library Home Page, Last Date Accessed:18/04/2005, Owner: Sourceforge.net, Url: http://sourceforge.net/projects/opencvlibrary/

[49] SPOT Image, S.A. Home Page, Last Date Accessed: 18/04/2005, Owner: SPOT Image S.A, Url: http://www.spot.com

APPENDIX

A. Sample Results

Several sample results for each input dataset used in this study is given in this section. The results include the field images, the Canny Operator outputs, the automated segmentation and the manual segmentation outputs for several fields with the given ID numbers.

Table A.1 – The Sample Outputs for the SPOT_5_PCA Image

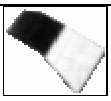
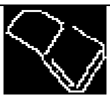
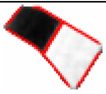
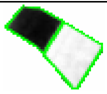
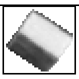
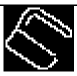
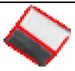
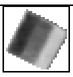
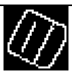
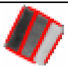
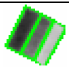
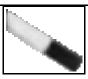
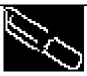
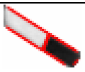
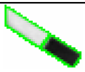
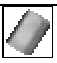
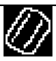
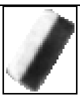
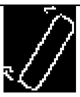
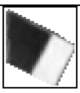
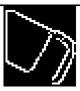
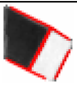
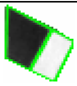
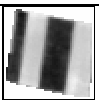
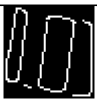
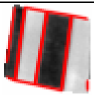
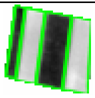
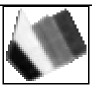
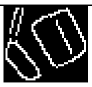
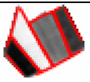
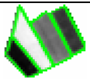
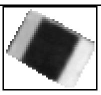
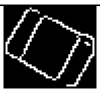
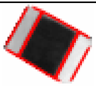
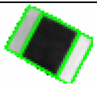
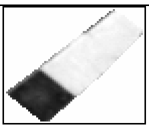
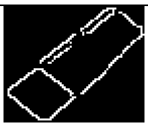
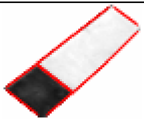
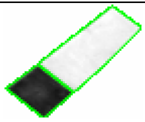
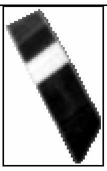
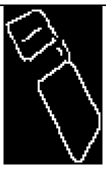
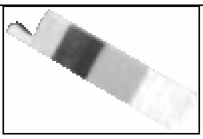
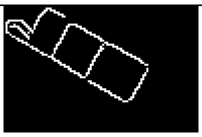
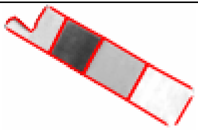
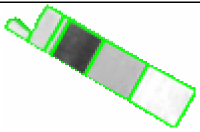
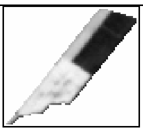
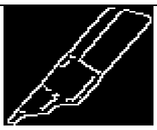
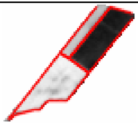
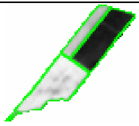
#	Field	Canny	Result	Truth	ID
1					212020
2					520
3					1620
4					620
5					1310
6					212090
7					212050
8					212400
9					212130
10					212160
11					212040
12					2480
13					2320
14					212370

Table A.1 – The Sample Outputs for the SPOT_5_PCA Image Continued

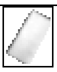
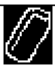
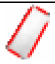
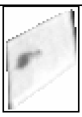
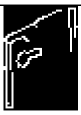
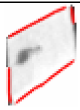
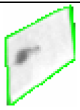
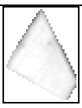
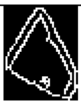
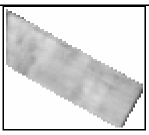
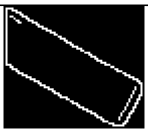
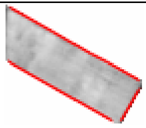
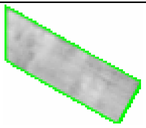
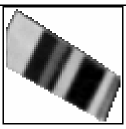
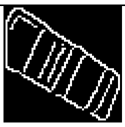
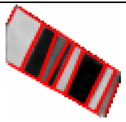
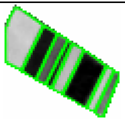
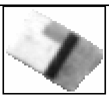
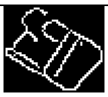
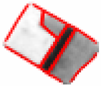
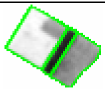
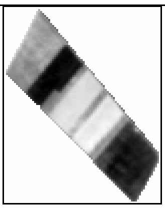
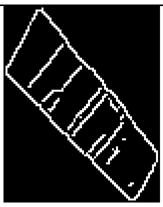
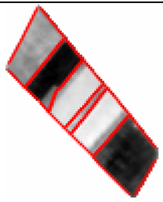
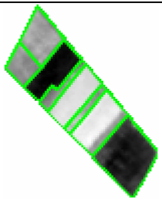
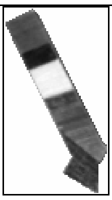
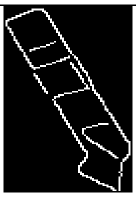
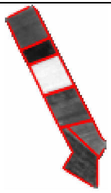
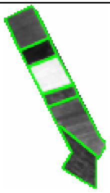
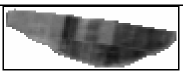
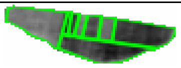
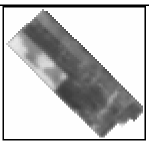
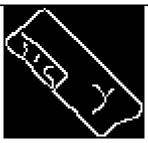
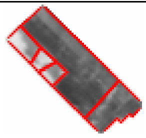
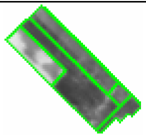
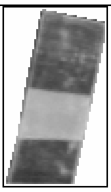
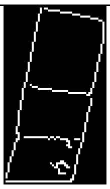
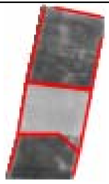
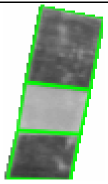
#	**Field**	**Canny**	**Result**	**Manual**	**ID**
15					39450
16					5330
17					2400
18					1910
19					70320
20					3080
21					6900
22					2410
23					10
24					2590
25					4310

Table A.1 – The Sample Outputs for the SPOT_5_PCA Image Continued

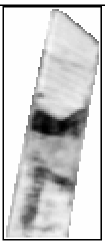
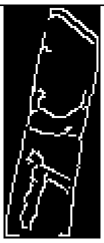
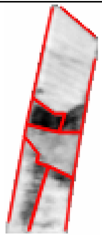
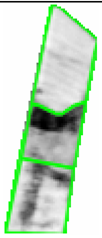
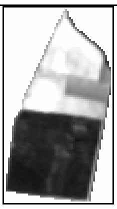
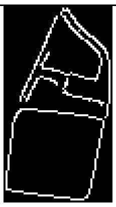
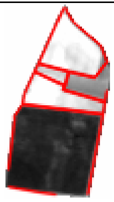
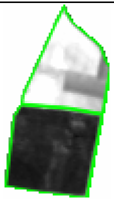
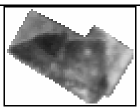
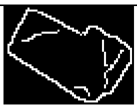
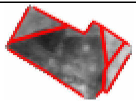
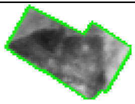
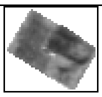
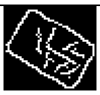
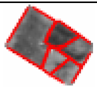
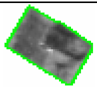
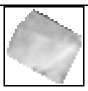
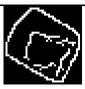
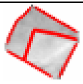
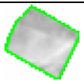
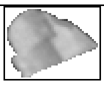
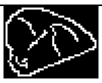
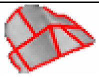
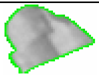
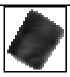
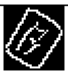
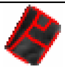
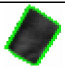
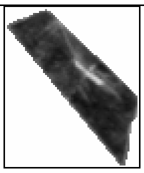
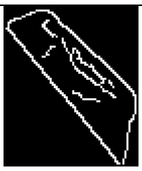
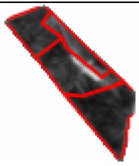
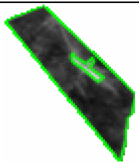
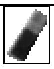
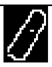
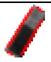
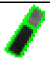
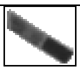
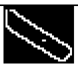
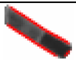
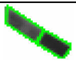
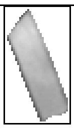
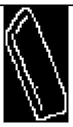
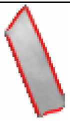
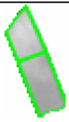
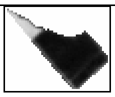
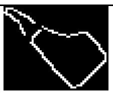
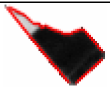
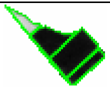
#	Field	Canny	Result	Manual	ID
26					4330
27					20
28					43690
29					43700
30					1580
31					4760
32					3520
33					4520
34					2940
35					640
36					5120
37					6940

Table A.2 – The Sample Outputs for the SPOT_5_Intensity Image

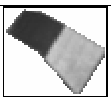
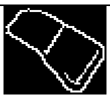
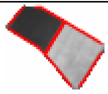
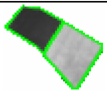
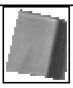
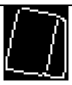
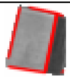
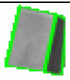
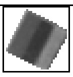
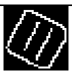
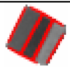
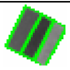
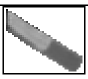
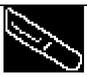
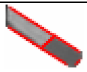
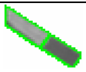
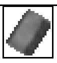
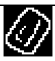
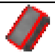
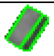
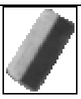
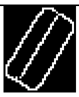
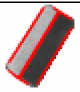
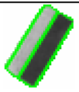
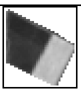
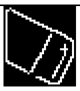
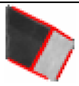
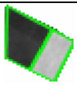
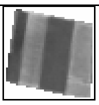
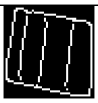
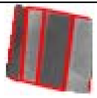
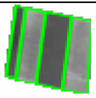
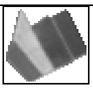
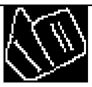
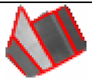
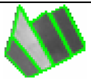
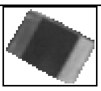
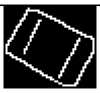
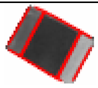
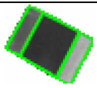
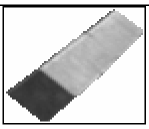
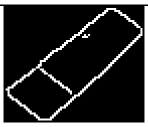
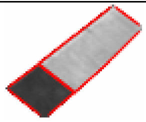
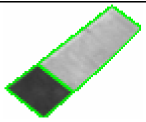
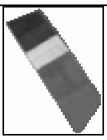
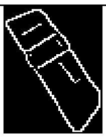
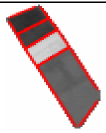
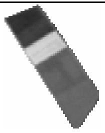
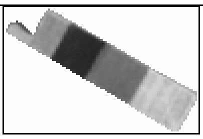
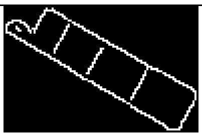
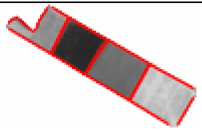
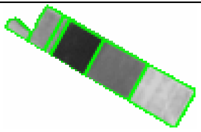
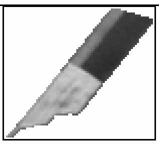
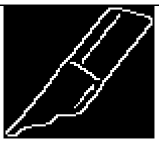
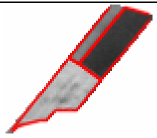
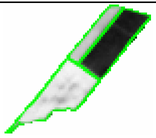
#	Field	Canny	Result	Truth	ID
1					212020
2					212510
3					1620
4					620
5					1310
6					212090
7					212050
8					212400
9					212130
10					212160
11					212040
12					2480
13					2320
14					212370

Table A.2 – The Sample Outputs for the SPOT_5_Intensity Image Continued

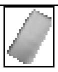
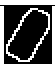
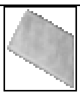
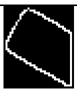
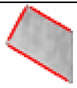
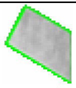
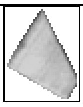
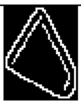
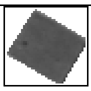
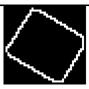
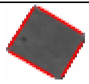
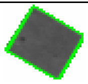
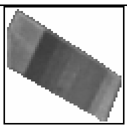
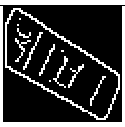
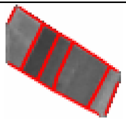
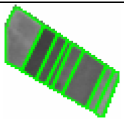
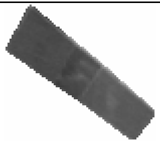
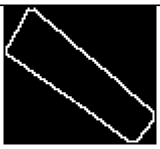
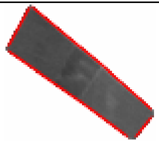
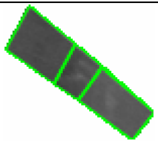
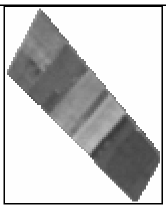
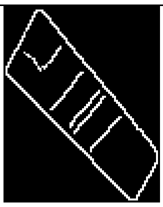
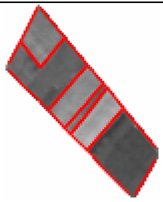
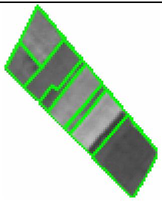
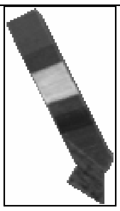
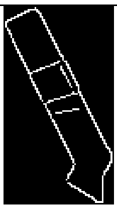
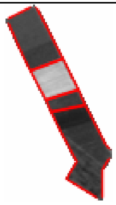
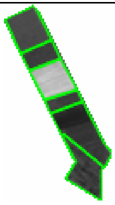
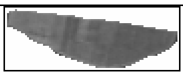
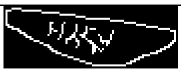
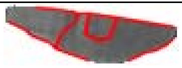
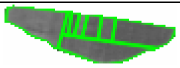
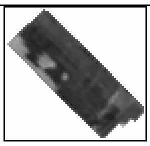
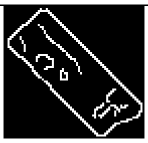
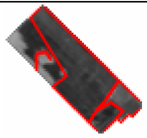
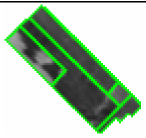
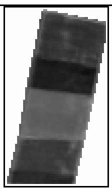
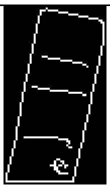
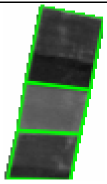
#	Field	Canny	Result	Truth	ID
15					39450
16					2330
17					2400
18					2870
19					70320
20					2490
21					6900
22					2410
23					10
24					2590
25					4310

Table A.2 – The Sample Outputs for the SPOT_5_Intensity Image Continued

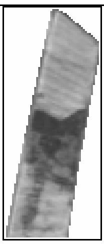
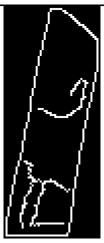
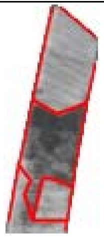
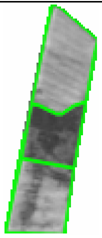
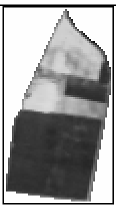
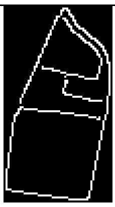
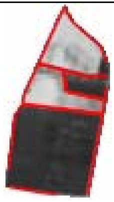
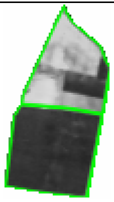
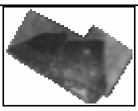
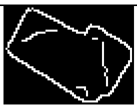
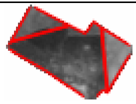
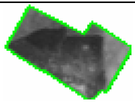
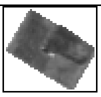
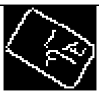
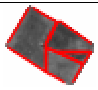
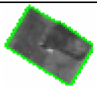
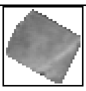
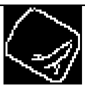
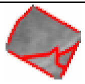
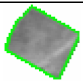
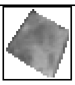
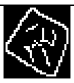
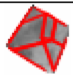
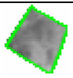
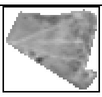
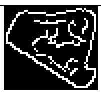
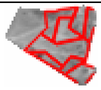
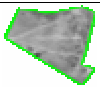
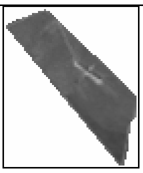
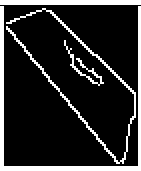
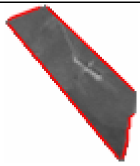
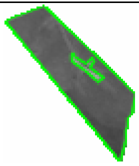
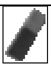
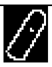
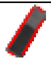
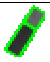
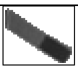
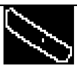
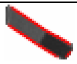
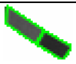
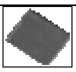
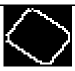
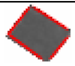
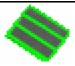
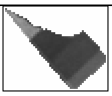
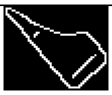
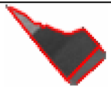
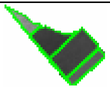
#	Field	Canny	Result	Truth	ID
26					4330
27					20
28					43690
29					43700
30					1580
31					1400
32					1420
33					4520
34					2940
35					640
36					5120
37					6940

Table A.3 – The Sample Outputs for the SPOT_4_PCA Image

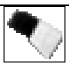
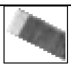
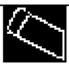
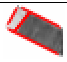
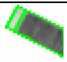
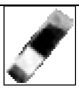
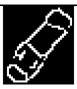
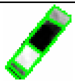
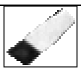
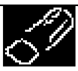
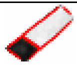
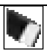
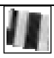
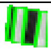
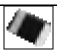
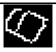
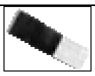
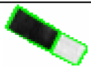
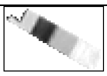
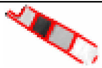
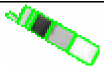
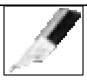
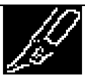
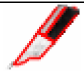
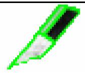
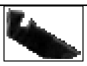
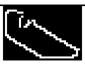
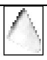
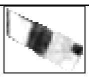
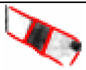
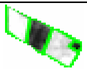
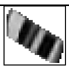
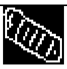
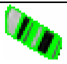
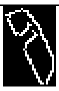
#	Field	Canny	Result	Truth	ID
1					212020
2					520
3					212510
4					212450
5					212150
6					212090
7					212040
8					212050
9					212400
10					212130
11					212160
12					2290
13					2320
14					212370
15					39450
16					3410
17					2400
18					2140
19					70320
20					2480

Table A.3 – The Sample Outputs for the SPOT_4_PCA Image Continued

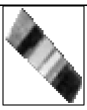
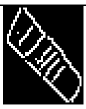
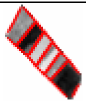
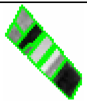
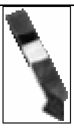
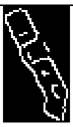
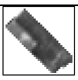
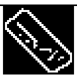
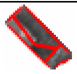
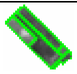
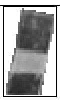
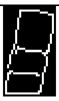
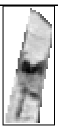
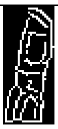
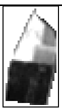
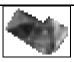
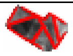
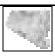
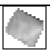
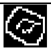
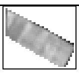
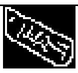
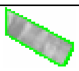
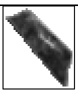
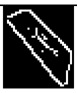
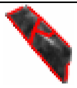
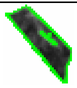
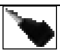
#	Field	Canny	Result	Truth	ID
21					6900
22					2410
23					10
24					2590
25					4310
26					4330
27					20
28					43690
29					1420
30					1580
31					4760
32					1910
33					4520
34					620
35					6940
36					212460
37					510

Table A.4 – The Sample Outputs for the SPOT_5_Intensity Image

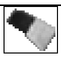
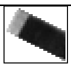
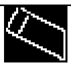
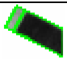
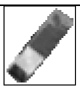
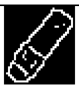
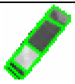
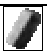
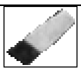
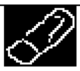
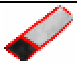
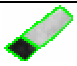
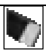
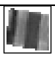
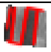
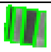
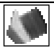
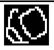
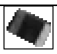
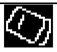
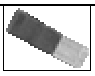
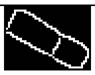
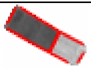
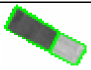
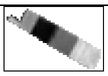
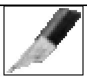
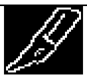
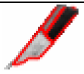
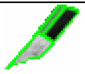
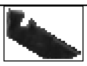
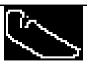
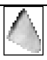
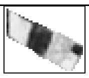
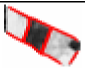
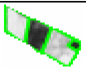
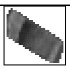
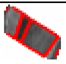
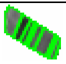
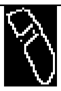
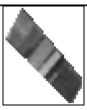
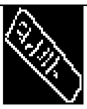
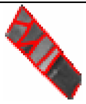
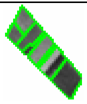
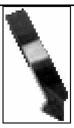
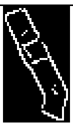
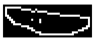
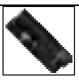
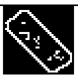
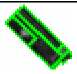
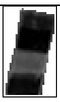
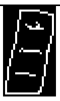
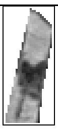
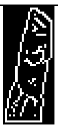
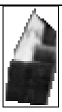
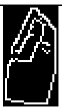
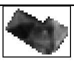
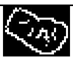
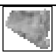
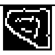
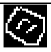
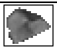
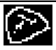
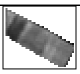
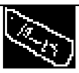
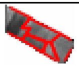
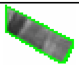
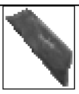
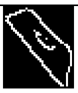
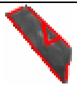
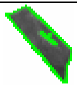
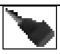
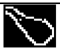
#	Field	Canny	Result	Truth	ID
1					212020
2					520
3					212510
4					212450
5					212150
6					212090
7					212040
8					212050
9					212400
10					212130
11					212160
12					2290
13					2320
14					212370
15					39450
16					3410
17					2400
18					2140
19					70320
20					2480

Table A.4 – The Sample Outputs for the SPOT_5_Intensity Image Continued

#	Field	Canny	Result	Truth	ID
21					6900
22					2410
23					10
24					2590
25					4310
26					4330
27					20
28					43690
29					1420
30					1580
31					4760
32					1910
33					4520
34					620
35					6940
36					212460
37					510