

**A METHODOLOGY FOR DETECTION AND EVALUATION OF LINEAMENTS
FROM SATELLITE IMAGERY**

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

BY

ARMAN KOÇAL

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MINING ENGINEERING**

JULY 2004

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.

Prof. Dr. Ümit Atalay
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.

Assist. Prof. Dr. H. Şebnem Düzgün
Co-Supervisor

Prof. Dr. Celal Karpuz
Supervisor

Examining Committee Members

Assoc. Prof. Dr. H. Aydın Bilgin	(METU, MINE)	_____
Prof. Dr. Celal Karpuz	(METU, MINE)	_____
Assist. Prof. Dr. H. Şebnem Düzgün	(METU, GGIT)	_____
Assoc. Prof. Dr. Vedat Toprak	(METU, GEOE)	_____
Assist. Prof. Dr. Mehmet Lütfi Süzen	(METU, GEOE)	_____

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 : Arman Koal

Signature :

ABSTRACT

A METHODOLOGY FOR DETECTION AND EVALUATION OF LINEAMENTS FROM SATELLITE IMAGERY

Koçal, Arman

M.Sc., Department of Mining Engineering

Supervisor: Prof. Dr. Celal Karpuz

Co-supervisor: Assist. Prof. Dr. H. Şebnem Düzgün

June 2004, 122 pages

The discontinuities play an important role both in design and development stages of many geotechnical engineering projects. Because of that considerable time and capital should be spent to determine discontinuity sets by conventional methods. This thesis present the results of the studies associated with the application of the Remote Sensing (RS) and the development of a methodology in accurately and automatically detecting the discontinuity sets.

For detection of the discontinuities, automatic lineament analysis is performed by using high resolution satellite imagery for identification of rock discontinuities. The study area is selected as an Andesite quarry area in Gölbaşı, Ankara, Turkey. For the high resolution data 8-bit Ikonos Precision Plus with 1 meter resolution orthorectified image is used. The automatic lineament extraction process is carried out with LINE module of PCI Geomatica v8.2. In order to determine the most accurate parameters of LINE, an accuracy assessment is carried out. To be the reference of the output, manual lineament extraction with directional

filtering in four principal directions (N-S, E-W, NE-SW, NW-SE) is found to be the most suitable method. For the comparison of automatic lineament extraction and manual lineament extraction processes, LINECOMP program is coded in java environment. With the written code, a location and length based accuracy assessment is carried out. After the accuracy assessment, final parameters of automatically extracted lineaments for rock discontinuity mapping for the study area are determined. Besides these, field studies carried out in the study area are also taken into consideration.

Key words: Remote Sensing (RS), High Resolution Satellite Image, Automatic Lineament Analysis, Accuracy Assessment

ÖZ

UYDU GÖRÜNTÜSÜNDEN ÇİZGİSELLİKLERİN BELİRLENMESİNE VE DEĞERLENDİRMESİNE DAİR BİR YÖNTEM

Koçal, Arman

Yüksek Lisans, Maden Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Celal Karpuz

Yardımcı Tez Yöneticisi: Yrd. Doç. Dr. H. Şebnem Düzgün

Temmuz 2004, 122 sayfa

Jeoteknik ile ilgili mühendislik uygulamalarında süreksizliklerin hem tasarım hemde işletme aşamasında büyük önemi vardır. Fakat geleneksel yöntemlerle süreksizliklerin belirlenmesi zaman alıcı olabilir ve ekonomik olmayabilir. Bu tezde, Uzaktan Algılama (UA) uygulamaları ile süreksizliklerin otomatik ve doğru olarak belirlenmesine için bir yöntem geliştirilmesine dair çalışmaların sonuçları sunulmuştur.

Süreksizliklerin belirlenmesinde yüksek çözünürlüklü uydu görüntüsüne kaya süreksizliklerini belirlemek için otomatik çizgisellik analizi uygulanmıştır. Çalışma sahası olarak Türkiye, Ankara, Gölbaşı andezit ocakları sahası seçilmiştir. Yüksek çözünürlüklü veri olarak 1 metre çözünürlüklü 8-bit Ikonos Precision Plus ortorektifiye görüntü kullanılmıştır. Otomatik çizgisellik ayıklama işlemi PCI Geomatica v8.2 programında LINE seçeneği kullanılarak gerçekleştirilmiştir. LINE opsiyonunun en uygun değişkenlerini belirlemek için doğruluk analizi yapılmıştır. Sonuca referans olması için 4 ana yönde (K-G, D-B, KD-GB, KB-GD) yönlü filtre kullanarak el ile çizgisellik haritası oluşturmak en uygun yol olarak belirlenmiştir.

El ile ve otomatik olarak oluşturulan çizgisellik haritalarını karşılaştırmak için java ortamında LINECOMP adlı program kodlanmıştır. Yazılan kodla, yer ve uzunluk esaslı doğruluk analizi yapılmıştır. Doğruluk analizinden sonra LINE opsiyonunun en uygun değişkenleri çalışma sahası için belirlenmiştir. Bunların yanında, çalışma sahasında yürütülen arazi çalışmaları da dikkate alınmıştır.

Anahtar kelimeler: Uzaktan Algılama (UA), Yüksek Çözünürlüklü Uydu Görüntüsü, Otomatik Çizgisellik Analizi, Doğruluk Analizi

To My Family

ACKNOWLEDGEMENTS

I would like to thank to the following persons and institutions for their support and advice during the course of this research.

My supervisor, Prof. Dr. Celal Karpuz and Co-supervisor Assist. Prof. Dr. H. Şebnem Düzgün for their kind supervision, valuable comments, suggestions and friendship throughout this research.

Assoc. Prof. Dr. H. Aydın Bilgin, Assoc. Prof. Dr. Vedat Toprak and Assist. Prof. Dr. Mehmet Lütfi Süzen for their suggestions, comments, guidance and valuable contributions during various stages of the research and/or for serving on the M.Sc. Thesis committee.

Assoc. Prof. Dr. Yusuf Kağan Kadioğlu for providing the geological map of the area of research.

Research Assistants Deniz Gerçek and Çağıl Kolat for their valuable help in Geological Engineering RS/GIS Laboratory.

METU Research Fund for providing the financial support under the project BAP 2003-03-05-01.

Perit Çakır and Murat Hazer for their valuable contribution and help in writing the Java code and Ömer Can Özcan, Ayşe Yasemin Yeşilay, Ayşe Alpagut, Onat Başbay, Emre Altun, Koray Önal and Reza R. Osgoui for their valuable contribution and help during various stages of this study.

Finally, but most deeply, I am indebted to all members of my great family for their continual encouragement in every stage of the entire study.

TABLE OF CONTENTS

PLAGIARISM.....	İii
ABSTRACT.....	iv
ÖZ.....	vi
DEDICATION.....	viii
ACKNOWLEDGEMENTS.....	ix
TABLE OF CONTENTS.....	xi
LIST OF TABLES.....	xiv
LIST OF FIGURES.....	xv
CHAPTER	
I. INTRODUCTION.....	1
1.1. General Remarks.....	1
1.2. Objectives and Scope of the Thesis.....	2
1.3. Thesis Outline.....	3
II. BASIC CONCEPTS ABOUT THE LINEAMENT ANALYSIS.....	4
2.1. Introduction.....	4
2.2. Lineament Analysis from Satellite Imagery.....	5
2.2.1. Spatial Convolution Filtering.....	6
2.2.2. Multiband Analysis.....	16
2.2.3. Automatic Lineament Extraction.....	19
2.3. Selection of Pixel Size.....	23
III. DESCRIPTION OF THE STUDY AREA AND DATA.....	24
3.1. Description of the Study Area.....	24

3.2. Description of Data.....	26
IV. EXTRACTION OF LINEAMENTS	29
4.1. Introduction.....	29
4.2. Automatic Lineament Extraction.....	31
4.2.1. Details of LINE.....	31
4.2.2. Algorithm of LINE.....	32
4.2.3. Results of Automatic Lineament Extraction.....	34
4.3. Manual Extraction of Lineaments.....	35
4.3.1. Image Smoothing.....	36
4.3.2. Directional Filtering.....	37
4.3.3. Digitizing the Lineaments.....	39
V. ACCURACY ASSESSMENT.....	42
5.1. Introduction.....	42
5.2. Directional Accuracy.....	43
5.3. Location and Length Based Accuracy.....	52
5.3.1. Algorithm of the Methodology.....	53
5.3.1.1. Algorithm of LINE ADDRESS.....	55
5.3.1.2. Algorithm of LINE COMPARE.....	56
5.3.1.3. Algorithm of LINE CLASSIFICATION.....	59
5.4. Discussion of the Results.....	68
VI. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY.....	70
6.1. Conclusions.....	70
6.2. Recommendations for Further Study.....	71
REFERENCES.....	72
APPENDICES	
A.....	77
B.....	92
C.....	97

D.....	102
E.....	107
F.....	110
G.....	113

LIST OF TABLES

TABLE

3.1. Corner Coordinates of the Study Area.....	25
4.1. Parameters, Total Length and Number of Lineaments Formed By Automatic Extraction.....	35
4.2. Sobel Filter Kernels.....	37
5.1. Sample Output of LINE COMPARE.....	56
5.2. Number of Lineaments Classified According to Matching Pixels with Tolerance Value 1.....	60
5.3. Matching Percentages of LINE 2 with Different Tolerance Values.....	59

LIST OF FIGURES

FIGURE

2.1.	Compass Gradient Masks of Pratt and Jain.....	9
2.2.	The Applied Filter Values by Mat Akhir et al. (1997).....	13
2.3.	Simple Illustration of Hough Transform.....	20
2.4.	Algorithm of Haar Transform.....	21
3.1.	Location Map of the Study Area.....	25
3.2.	Lithological Map of the Study Area.....	26
3.3.	Pansharpened True Color Composite Image of the Study Area.....	28
4.1.	Flowsheet of the Study.....	30
4.2.	Parameter Entry Screen of Automatic Extraction Process.....	34
4.3.	Filtered Image in N-S Direction.....	37
4.4.	Filtered Image in NW-SE Direction.....	38
4.5.	Filtered Image in E-W Direction.....	38
4.6.	Filtered Image in NE-SW Direction.....	39
4.7.	The Manually Extracted Lineament Map of the Area of Interest.....	40
5.1.	Simple Illustration of Node to Node Method.....	43
5.2.	Rose Diagram of LINE 1.....	44
5.3.	Rose Diagram of LINE 2.....	44
5.4.	Rose Diagram of LINE 5.....	45
5.5.	Rose Diagram of LINE 16.....	45
5.6.	Rose Diagram of LINE 19.....	46
5.7.	Rose Diagram of LINE 27.....	46

5.8.	Rose Diagram of LINE 29.....	47
5.9.	Rose Diagram of Manually Extracted Lineaments.....	47
5.10.	Pole Distribution of the Gölbaşı District.....	48
5.11.	Raw Image of the Quarry Face.....	50
5.12.	Lineaments of the Face.....	50
5.13.	Near Face Lineaments.....	52
5.14.	Algorithm of LINECOMP.....	54
5.15.	Simple Illustration of Intersecting Lineaments in LINE ADDRESS.....	55
5.16.	Illustration of Matching of Divided Lineaments.....	57
5.17.	Sample Illustration of Matching of Tolerance Value in LINECOMPARE Algorithm.....	58
5.18.	Assigned Colors for Pixel Values.....	59
5.19.	Sample Classified Map	60
5.20.	True Color Composite Image Showing Zones A, B and C.....	63
5.21.	Classified Image of LINE 2 in Zone A with Tolerance Value = 0.....	64
5.22.	Classified Image of LINE 2 in Zone A with Tolerance Value = 1.....	64
5.23.	Classified Image of LINE 2 in Zone A with Tolerance Value = 2.....	65
5.24.	Classified Image of LINE 2 in Zone A with Tolerance Value = 3.....	65
5.25.	Classified Image of LINE 2 in Zone A with Tolerance Value = 4.....	66
5.26.	Classified Image of LINE 2 in Zone A with Tolerance Value = 5.....	66
5.27.	Classified Image of LINE 2 in Zone A with Tolerance Value = 6.....	67
5.28.	Classified Image of LINE 2 in Zone A with Tolerance Value = 7.....	67
A1.	Lineament Map of LINE 1.....	77
A2.	Lineament Map of LINE 2.....	78
A3.	Lineament Map of LINE 3.....	78
A4.	Lineament Map of LINE 4.....	79
A5.	Lineament Map of LINE 5.....	79
A6.	Lineament Map of LINE 6.....	80
A7.	Lineament Map of LINE 7.....	80

A8.	Lineament Map of LINE 8.....	81
A9.	Lineament Map of LINE 9.....	81
A10.	Lineament Map of LINE 10.....	82
A11.	Lineament Map of LINE 11.....	82
A12.	Lineament Map of LINE 12.....	83
A13.	Lineament Map of LINE 13.....	83
A14.	Lineament Map of LINE 14.....	84
A15.	Lineament Map of LINE 15.....	84
A16.	Lineament Map of LINE 16.....	85
A17.	Lineament Map of LINE 17.....	85
A18.	Lineament Map of LINE 18.....	86
A19.	Lineament Map of LINE 19.....	86
A20.	Lineament Map of LINE 20.....	87
A21.	Lineament Map of LINE 21.....	87
A22.	Lineament Map of LINE 22.....	88
A23.	Lineament Map of LINE 23.....	88
A24.	Lineament Map of LINE 24.....	89
A25.	Lineament Map of LINE 25.....	89
A26.	Lineament Map of LINE 26.....	90
A27.	Lineament Map of LINE 27.....	90
A28.	Lineament Map of LINE 28.....	91
A29.	Lineament Map of LINE 29.....	91
B1.	Rose Diagram of LINE 3.....	92
B2.	Rose Diagram of LINE 4.....	92
B3.	Rose Diagram of LINE 6.....	93
B4.	Rose Diagram of LINE 7.....	93
B5.	Rose Diagram of LINE 8.....	93
B6.	Rose Diagram of LINE 9.....	93
B7.	Rose Diagram of LINE 10.....	93

B8.	Rose Diagram of LINE 11.....	93
B9.	Rose Diagram of LINE 12.....	94
B10.	Rose Diagram of LINE 13.....	94
B11.	Rose Diagram of LINE 14.....	94
B12.	Rose Diagram of LINE 15.....	94
B13.	Rose Diagram of LINE 17.....	94
B14.	Rose Diagram of LINE 18.....	94
B15.	Rose Diagram of LINE 20.....	95
B16.	Rose Diagram of LINE 21.....	95
B17.	Rose Diagram of LINE 22.....	95
B18.	Rose Diagram of LINE 23.....	95
B19.	Rose Diagram of LINE 24.....	95
B20.	Rose Diagram of LINE 25.....	95
B21.	Rose Diagram of LINE 26.....	96
B22.	Rose Diagram of LINE 28.....	96
C1.	Classified Image of LINE 2 in Zone B with Tolerance Value = 0.....	97
C2.	Classified Image of LINE 2 in Zone B with Tolerance Value = 1.....	98
C3.	Classified Image of LINE 2 in Zone B with Tolerance Value = 2.....	98
C4.	Classified Image of LINE 2 in Zone B with Tolerance Value = 3.....	99
C5.	Classified Image of LINE 2 in Zone B with Tolerance Value = 4.....	99
C6.	Classified Image of LINE 2 in Zone B with Tolerance Value = 5.....	100
C7.	Classified Image of LINE 2 in Zone B with Tolerance Value = 6.....	100
C8.	Classified Image of LINE 2 in Zone B with Tolerance Value = 7.....	101
D1.	Classified Image of LINE 2 in Zone B with Tolerance Value = 0.....	102
D2.	Classified Image of LINE 2 in Zone B with Tolerance Value = 1.....	103
D3.	Classified Image of LINE 2 in Zone B with Tolerance Value = 2.....	103
D4.	Classified Image of LINE 2 in Zone B with Tolerance Value = 3.....	104
D5.	Classified Image of LINE 2 in Zone B with Tolerance Value = 4.....	104
D6.	Classified Image of LINE 2 in Zone B with Tolerance Value = 5.....	105

D7.	Classified Image of LINE 2 in Zone B with Tolerance Value = 6.....	105
D8.	Classified Image of LINE 2 in Zone B with Tolerance Value = 7.....	106

CHAPTER I

INTRODUCTION

1.1. General Remarks

In mining processes, determination of discontinuities plays an important role both in design and development stages, which requires systematic field studies to update the mining plans for future. It is obvious that these field studies are time and money consuming operations. Therefore it is important to minimize the cost and maximize the incomes and also to take into account the time spent for the field studies.

In this respect, detection of discontinuities from satellite images have great potential to reduce the cost of field studies in terms of time and money.

The extraction process of linear features from remotely sensed images are called lineament analysis. Lineaments are any linear features that can be picked out as lines (appearing as such or evident because of contrasts in terrain or ground cover on either side) in aerial or space imagery. If they appear geological, these are usually faults, joints, or boundaries between stratigraphic formations. Other causes of lineaments include roads and railroads, contrast-emphasized contacts between natural or man-made geographic features (e.g., fence lines). The human eye tends to single out both genuine and spurious linear features, so that some thought to be geological may, in fact, be of other origins. (NASA Remote Sensing Tutorial Web Page)

In literature there are plenty of studies related to lineament analysis [Vanderbrug (1976), Koopmans (1986), Majumdar and Bhattacharya (1988), Wang and Howarth (1990), Clark and Wilson (1994), Mah et al. (1995), Koike et al. (1995), Kane et al. (1996), Mat Akhir and Abdullah (1997), Süzen and Toprak (1998), Novak and Soulakellis (2000), Vasillas et al. (2000)] but these studies do not specifically focus on detecting smaller line features where mining activities require detailed discontinuity maps. In the previously mentioned studies, commonly middle to low resolution images are used covering as compared to relatively larger areas such as possible mine sites. Moreover, these studies lack in an accuracy assessment, in which the reliability of the detected lineaments is determined. In literature, usually the lineaments are extracted based on manual digitizing using either satellite images or aerial photos [Koopmans (1986), Majumdar et al. (1988), Wang and Howarth (1990), Clark and Wilson (1994), Mah et al. (1995), Kane et al. (1996), Mat Akhir and Abdullah (1997), Süzen and Toprak (1998), Novak and Soulakellis (2000), Vasillas et al. (2000),]. There are a few studies on pattern recognition about the lineaments in the literature [Koike et al. (1995)].

1.2. Objectives and Scope of the Thesis

This study has three main objectives. The first one is to detect the line features, which are possible discontinuities in the mine area with Remote Sensing techniques using multispectral Ikonos pansharpened image (with 1 meter resolution, ortho-rectified) and to compare it with the results obtained from conventional methods, which is the manual digitizing the lineaments on the image. The image processing is held by PCI Geomatica v 8.2 software. Geomatica has an option of extracting lineaments automatically within an algorithm (LINE). The vector product obtained by LINE is compared with manually extracted lineaments.

The second objective of the thesis is to create the lineament map to be used as a reference in order to compare the vector product of LINE. The comparison of

manually obtained lineament map (reference map) with the output of LINE (automatically produced map) forms a basis of accuracy assessment.

The third objective is to develop a methodology for accuracy assessment of automatically produced lineament map to determine the reliability and efficiency of the analyses.

1.3. Thesis Outline

Following the introduction, Chapter I, basic concepts associated with lineament analysis is reviewed in Chapter II as a part of the literature survey carried out in this study.

Chapter III includes description of the study area, which is located in andesite quarry area in Gölbaşı district of Ankara, Turkey. This chapter also gives information about the data used in the thesis. The remote sensing analyses used for automatic and manual extraction of lineaments are presented in Chapter IV.

Chapter V includes the accuracy assessment of the automatically extracted lineament maps. The algorithm developed for assessing the reliability of automatically extracted lineament map is explained in this chapter.

CHAPTER II

BASIC CONCEPTS ABOUT THE LINEAMENT ANALYSIS

2.1. Introduction

In the early days of Landsat, detecting linear features that appeared as tonal discontinuities was the most common use of space imagery. Just about anything that showed as a roughly straight line in an image was supposed to be geological. Most of these lineaments were ascribed either to faults or to fracture systems. (Nasa Remote Sensing Tutorial Web Page)

O’Leary et al. (1976) described the term lineament as “a mappable, simple or composite linear feature of a surface whose parts are aligned in a rectilinear or slightly curvilinear relationship and which differ from the pattern of adjacent features and presumably reflects some sub-surface phenomenon”.

Continuous subsurface fracture planes, which extend over large distances and intersect the land surface, produce linear traces (lineaments). A linear feature in general can show up in an aerial photo as discontinuity in brightness that is either darker in the middle and lighter on both sides; or, it is lighter on one side and darker on the other side. Obviously, some of these features may not be geological. These could be fence lines between crop fields, roads, or variations in land use. Others may be morphological, such as ridge crests, set off by shadowing. But those that are structural (joints and faults) are visible in several ways. They commonly are opened up and enlarged by erosion. (Nasa Remote Sensing Tutorial Web Page)

In the following sections different remote sensing techniques for lineament extraction is provided. Furthermore, lineament extraction from satellite imagery in the literature is reviewed.

2.2. Lineament Analysis from Satellite Imagery

A class of important features in satellite imagery are the linear features (Vanderbrug, 1976). Lineament analyses in remote sensing always represent an important part of Earth science based studies. Lineament maps prepared by conventional field mapping techniques cannot identify all of the lineaments existing in the area, because of the large scale. However, they are the most accurate and reliable maps (Süzen and Toprak, 1998). Furthermore, in the aerial photographic surveys, large-scale regional lineaments that extend the coverage of the photos cannot be detected due to the lack of continuity (Süzen and Toprak, 1998). Relatively more reliable lineament maps can be prepared by the use of digital remote sensing with the availability of multi-spectral and radar images and progressive development in the image enhancement techniques (Süzen and Toprak, 1998). The wide ground coverage and relative high resolution with respect to scale presented by the satellite images are other advantages in regional and local lineament analysis. (Süzen and Toprak, 1998)

There are basically three methods for obtaining a lineament map from remotely sensed images. The first method is called spatial convolution filtering methods for lineament extraction. Second method is called multiband analysis and finally the third one is automatic lineament extraction methods.

2.2.1. Spatial Convolution Filtering

A linear spatial filter is a filter for which the brightness value at a certain location in the output image is a function of some weighted average of brightness values located in a particular spatial pattern around the certain location in the input image. This process of evaluating the weighted neighboring pixel values is called two-dimensional convolution filtering (Pratt, 1991 in Jensen, 1996). This procedure is often used for changing the spatial frequency characteristics of an image. For example, a high-frequency linear spatial filter may sharpen the edges within an image; however, a low-frequency linear spatial filter may be used to reduce noise within an image (Jensen, 1996).

Image enhancements that deduct or block the high spatial frequency detail are low-frequency or low-pass filters. The simplest low-pass filter evaluates a particular input pixel brightness value and the pixels surrounding the input pixel, and outputs a new brightness value that is the mean of this convolution. The size of the neighborhood convolution kernel is usually 3x3, 5x5, 7x7, or 9x9. (Jensen, 1996)

For example, the coefficients in a low-frequency convolution mask might all be set equal to 1:

Original Image

0	1	3	4	7
2	5	4	9	7
8	3	1	6	4

$$\text{Mask A} = \begin{matrix} & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{matrix}$$

Filtered Image

	3	4	5

$$\text{i.e. } 3 = (0 \times 1 + 1 \times 1 + 3 \times 1 + 2 \times 1 + 5 \times 1 + 4 \times 1 + 8 \times 1 + 3 \times 1 + 1 \times 1) / 9$$

$$4 = (1 \times 1 + 3 \times 1 + 4 \times 1 + 5 \times 1 + 4 \times 1 + 9 \times 1 + 3 \times 1 + 1 \times 1 + 6 \times 1) / 9$$

$$5 = (3 \times 1 + 4 \times 1 + 7 \times 1 + 4 \times 1 + 9 \times 1 + 7 \times 1 + 1 \times 1 + 6 \times 1 + 4 \times 4) / 9$$

Such image smoothing is useful for removing periodic “salt and pepper” noise in the image which is recorded by the sensor. This simple smoothing operation will, however blur the image, especially at the edges of objects. Blurring becomes harsher as the size of the kernel increases. To reduce blurring, unequal-weighted smoothing masks have been introduced. Using a 3x3 kernel can result in the low-pass image being two lines and two columns smaller than the original image.

High-pass filtering, on the other hand, emphasizes the high frequency components of a signal while reducing the low frequency components. A high-pass filter is computed by subtracting the output of the low-pass filter from twice the value of the original central pixel value:

$$\text{HPF}_{5, \text{out}} = (2 \times \text{BV}_5) - \text{LPF}_{5, \text{out}}$$

Where;

BV = Brightness Value

LPF_{5, out} = Output of Low-pass Filter

Brightness values tend to be highly correlated in a nine-element window. Thus, the high-pass filtered image will have a relatively narrow intensity histogram. Because of this, the high pass filtered images should be contrast stretched for the purpose of visual analysis.

Edge detection in remote sensing is one of the most important method in lineament extraction. In an image, edges surrounding various objects contain the most valuable information. Edge enhancement techniques make the shapes and details more conspicuous and easier to analyze. In general, edges are simply sharp changes in brightness values between two adjacent pixels. The edges may be enhanced by both linear and non-linear edge enhancement techniques.

Application of a directional first-difference algorithm that approximates the first derivative between two adjacent pixels is a basic method of extracting edges in remotely sensed imagery. The algorithm produces the first difference of the image input in the horizontal, vertical and diagonal directions. The algorithms for enhancing horizontal, vertical, and diagonal edges for a pixel (x,y) are, respectively;

$$\text{Vertical: } BV_{x,y} = BV_{x,y} - BV_{x,y+1} + K$$

$$\text{Horizontal: } BV_{x,y} = BV_{x,y} - BV_{x-1,y} + K$$

$$\text{NE Diagonal: } BV_{x,y} = BV_{x,y} - BV_{x+1,y+1} + K$$

$$\text{SE Diagonal: } BV_{x,y} = BV_{x,y} - BV_{x-1,y+1} + K$$

In order to make all values positive and ranged between 0 and 255, a constant K is added (usually 127) because the result of the subtraction can be either positive or negative (Jensen, 1996). This causes adjacent pixels with very little difference in brightness value to obtain a brightness value of around 127 and any

dramatic change between adjacent pixels to migrate away from 127 in either direction.

Jensen (1996) refers to Pratt (1991) and Jain (1989) introduced the compass gradient masks, which is shown in Figure 2.1, to perform two-dimensional, discrete differentiation directional edge enhancement.

North Mask	1	1	1	Northeast Mask	1	1	1
	1	-2	1		-1	-2	1
	-1	-1	-1		-1	-1	1
East Mask	-1	1	1	Southeast Mask	-1	-1	1
	-1	-2	1		-1	-2	1
	-1	1	1		1	1	1
South Mask	-1	-1	-1	Southwest Mask	1	-1	-1
	1	-2	1		1	-2	-1
	1	1	1		1	1	1
West Mask	1	1	-1	Northwest Mask	1	1	1
	1	-2	-1		1	-2	-1
	1	1	-1		1	-1	-1

Figure 2.1. Compass Gradient Masks of Pratt and Jain

Jensen (1996) refers to Richards (1986) identified four additional 3x3 templates that may be used to detect edges in the images:

$$\begin{array}{rcl}
 & -1 & 0 & 1 & & -1 & -1 & -1 \\
 \text{Vertical} = & -1 & 0 & 1 & & \text{Horizontal} = & 0 & 0 & 0 \\
 & -1 & 0 & 1 & & & 1 & 1 & 1
 \end{array}$$

$$\text{Diagonal} = \begin{matrix} & 1 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{matrix} \qquad \text{Diagonal} = \begin{matrix} & 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & -1 \end{matrix}$$

Nonlinear edge enhancements are performed using nonlinear combinations of pixels. Many algorithms are applied using either 2x2 or 3x3 kernels. The Sobel edge detector is based on the notation of the 3x3 window previously described and is computed according to the relationship:

$$\text{Sobel}_{5, \text{out}} = \sqrt{X^2 + Y^2}$$

Where

$$X = (BV_3 + 2BV_6 + BV_9) - (BV_1 + 2BV_4 + BV_7)$$

And

$$Y = (BV_1 + 2BV_2 + BV_3) - (BV_7 + 2BV_8 + BV_9)$$

The Sobel operator may also be computed by simultaneously applying the following 3x3 templates across the image. This procedure detects horizontal, vertical, and diagonal edges. (Jain, 1989):

$$X = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}, \qquad Y = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

The Roberts edge detector is based on the use of only four elements of a 3x3 mask. The new pixel value at pixel location $BV_{5, out}$ is computed according to the equation (Peli and Malah (1982) in Jensen (1996)]

$$Roberts_{5, out} = X+Y$$

Where

$$X = |BV_5 - BV_9|$$

$$Y = |BV_6 - BV_8|$$

Canny Edge Detection Algorithm is another effective tool for detecting edges in an image. In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image with Gaussian filter before trying to locate and detect any edges (Green, 2002). 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. Then a simple 2-D first derivative operator (somewhat like the Sobel Operator) is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non-maximal suppression. The tracking process exhibits hysteresis controlled by two thresholds: $T1$ and $T2$ with $T1 > T2$. Tracking can only begin at a point on a ridge higher than $T1$. Tracking then continues in both directions out from that point until the height of the ridge falls below $T2$. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments (The University of Edinburgh, Hyper Media Image Processing Reference).

There are several different combinations of above mentioned image filtering methods for lineament analysis purposes. Mat Akhir and Abdullah (1997)

proposed the following algorithm for mapping and analysis of lineaments in NW Peninsula Malaysia. The study area is approximately 2500 km². The Landsat Thematic Mapper (TM) which is covering part of north-west Peninsula Malaysia has been digitally processed to enhance the lineaments. TM band 4 which is contrast stretched was found to be the most suitable data for lineament extraction. The directional and non-directional filtered images were used together with the contrast stretched image of TM band 4 for lineament mapping and analysis for the study area. After all the lineaments were traced, their orientations and lengths were determined and measured, and then analyzed by preparing rose diagrams for analysis and interpreted with the published geological map. After following a number of processing techniques of Jensen (1986) and Mather (1987), the contrast of six bands data were digitally enhanced. As a result of visual interpretation of contrast and definition of geological features (lineaments), Landsat TM band-4 data were selected. In order to enhance the lineaments further, filtering techniques were applied to this band 4 of Landsat TM data. The applied filter values are given in Figure 2.2.

Non-directional filter (Laplacian convolution filter)

0	-1	0
1	5	-1
0	-1	0

Directional filter

1	0	1	-1	-2	-1
2	0	2	0	0	0
1	0	1	1	2	1
(north-south)			(east-west)		

2	-1	0	0	-1	-2
1	0	1	1	0	-1
0	1	2	2	1	0
(northeast-southwest)			(northwest-southeast)		

Figure 2.2. The applied filter values by Mat Akhir and Abdullah (1997)

Therefore, the following images are used in the study of Mat Akhir and Abdullah (1997) that is:

- Image of digitally contrast enhanced Band-4
- Image of laplacian convolution filtered band-4
- Four images of directional filtered band-4, N-S, E-W, NE-SW and NW-SE respectively.

After filtering operations the below given methods were followed:

- All lineaments in contrast enhanced TM band 4 image and filtered TM band 4 were traced separately. The tracings were overlapped and all the lineaments which were present in both tracings were traced again in the third tracing.
- The similar method was applied to all images of directional filtered TM band 4 images (Filter N-S, E-W, NW-SE, NE-SW). Only lineaments present in any two images were traced into the third tracing and were considered in analysis and mapping later.

The traced lineaments showed a similar pattern compared to the published geologic data of the area. It was seen that the main features of fault pattern were enhanced in the lineament map. However, in terms of total number and length, the traceable lineaments were much more greater than already mapped fault.

After the analysis Mat Akhir and Abdullah (1997) agreed on some following conclusions:

1. Contrast enhanced TM band-4 and Laplacian convolution filtered and directional filtered band-4 were the best to enhance the structural information (lineaments) of the area.
2. The interpreted lineaments are closely correlated to the faults in the published geological map of the area.
3. If other remote sensing data, which have higher resolution, of the area are available, possible better and more detailed results could be obtained.

As another study related to filtering operation, Suzen and Toprak (1998) proposed another algorithm for the lineament analysis of southern part of the Galetean Volcanic Province, Turkey. The study area is 1500 km².

In order to overcome the masking of dominant directions over minor ones in multiband applications, basically the spatial frequency and the contrast would have to be increased. Applying spatial domain filtering to the image was selected. Band 7 is for filtering analyses because of its greater wavelength and to verify the

outcome of filtering in the same band. The types of the filters were selected as directional high pass filters.

Firstly Prewitt kernels were convolved over band 7 in four principal directions (N-S, NE-SW, E-W, and NW-SE). However, there was no significant increase in the contrast and frequency relative to the unfiltered original images. Because of that reason for the next step, Sobel filters were used to increase the frequency and contrast. The result is more satisfactory than the Prewitt kernels but the lineaments have a segmented appearance, due to the increased frequency. In order to combine these lineaments and reduce the noise present in TM band 7 12x12 low pass filters were used. Vertical gradient filter was convolved over the smoothed image because homogenization was preferred in fault zones.

As a conclusion, the directional property of Prewitt and Sobel filters results in an effective and faster way to evaluate lineaments. Spatial domain filtering analysis with single band input seems to be the most cost-time efficient and fast method for regional lineament analysis.

Besides directional filters, filtering with different illumination directions is another method used in lineament analysis such as the study of Mah et al. (1995) in Palm Valley, Australia.

First TM band 7 was filtered with a NE-SW, TM band 5 with a NW-SE and TM band 4 with E-W and N-S trending illuminated filters. In filtered image of band 7 highlighted not only all the NW-SE trending but also NNW-SSE and WNW-ESE trending lineaments because the illumination direction was from NE-SW. Similar to this NW-SE filtered band 5 highlighted the NE-SW, NNE-SSW, ENE-WSW trending and E-W and N-S filtered band 4 highlights the E-W, ENE-WSW, ESE-WNW and N-S, NNW-SSE, NNE-SSW trending lineaments respectively.

2.2.2. Multiband Analysis

Principal components analysis and false color composites constitute an important part for the lineament analysis with multiband applications in literature. There are applications of manually digitizing the color composite images [Minor et al. (1994), Kane et al. (1996), Cortes et al. (2003), Leech et al. (2003), Williamson (2003)].

Kane et al. (1996) studied this subject by using Landsat MSS digital imagery. Bands 457 combined in false color composite form. The study area is approximately 185 km by 185 km and the data has a ground resolution of 79 m. After forming the composite images the lineaments are manually plotted.

Cortes et al. (2003) used the Landsat TM data for the lineament analysis of the Duero Basin, Spain. A subscene of 180 km x 210 km was selected. A color composite of bands 542 (RGB) was formed and %1 of this image was linear contrast stretched. After remote sensing analysis lineaments were drawn and mapped by hand.

Leech et al. (2003) used Landsat TM data with an area of 100 km x 80 km for the analysis of fracture patterns of Coastal Cordillera of Northern Chile. According to the authors, the optimum band combination is bands 721 (RGB) image with a Gaussian nonlinear stretch. Following these image processing techniques, lineaments were manually digitized.

Minor et al. (1994) studied this subject by using Landsat TM data and constructed the color composites with band combinations of 453 and 743 in RGB. In addition to this, each color composite was then edge enhanced using 3x3 edge sharpening filter. In addition to generating edge enhanced color composites, directionally filtered images were constructed using TM band 4.

Producing mosaic images and combining with other data is another method studied in lineament analysis. Williamson (2003) refer to Sabins (1996) in his study that ratio images combined in RGB offer great contrast between geologic units than

do individual TM band false color images. For this reason 5/7-5/1-3/4*5/4 image was used besides 7-4-2 composite image. In order to interpret the bulk of the structure the most suitable image was found to be the SLAR image and an image mosaic of SLAR was scaled and overlaid with TM data.

Principal component analysis (also known as Hotelling, the Karhunen - Loeve transformation, or Eigenchannel transformation) is an image enhancement technique for compressing the information of n spectral bands into just two or three transformed principal component images (Jensen, 1996). Jensen refers to Singh and Harrison (1985) in the way that new principal component images may be more interpretable than the original data. Also it can be stated that PCA can also be useful for reducing the dimensionality of multispectral datasets (Jensen, 1996). Another advantage of PCA is that spectral differences between objects may be more obvious in PC images (Sabins, 1996).

In the study of Süzen and Toprak (1998) principal components analysis was used to detect lineaments after single band (band 7 of TM) analysis, which consists of equalizing and linearly stretching the histograms with manually digitizing. The image was then improved by PCA. The objective of this was to get nearly the whole of the available information in three bands (Süzen and Toprak, 1998). After calculating the seven eigen values and eigen vectors, the information in these three bands was found to be sufficient enough, which is 96.31 per cent. Visually interpretation of the resultant color composite yielded to a lineament map which has increased approximately about 17 per cent. The maximum concentration was found in the same direction as the single band analysis but the density decreased. As a result of this the authors decided to take directional filtering into account.

Novak and Soualakellis (2000) used PCA and false color composite (FCC) image analysis techniques to identify geomorphic features of Lesvos, Greece using Landsat-5 TM data. The study area is 1632 km².

Principal component images of the three visible and three reflected infrared bands were prepared. The authors refer to Sabins (1996) for determining the per

cent variation of the PC images to original six bands. This was 97 per cent for first three PC images (PC 1, 2, 3) and 2.6 per cent for the last three (PC 4, 5, 6).

The PC 1 signifies a panchromatic view of the area with 88 per cent of the data variance, whereas the PC 2 signifies the difference between infrared and visible bands.

According to Novak and Soualakellis (2000) FCC image is the most effective means of visual presentation of multiband image. Because a standard FCC image contains 73 per cent of variance and PC image has 97 per cent, it was decided to use first two PC images in creating the FCC image. After a trial of several combinations two most appropriate RGB color composites were determined. These were TM band 3, PC 1 and PC 2; and the other one was TM band 7, PC 1, PC 2. The second combination has advantages over the first one because of the high applicability of band 7 with geologic investigations.

Some lineaments determined with the above methods not only represent the previously mapped faults or tectonic contacts but also some extensions of these and previously undetected ones, which could be faults, tectonic and formational contacts.

In the study of Won-In and Charusiri (2002) geological mapping of Cho Dien area in Northern Vietnam different enhancement techniques were applied to Landsat TM bands 4, 5 and 7 according to their better applicability to geological interpretations. The area of study is 130 km².

For the image enhancement, dark-pixel correction was used as the first step. Following that high-pass filtering is the second step. It was believed that high-pass filtering shows features clearer and more distinct than unenhanced ones. Another step of the study is ratioing. According to the author ratios of bands 4/5, 5/7, 7/5 show “good-looking” in geological interpretations. Besides these analyses, image classification was also applied in order to get a good contrast of lithologies and lineaments. Final enhancement step is to create a false color composite by assigning band 7 to blue, band 5 to green and band 4 to red.

After analyzing different methods, false color composite is recommended for geological and mineral investigations.

The lineaments of the study area were classified into two types that are major (>1 km) and minor (<1 km) according to the method of Udomratn (1987). The major ones were again classified into four types; N, NNE to NE, NNW to NW and E trending lineaments. There are also several minor lineaments having directions of N, NE and NW.

2.2.3. Automatic Lineament Extraction

Besides the methods mentioned above there are some studies of automatic lineament extraction by applying different transformation techniques to remotely sensed images. Hough Transform, Haar Transform, Segment Tracing Algorithm and Rotation Transformation are examples of such techniques.

The Hough transform is a technique which can be used to separate features of specific shape within an image. It is required that the specific feature must be specified in some parametric form, the classical Hough transform is most commonly used for the detection of lines, circles, ellipses, etc. Also generalized Hough transform can be applied where a simple analytic description is not possible, but the calculation becomes much more complex than the classical Hough transform.

A line segment, which is shown in Figure 2.3, can be defined as a convenient equation:

$$X \cos\theta + Y \sin\theta = r$$

Where

r: the length of a normal from the origin to the line

θ : orientation of r with respect to x-axis

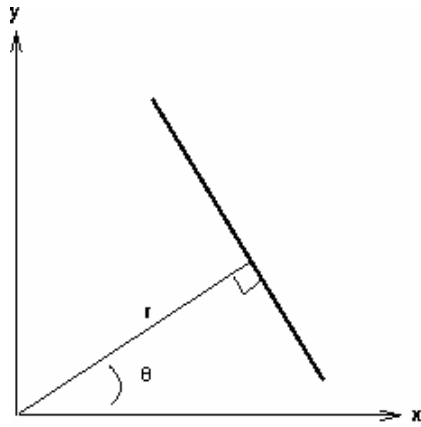


Figure 2.3. Simple Illustration of Hough Transform

In an image analysis, the coordinates of the points of edge segments in the image are known. If possible (r, θ) values defined by each pair of coordinates (x, y) is plotted, points in Cartesian image space map to curves in the Hough parameter space. This point to curve transformation is the Hough transformation for straight lines. The transform is implemented by quantizing the Hough parameter space into finite intervals. As the algorithm runs, each (x, y) is transformed into a discretized (r, θ) curve and the accumulator cells which lie along this curve are incremented. Resulting peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image.

There are some studies in literature, which used the Hough transform for automated lineament detection such as Wang and Howarth (1990). The study area is a part of Canadian Shield near Sudbury, Ontario, which covers a 225 km². Landsat TM 5 band 4 is used for the lineament analysis after a visual comparison of all TM bands.

The original image was preprocessed using a 3x3 median filter and for detecting the edges in the image, edges following as graph searching (EFGS) algorithms were applied. In EFGS algorithm first an edge operator is used to determine the locations where maximum changes in pixel values occur and their direction. After that the starting points of the edges are determined.

After edge detection, Hough transform was applied. Brighter tones represent higher counts in the accumulator arrays, which were produced after Hough transform. From the accumulator arrays local maxima were selected and following that inverse Hough transform and straight-line profile analysis were applied and lineament plots obtained.

The effectiveness of the lineament plots are determined by comparing them with the geologic map. The geologic map effectiveness is accepted as 100 per cent and according to this, visual interpretation has 50.15 per cent effectiveness. EFGS has 54.15 per cent effectiveness. Hough transform I (threshold for local maxima = 80, gap allowed = 40) has 53.72 per cent effectiveness. Hough transform II (threshold for local maxima = 75, gap allowed = 40) has 65.41 per cent effectiveness. Hough transform III (threshold for local maxima = 40) has 68.88 per cent effectiveness. Hough transform IV (threshold for local maxima = 75) has 69.18 per cent effectiveness.

Majumdar and Bhattacharya (1988) proposed a method for extraction of linear and anomalous patterns by application of Haar transform. The image was obtained using ISRO Multispectral scanner carried on an aircraft, which contains 256x256 pixels covering an area of approximately 17 km². In image processing applications the Haar transform provides a domain in which a type of differential energy is concentrated in local regions. The transform has both low and high frequency components and therefore can be used for image enhancement. The algorithm of the study is given in Figure 2.4.

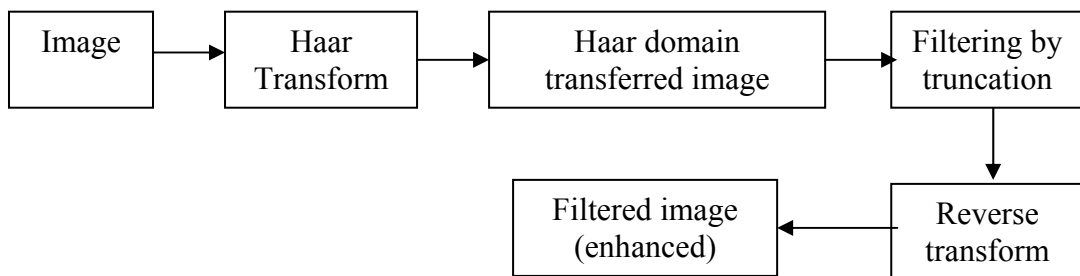


Figure 2.4. Algorithm of Haar Transform

As a conclusion, linear patterns have been extracted with sharp description and in actual case almost all the relevant linear patterns have been extracted. The results were finally compared with existing information such as geological maps and the method proposed had accuracy over 60 per cent.

The Segment Tracing Algorithm (STA), which is developed by Koike et al. (1995), is a method to automatically detect a line of pixels as a vector element by examining local variance of the gray level in a digital image. The test area is in central Kyushu in Japan. The area is mountainous and slopes are covered with broad and needle-leaved trees. Because of the difficulty of using filtering techniques for extraction of lineaments in mountain shadows, STA is used to improve the extraction accuracy of lineaments. For the analysis, Landsat TM is used for an area of approximately 200 km².

It is decided as TM band 4 is the most suitable band for lineament analysis of tree-covered area. The STA determines a line element (segment) which identifies a lineament by two principles. The first is to determine the orientation of a valley by examining the local deviation of gray levels $Z(x)$ along a line:

$$C(x) = \int w(x+h) \{Z(x) - Z(x+h)\}^2 dh$$

Where $w(x)$ is a weight coefficient, h is a distance from x , and a is the maximum distance from the x . The second is to evaluate the variation of $Z(x)$ by the differential:

$$\lambda = \{d^2Z(x)/dx^2\}^2 / Z(x)$$

The division of the secondary differential by $Z(x)$ equalizes the variations for areas of high and low contrast in the image. The detailed content of STA follows the equations above.

For the comparison of the algorithm with filtering techniques, Koike et al. (1995) extract the lineaments by using filtering techniques. From the comparison of

two methods, it was seen that line elements extracted by differential filtering operator are continuous as compared to STA's output. However, the line elements in the shadow area are smaller in number.

2.3. Selection of Pixel Size

From the literature studies discussed in this chapter, it can be clearly seen that the image data used is generally Landsat TM. Because of the large extend of the research area and size of the lineaments, Landsat TM data is probably the most suitable one with its 30 m ground resolution.

The literature studies on lineament analysis covers large areas, where there are relatively less detail required for mining applications. For this reason low resolution image such as Landsat TM data is not sufficient for mining applications. As in mining activities the area is relatively smaller than the previous case studies and a higher spatial resolution is required. In this study Ikonos image with higher ground resolution (1 m), which is one of the highest ground resolution provided by earth observing satellites, is selected for producing lineament map.

CHAPTER III

DESCRIPTION OF THE STUDY AREA AND DATA

3.1 Description of the Study Area

The area of research which is within Ankara province, is located 7 km to the east of Gölbaşı town. There are plenty of working and abandoned andesite quarries in the region. The area is approximately 15 km². The general location of the study area in terms of its location in Turkey and Ankara is given in Figure 3.1. The coordinates of the study area are given in Table 3.1 as Easting and Northing in UTM projection and geographical Longitude and Latitude.

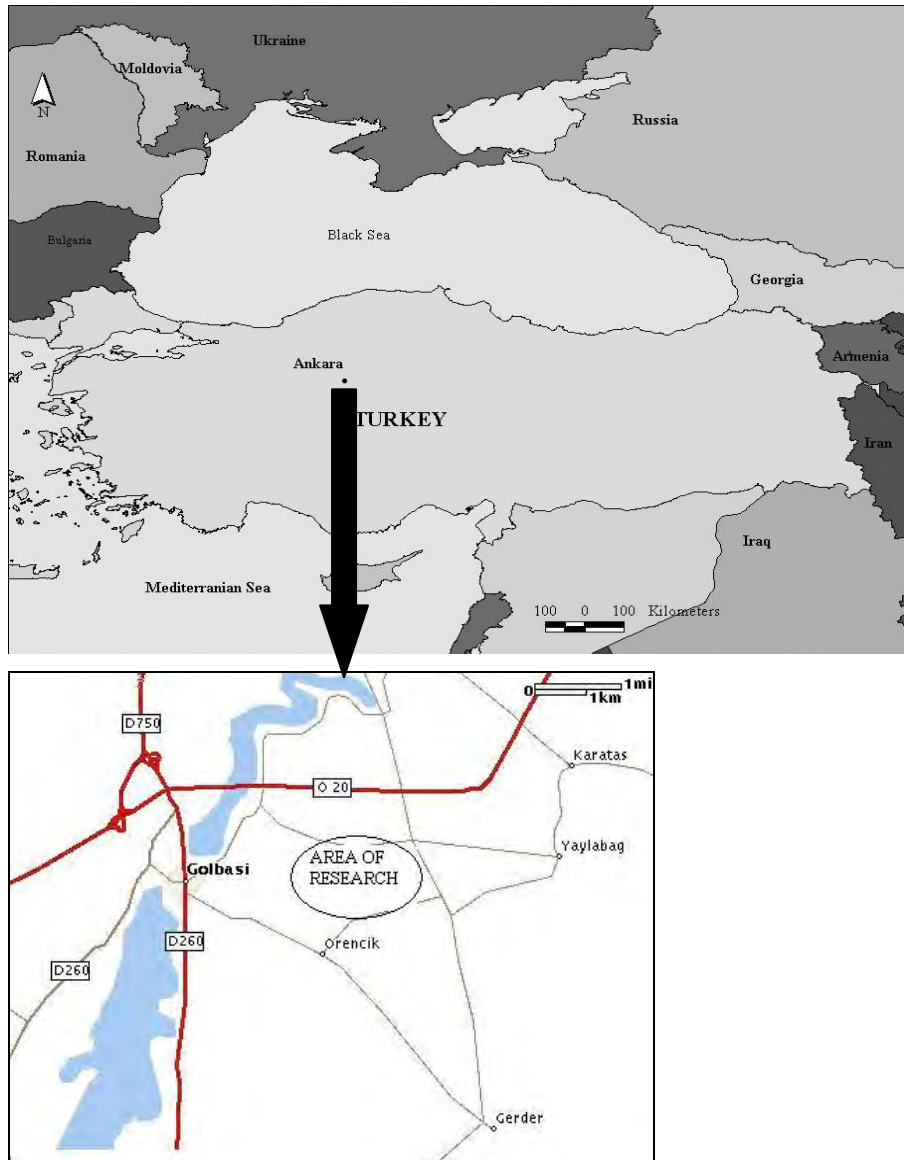


Figure 3.1. Location map of the study area

Table 3-1. – Corner coordinates of the study area

Corner	Easting	Northing	Longitude	Latitude
North-west	488234	4405640	32°51'45"	39°48'2"
North-east	492339	4405640	32°54'38"	39°48'3"
South-east	492339	4401954	32°54'38"	39°46'3"
South-west	488234	4401954	32°51'45"	39°46'3"

The nearest villages to the study area is Örencik and Yurtbeyi. There are six general formations such as tuff, andesite, alluvial, limestone in complex series, ofiolitic dominant complex series (Personal interview with Assoc. Prof. Dr. Yusuf Kağan Kadioğlu). The lithological map of the area is given in Figure 3.2.

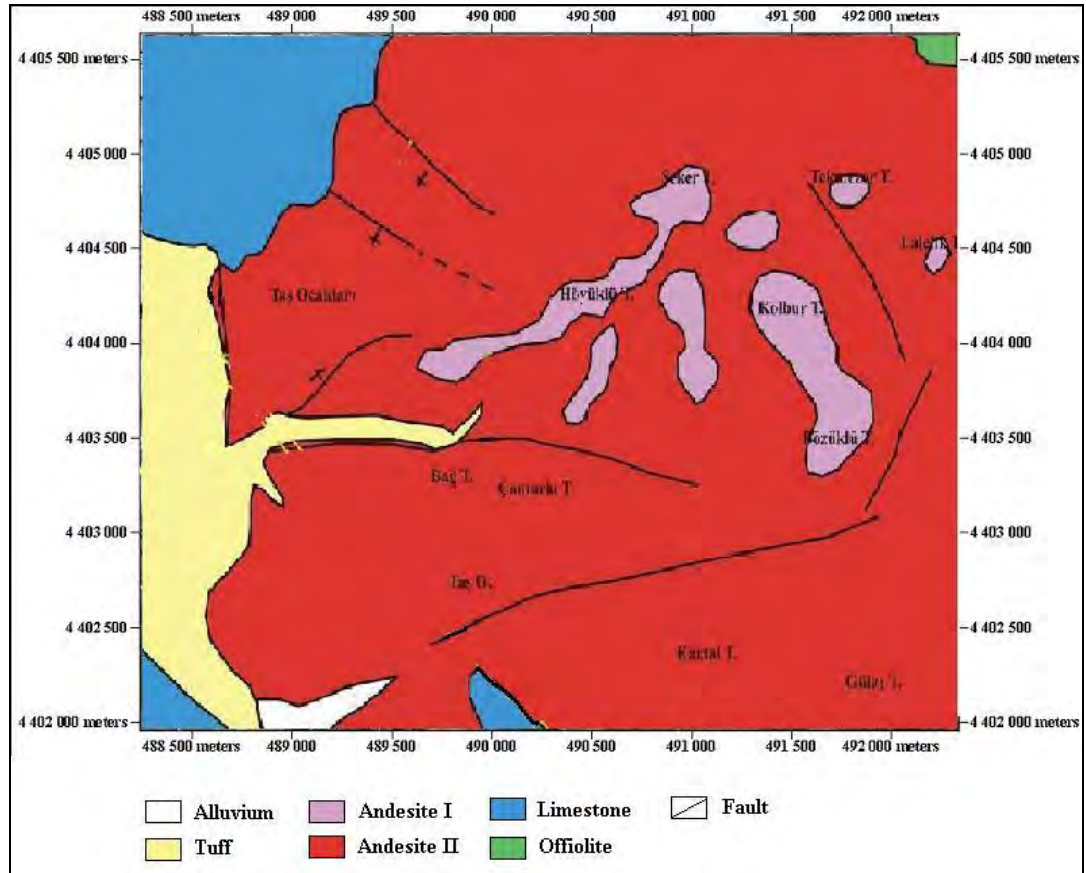


Figure 3.2. Lithological map of the study area

3.2. Description of Data

Discontinuities have important role in mining industry. In order to evaluate them for mine design purposes, they have to be determined in large scale maps. Therefore using a high resolution satellite imagery for the lineament extraction is preferred in this study. For this reason, 8-bit Ikonos Precision Plus with 1 meter resolution orthorectified image, which was captured on October 2002, of the andesite quarry area is used. This image combines the details of 1 meter

panchromatic data and color content of 4 meter multi-band data. In order to evaluate the automatically extracted lineament maps with the manually extracted one, the data is selected as 8-bit because of the reason that Geomatica's LINE option works with 8-bit data as it will be explained in the following chapter.

The orthorectification process removes image distortions introduced by the collection geometry and by relief of the terrain, and re-samples the imagery to a uniform ground sample distance and a user-specified map projection. Precision and Precision Plus, the highest-level accuracy products, requires the use of ground control points. Space Imaging is providing ground control points. Also the Precision Plus products provide minimum data collection angle of 75°.

Image of the study area is obtained from INTA Space Systems, Inc. The image is 8 bit with 1-m resolution pan-sharpened multiband. Band 1 is blue, band 2 is green and band 3 is red. The pansharpened true color composite image of the research area is given in Figure 3.3.

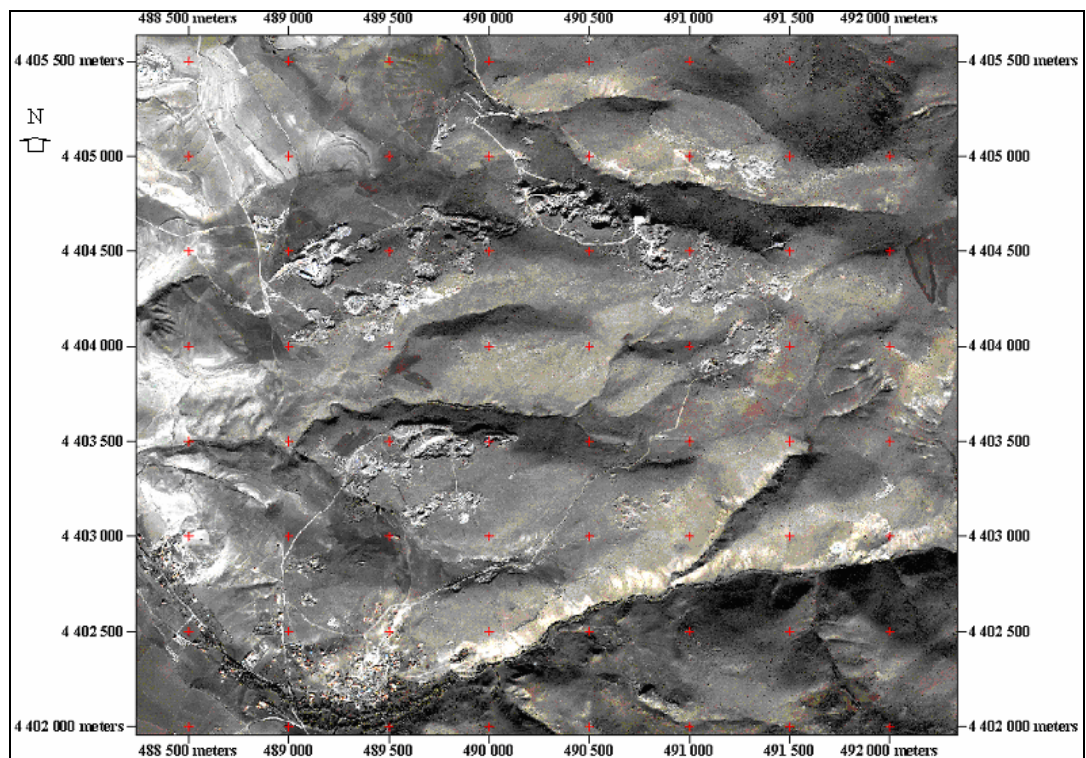


Figure 3.3. Pansharpened True Color Composite Image of the Study Area

CHAPTER IV

EXTRACTION OF LINEAMENTS

4.1. Introduction

As it is mentioned before for the image analysis, PCI Geomatica Version 8.2 and TNTmips Version 6.8 were used. The most important factor for using Geomatica is the ability to extract lineaments from raster images automatically with the LINE option. For testing the reliability of the software, the lineaments are extracted manually by directional filtering. Furthermore, the lineaments detected are compared with those appear in the the slope face. The face discontinuities are determined again by directional filtering and followed by manual extraction. The flowchart of the study is given in Figure 4.1.

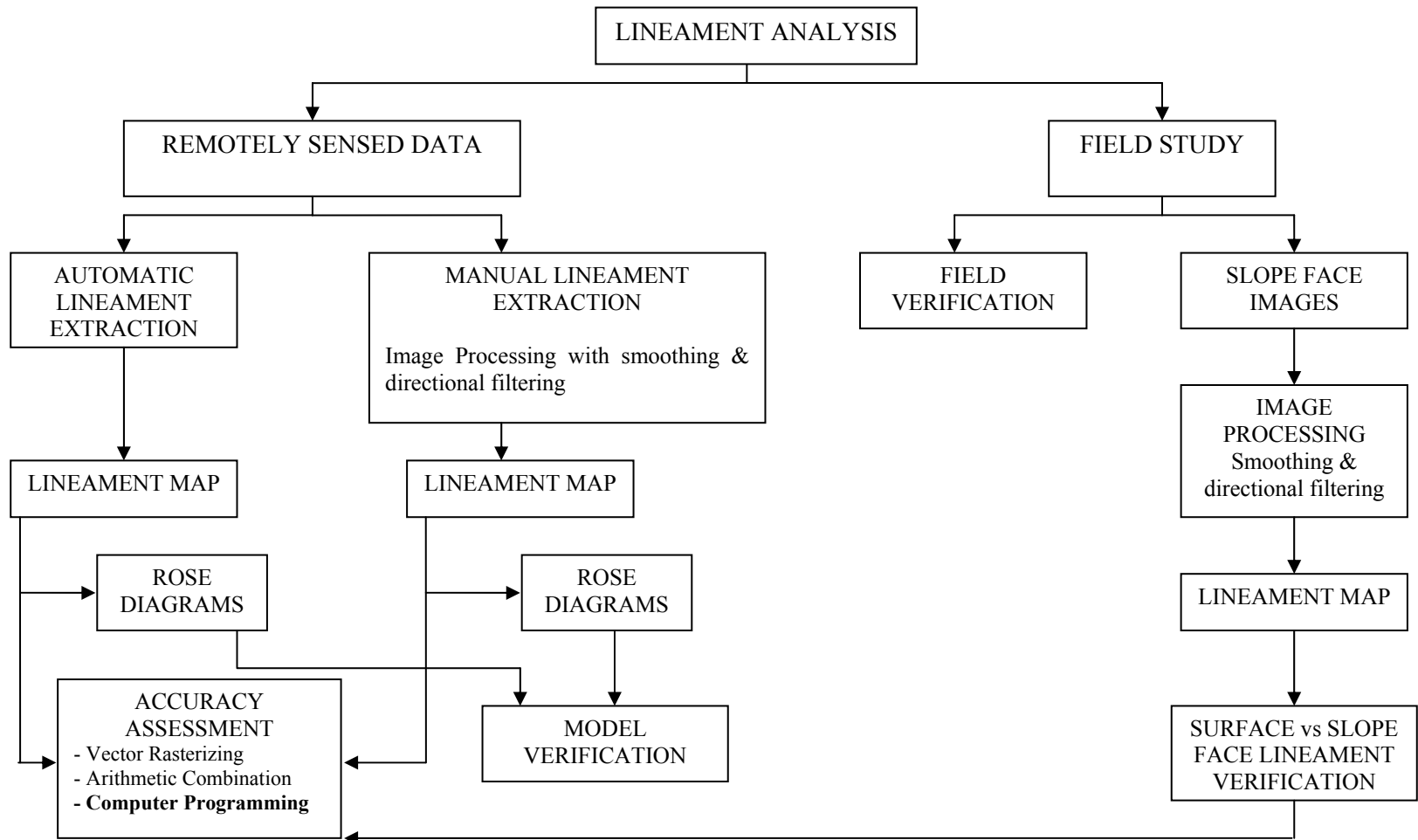


Figure 4.1. Flowchart of the Study

4.2. Automatic Lineament Extraction

LINE option of Geomatica extracts linear features from an image and records the polylines in a vector segment.

LINE is controlled by the following global parameters:

Name	Prompt	Count	Type
FILI	Database Input File Name	1-64	Char
FILO	Database Output File Name	1-64	Char
DBIC	Database Input Channel List	1	Int
DBOC	Database Output Channels	0-1	Int
DBVS	Database Vector Segment	0-1	Int
RADI	Radius of filter in pixels	1	Int
GTHR	Threshold for edge gradient	1	Int
LTHR	Threshold for curve length	1	Int
FTHR	Threshold for line fitting error	1	Int
ATHR	Threshold for angular difference	1	Int
DTHR	Threshold for linking distance	1	Int

4.2.1. Details of LINE

The LINE program takes a single image channel as input. If it is 16-bit or 32-bit, the image is first scaled to 8 bit using a nonlinear scaling routine. The output of the program is a vector segment which contains linear features as extracted from the image. If DBOC is specified, a binary edge image (which is the result of thresholding the gradient) will be saved in the specified channel.

RADI specified the size of Gaussian kernel which is used as a filter during edge detection. The larger the RADI value, the less noise and less details in the edge detection result.

The value for thresholding the gradient image is given by the parameter GTHR. This value should be in the range 0 to 255. The user can experiment with different GTHR values and choose one which produces a suitable binary image in DBOC. If the pixels representing the edges in the image appear to be too sparse, the GTHR value should be decreased. On the other hand, if the pixels representing the edges are dense and noisy, the GTHR value should be increased. Note that it is important to have sufficient information in the edge image as the subsequent lineament extraction process is based on this edge image as input.

Various other parameters control the curve extraction process. FTHR is the tolerance for fitting line segments to a (curved) lineament. It is specified in number of pixels. A lower value results in many shorter line segments that better approximate the lineament; however, a larger value better average out noise giving longer straighter lineaments. Typically, a value between 2 and 3 is ideal. LTHR is the minimum length of a curve (in pixels) to be considered as lineament for further consideration. ATHR is the maximum angle (in degrees) between two vectors for them to be linked. DTHR is the maximum distance (in pixels) between two vectors for them to be linked.

4.2.2. Algorithm of LINE

The algorithm of LINE consists of three stages: edge detection, thresholding, and curve extraction.

In the first stage, the Canny edge detection algorithm is applied to produce an edge strength image. The Canny edge detection algorithm has three substeps. First, the input image is filtered with a Gaussian function whose radius is given by the RADI parameter. Then gradient is computed from the filtered image. Finally, those pixels whose gradient are not local maximum are suppressed (by setting the edge strength to 0).

In the second stage, the edge strength image is thresholded to obtain a binary image. Each ON pixel of the binary image represents an edge element. The threshold value is given by the GTHR parameter.

In the third stage, curves are extracted from the binary edge image. This step consists of several substeps. First, a thinning algorithm is applied to the binary edge image to produce pixel-wide skeleton curves. Then a sequence of pixels for each curve is extracted from the image. Any curve with the number of pixels less than the parameter value LTHR is discarded from further processing. An extracted pixel curve is converted to vector form by fitting piecewise line segments to it. The resulting polyline is an approximation to the original pixel curve where the maximum fitting error (distance between the two) is specified by the FTHR parameter. Finally, the algorithm links pairs of polylines which satisfy the following criteria:

- (1) two end-segments of the two polylines face each other and have similar orientation (the angle between the two segment is less than the parameter ATHR);
- (2) the two end-segments are close to each other (the distance between the end points is less than the parameter DTHR).

The final polylines are saved in a vector segment. An example of parameter entry of the automatic extraction is given in Figure 4.2.



Figure 4.2. Parameter Entry Screen of Automatic Extraction Process

4.2.3. Results of Automatic Lineament Extraction

By changing the six controlling parameters of the automatic extraction process, some results are obtained. The main focus is given to the number of the lineaments, the length of the lineaments and the orientation of the lineaments. The parameters with total length and number of lineaments together with the corresponding parameters are given in Table 4.1. The resultant maps and the related parameters are given in Figure A1-A29 in Appendix A.

Table 4.1. Parameters, total length and number of lineaments formed by automatic extraction

LINE ID	PARAMETERS						LENGTH (m)	NUMBER
	RADI	GTHR	LTHR	FTHR	ATHR	DTHR		
1	5	20	10	2	20	1	112195	2883
2	5	20	20	2	20	1	122827	3056
3	5	20	15	2	20	1	165649	5404
4	5	20	25	2	20	1	89641	1901
5	5	30	15	2	20	1	104321	3312
6	5	30	20	2	20	1	74361	1887
7	5	40	15	2	20	1	69366	2265
8	12	40	25	2	20	1	117572	2368
9	12	20	25	3	20	1	210940	4418
10	12	20	25	3	30	1	252699	4846
11	12	20	25	3	40	1	298381	5468
12	12	30	20	3	30	1	245697	5235
13	12	30	30	3	20	1	135713	2415
14	12	30	25	3	20	1	172840	3520
15	12	30	20	3	20	1	225425	5213
16	12	40	25	3	20	1	139668	2776
17	12	50	25	3	20	1	115504	2256
18	12	40	20	3	20	1	181198	4101
19	12	50	20	3	20	1	147397	3271
20	12	40	15	3	20	1	235470	5986
21	12	50	15	3	20	1	189130	4692
22	12	60	15	3	20	1	153974	3752
23	12	60	20	3	20	1	120410	2589
24	12	60	25	3	20	1	95778	1833
25	12	100	15	3	20	1	78245	1807
26	7	20	25	2	20	1	140652	2496
27	7	30	20	2	20	1	135568	2736
28	9	30	20	2	20	1	145870	3410
29	9	30	25	2	20	1	150876	3258

4.3. Manual Extraction of Lineaments

Manual extraction of the lineaments is decided to be the reference of the applicability of Automatic Lineament Extraction with Canny Edge Detection Algorithm. The main advantage of manual extraction is the ability to detect the

non-geological lineaments such as roads, fences, field boundaries with human eye. As a result of this, the reliability will be determined more accurately.

4.3.1. Image Smoothing

The image is first smoothed with an average low pass filter. The average filter computes the mean (average) of the grey-level values within a rectangular filter window surrounding each pixel. This has the effect of smoothing the image (eliminating noise).

Average filter performs spatial filtering on each individual pixel in an image using the grey-level values in a square or rectangular window surrounding each pixel. The dimensions of the filter size must be odd and can be between 1x3 or 3x1 to 1001x1001. The filter size must not exceed the database size.(PCI Geomatica users' manual, 2001)

All pixels are filtered. In order to filter pixels located near the edges of the image, edge pixel values are replicated to give sufficient data.

```
+-----+
|a1 a2 a3 |
|a4 a5 a6 |  <---  Filter window 3 X 3
|a7 a8 a9 |
+-----+
```

The averaging filter calculates the sum of all pixels in the filter window, then divides by the number of pixels in the filter window:

$$\text{Filtered pixel } r = (a1 + a2 + \dots + a9) / 9$$

4.3.2. Directional Filtering

For the manual extraction process, directional filtering of the image in N-S, E-W, NE-SW and NW-SE directions are preferred. The directional filters in four principal directions are given in Table 4.2 and the samples of filtered images are given in Figure 4.3-6.

Table 4.2. Directional filter kernels

N-S			NE-SW			E-W			NW-SE		
-1	0	1	-2	-1	0	-1	-2	-1	0	1	2
-2	0	2	-1	0	1	0	0	0	-1	0	1
-1	0	1	0	1	2	1	2	1	-2	-1	0

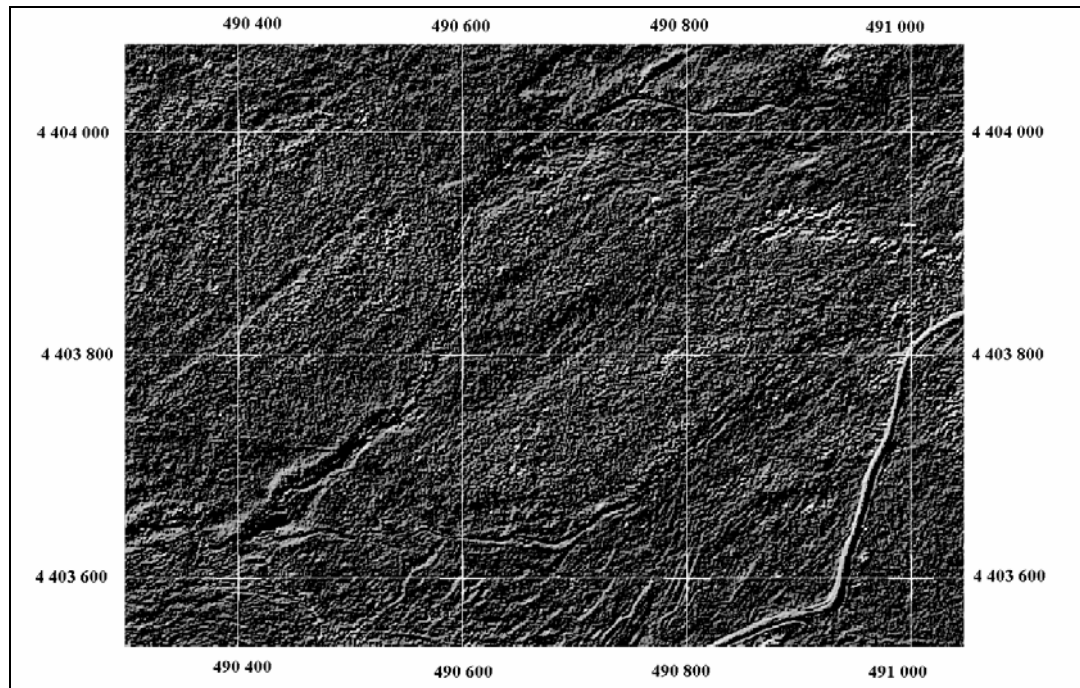


Figure 4.3. Filtered Image in N-S direction

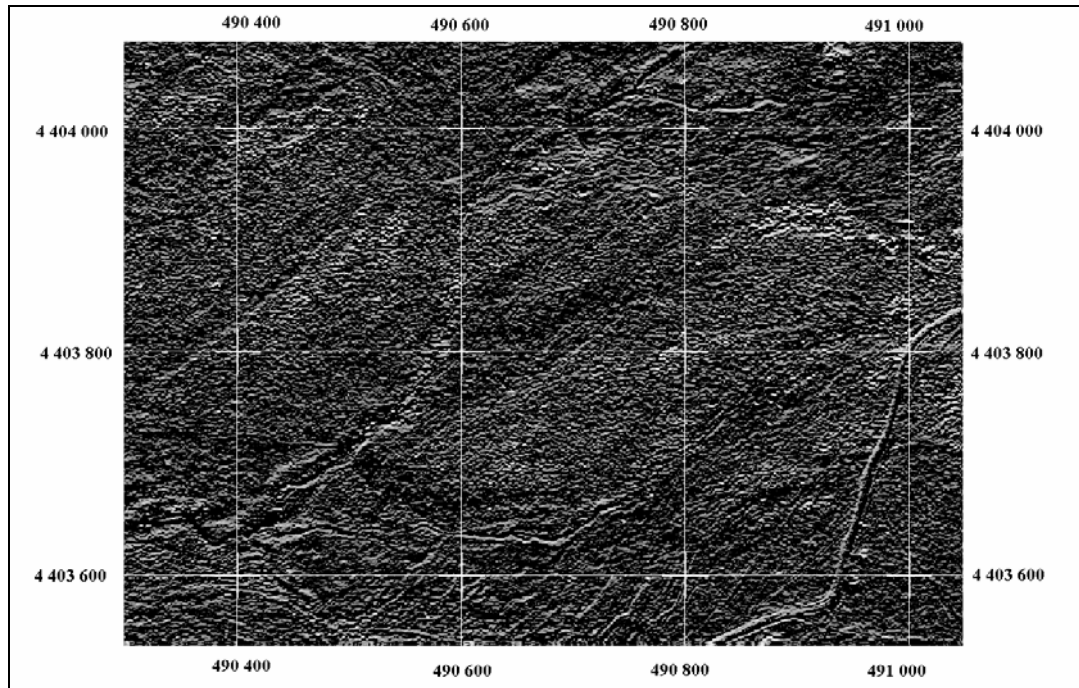


Figure 4.4. Filtered Image in NW-SE direction

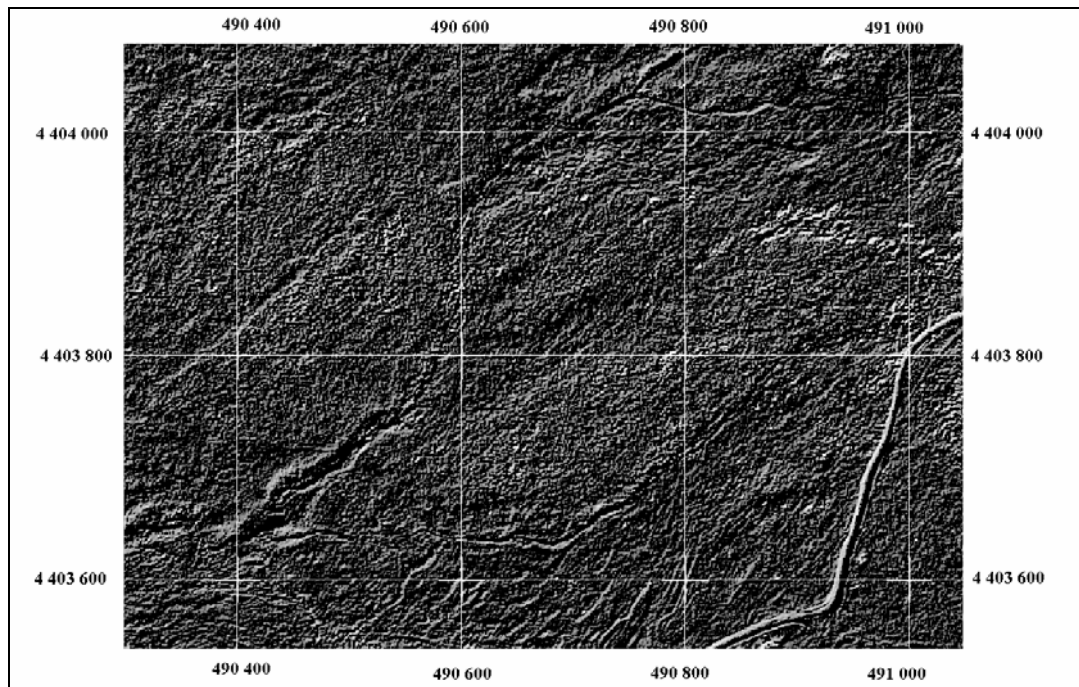


Figure 4.5. Filtered Image in E-W direction

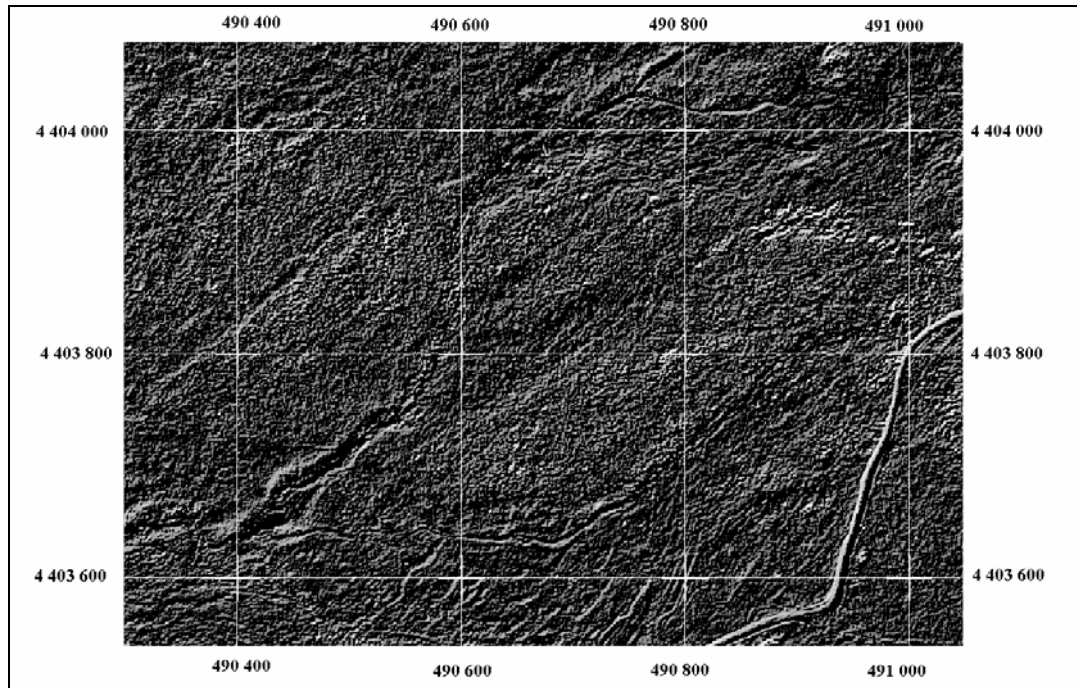


Figure 4.6. Filtered Image in NE-SW direction

4.3.3. Digitizing the Lineaments

It is mentioned before that a lineament in an aerial photo or a space image can show up either darker pixels in the middle and lighter on both sides: or, is lighter on one side and darker on the other side. When directional filtering is applied to the image the lineaments show up in light color which is surrounded by dark colored pixels.

After the filtering operation, the lineaments are digitized manually with a scale of 1:1500. First, the image that is filtered in N-S direction is digitized. Following that, the lineaments in the E-W filtered image are added to the previously extracted lineaments in the N-S filtered image. Similarly the lineaments in the NW-SE and NE-SW directions are added to the vector layer respectively. The result is the lineament map of the study area. As the digitizing is done manually the non-geological lineaments are not considered. After digitizing the

lineaments total number of 3065 lineaments with total 130040 m in length. The manually extracted lineament map of the area is shown in Figure 4.7.

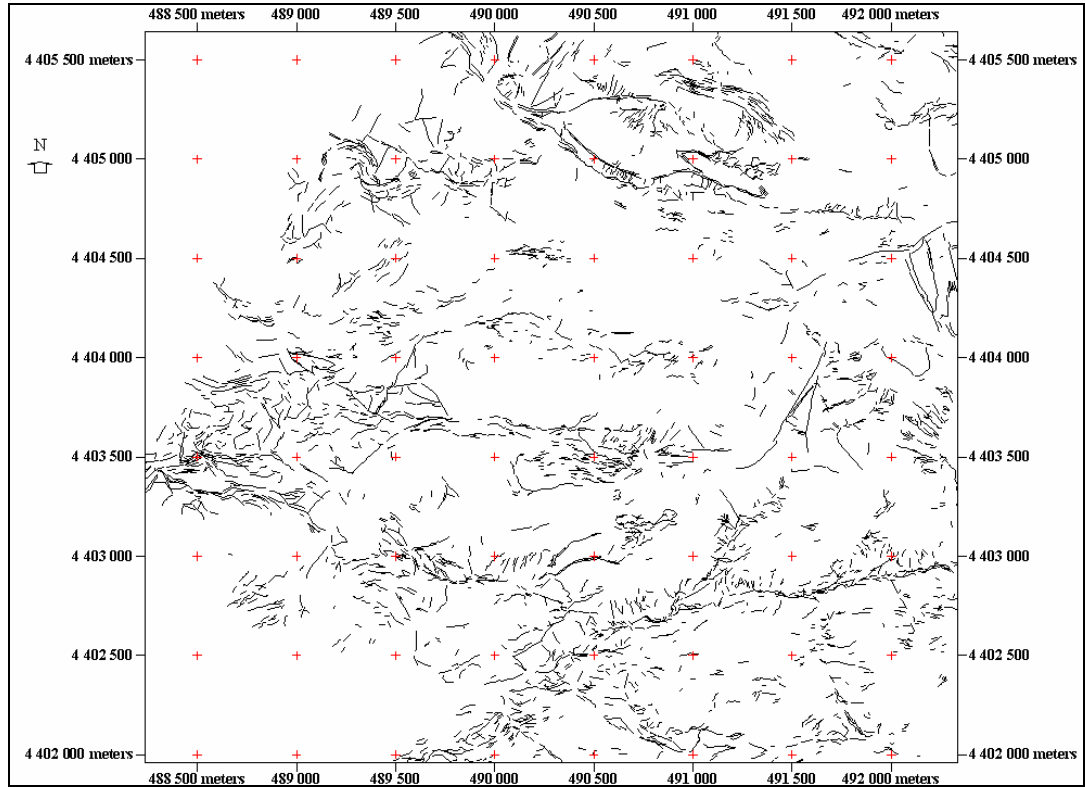


Figure 4.7. The manually extracted lineament map of the area of interest

The raw image contains agricultural fields in the north west part and village houses in the south west part. Because of this an area of interest is produced by excluding these areas by digitizing. In the reference lineament map, that is produced by manually, a boundary for the area of interest is formed. The automatically extracted lineaments are overlaid with the reference data and the lineaments of automatic extraction process in the restricted area are deleted.

Not only the field boundaries and village but also the mine roads are seen as a lineament in the automatically extraction process. Deleting the vectors that corresponds to the roads of the mines will increase accuracy of measuring the applicability of the methodology in terms of detecting the discontinuities.

For the comparison of the manually and automatically extracted lineament maps, the automatically extracted lineament maps with their total numbers are within the 10 % of manually extracted lineament map are selected for further applications. From this elimination LINE 1, LINE 2, LINE 5, LINE 16, LINE 19, LINE 27 and LINE 29 are selected for analysis.

CHAPTER V

ACCURACY ASSESSMENT

5.1. Introduction

In order to verify the methodology's applicability, the results obtained from remote sensing analysis need to be checked in some manner. To test the applicability two different perspectives are considered. The primary application is to compare the results with previously determined ones and field studies. The second one is comparison of manually digitized discontinuity map with automatically extracted lineament map.

In the first method, rose diagrams of automatically and manually extracted lineament maps are compared and the study of Karpuz (1982), as well as field investigations in the study area are utilized. Moreover, near-face lineaments are compared with the face discontinuities of an abandoned quarry. This approach is mainly for analyzing the directional accuracy of the lineaments.

In the second method, the accuracy of the automatic detection algorithm is computed by rasterizing both manually and automatically extracted maps and then arithmetically finding the differences between the cell values of rasterized images. In the second approach, accuracy of the location and length of lineaments are assessed.

After examining the results obtained from the above operations, the applicability of the methodology is assessed.

5.2. Directional Accuracy

The directional analysis of the lineaments are done in TNT mips v.6.8 software. For the analysis, rose diagrams are produced. The rose diagram is logically a 180 degree arc. The rose diagram window displays it in a 360 degree presentation that is centrally symmetric. For purposes of directional analysis, the start/end order of nodes in a line element is ignored. Thus for example, a line at a 45 degree angle and a line at a 225 degree angle have an equivalent orientation. (TNTmips 6.8 User's Manual, 2003)

For the rose diagram applications, the 'Node to Node' method is used in order to evaluate a lineament independently from the line segments forming it. This method uses the direction of a simple line segment that connects the endpoints (nodes) of the line element. Node to Node gives satisfactory results because the extracted lineaments do not have a 'fish hook' or 'horseshoe' shaped lines. Simple illustration of the method is given in Figure 5.1.



Figure 5.1. Simple illustration of Node to Node Method

The sample rose diagrams for automatically extracted lineaments are shown in Figure 5.2-8. The rest of the rose diagrams are given in Figure B1-B22 in Appendix B.

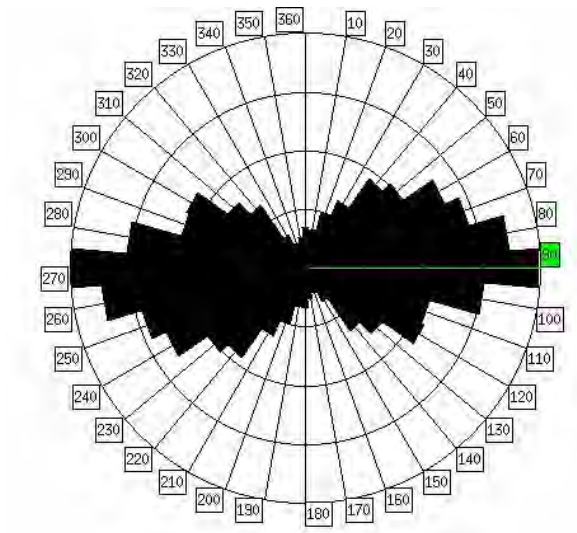


Figure 5.2. Rose Diagram of LINE 1

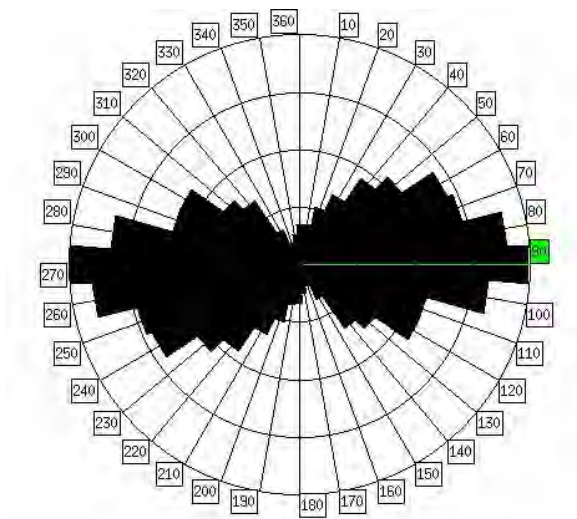


Figure 5.3. Rose Diagram of LINE 2

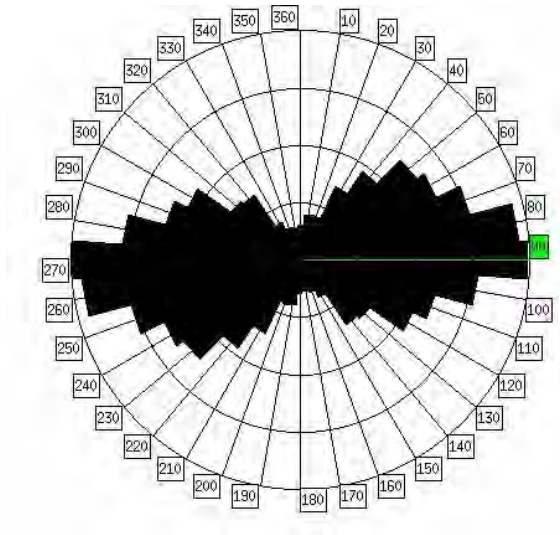


Figure 5.4. Rose Diagram of LINE 5

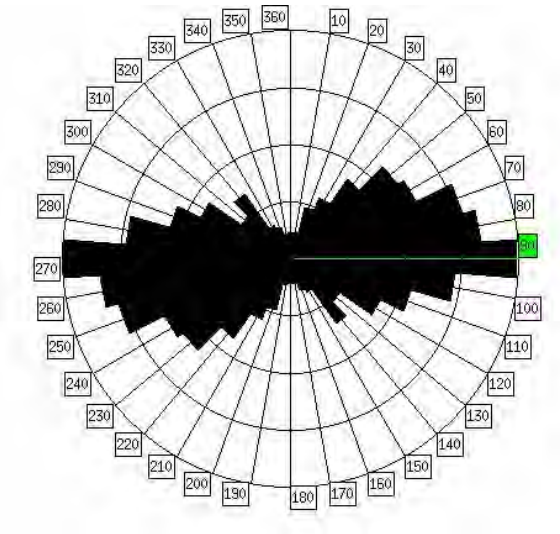


Figure 5.5. Rose Diagram of LINE 16

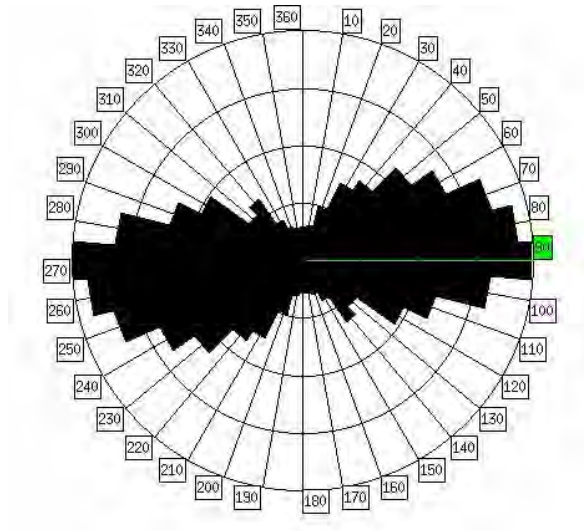


Figure 5.6. Rose Diagram of LINE 19

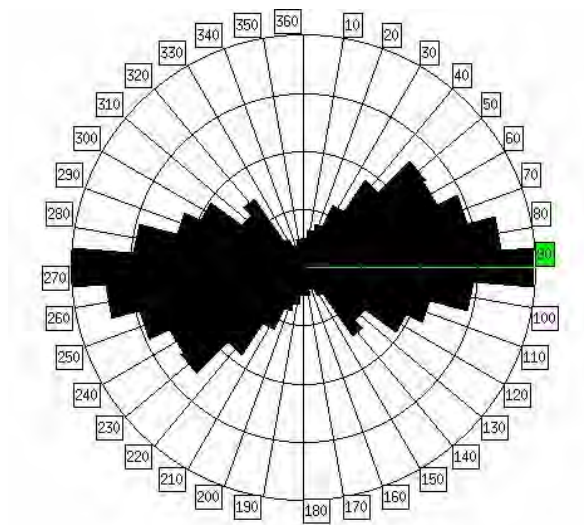


Figure 5.7. Rose Diagram of LINE 27

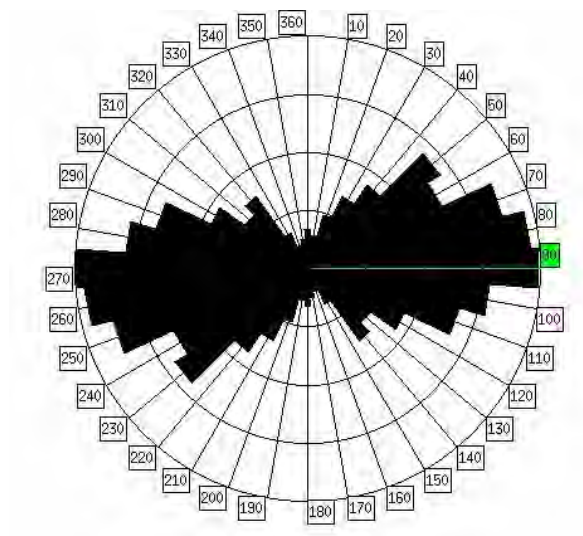


Figure 5.8. Rose Diagram of LINE 29

The rose diagram of the manually extracted lineaments is shown in Figure 5.9.

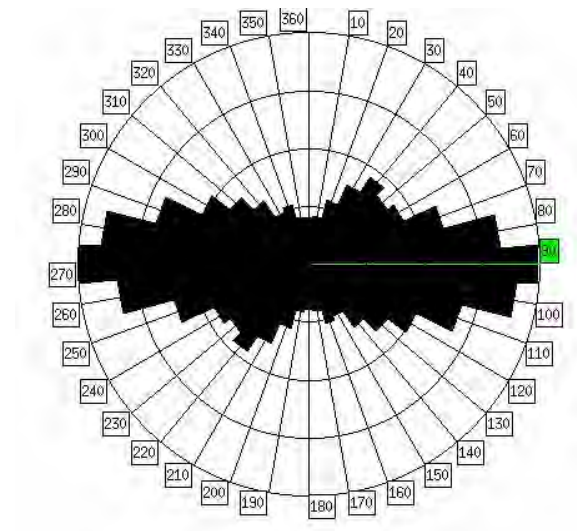


Figure 5.9. Rose Diagram of Manually Extracted Lineaments

Rose diagrams shows the similarity in major discontinuity orientations. The reference lineament map, which is drawn manually has the major orientation of E-W. Similarly the automatically extracted lineament maps has the same major orientation.

Besides the similarities of the rose diagrams, a site investigation is necessary for more accurate analysis. The study of Karpuz (1982) is considered for obtaining the major discontinuity orientations. The orientations are found from the pole distribution data defined by Karpuz (1982). The pole distribution is shown in Figure 5.10.

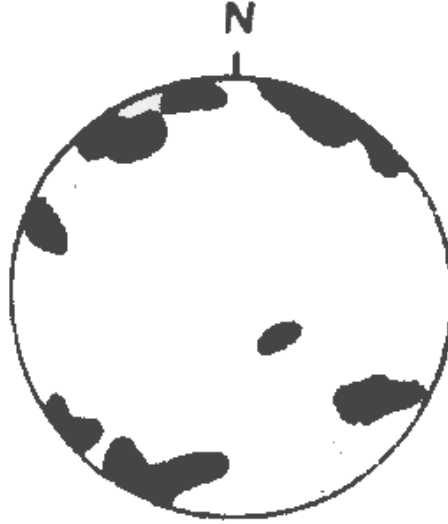


Figure 5.10. Pole Distribution of the Gölbaşı District (After Karpuz, 1982)

From the pole distribution of the discontinuity sets, three major discontinuity orientation is determined as N 80° E, N 70° W and a minor set of N 20° E (Karpuz,1982). From these results, it can be easily seen that the results of the study of Karpuz (1982) and the extracted lineaments have the same orientation. Except one set that is accepted as the minor discontinuity set in the field.

For another application to verify the directional accuracy of the lineaments, a study in mining sites are carried out by extracting face discontinuity sets of a quarry and comparing the strikes with the directions of the manually extracted lineaments from Ikonos image, which appear near the quarry. From the field studies carried out, the most suitable quarry for detecting lineaments is found for further analysis. The quarry is selected to be an abandoned quarry in order not to detect a change from the capturing time of the image and the field studies.

First application of this procedure is capturing the face photograph of the quarry with a 2.0 megapixel digital camera from coordinates of 4 404 212 N – 489 200 E. The raw image of the quarry face is given in Figure 5.11. Secondly, the face discontinuities are extracted on the image by the same image processing techniques that were done for manually extraction of lineaments from Ikonos image, that is directional filtering. Except the E-W direction, all three main directions (N-S, NW-SE, NE-SW) are considered. E-W direction is not considered because of the reason that horizontal lineaments are difficult to match with the surrounding lineaments. Thirdly, the dip amount and dip direction of the face discontinuities are determined. Finally, from the determined dip direction, the strike of the discontinuities, which are the lineament directions found from Ikonos image, are determined, and compared with the direction of lineaments detected manually from the Ikonos image. The digitized face discontinuity sets are given in Figure 5.12.

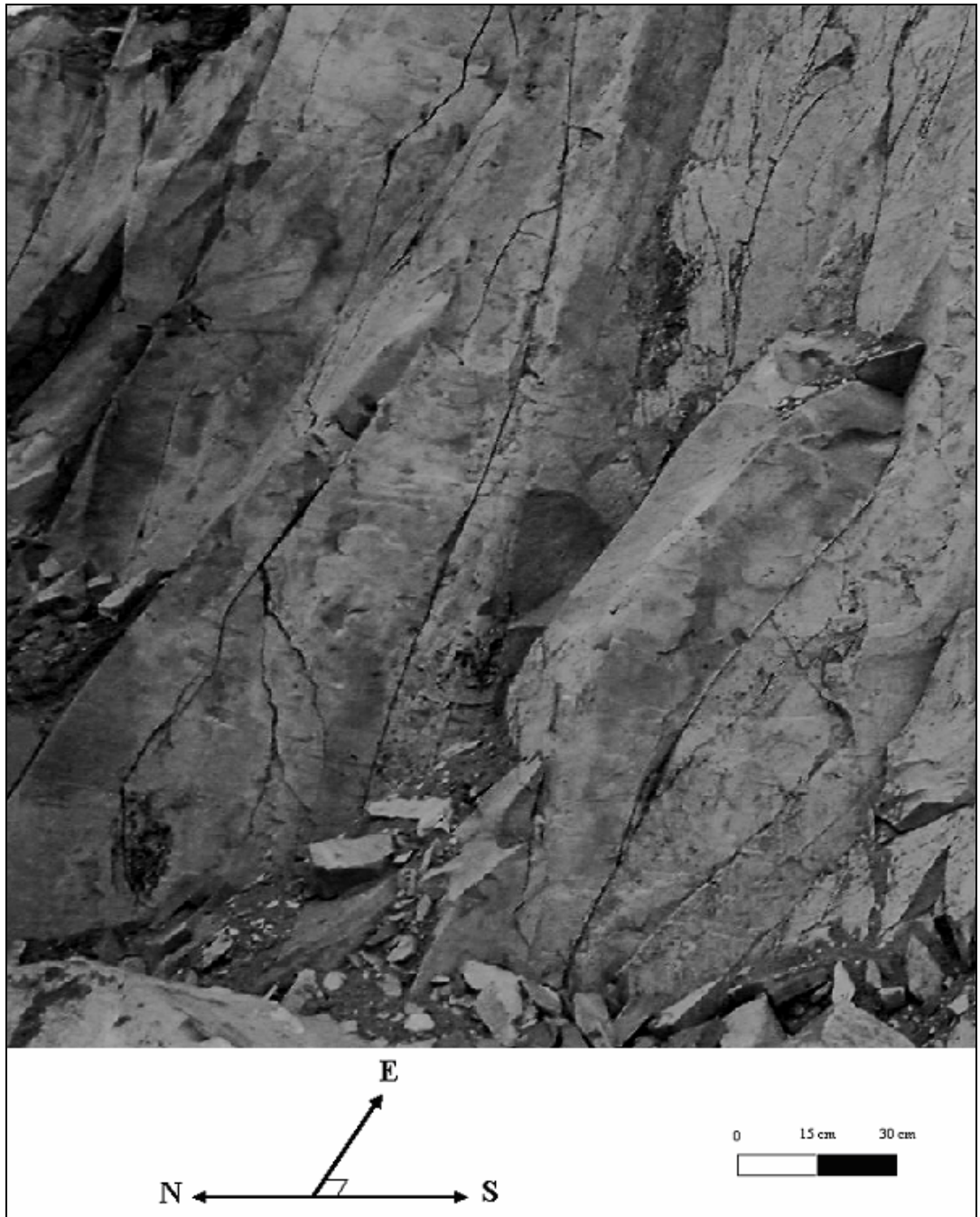


Figure 5.11. Raw Image of the Quarry Face

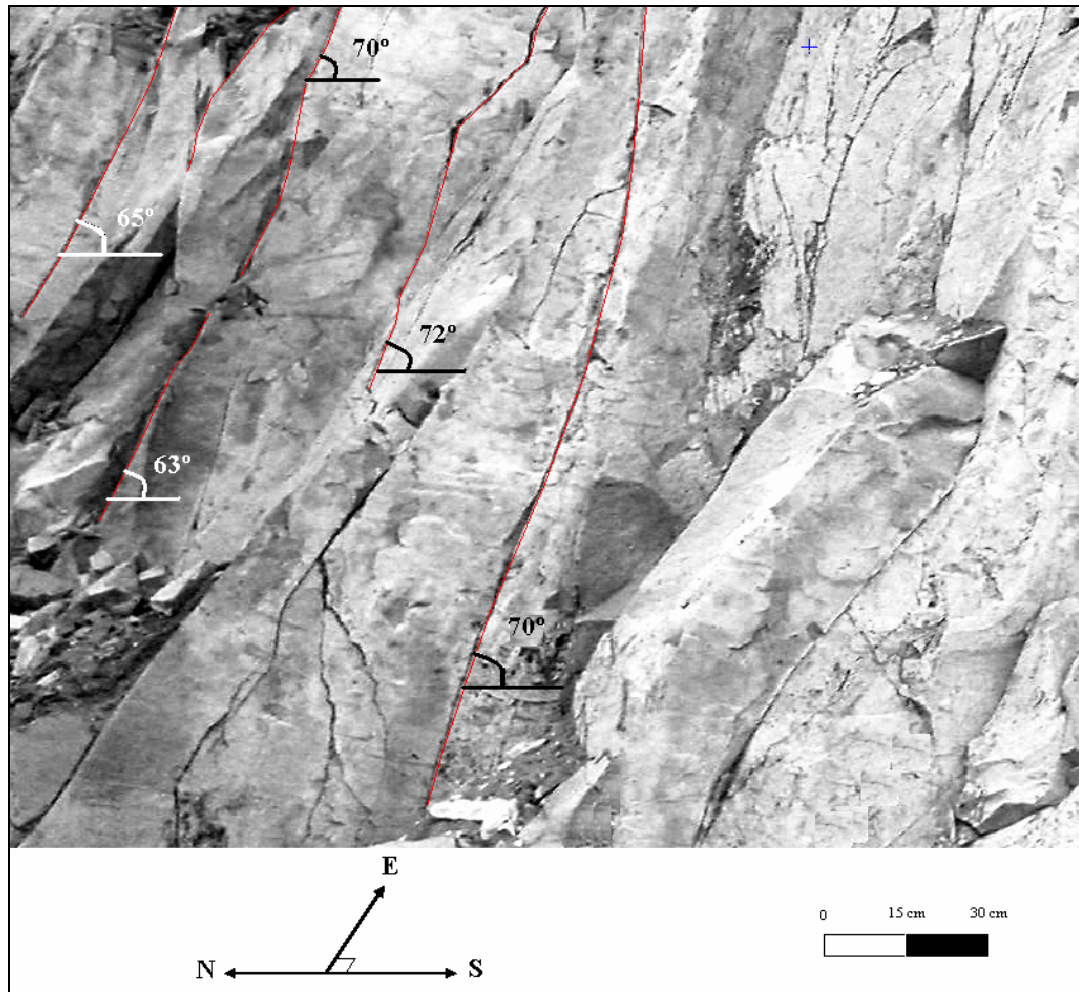


Figure 5.12. Lineaments of the Quarry Face

From the field studies carried out, the dip angles of the face discontinuities are found to be nearly 70° and a dip direction of 190° . From the dip directions the strikes are found as adding or subtracting 90° . According to this procedure the strike of the discontinuities are found to be $N\ 80^\circ\ W$ (or $S\ 80^\circ\ E$), which also verifies the near face lineament strikes that are shown in Figure 5.13. From the analyses carried out, it is seen that the digitized lineaments near the quarry corresponds to rock discontinuities.



Figure 5.13. Near face lineaments

5.3. Location and Length Based Accuracy

This section basically deals with evaluation of the matching level of the manually and automatically extracted lineament maps. To perform such an accuracy assessment, it is necessary to compare two sources of information: (1) the automatically extracted lineaments and (2) manually extracted lineaments which is considered to be the reference information. For this purpose an algorithm in this thesis is developed.

5.3.1. Algorithm of the Methodology

The resultant lineament maps are in vector format. For the arithmetical evaluation of the lineaments, the maps are converted into raster format. Therefore, each pixel forming the lineament is represented by a digital number.

The vector layers are rasterized in TNT mips software and for the manually extracted map, digital number of 10 is assigned to the vectors. The result is a raster map with pixel value 10 for lineaments and 0 for the others. Similarly for the automatically extracted vector maps, a raster map with pixel value 10 for lineaments and 0 for the others are formed.

In order to make an accuracy assessment it is not necessary to compare only the numbers found by two different methods. As the location of each lineament is important, it should be compared with respect to their location. The location of the lineaments are defined as the pixels addresses. The raster maps obtained from vector rasterization has a size of 3685 x 4104 (row x column) pixels. Every lineament pixels has an address $i \times j$, i for row number and j for column number.

For the comparison of the two data, first the pixel addresses should be determined. For that reason a computer program called LINECOMP is developed in java environment. LINECOMP has three modules. Those are LINE ADDRESS for identification of pixel addresses of lineaments, LINE COMPARE for comparing two lineament maps with respect to their lineament pixels and LINE CLASSIFICATION for classifying the results with respect to the matching amount of the lineaments. The algorithm of LINECOMP is given in Figure 5.14.

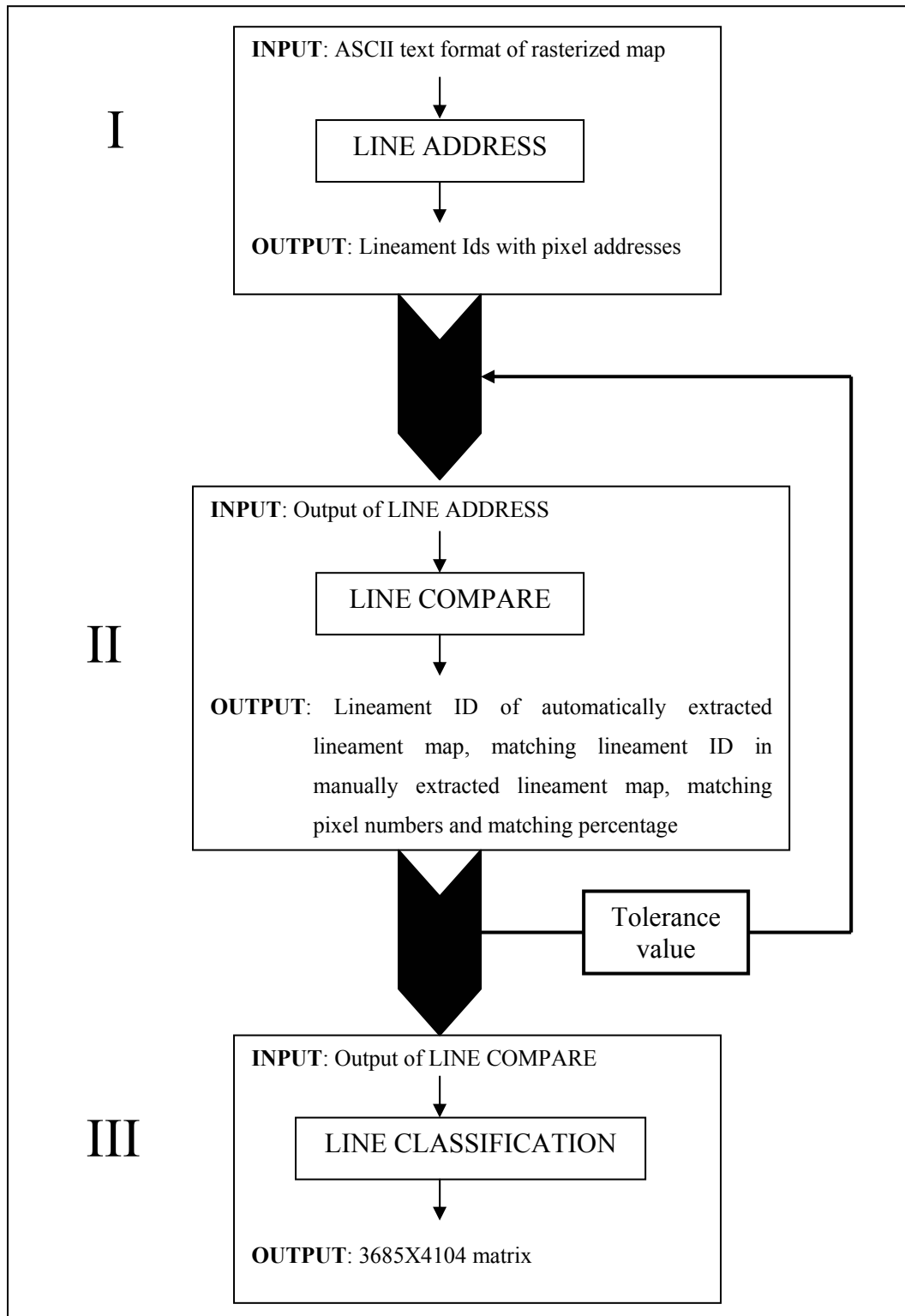


Figure 5.14. Algorithm of LINECOMP

5.3.1.1. Algorithm of LINE ADDRESS

The algorithm of the LINE ADDRESS is simply as follows:

- 1- Scan the lineament map in raster format
- 2- Catch the first pixel with value 10 and note the address as Lineament 0 = $[i_1, j_1]$
- 3- Search the neighbourhood of that pixel and find the other 10 and add that pixels' address to Lineament 0 as Lineament 0 = $[i_1, j_1], [i_2, j_2]$
- 4- Repeat steps 2 and 3 until there is no neighbour pixel with value 10.
- 5- Continue to scan the image from the first pixel for the other lineaments.
Do not record one lineament more than once.
- 6- Record the number of lineaments and pixel addresses of each lineament.

For a special case of intersecting lineaments, LINE ADDRESS detects the two or more intersecting lineaments as one lineament. Simple illustration of this case is given in Figure 5.15.

$\begin{matrix} j \\ i \end{matrix}$	1	2	3	4	5	6	7	8
1	0	0	10	0	0	0	10	0
2	0	0	0	10	0	10	0	0
3	0	0	0	0	10	0	0	0
4	0	0	0	10	0	10	0	0
5	0	0	10	0	0	0	10	0
6	0	0	0	0	0	0	0	0

OUTPUT

Lineament_1
[1,3][2,4][3,5][4,4][5,3][2,6]
[1,7][4,6][5,7]

Figure 5.15 Simple Illustration of Intersecting Lineaments in LINE ADDRESS

The second step is to compare the two lineament maps. For this purpose the output of LINE ADDRESS is used as input of LINE COMPARE.

5.3.1.2. Algorithm of LINE COMPARE

After the pixel addresses are determined, the automatically extracted lineament maps are compared with the manually extracted lineament map with the second module LINE COMPARE. The output gives the percent matching of lineaments. The algorithm of the LINE COMPARE part is simply as follows:

- 1- The first vector in the automatically extracted lineament map is compared with all of the lineaments in the reference map and the one that has maximum number of matching pixels is selected.
- 2- The matching lineament id in automatically extracted lineament map, number of pixels, matching lineament id in manually extracted lineament map, matching pixel numbers and percent of matching are recorded.
- 3- The same procedure is applied until the last lineament in the automatically extracted lineament map is compared.

A sample output of LINE COMPARE is given in Table 5.1.

Table 5.1. Sample Output of LINE COMPARE

Lineament Id	Number of pixels	Lineament Id in the reference	Number of matching pixels	Matching percentage
1	30	2	10	33
2	100	4	70	70
3	5	8	2	40
4	6	9	4	67

A lineament in the reference map can be divided into one or more lineaments in the other map. That is if a lineament in reference map covers more than one lineament, it is compared with both of the lineaments. Simple illustration

<div><div>j \ i</div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div><div>7</div><div>8</div></div></div>								
<div>1</div>	0	0	10	0	0	0	0	0
<div>2</div>	0	0	0	10	0	0	0	0
<div>3</div>	0	0	0	0	10	0	0	0
<div>4</div>	0	0	0	0	0	10	0	0
<div>5</div>	0	0	0	0	0	0	10	0
<div>6</div>	0	0	0	0	0	0	0	10

Reference Matrix

<div><div>j \ i</div><div><div>1</div><div>2</div><div>3</div><div>4</div><div>5</div><div>6</div><div>7</div><div>8</div></div></div>								
<div>1</div>	0	0	10	0	0	0	0	0
<div>2</div>	0	0	0	10	0	0	0	0
<div>3</div>	0	0	0	0	10	0	0	0
<div>4</div>	0	0	0	0	0	0	0	0
<div>5</div>	0	0	0	0	0	0	10	0
<div>6</div>	0	0	0	0	0	0	0	10

Matrix to be Compared

OUTPUT

Lineament Id	Number of pixels	Lineament Id in the reference	Number of matching pixels	Matching percentage
1	3	1	3	100
2	2	1	2	100

of this case is given in Figure 5.16.

Figure 5.16. Illustration of Matching of Divided Lineaments

For eliminating the small pixel address differences of the vectors, a tolerance value option is available in the program. The user defined tolerance value is the number of surrounding pixels of the manually extracted lineaments that is accepted as the vector pixel. The simple illustration of this algorithm is shown in Figure 5.17 as a 8x6 matrix.

$\begin{smallmatrix} j \\ i \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	0	0	0	0	10	10	10	0
2	0	0	0	10	10	10	0	0
3	0	0	0	0	10	10	10	0
4	0	0	0	0	0	10	10	10
5	0	0	0	0	0	10	10	10
6	0	0	0	0	0	10	10	10

Reference matrix

$\begin{smallmatrix} j \\ i \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	10	0	0	0	0
3	0	0	0	0	10	0	0	0
4	0	0	0	0	0	10	0	0
5	0	0	0	0	0	10	0	0
6	0	0	0	0	0	0	0	0

Matrix to be compared

Figure 5.17. Sample illustration of Tolerance Value in LINE COMPARE Algorithm

The value 10 indicates the vector pixels. The 10 values that are bold indicates the surrounding pixels when the tolerance value is equal to 1. The summary of this example is 100 % matching.

After the matching percentages are determined, it is important to see the matching pixels in the image. For this purpose, the output of the LINE COMPARE is classified according to the percentages by LINE CLASSIFICATION. 0 % matching means that there is no vector in the manually extracted lineament map corresponding to that vector in automatically extracted lineament map and it will be mentioned as non-matching lineaments. 100 % matching gives the result of completely matching vectors and it will be mentioned as perfectly matching lineaments. If the matching percent is between 0 % and 100 % matching means that automatically extracted lineaments are greater in length and it will be mentioned as longer lineaments. Finally values greater than 100 % means lineaments in automatically extracted map is smaller in length than manually extracted lineaments and it will be mentioned as shorter lineaments.

5.3.1.3. Algorithm of LINE CLASSIFICATION

After the classification of the results into four, the matrixes are again constructed according to the classification. The non-matching lineament pixels are valued as 1, the perfectly matching lineament pixels are valued as 2, longer lineament pixels are valued as 3 and pixels of shorter lineaments as 4. The non-lineament pixels are valued as 0 and the 3685x4104 matrix is formed in java.

The matrix is formed in ASCII text format. It is then imported to TNT mips software as a raster image with cell size of 1 meter. After the raster image is formed, it is important to see the classified pixels in the image. For this reason, different colors are assigned to different pixel values. The assigned colors for pixel values and a sample classified image are given in Figure 5.18 and Figure 5.19 respectively. The codes of LINE ADDRESS, LINE COMPARE AND LINE CLASSIFICATION is given in Appendix respectively.






	White → Pixel value = 0 (Non-lineament pixels)
	Red → Pixel value = 1 (non-matching lineaments)
	Black → Pixel value = 2 (perfectly matching lineaments)
	Green → Pixel value = 3 (longer lineaments)
	Blue → Pixel value = 4 (shorter lineaments)

Figure 5.18. Assigned Colors for Pixel Values

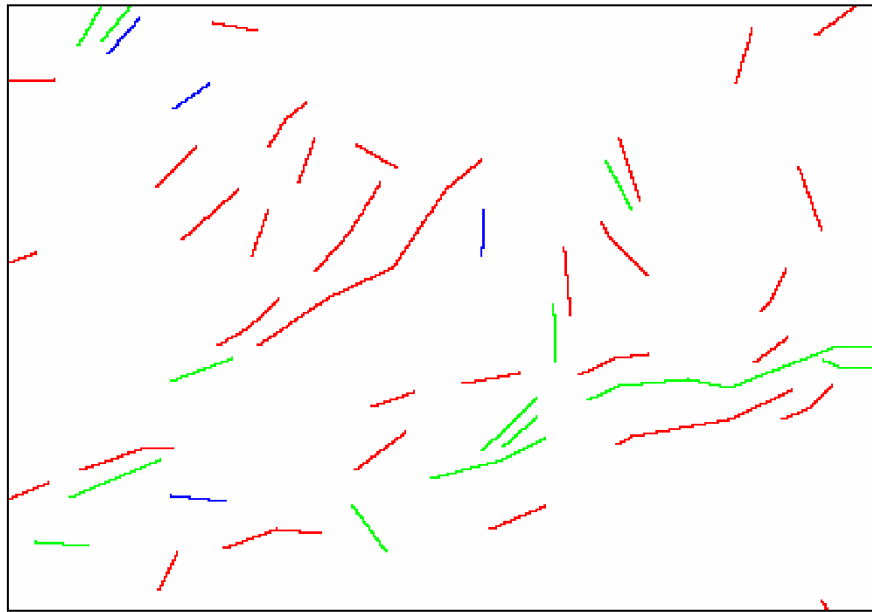


Figure 5.19. Sample Classified Map

In order to determine the most suitable parameters of Geomatica's LINE option the previously selected seven automatically extracted lineament maps (LINE 1, LINE 2, LINE 5, LINE 16, LINE 19, LINE 27 and LINE 29) are compared with manually extracted lineament map with LINE COMPARE with tolerance value 1. Following that, the upper most value of the tolerance value in this study is determined. The results of the comparison of the automatically extracted lineament maps with the manually extracted lineament map with tolerance value equal to 1 are given in table 5.2.

Table 5.2. Number of lineaments classified according to matching pixels with tolerance value 1

LINE ID	Number of lineaments	Number of non-matching lineaments	Number of perfectly matching lineaments	Number of longer lineaments	Number of shorter lineaments	Percent of non-matching lineaments
1	2883	1630	6	1010	237	56,54
2	3056	1703	11	1071	271	55,73

5	3312	2093	10	1056	150	63,19
LINE ID	Number of lineaments	Number of non- matching lineaments	Number of perfectly matching lineaments	Number of longer lineaments	Number of shorter lineaments	Percent of non- matching lineaments
16	2776	1873	2	855	46	67,47
19	3271	2266	2	936	67	69,28
27	2736	1720	7	920	89	62,87
29	3258	2009	9	1083	157	61,66

From the results, the automatically extracted lineament map with most suitable parameters is selected according to the total number of lineaments, number of perfectly matching lineaments and the percent of non-matching lineaments. For the total number of lineaments, a $\pm 5\%$ limit is assumed and only lineament map LINE 2 is in that interval. LINE 2 also gives the greatest number of perfectly matching lineaments. Again according to the percentage of non-matching lineaments, LINE 2 is one of the lineament map that has lower percentage of non-matching lineaments.

From these results, LINE 2 is selected as the most suitable automatically extracted lineament map in this study and is analyzed for further study to determine the upper most value of tolerance value for the study area of Gölbaşı andesite quarry region.

The matching percentages of LINE 2 with different tolerance values are given in Table 5.3.

Table 5.3. Matching Percentages of LINE 2 with Different Tolerance Values

Tolerance Value	Number of non-matching lineaments	Number of perfectly matching lineaments	Number of longer lineaments	Number of shorter lineaments	Percent of non-matching lineaments
0	2179	0	0	877	71,30
1	1703	11	1071	271	55,73
2	1448	14	1202	392	47,21
3	1257	11	850	938	40,41
4	1104	11	703	1238	36,13
5	1004	7	579	1466	32,49
6	918	5	490	1643	30,04
7	872	3	423	1758	28,53

In this study three zones (Zone A, Zone B, Zone C) are determined and special attention is given to those areas during the location based accuracy assessment. These areas are selected near the faults in the area. The zones are shown in Figure 5.20.

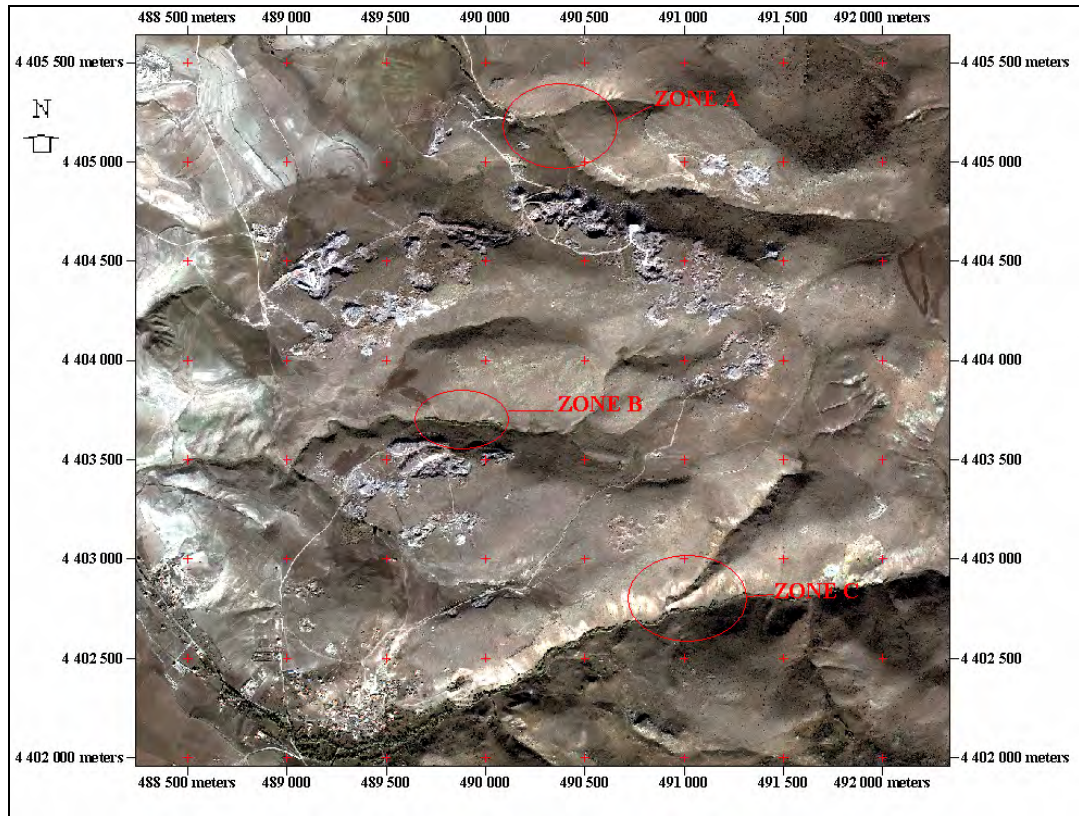


Figure 5.20. True Color Composite Image Showing Zones A, B and C

From the results of LINE COMPARE, the matrixes are formed by LINE CLASSIFICATION and the ASCII text format data is imported to TNTmips as a raster data. The resultant classified images in Zone A is given in Figures 5.21-28

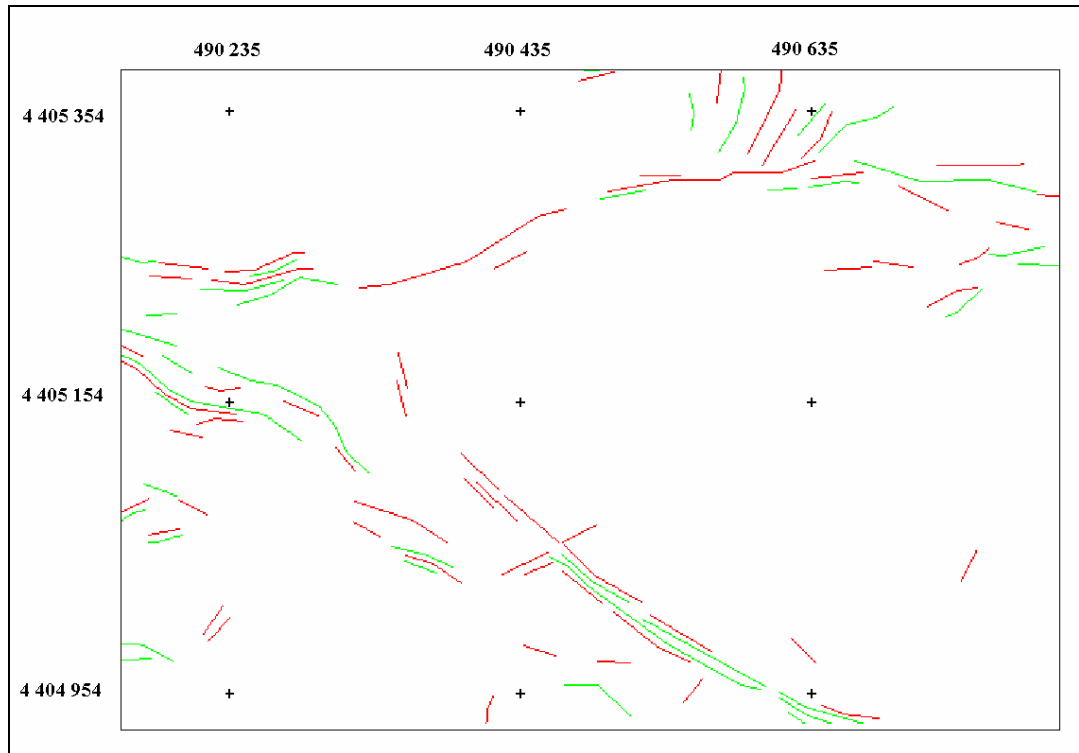


Figure 5.21. Classified Image of LINE 2 in Zone A with Tolerance Value = 0

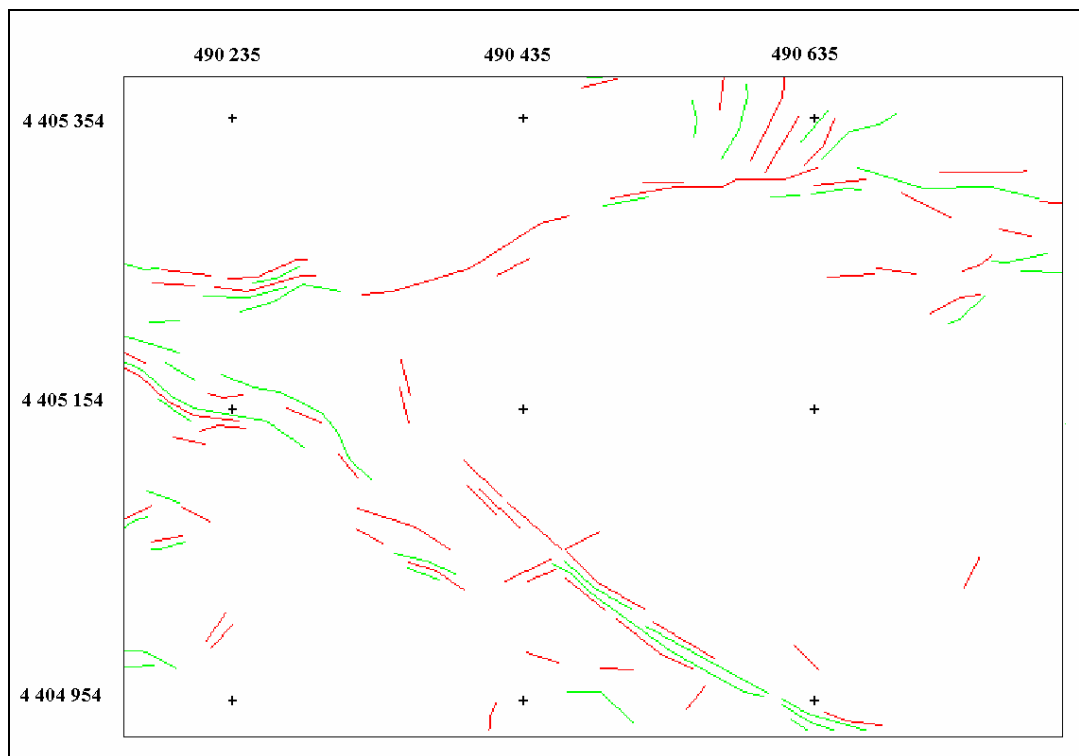


Figure 5.22. Classified Image of LINE 2 in Zone A with Tolerance Value = 1

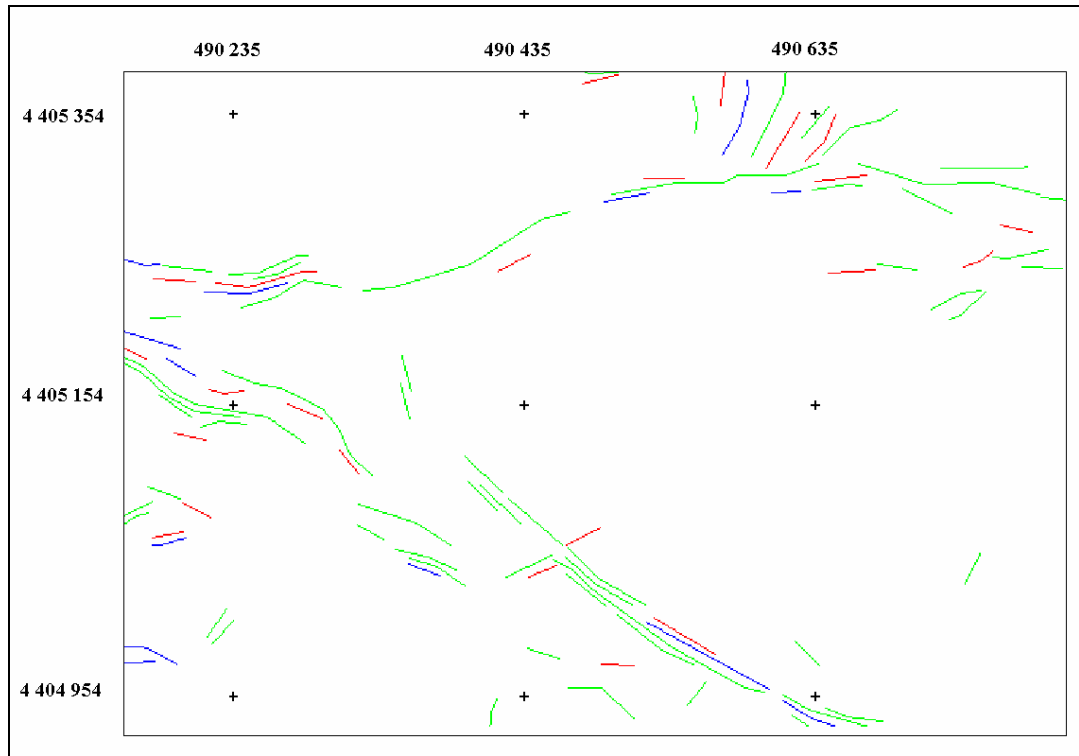


Figure 5.23. Classified Image of LINE 2 in Zone A with Tolerance Value = 2

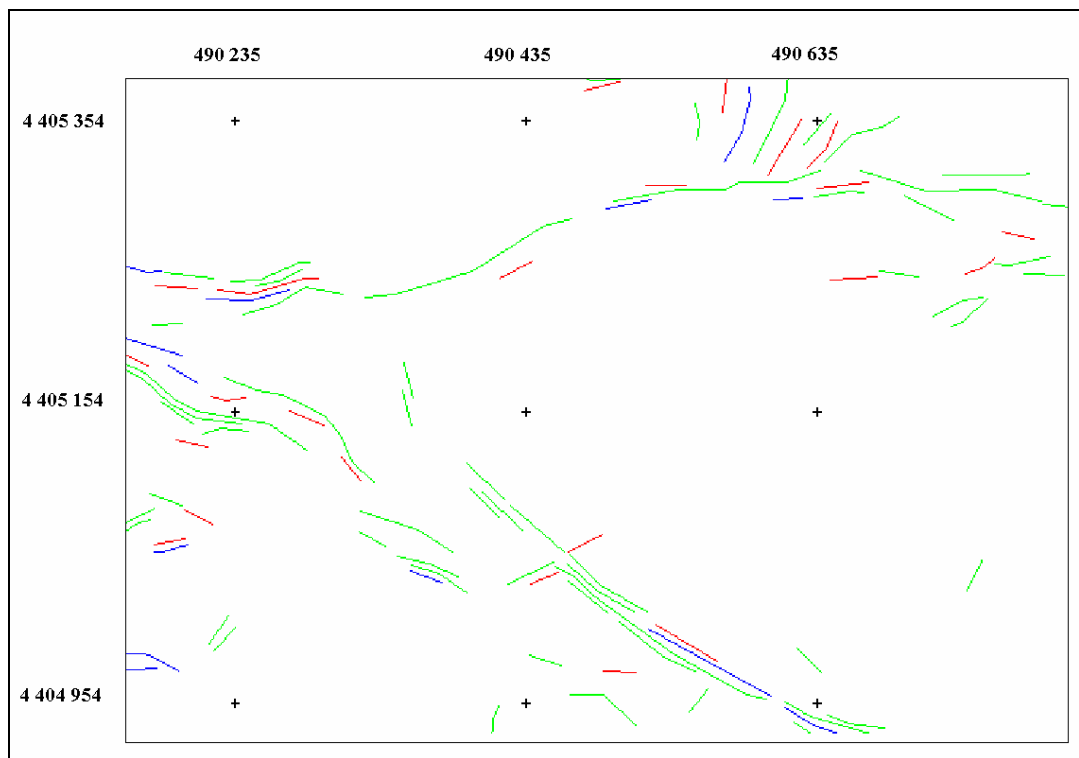


Figure 5.24. Classified Image of LINE 2 in Zone A with Tolerance Value = 3

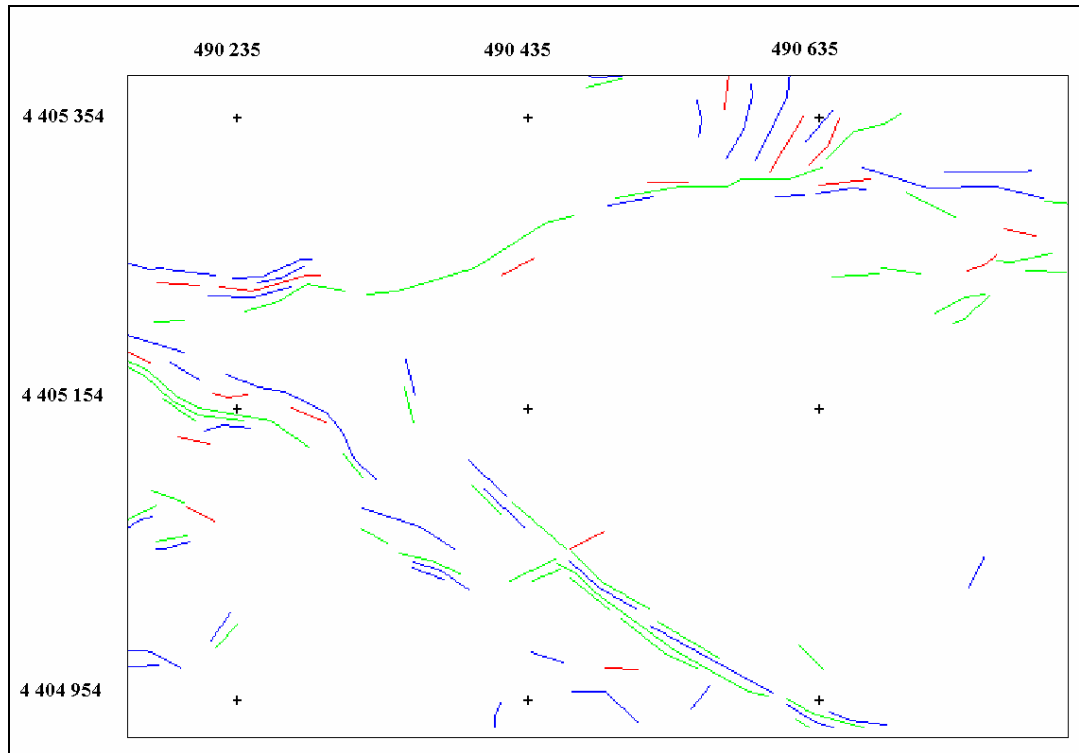


Figure 5.25. Classified Image of LINE 2 in Zone A with Tolerance Value = 4

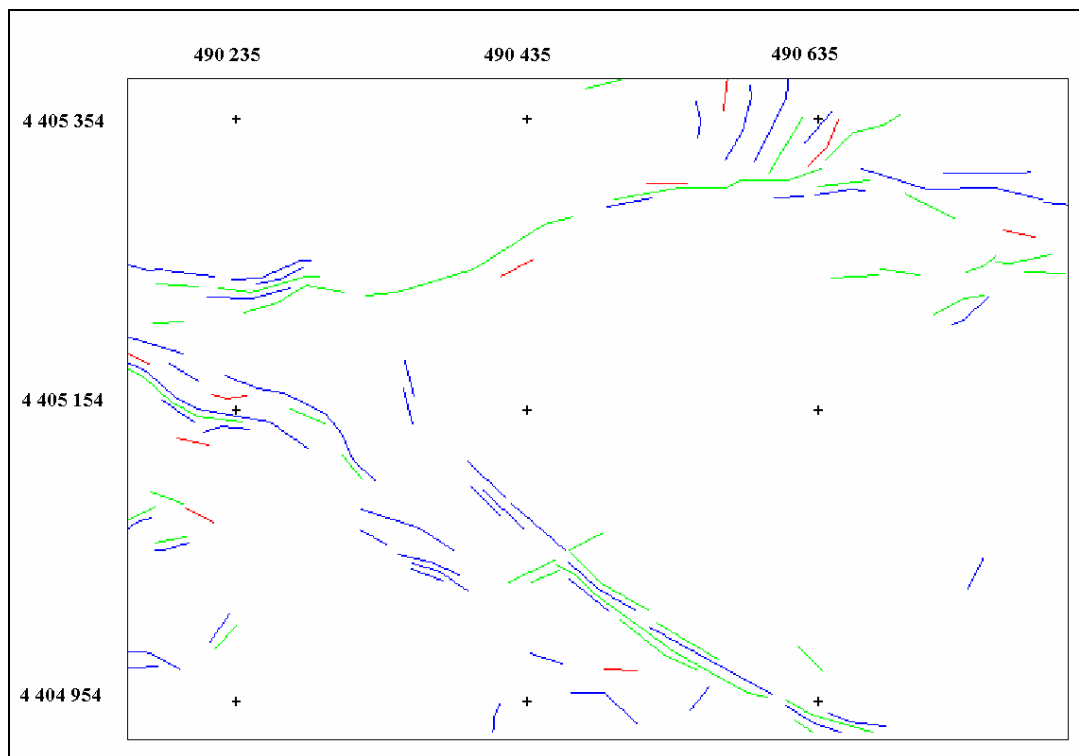


Figure 5.26. Classified Image of LINE 2 in Zone A with Tolerance Value = 5

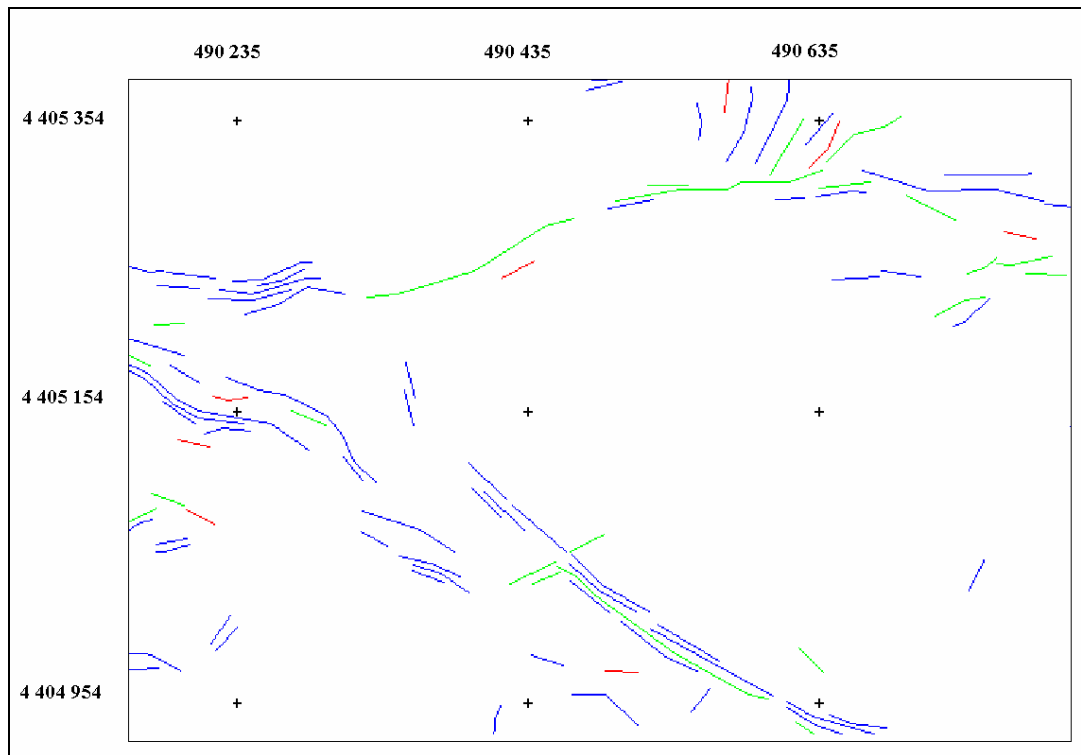


Figure 5.27. Classified Image of LINE 2 in Zone A with Tolerance Value = 6

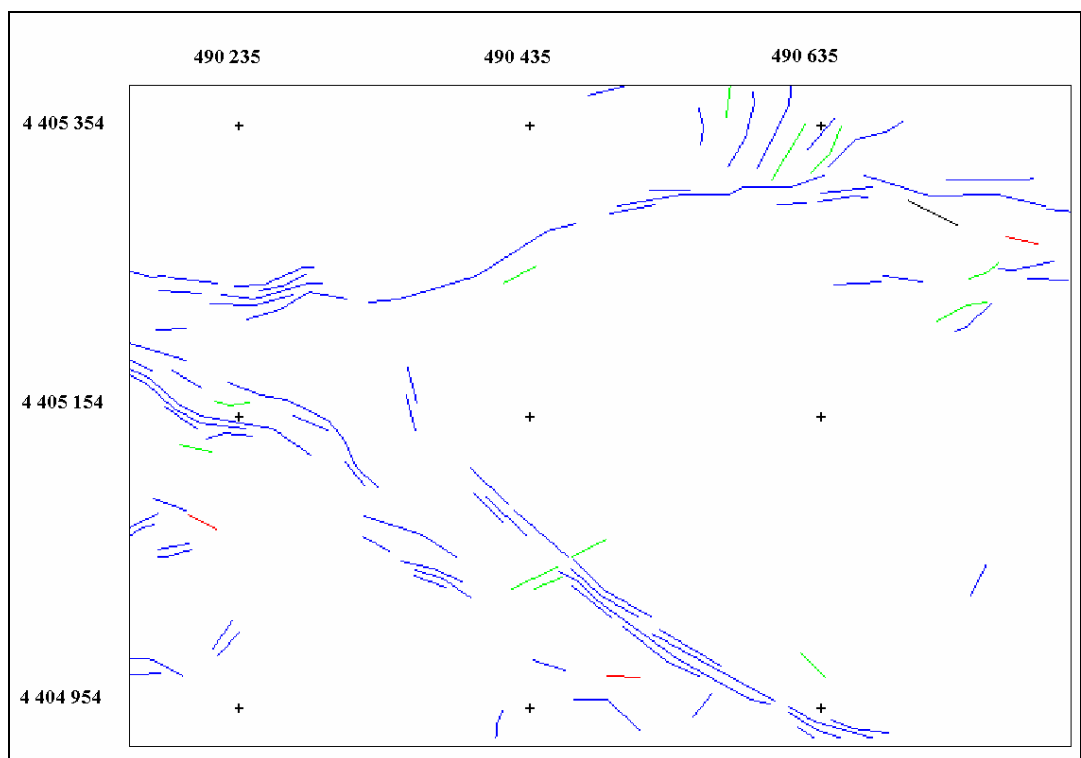


Figure 5.28. Classified Image of LINE 2 in Zone A with Tolerance Value = 7

The classified maps of LINE 2 in Zone B are given in Figures C1-C8 in Appendix C and in Zone C are given in Figures D1-D8 in Appendix D.

5.4. Discussion of the Results

From the analysis carried out to determine the accuracy and applicability of the automatic extraction of lineaments to determine rock discontinuities, it can be generally concluded that, the directional behaviour of the lineaments are appear to be similar.

Besides the general directional behaviour, the analytical comparison of the methodology gives more reliable results. First impression about the results can be obtained from the percentage of the non-matching pixels. As the number of the lineaments increase, the percentage of non-matching pixels also increase.

Another factor which effects the accuracy, is the tolerance value. As the tolerance value increases, the percent of matching pixels also increases. On the other hand, increasing tolerance value decreases the sensitivity of the analysis.

When the tolerance value is equal to 1, pixels with red color appear to be dominant in the classified image. As it is mentioned in Chapter IV, lineaments that refer to mine roads, field boundaries, etc. are deleted. Because of some possible errors occurred during the deleting of the lineaments red color in the classified image also appear near areas of those deleted lineaments. Green pixels appear to be widely spaced. Blue pixels have a sparse distribution especially in the three zones (Zone A, B and C).

The results when the tolerance value is increased are slightly different from those with tolerance is equal to 1. The number of red pixels decrease with respect to previous result. Green pixels are again widely spaced but the number decreases. The red and green pixels seem to become blue when the tolerance value is increased.

When the tolerance value is increased to 4, the red pixels continue to decrease. Similar to the previous stage, the number of blue pixels increases, as well as the number of red and green pixels are decreasing.

Increasing the tolerance value to 5 gives similar results to 4, the rate of decrease in percent of non-matching lineaments decrease. In addition to that the number of perfectly matching lineaments start to decrease.

With the tolerance value 6, similar to the previous stage, that is change of tolerance value from 4 to 5, the rate of change in non-matching lineaments are smaller to the previous stages.

Finally with the tolerance value 7, the green lineaments turn into blue with the same rate of change.

From the analyses carried out in the andesite quarry area of Gölbaşı District of Ankara, a tolerance value of 4 is recommended for matching the manually extracted lineaments with automatically extracted lineaments. From tolerance value 0 (one to one matching) to tolerance value 4 the rate of change of non-matching lineaments decrease considerably. Tolerance values higher than 4 yields with decreasing of the number of perfectly matching lineaments and the rate of change in non-matching lineaments is getting smaller. Also the sensitivity of the application decreases. With the smaller rate of change of non-matching lineaments, it can still be seen that green lineaments (longer) turn into blue by increasing tolerance value. It can be concluded that in this study, nearly all of the lineaments that can be matched are evaluated with tolerance value 4.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY

In this thesis, the possibility of using Remote Sensing (RS) techniques in mapping rock discontinuities with automatic lineament extraction and an accuracy assessment methodology was presented. For this purpose, the andesite quarry area of Gölbaşı district of Ankara was selected. In this study, Ikonos ortho-rectified Precision Plus satellite imagery with 1 meter resolution, which was captured in October 2002, was used.

6.1. Conclusions

Main conclusions derived from this study are as follows:

1. A methodology for using automatic lineament extraction for discontinuity mapping have been acceptably developed.
2. The directional analysis of the automatically extracted lineament maps have been done with the reference of manually extracted lineaments and previous studies about the area of research. The results verify the similarity in the directional behaviour of the lineaments.
3. With the field studies, face discontinuities are found to be related with the near face lineaments.

4. Besides a quantity based accuracy assessment, a location-direction-length based accuracy assessment methodology have been developed and a program called LINECOMP have been coded in Java environment.
5. From the accuracy assessment procedure, the most applicable parameters of PCI Geomatica's for this study have been identified. Those are as follows:
 - RADI = 5
 - GTHR = 20
 - LTHR = 20
 - FTHR = 2
 - ATHR = 20
 - DTHR = 1

This study served the purpose of determining rock discontinuities automatically and checking the accuracy of the automatic lineament extraction process of PCI Geomatica's LINE option.

6.2. Recommendations for Further Study

The recommendations for further studies are as follows:

1. Special attention can be given to the determination of the continuity of the discontinuities.
2. Different areas having different geotechnical properties may be studied to reach more generalized conclusions.
3. Mine roads decrease the accuracy of the methodology, an area with less roads, field boundaries etc. may be selected to get better accuracy of the methodology.

REFERENCES

- Clark, Chris D. and Wilson, Colin, 1994. "Spatial Analysis of Lineaments", Computers and Geosciences, Vol. 20, No. 7/8, 1237-1258.
- Cortes, A.L., Soriano, M.A., Maestros, A., Casas, A.M., 2003, "The role of tectonic inheritance in the development of recent fracture systems, Duero Basin, Spain", International Journal of Remote Sensing, Vol. 24, No. 22, 4325-4345.
- Elias, K.M.Mohamed, 2003, "Multiple data set integration for structural and stratigraphic analysis of Oil and Gas bearing formation using GIS", Map India Conference 2003, Geology and Mineral Resource.
- Geomatica 8.2 User's Manual, 2001
- Jain, A. K., 1989, "Fundamentals of Digital Image Processing", Englewood Cliffs, NJ: Prentice-Hall, 342-357.
- Jensen, J. R., 1996. "Introductory Digital Image Processing", Prentice Hall Series in Geographic Information Science, New Jersey, 316 pages.
- Kane, W., Peters, D.C., Speirer, R.A., 1996, "Remote Sensing In Investigation of Engineered Underground Structures", Journal of Geotechnical Engineering, Vol. 122, No. 8, 674-681.

- Karpuz, C., 1982, "Rock Mechanics Characteristics of Ankara Andesites in Relation to Their Degree of Weathering", PhD Thesis, METU Mining Engineering Department, Ankara, Turkey, 157 pages.
- Koike, K., Nagano, S. And Ohmi, M., 1995. "Lineament Analysis of Satellite Images Using A Segment Tracing Algorithm (STA)", Computers and Geosciences, Vol. 21, No. 9, 1091-1104.
- Koopmans, B.N., 1986, "A comparative study of lineament analysis from different remote sensing imagery over areas in the Benue Valley and Jos Plateau Nigeria", International Journal of Remote Sensing, Vol. 7, No. 12, 1763-1771.
- Leech, D.P., Treloar, P.J., Lucas, N.S., Grocott, J., 2003, "Landsat TM analysis of fracture patterns: a case study from the Coastal Cordillera of northern Chile", International Journal of Remote Sensing, Vol. 24, No. 19, 3709-3726.
- Mah, A., Taylor, G.R., Lennox, P. and Balia, L., 1995. "Lineament Analysis of Landsat Thematic Mapper Images, Northern Territory, Australia", Photogrammetric Engineering and Remote Sensing, Vol. 61, No. 6, 761-773.
- Majumdar, T.J., Bhattacharya, B.B., 1988. "Application of the Haar transform for extraction of linear and anomalous over part of Cambay Basin, India", International Journal of Remote Sensing, Vol, 9, No. 12, 1937-1942.
- Mat Akhir, J., Abdullah, I., 1997, "Geological Applications of LANDSAT Thematic Mapper Imagery: Mapping and Analysis of Lineaments in NW Peninsula Malaysia", Asian Association of Remote Sensing.

Minor, T.B., Carter, J.A., Chesley, M.M., Knowles, R.B., 1994, "An integrated approach to groundwater exploration in developing countries using GIS and remote sensing", American Society for Photogrammetry and Remote Sensing and the American Congress on Surveying.

NASA Remote Sensing Tutorial Web Page

<http://rst.gsfc.nasa.gov>

Novak, I. D. And Soulakellis, N., 2000. "Identifying geomorphic features using Landsat-5/TM Data Processing Techniques on Lesvos, Greece", *Geomorphology*, Vol. 34, No. 7, 101-109.

O'leary, D. W. Friedman, J. D., Pohn, H. A., 1976. "Lineament, linear, lineation: some proposed new standards for old terms", *Geological Society America Bulletin*, Vol.87, 1463-1469.

Personal interview with Assoc. Prof. Dr. Yusuf Kağan Kadioğlu

Pratt, W. K., 1991, "Digital Image Processing, 2nd Ed.", New York: Wiley, 698 pages.

Remote Sensing and Photo Interpretation Tutorial Web Page,
<http://mercator.upc.es/tutorial>

Richards, J. A., 1986, "Remote Sensing Digital Image Analysis", New York: Springer-Verlag, 281 pages.

Rowan, Lawrence C. and Bowers, Timothy L., 1995. "Analysis of Linear Mapped in Landsat Thematic Mapper and Side-Looking Airborne Radar Images of the Reno 1° by 2° Quadrangle, Nevada and California: Implications for Mineral Resource Studies", *Photogrammetric Engineering and Remote Sensing*, Vol. 61, No. 6, 749-759.

- Sabins, F. F., 1996, "Remote Sensing: Principles and Interpretation, 3rd Ed.: W. H. Freeman and Company, New York, 494 pages.
- Singh, A., and A. Harrison, 1985, "Standardized Principal Components", International Journal of Remote Sensing, 6: 883-896.
- Süzen, M.L., and Toprak, V., 1998. "Filtering of Satellite Images in Geological Lineament Analyses: An Application to a Fault Zone in Central Turkey", International Journal of Remote Sensing, Vol. 19., No. 19, 1101-1114.
- TNTmips 6.8 User's Manual, 2003
- Trenc, N., Morgan, K.M., Donovan, R.N. and Busbey, A.B., 1999, "Remote Sensing Analysis of Selected Terrestrial Impact Craters and a Suspected Impact Structure in South Korea using Space Shuttle Photographs", Geol. Croat., Vol. 52, No. 2, 203-215.
- Udomratn, C., 1987, "Geologic Application and Utilization of Remote Sensing Images in Thailand", Geological Survey Division, Department of Mineral Resources, Bangkok, 151 pages.
- Vanderbrug, G.J., 1976, "Line Detection in Satellite Imagery", IEEE Transactions on Geoscience Electronics, Vol. GE-14, No. 1, 37-44.
- Vassilas, N., Perantonis, S., Charou, E., Seretis, K. And Tsenoglou Th., 2000, "Automatic Lineament Detection from Geophysical Data Using Efficient Clustering and Weighted Hough Transform Algorithms"
- Wang, Jinfei, Member, IEEE, Howarth, Philip J., 1990. "Use of the Hough Transform in Automated Lineament Detection", IEEE Transactions on Geoscience and Remote Sensing, Vol. 28, No. 4, 561-566.

Williamson, D., 2003, “ Structural Analysis of the Boktukola Syncline / Boktukola Fault Complex Mountains, Using Side-Looking Airborne Radar and Landsat Thematic Mapper”

Won-In, K., Charusiri, P., 2002, “Enhancement of thematic mapper satellite images for geological mapping of the Cho Dien area, Northern Vietnam”, International Journal of Applied Earth Observation and Geoinformation, Vol. 15, 1-11.

APPENDIX A

In this appendix, the automatically extracted lineament maps are presented in Figure A1-A29.

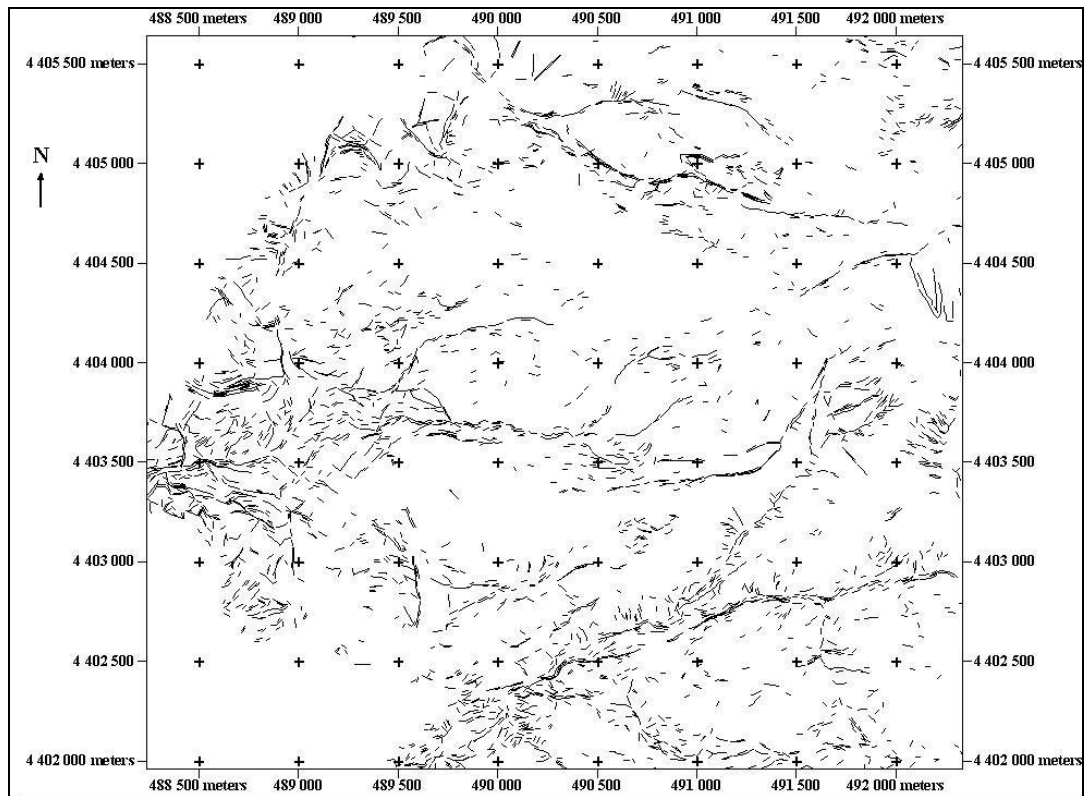


Figure A1. Lineament Map of LINE 1

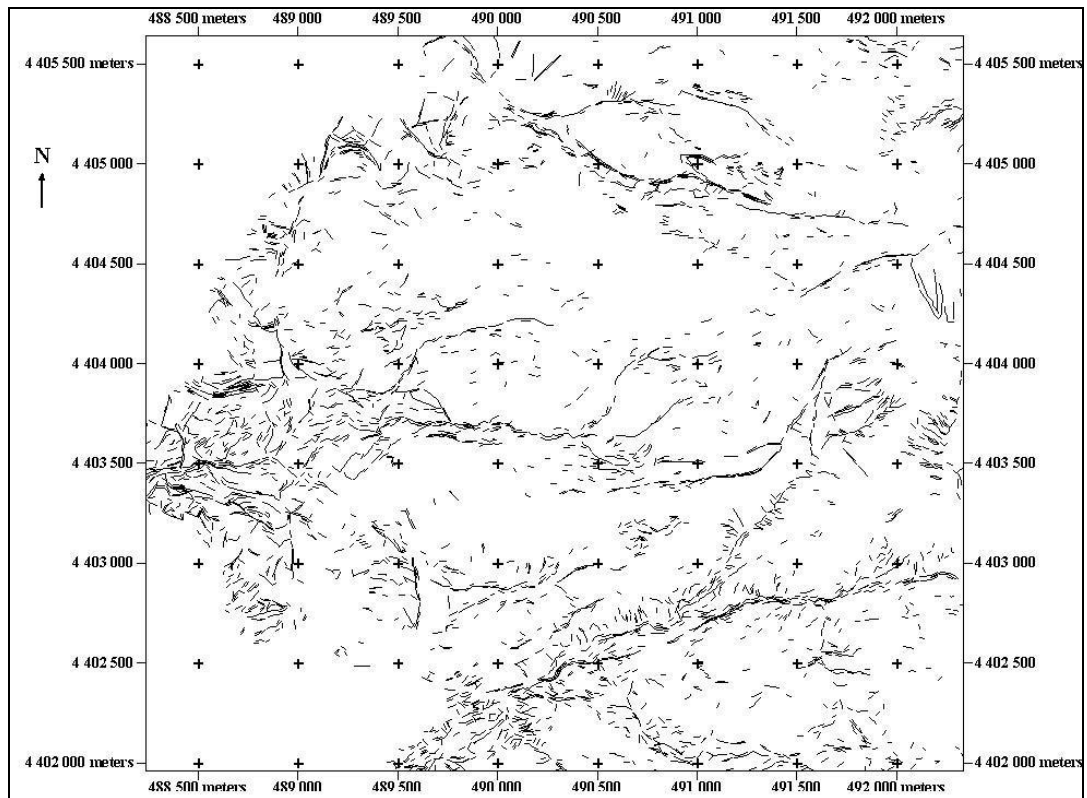


Figure A2. Lineament Map of LINE 2

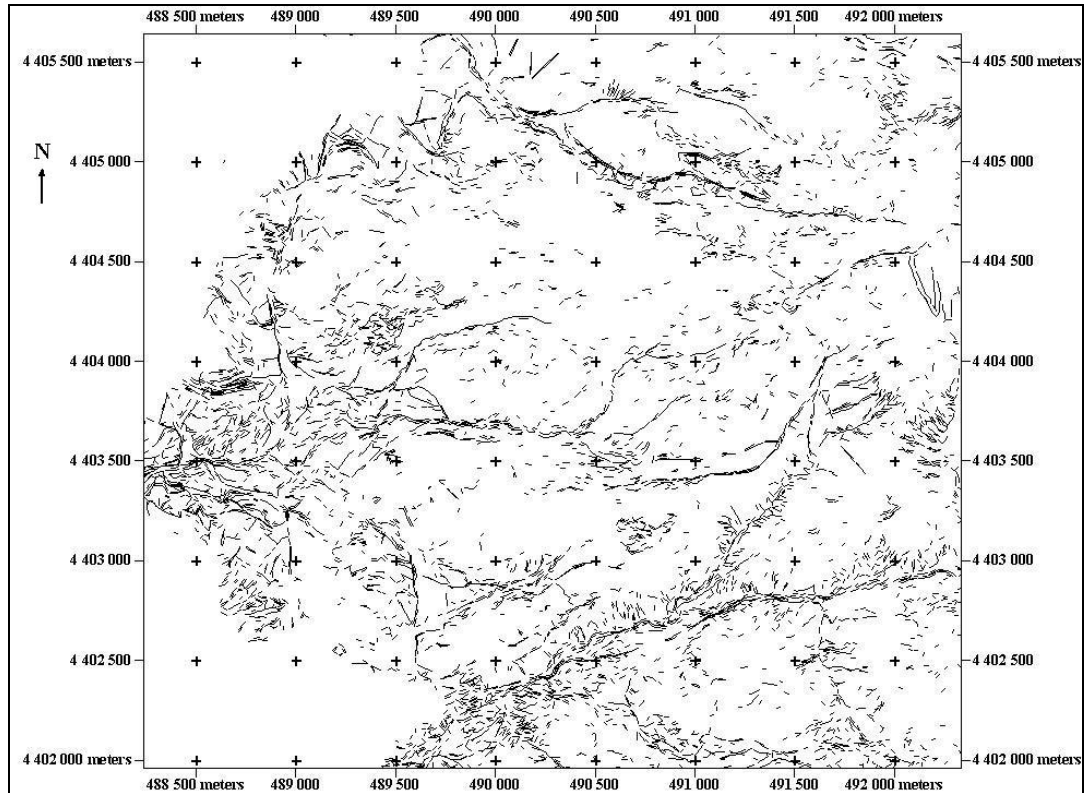


Figure A3. Lineament Map of LINE 3

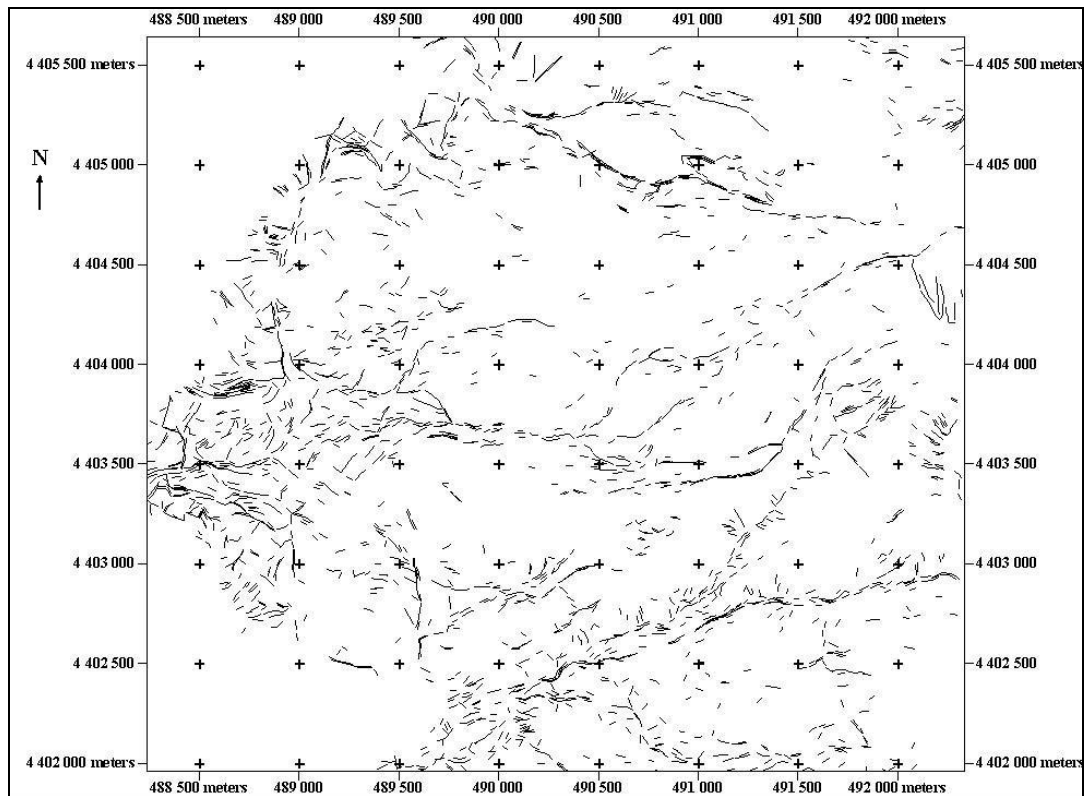


Figure A4. Lineament Map of LINE 4

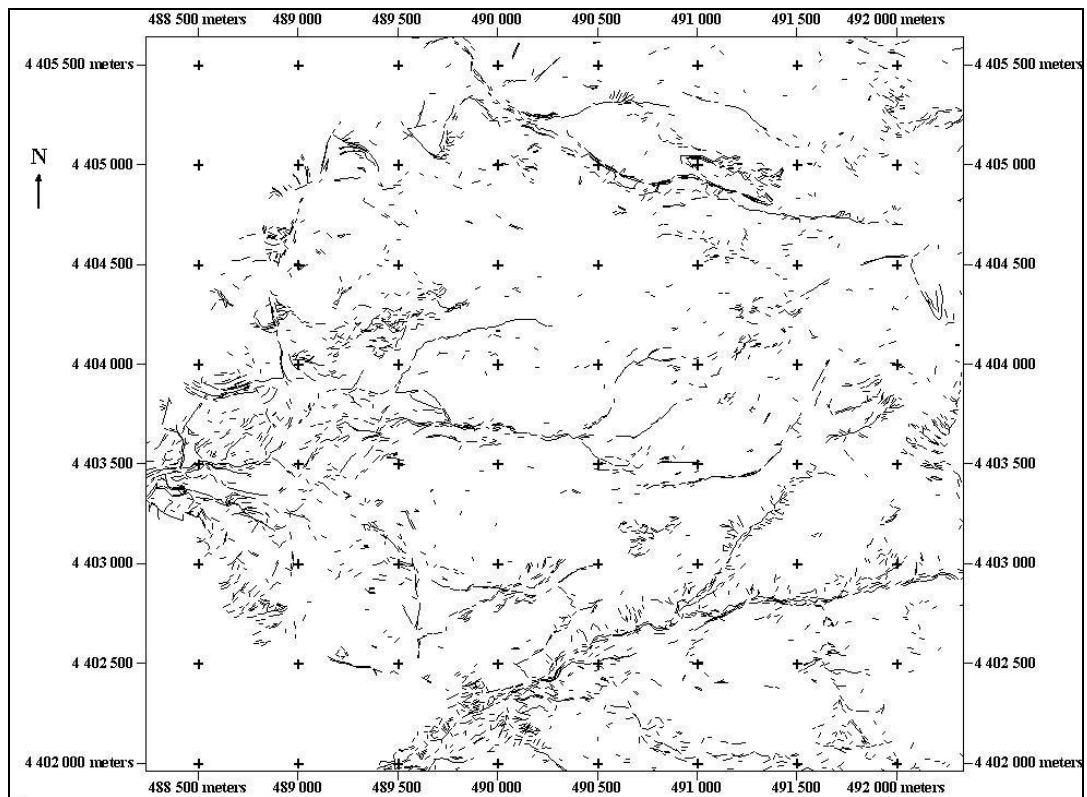


Figure A5. Lineament Map of LINE 5

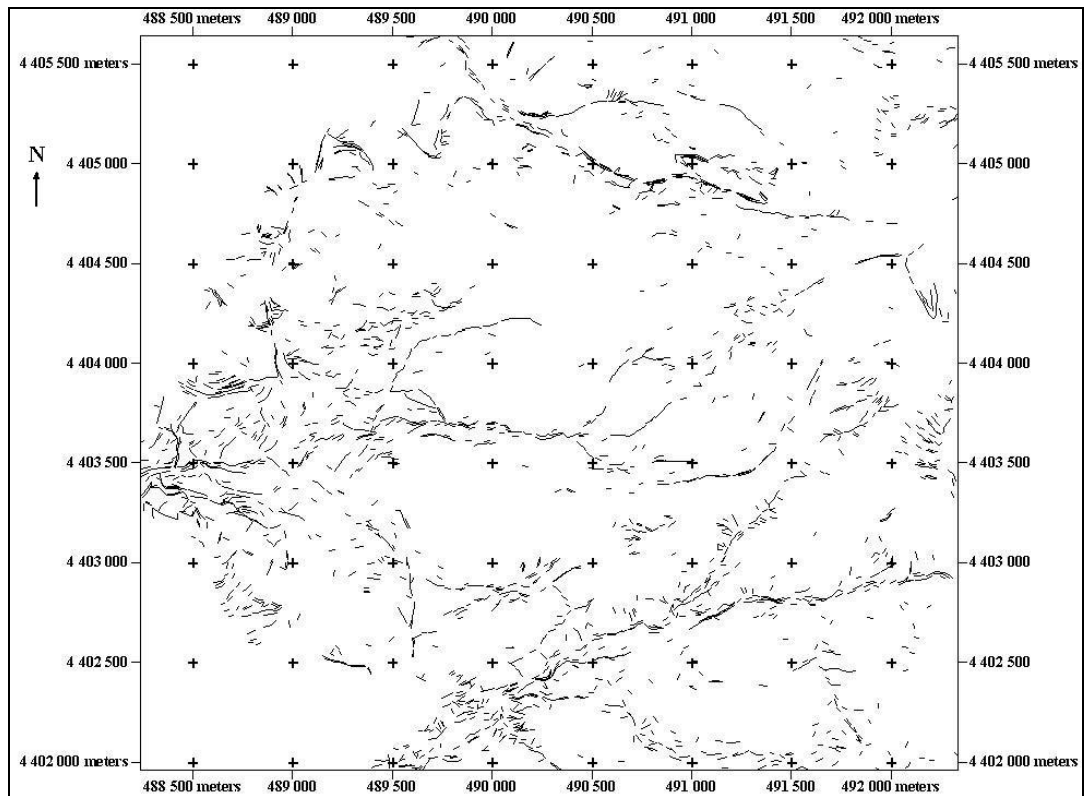


Figure A6. Lineament Map of LINE 6

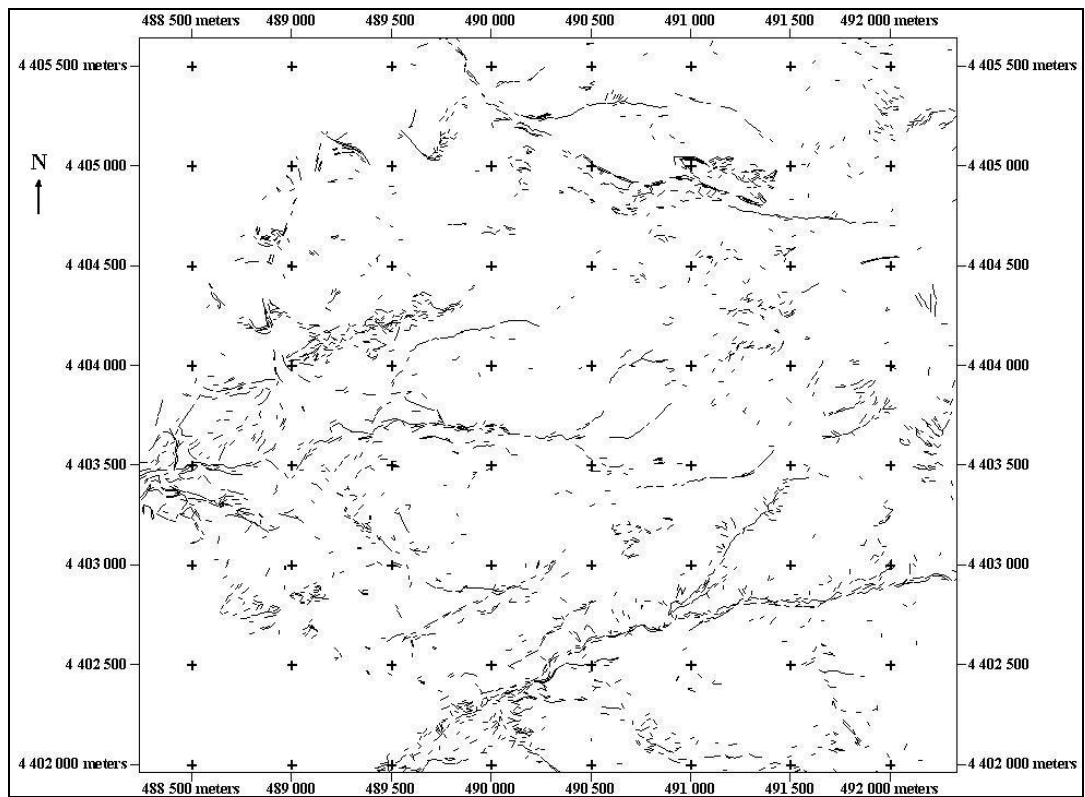


Figure A7. Lineament Map of LINE 7

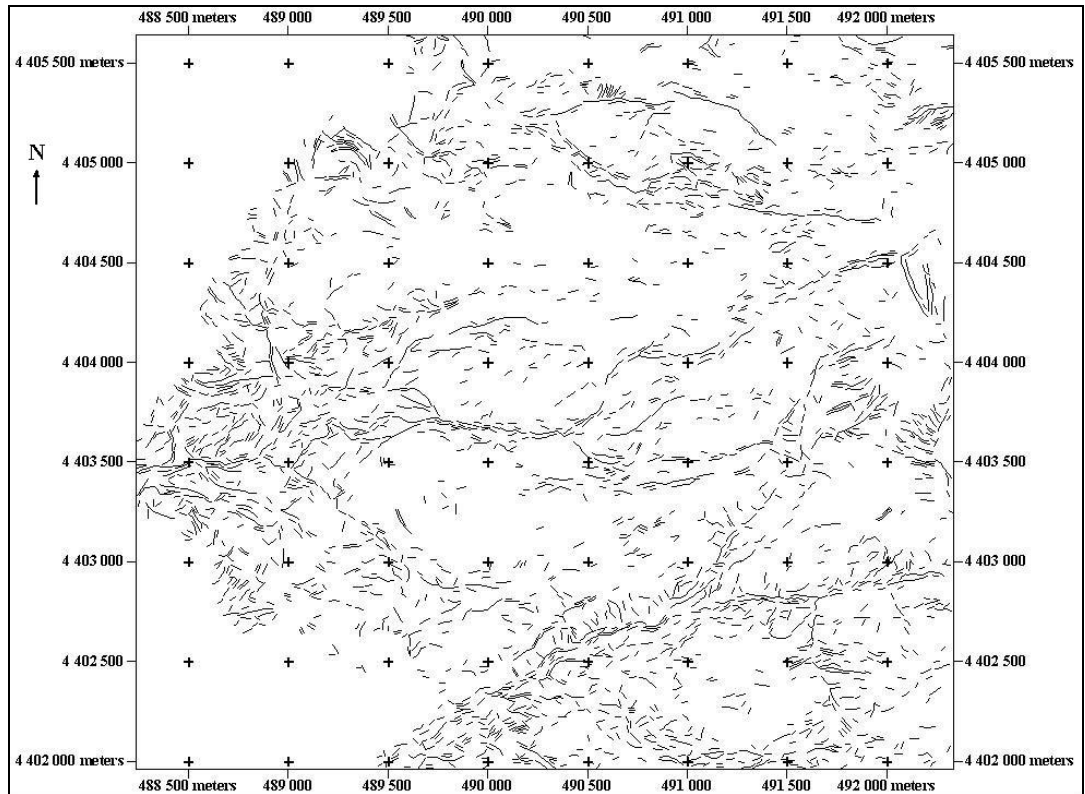


Figure A8. Lineament Map of LINE 8

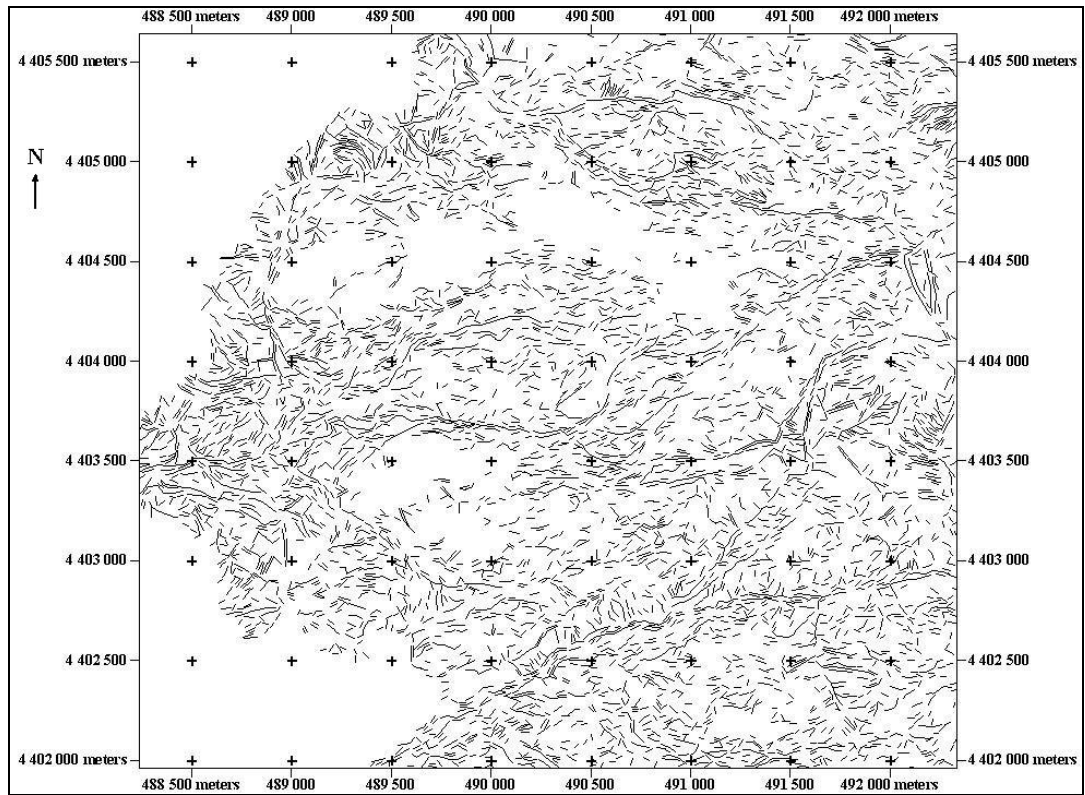


Figure A9. Lineament Map of LINE 9

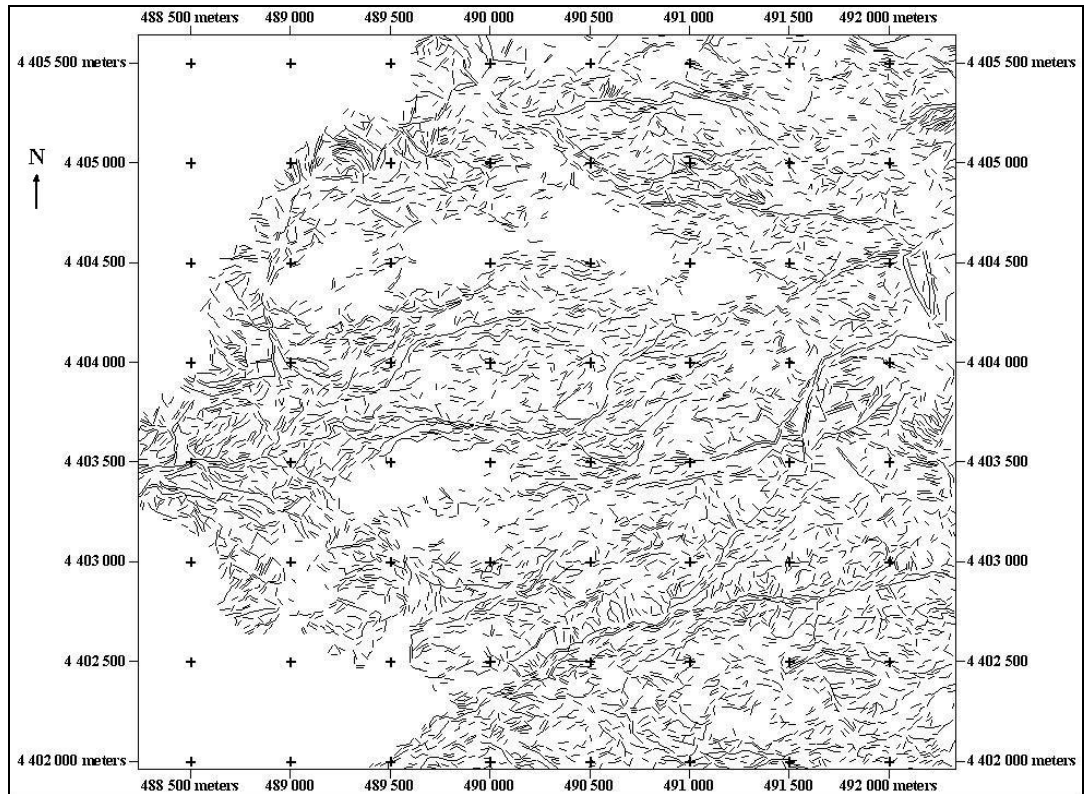


Figure A10. Lineament Map of LINE 10

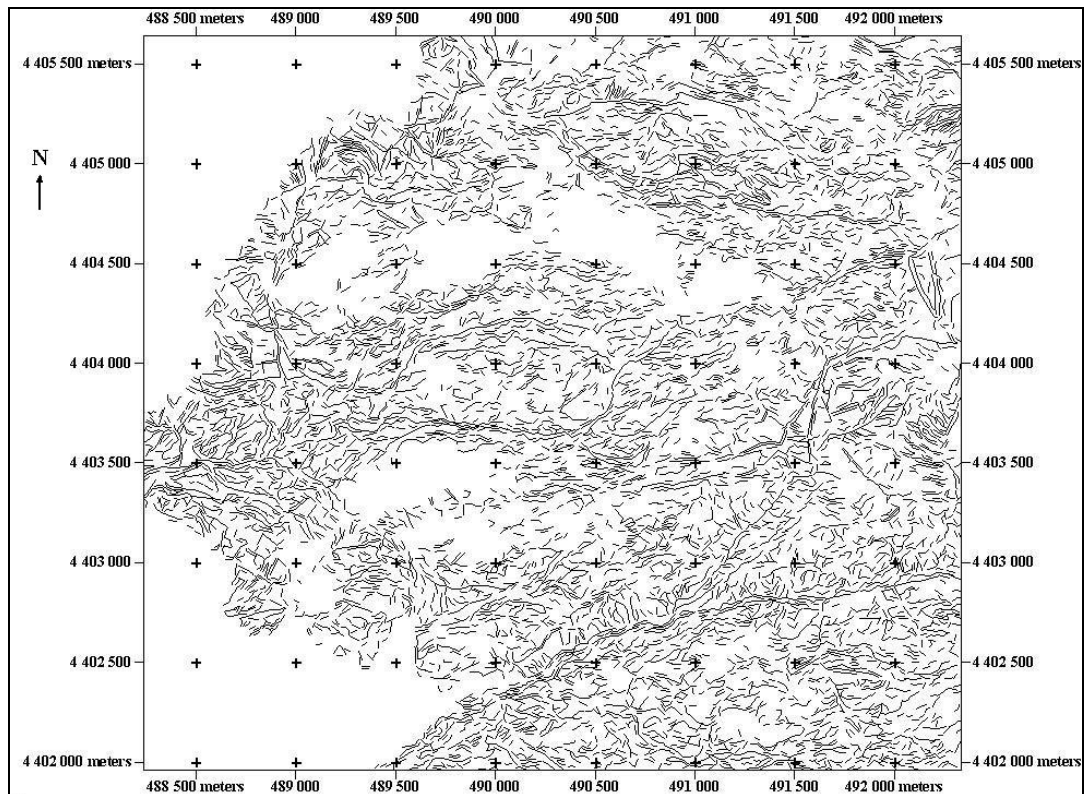


Figure A11. Lineament Map of LINE 11

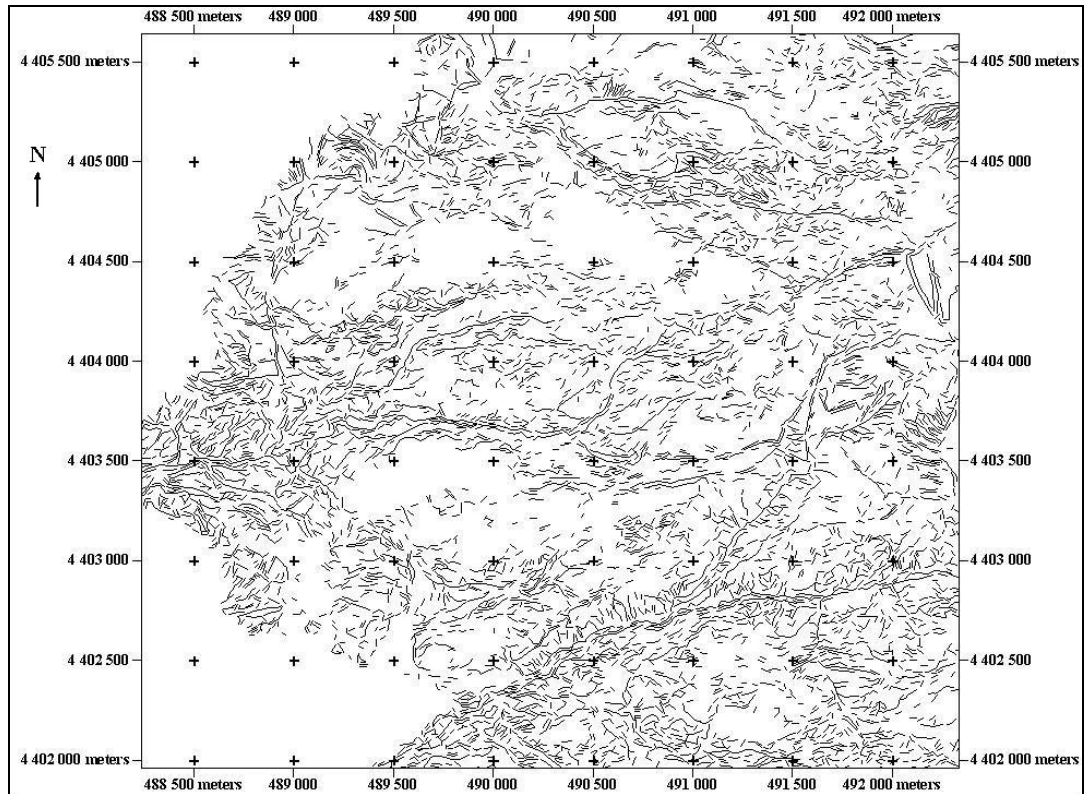


Figure A12. Lineament Map of LINE 12

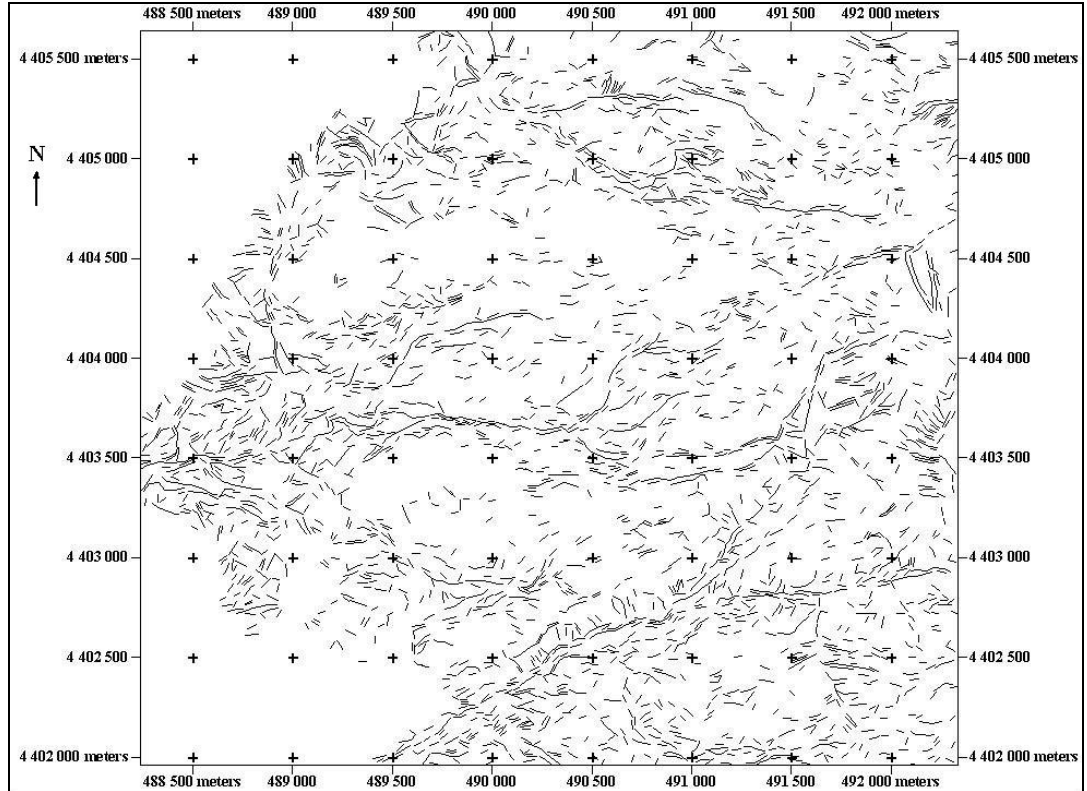


Figure A13. Lineament Map of LINE 13

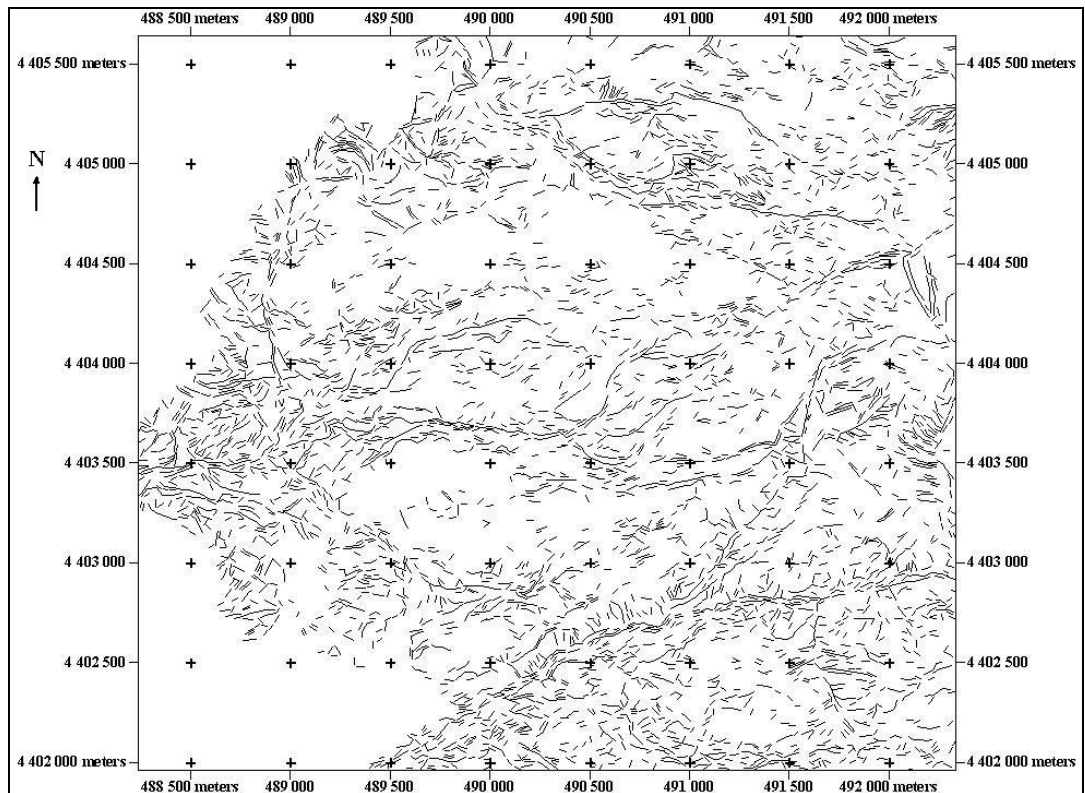


Figure A14. Lineament Map of LINE 14

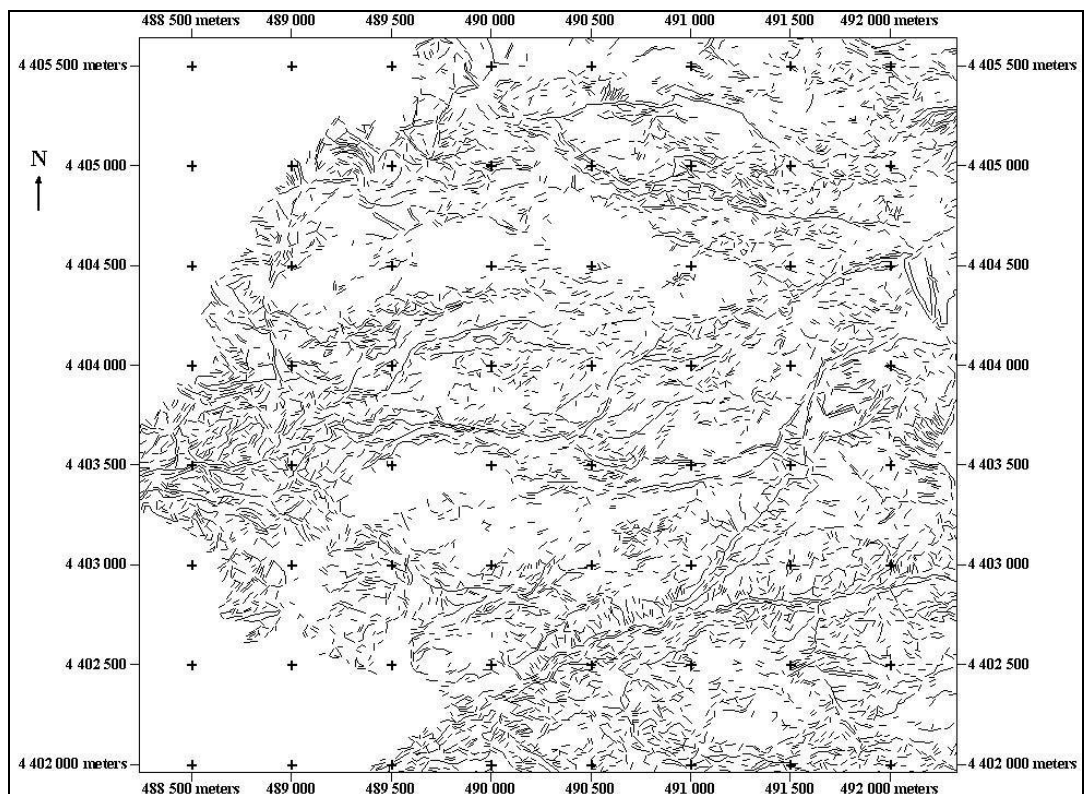


Figure A15. Lineament Map of LINE 15

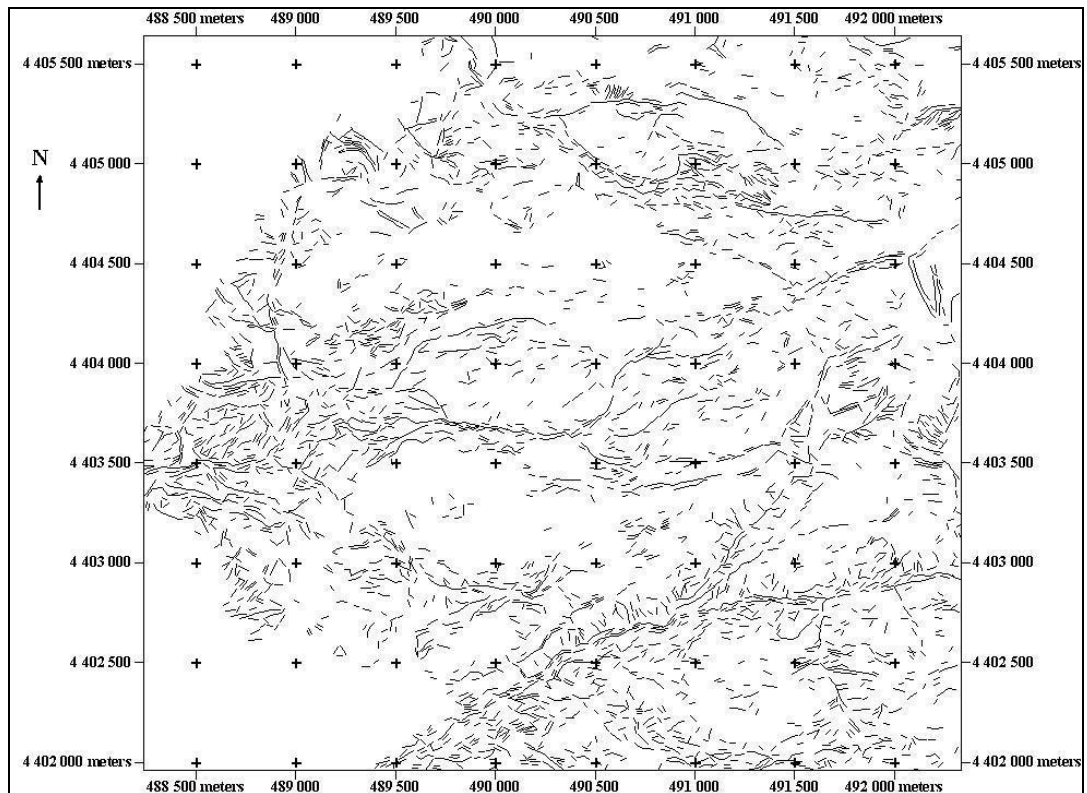


Figure A16. Lineament Map of LINE 16

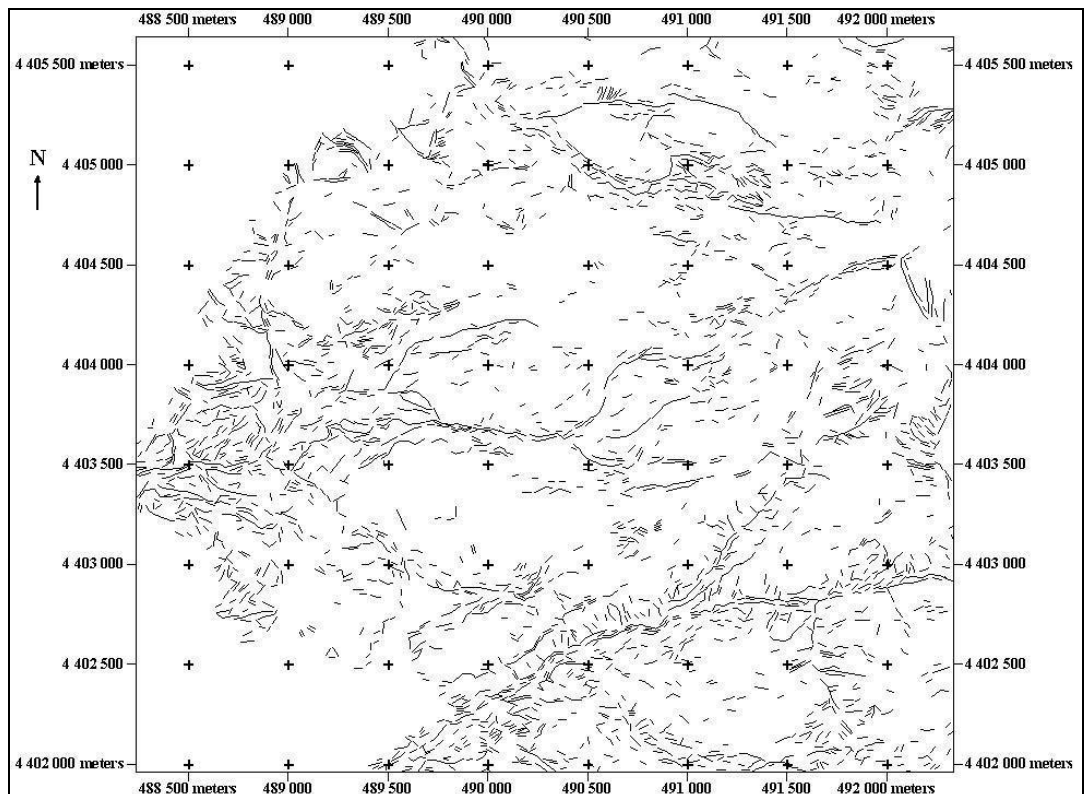


Figure A17. Lineament Map of LINE 17

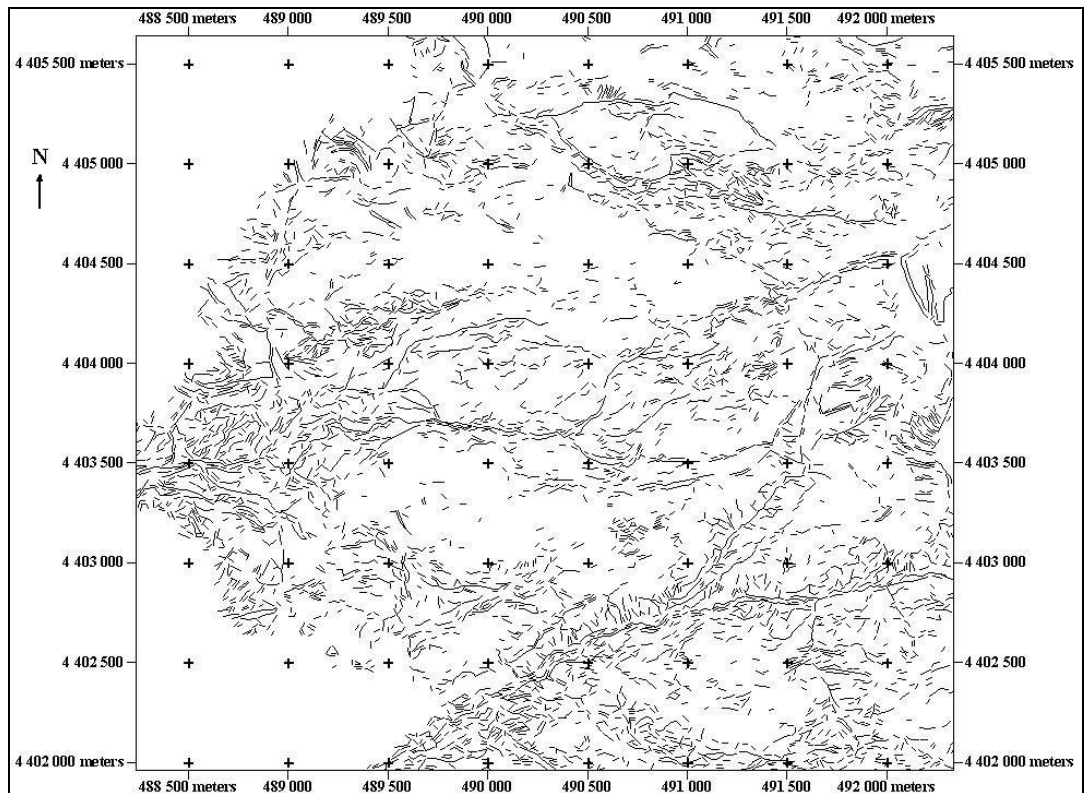


Figure A18. Lineament Map of LINE 18

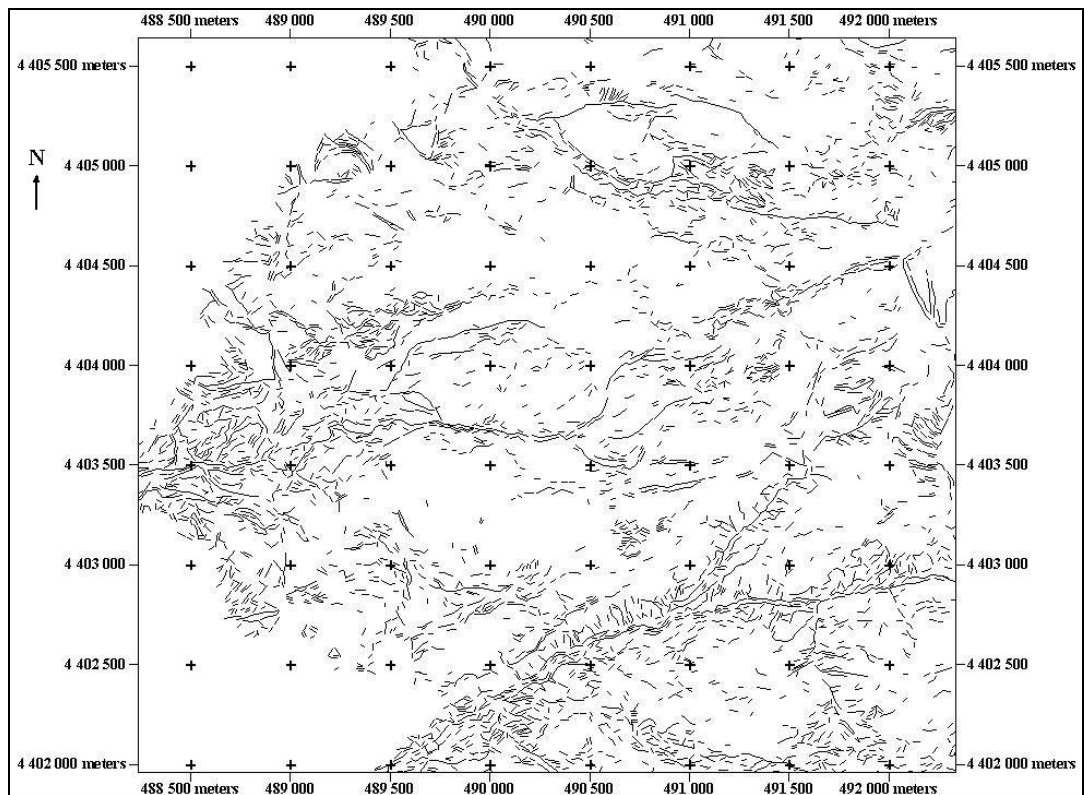


Figure A19. Lineament Map of LINE 19

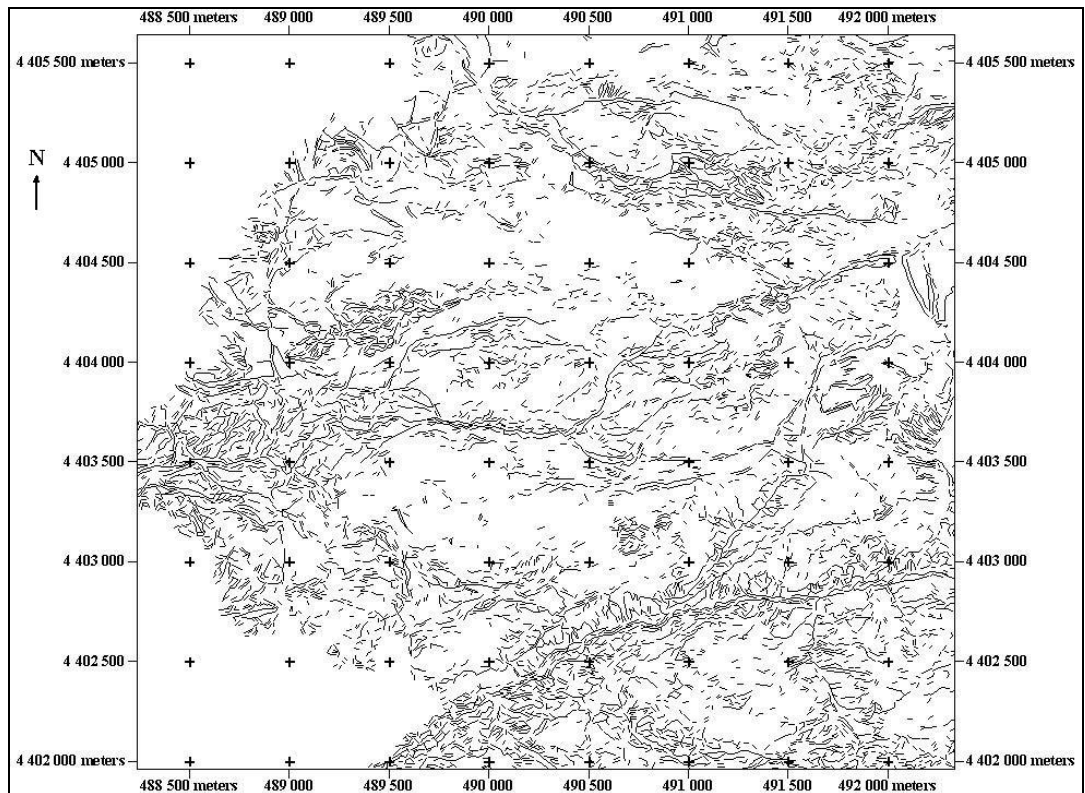


Figure A20. Lineament Map of LINE 20

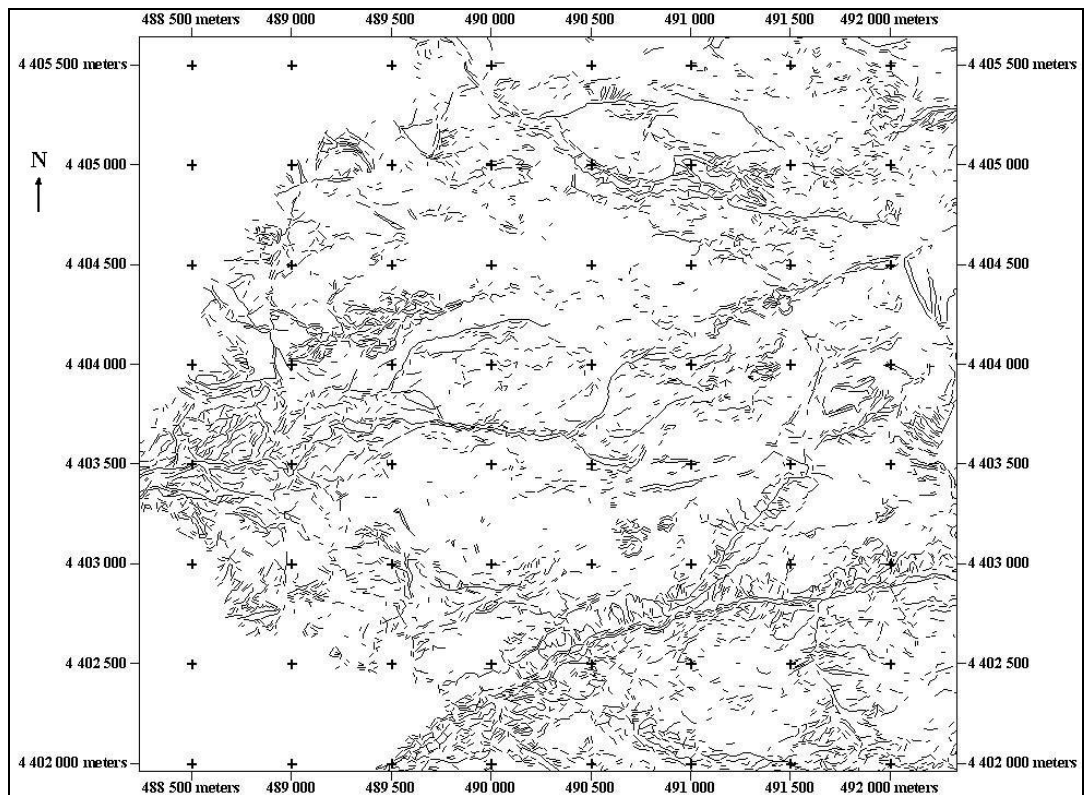


Figure A21. Lineament Map of LINE 21

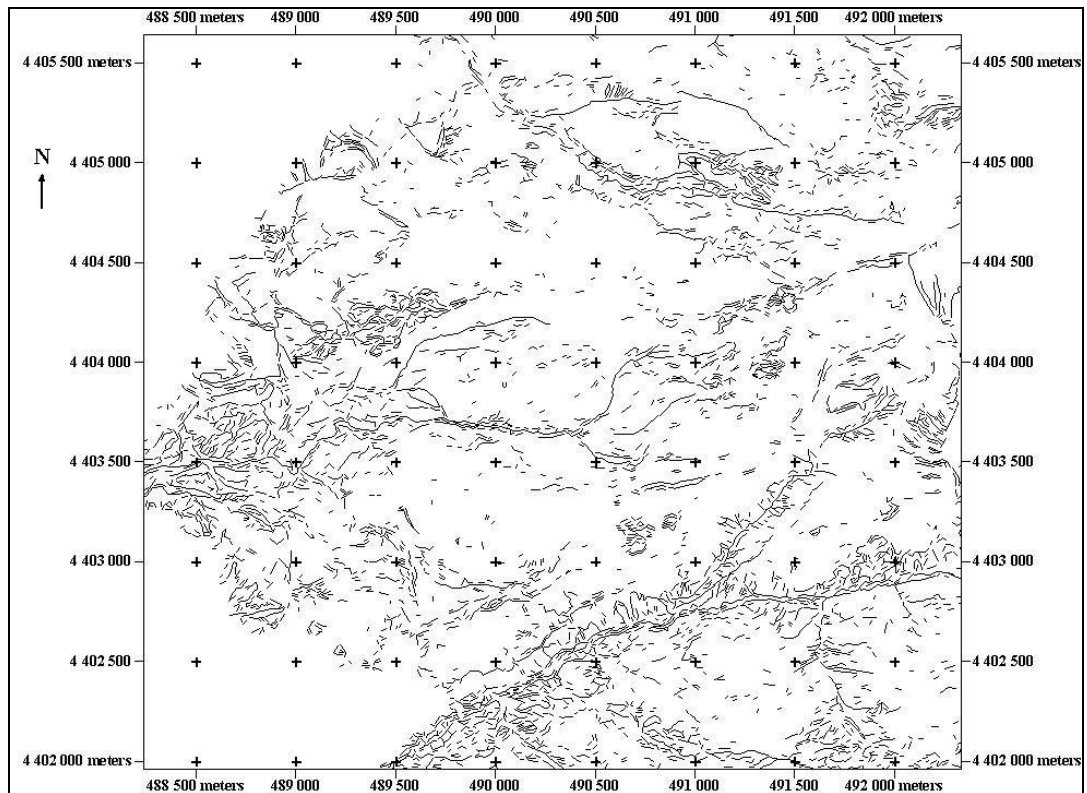


Figure A22. Lineament Map of LINE 22

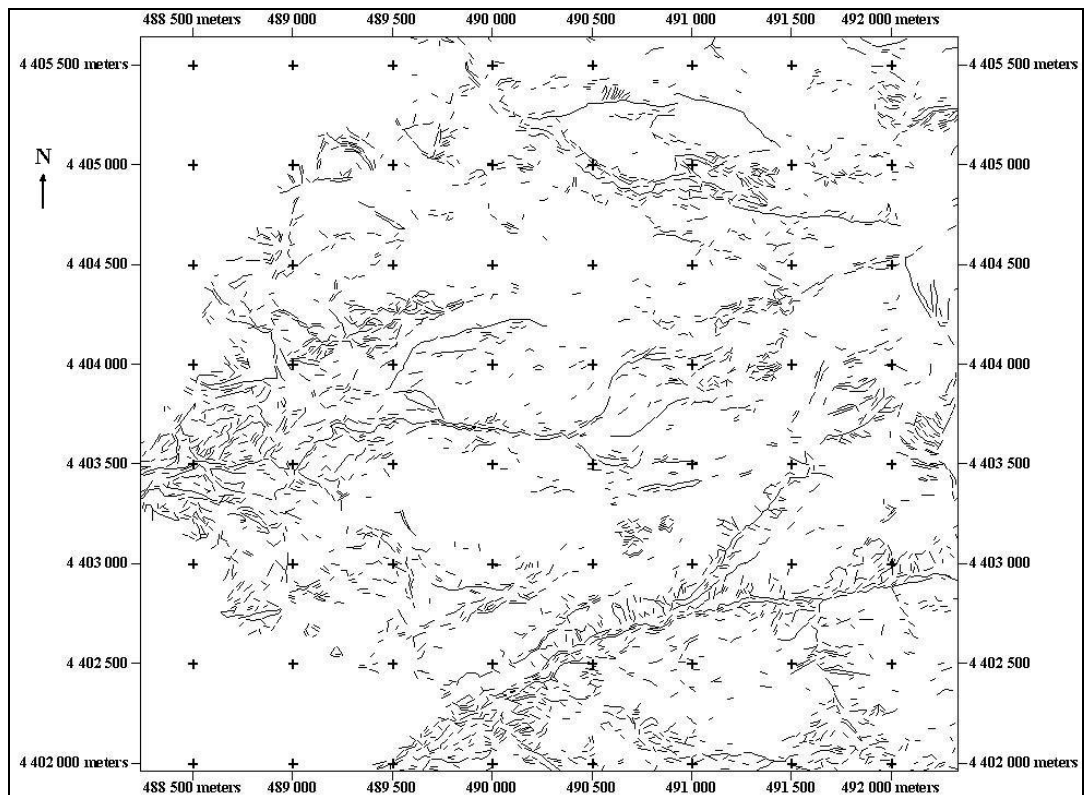


Figure A23. Lineament Map of LINE 23

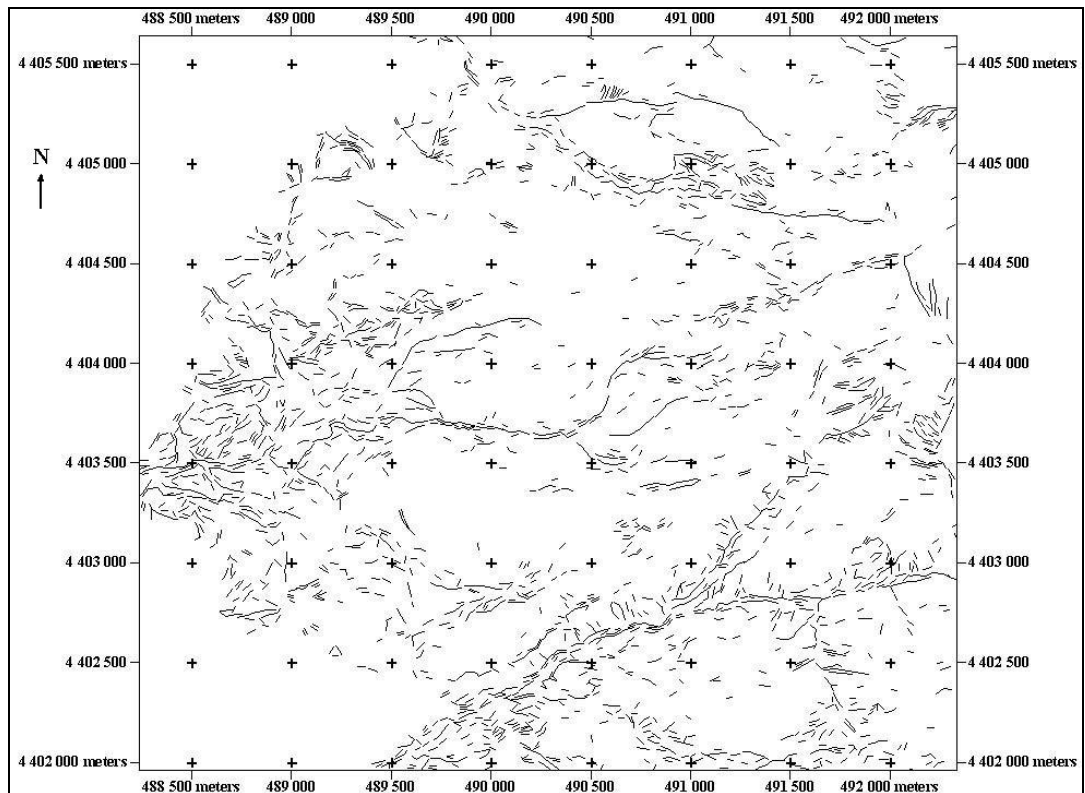


Figure A24. Lineament Map of LINE 24

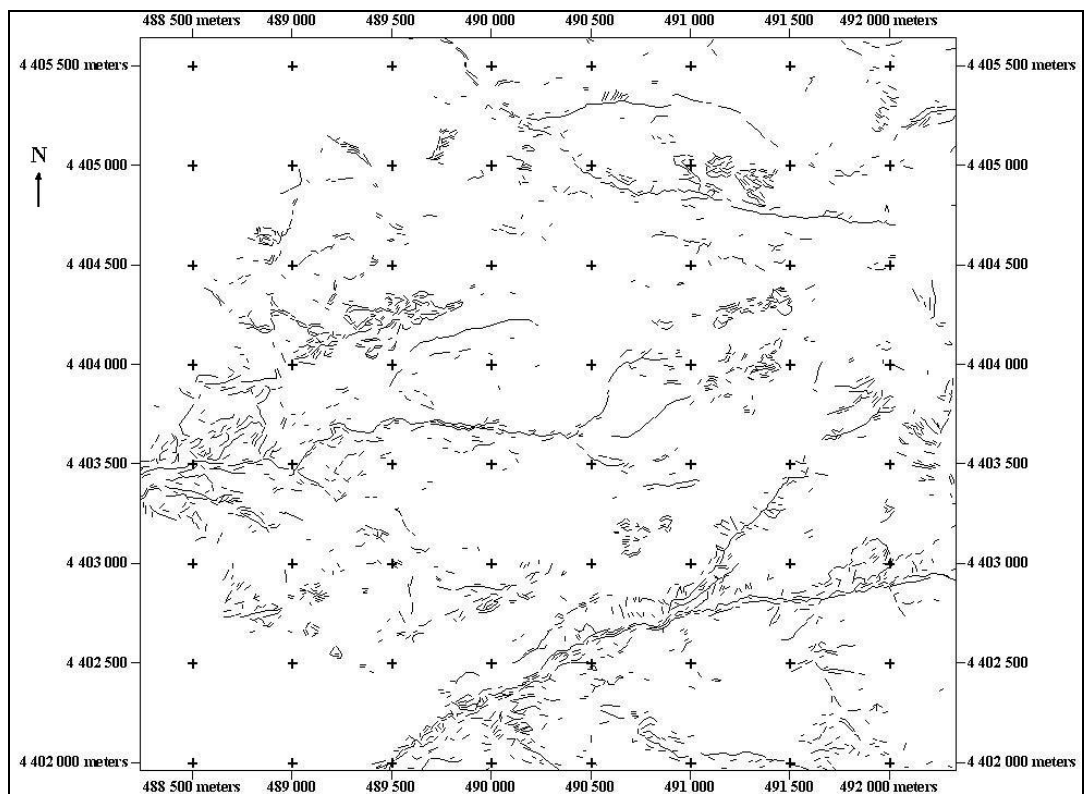


Figure A25. Lineament Map of LINE 25

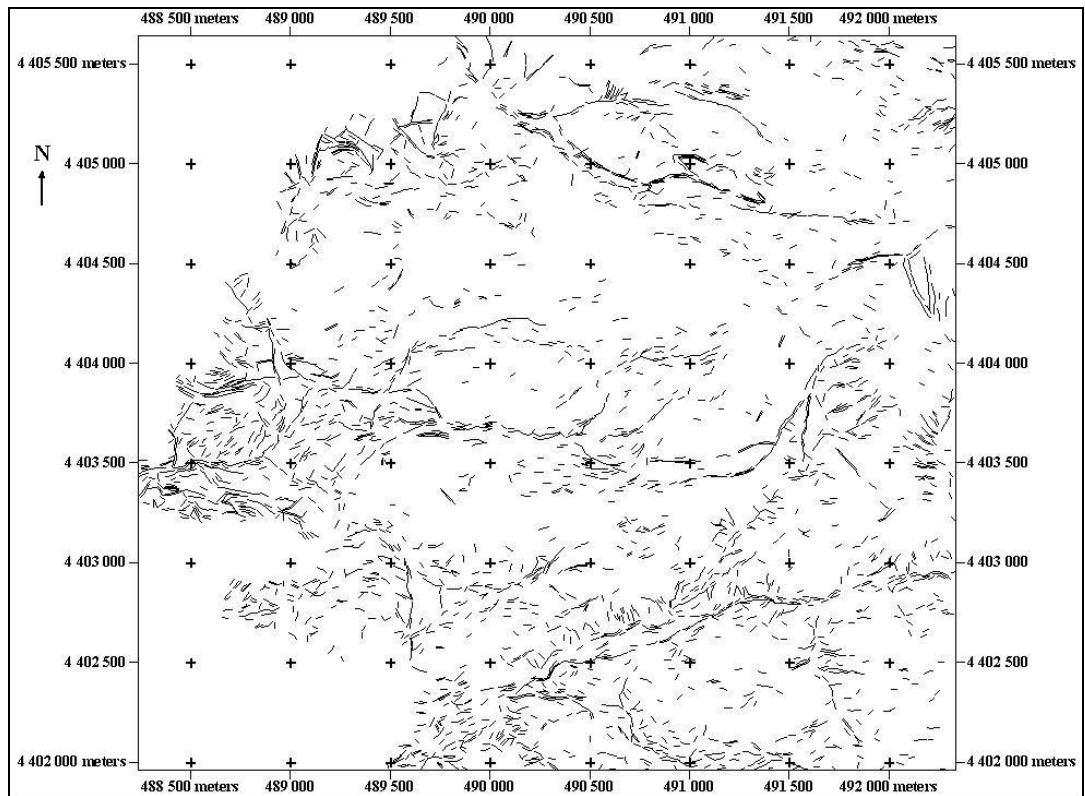


Figure A26. Lineament Map of LINE 26

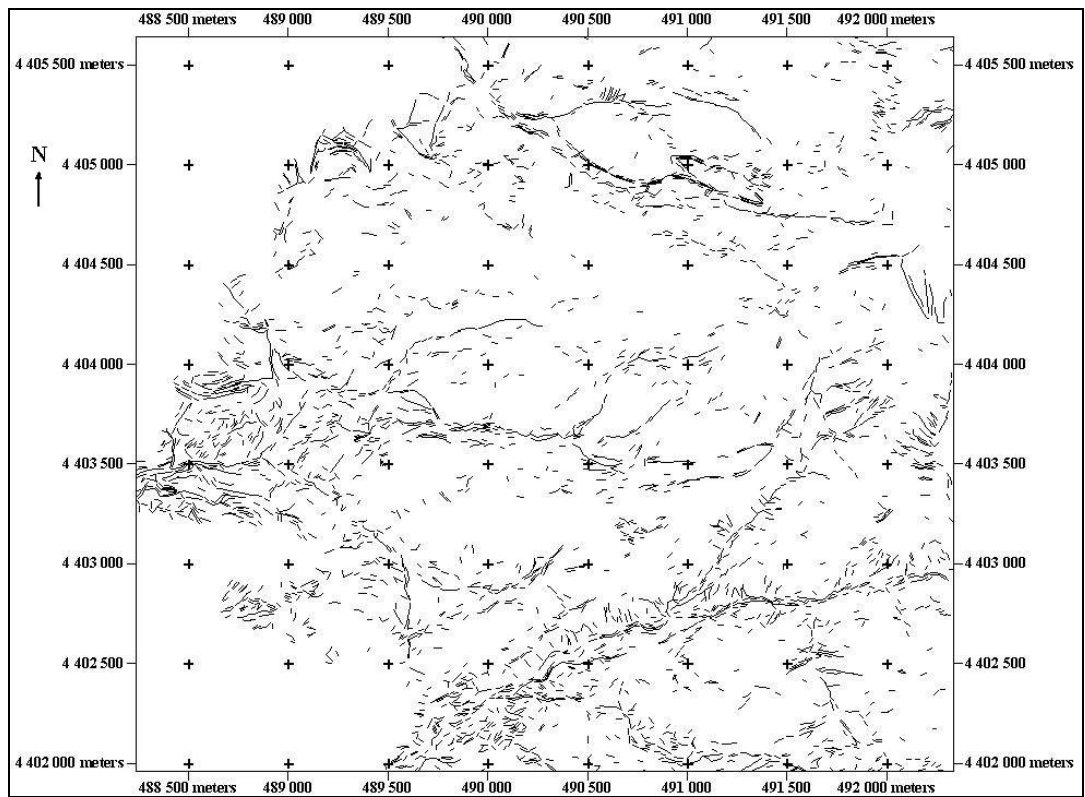


Figure A27. Lineament Map of LINE 27

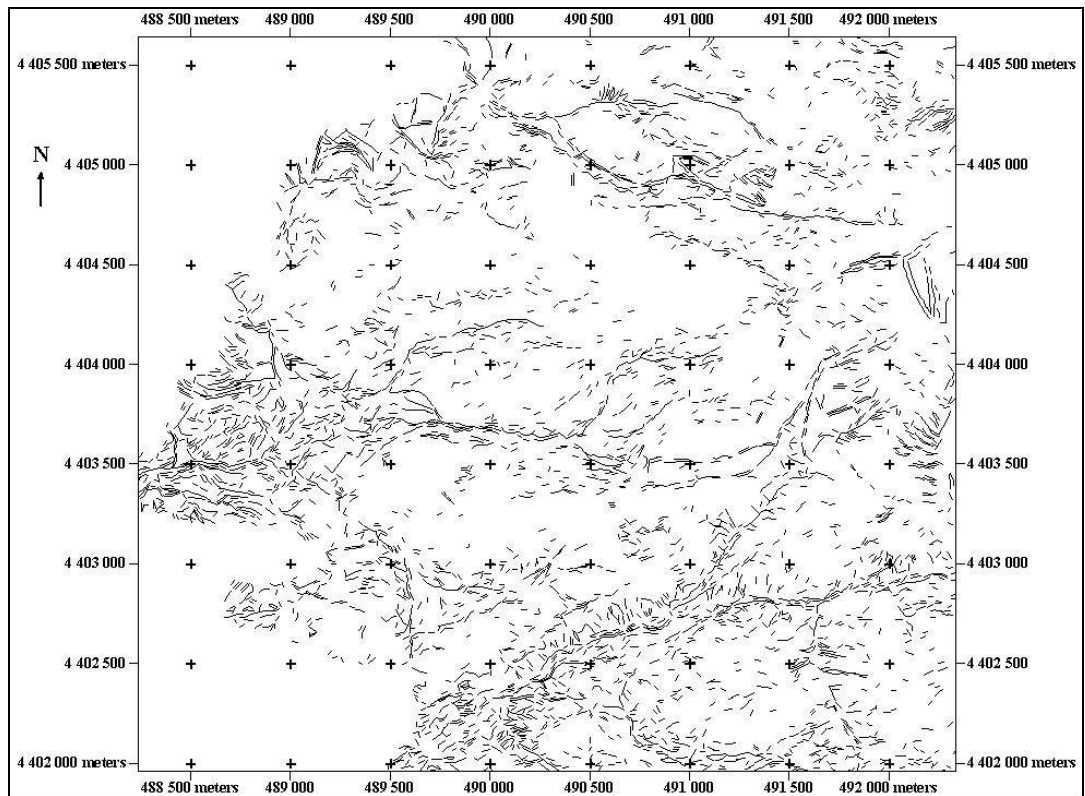


Figure A28. Lineament Map of LINE 28

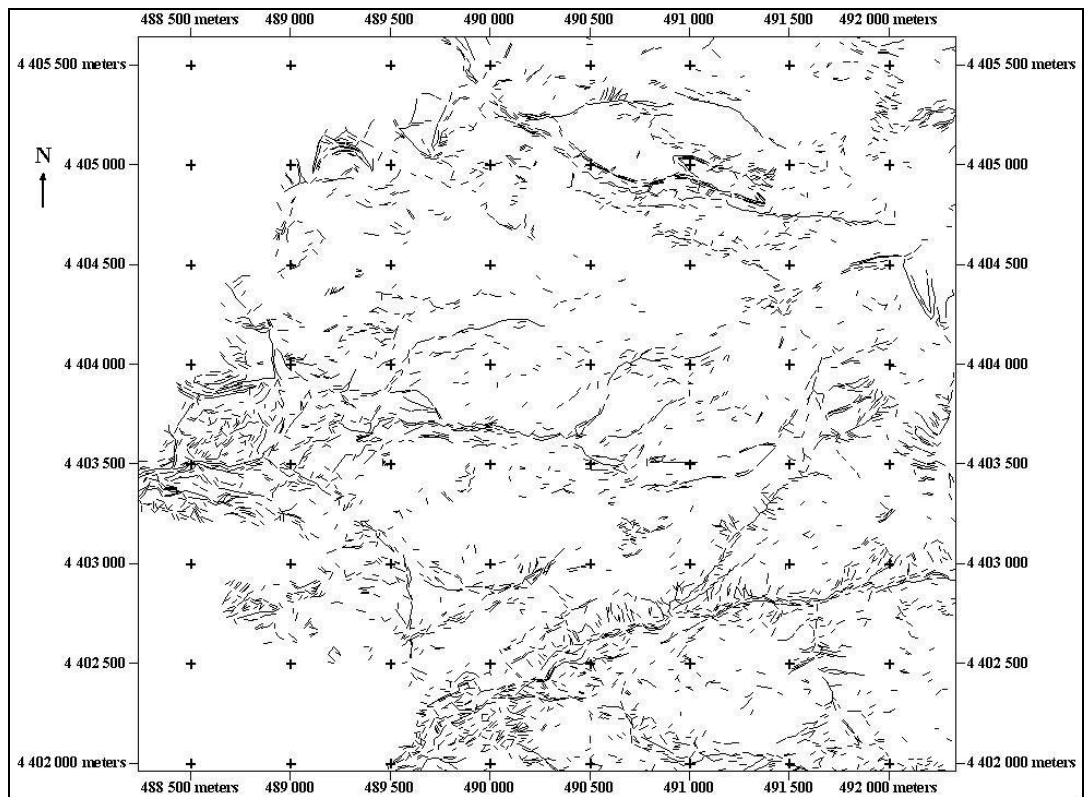


Figure A29. Lineament Map of LINE 29

APPENDIX B

In this appendix, the rose diagrams of automatically extracted lineaments are given in Figure B1-B22.

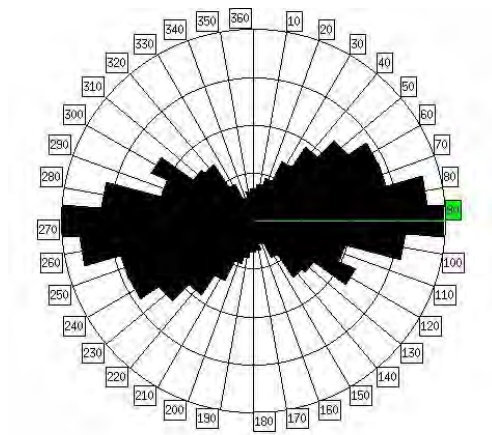


Figure B1. Rose Diagram of LINE 3

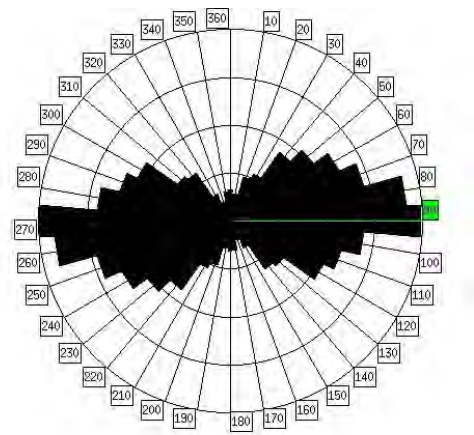


Figure B2. Rose Diagram of LINE 4

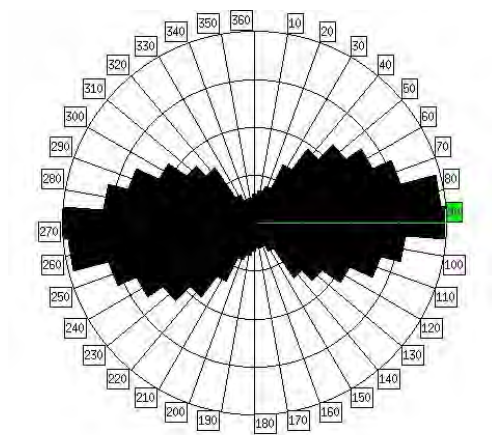


Figure B3. Rose Diagram of LINE 6

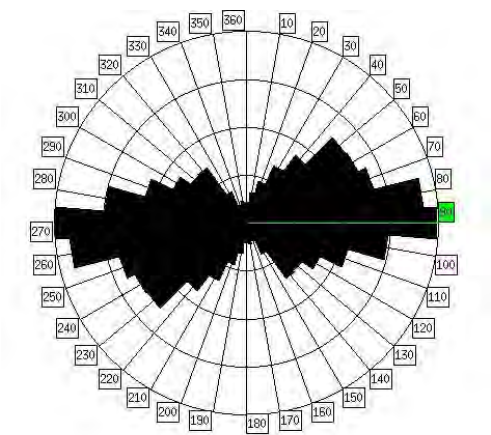


Figure B4. Rose Diagram of LINE 7

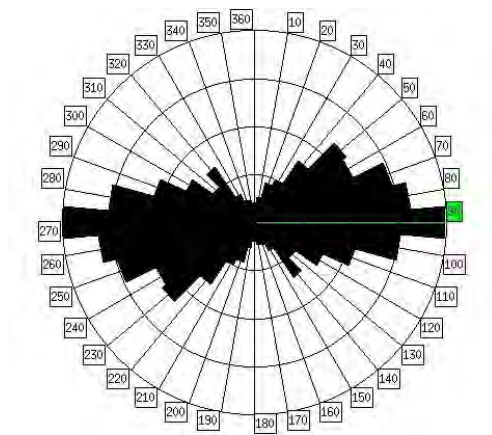


Figure B5. Rose Diagram of LINE 8

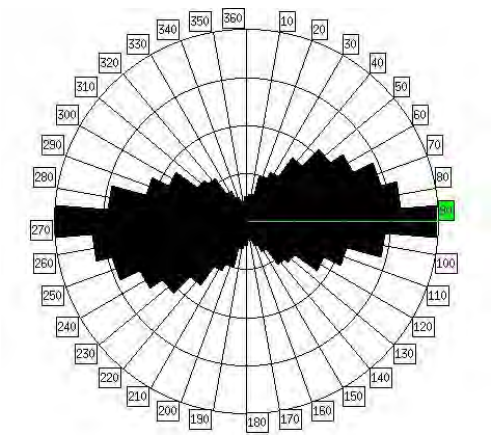


Figure B6. Rose Diagram of LINE 9

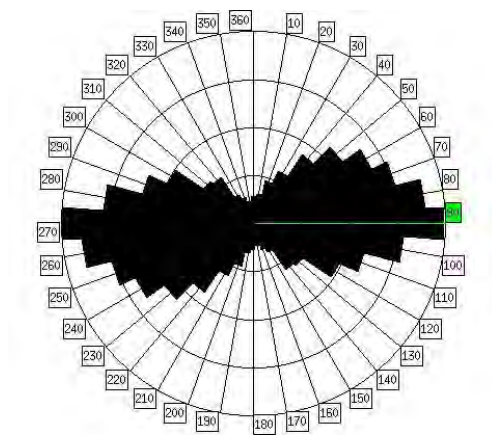


Figure B7. Rose Diagram of LINE 10

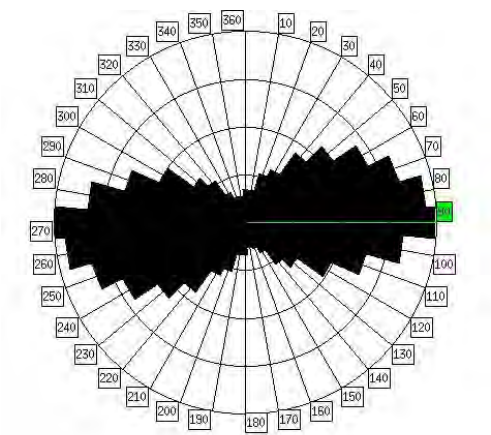


Figure B8. Rose Diagram of LINE 11

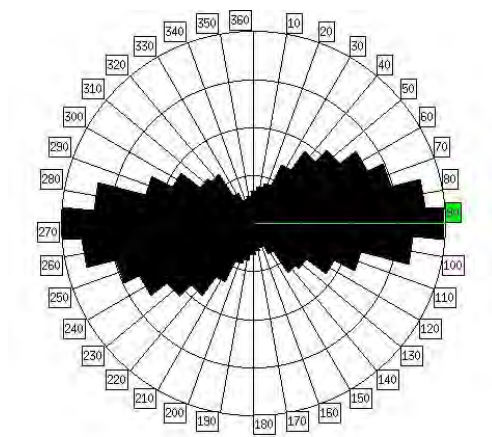


Figure B9. Rose Diagram of LINE 12

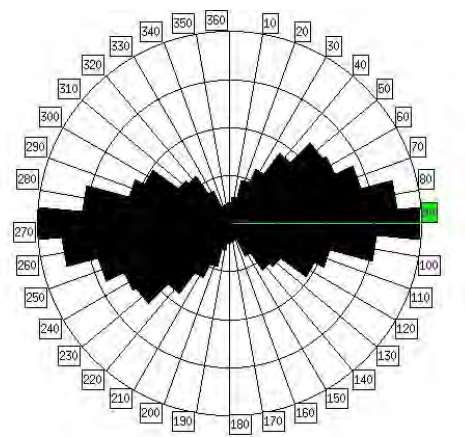


Figure B10. Rose Diagram of LINE 13

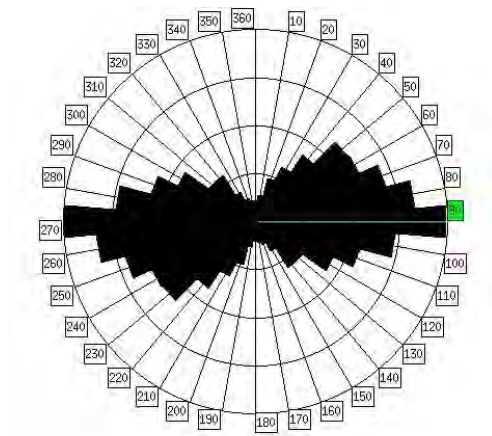


Figure B11. Rose Diagram of LINE 14

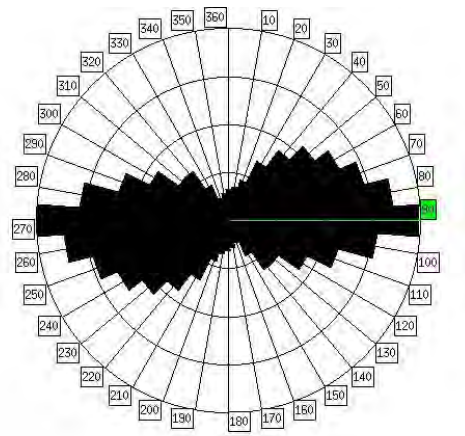


Figure B12. Rose Diagram of LINE 15

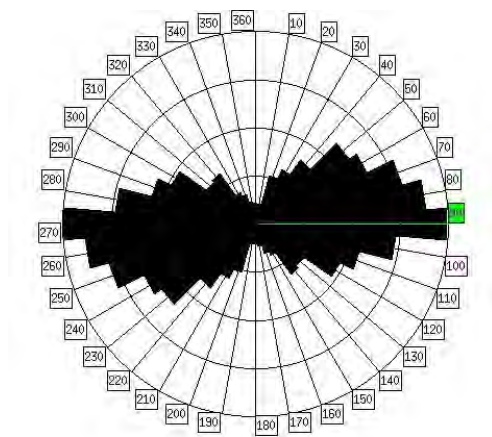


Figure B13. Rose Diagram of LINE 17

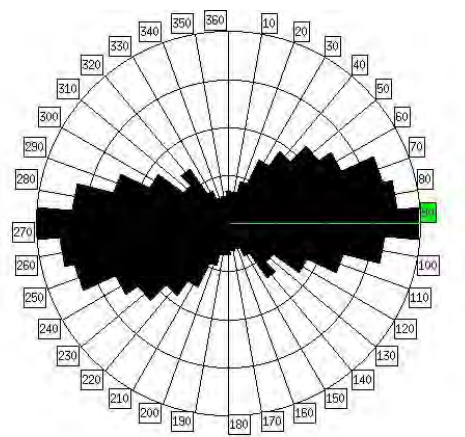


Figure B14. Rose Diagram of LINE 18

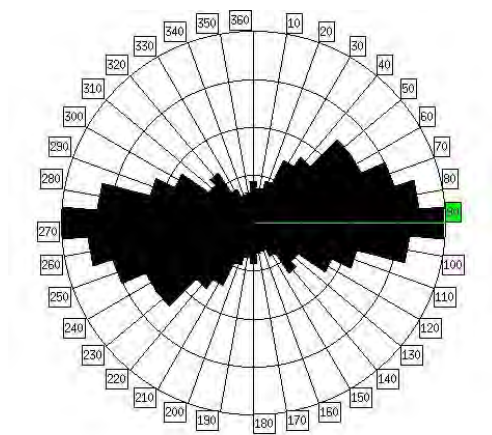


Figure B15. Rose Diagram of LINE 20

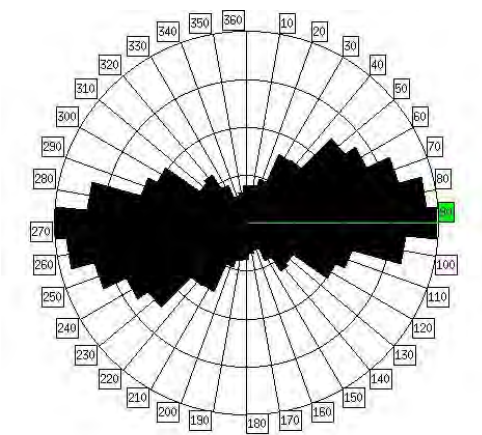


Figure B16. Rose Diagram of LINE 21

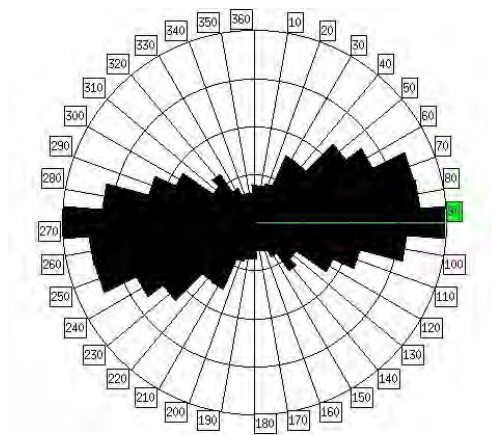


Figure B17. Rose Diagram of LINE 22

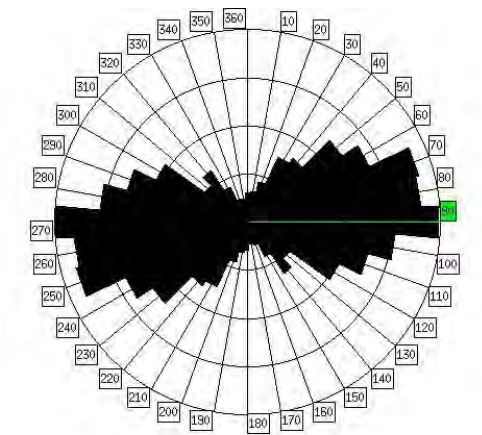


Figure B18. Rose Diagram of LINE 23

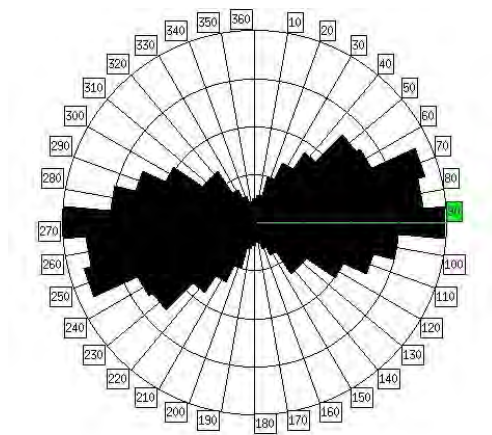


Figure B19. Rose Diagram of LINE 24

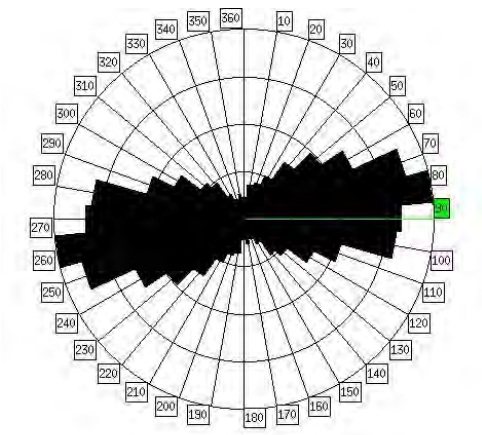


Figure B20. Rose Diagram of LINE 25

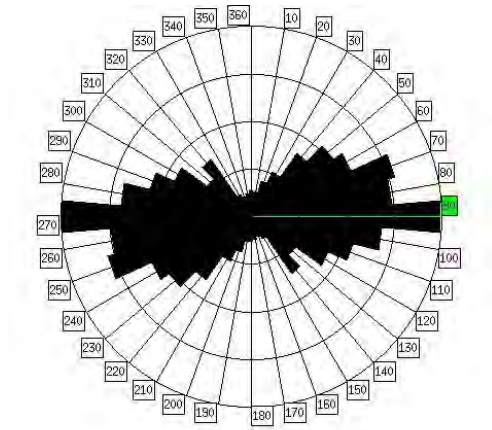


Figure B21. Rose Diagram of LINE 26

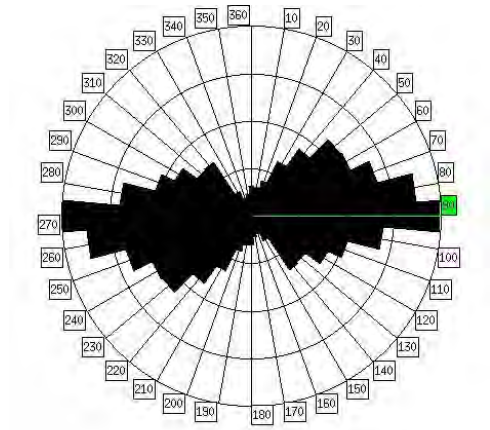


Figure B22. Rose Diagram of LINE 28

APPENDIX C

In this appendix, the classified images of LINE 2 within Zones B with different tolerance values are presented.

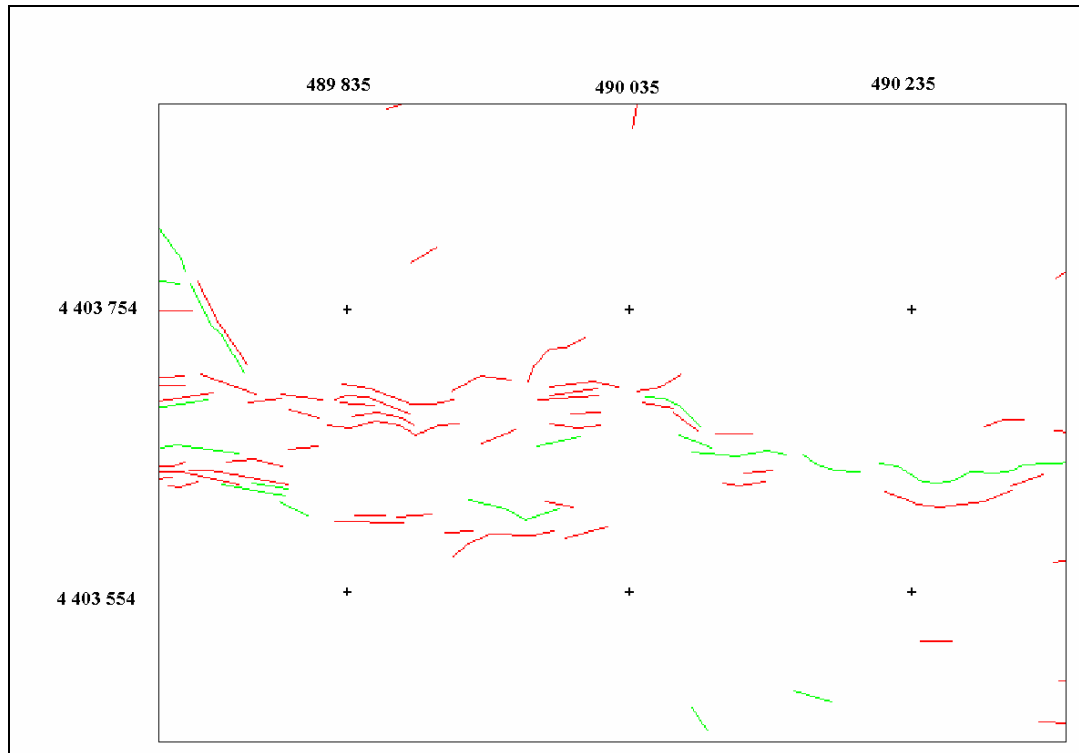


Figure C1. Classified Image of LINE 2 in Zone B with Tolerance Value = 0

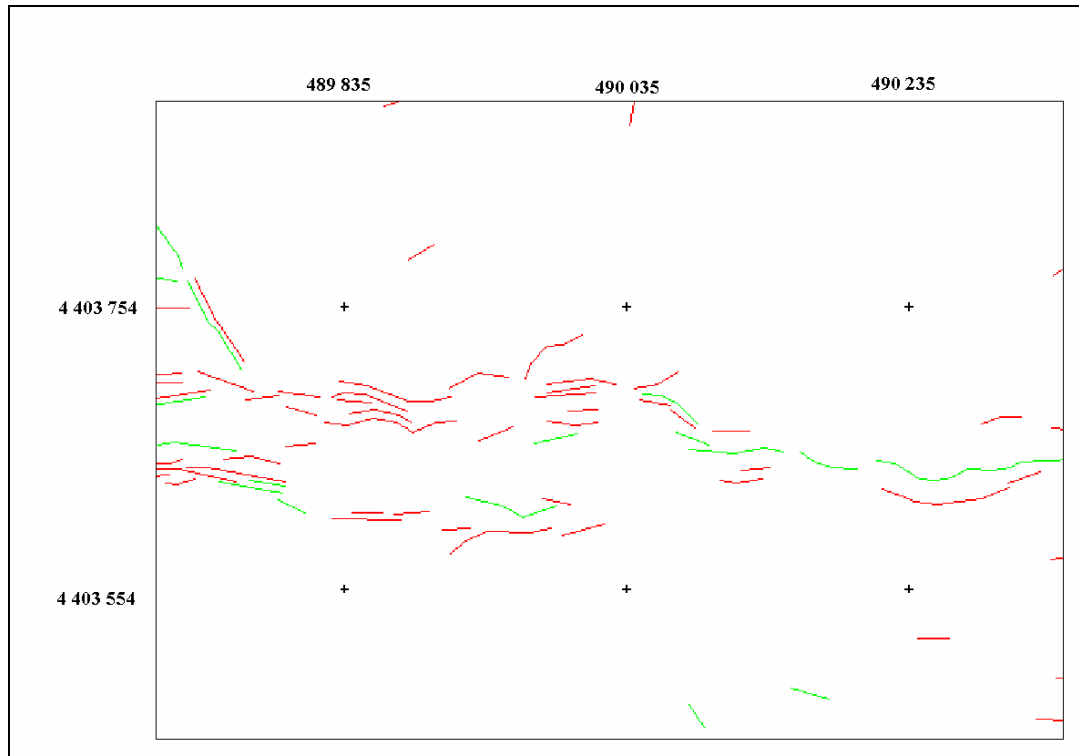


Figure C2. Classified Image of LINE 2 in Zone B with Tolerance Value = 1

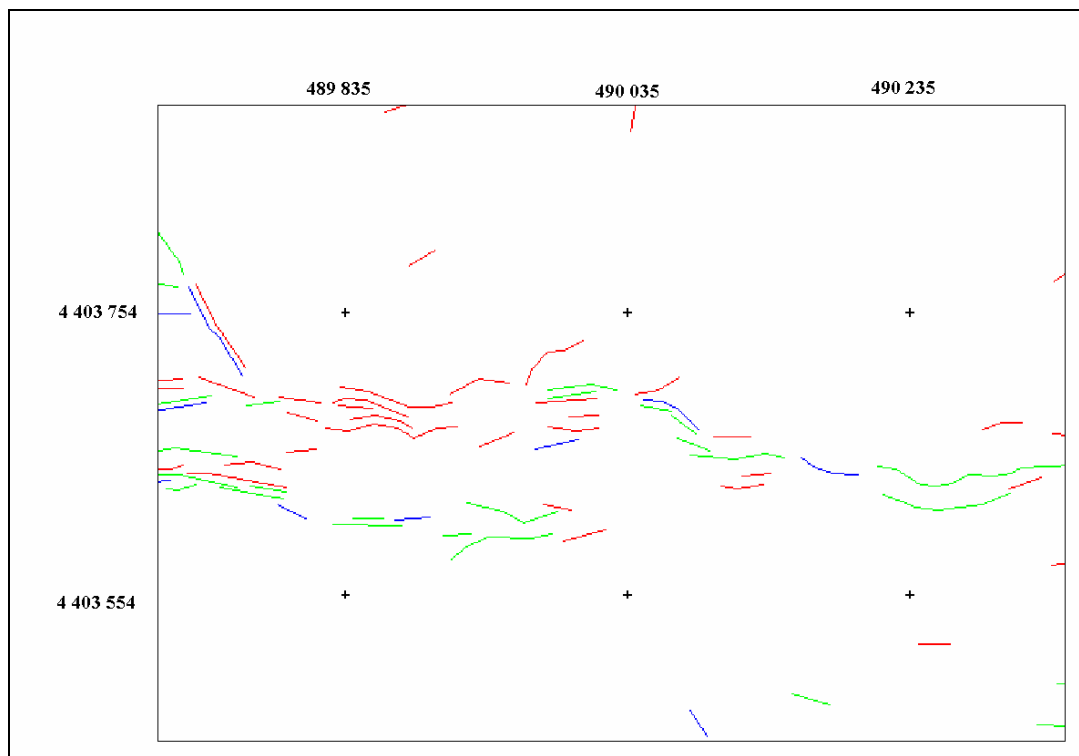


Figure C3. Classified Image of LINE 2 in Zone B with Tolerance Value = 2

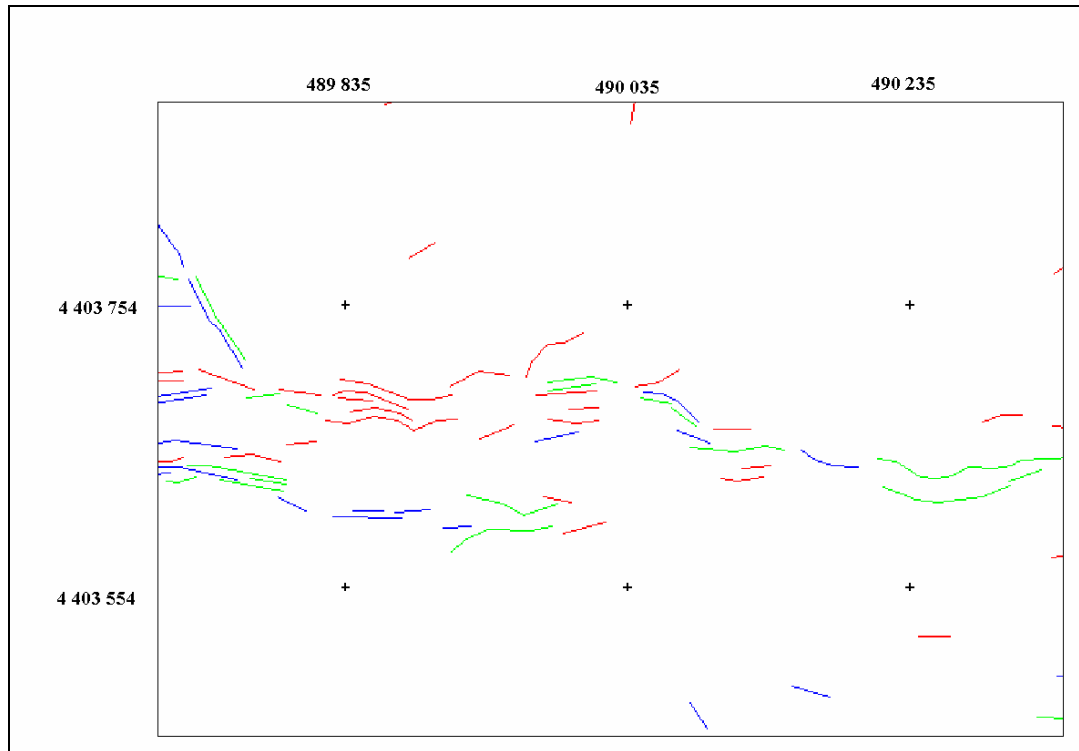


Figure C4. Classified Image of LINE 2 in Zone B with Tolerance Value = 3

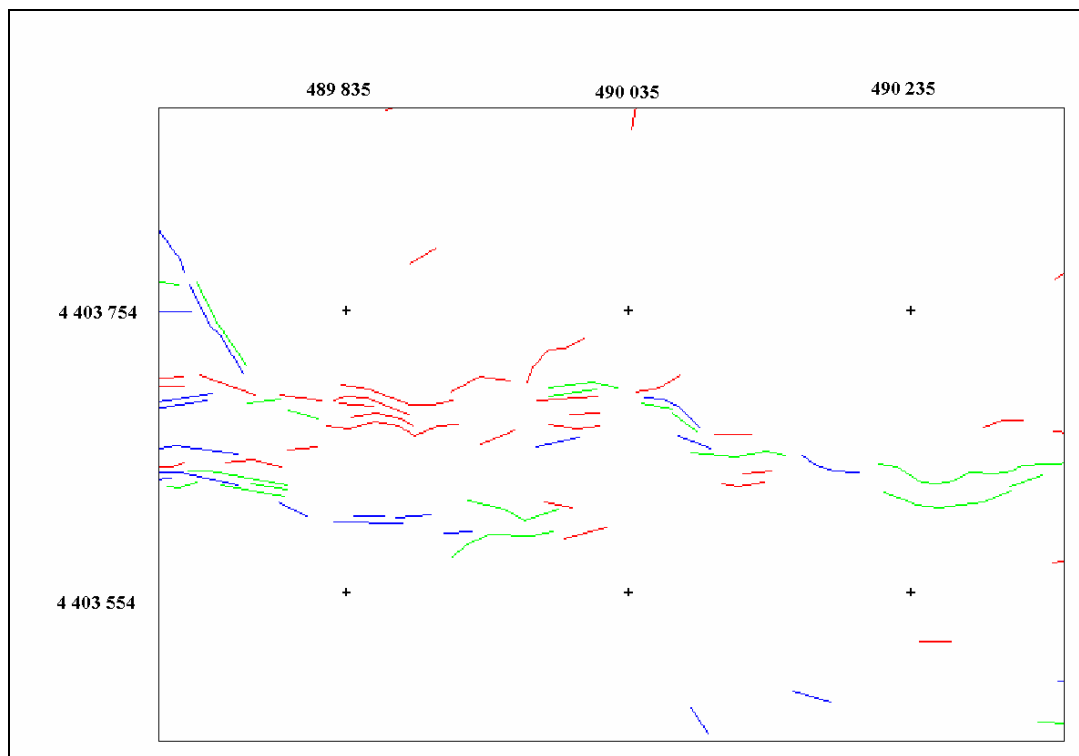


Figure C5. Classified Image of LINE 2 in Zone B with Tolerance Value = 4

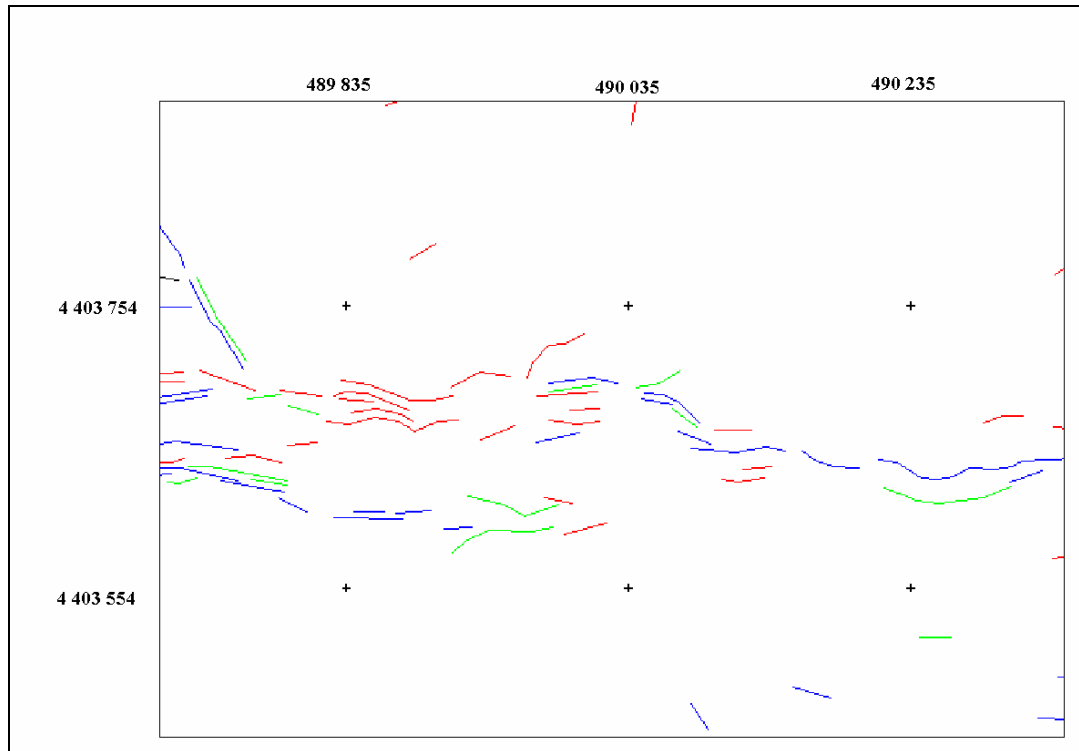


Figure C6. Classified Image of LINE 2 in Zone B with Tolerance Value = 5

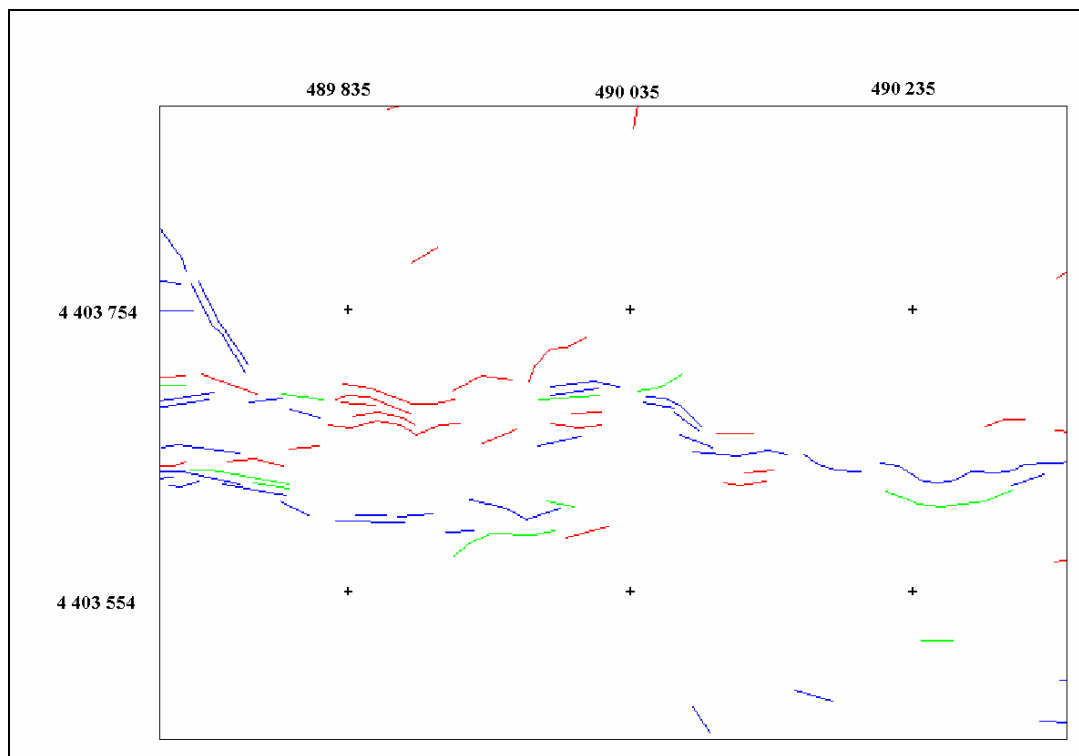


Figure C7. Classified Image of LINE 2 in Zone B with Tolerance Value = 6

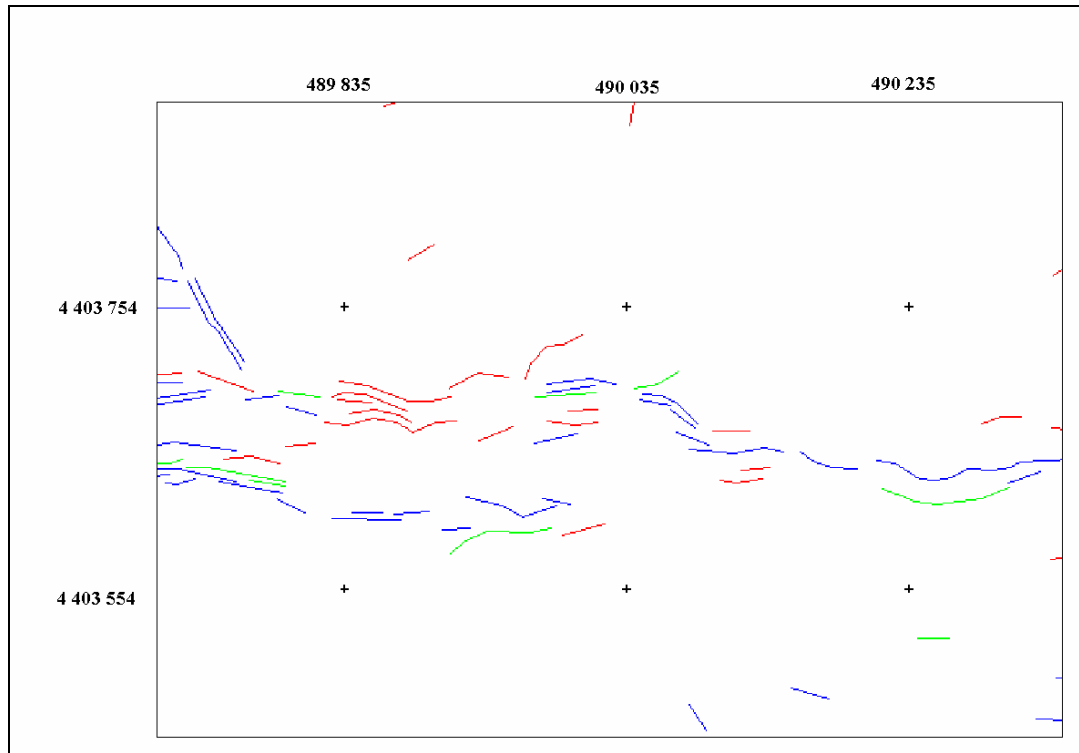


Figure C8. Classified Image of LINE 2 in Zone B with Tolerance Value = 7

APPENDIX D

In this appendix, the classified images of LINE 2 within Zone C with different tolerance values are presented.

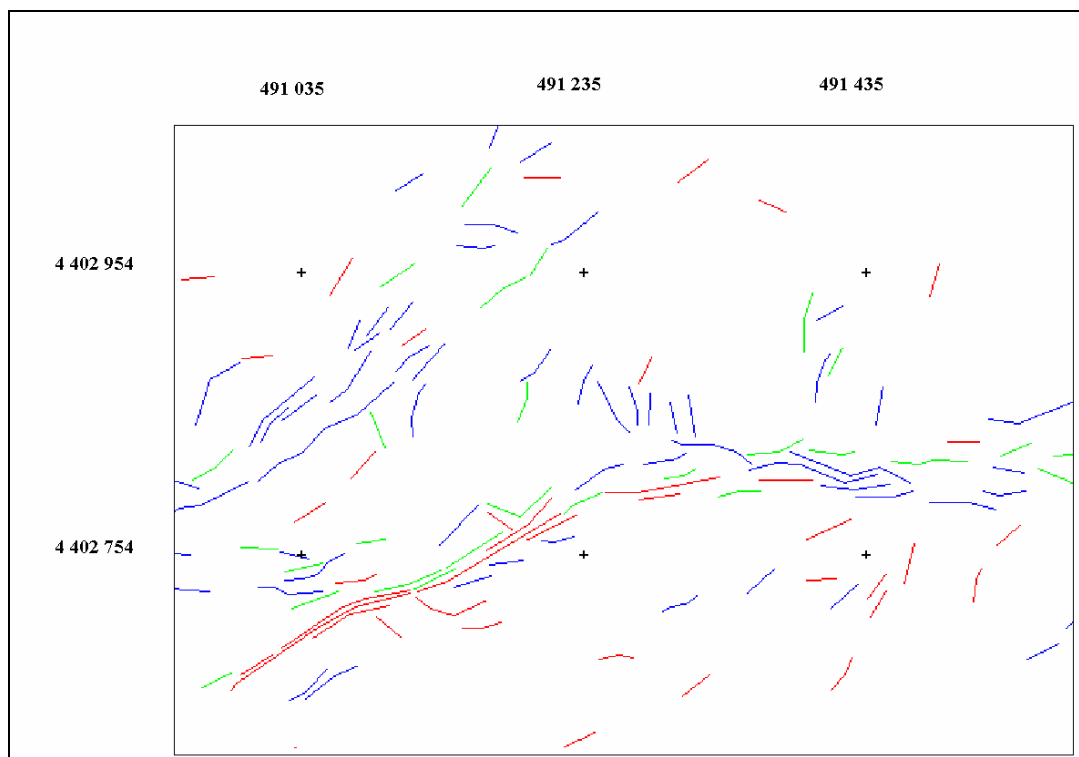


Figure D1. Classified Image of LINE 2 in Zone C with Tolerance Value = 0

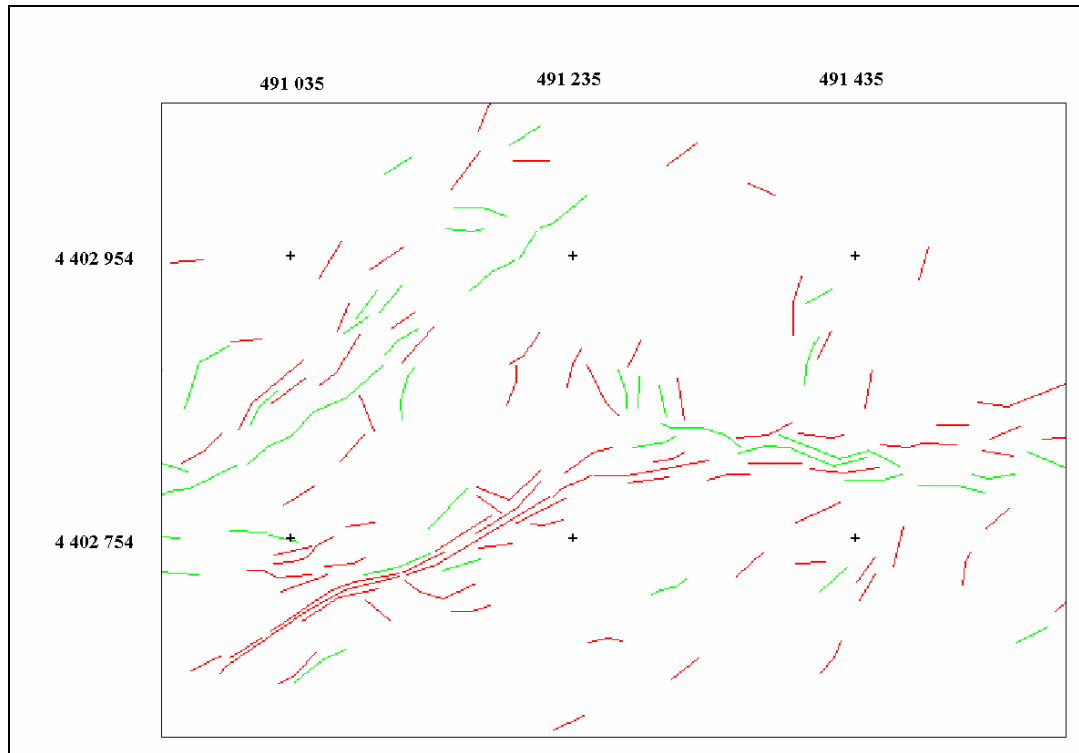


Figure D2. Classified Image of LINE 2 in Zone C with Tolerance Value = 1

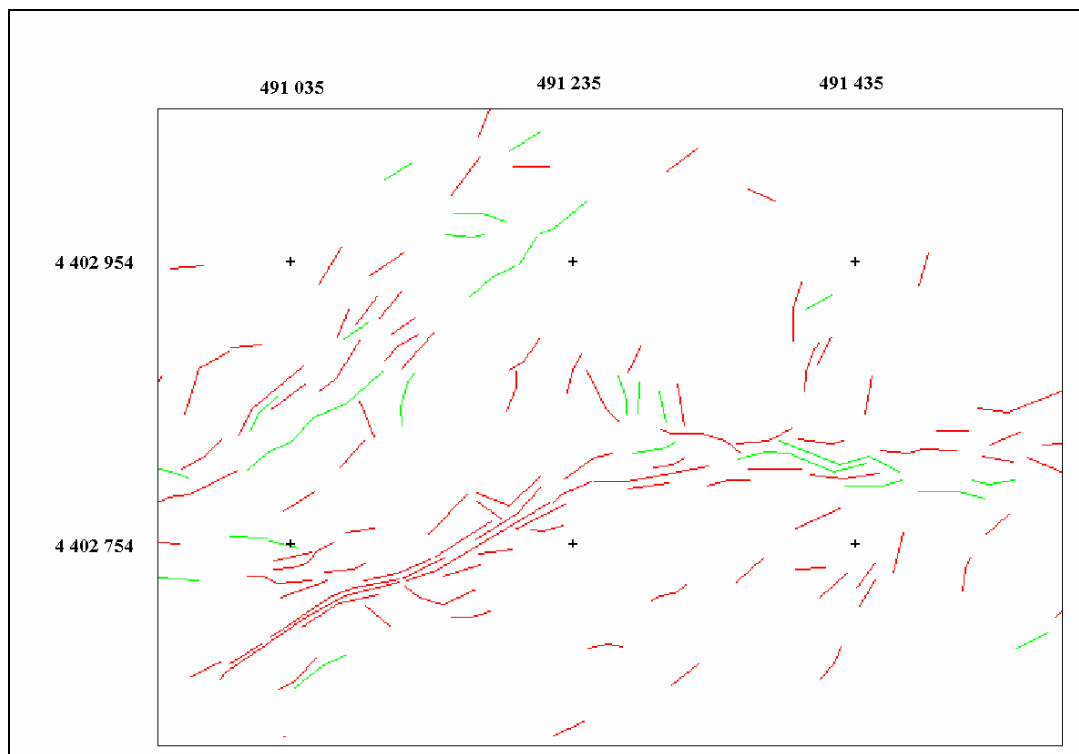


Figure D3. Classified Image of LINE 2 in Zone C with Tolerance Value = 2

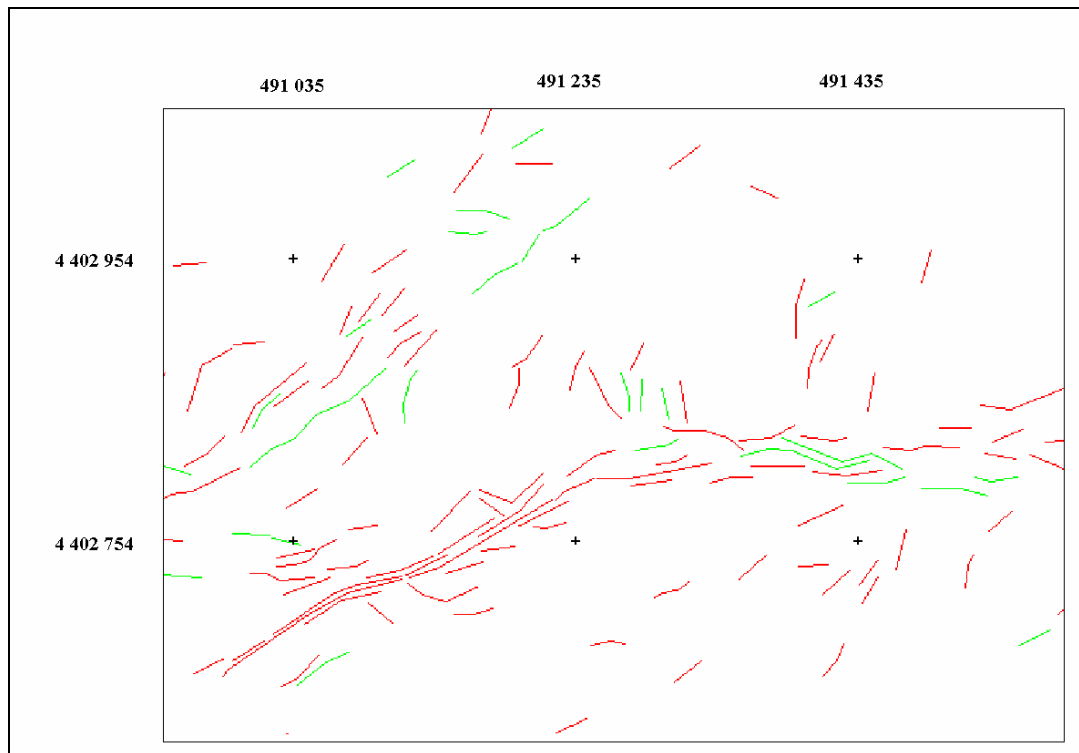


Figure D4. Classified Image of LINE 2 in Zone C with Tolerance Value = 3

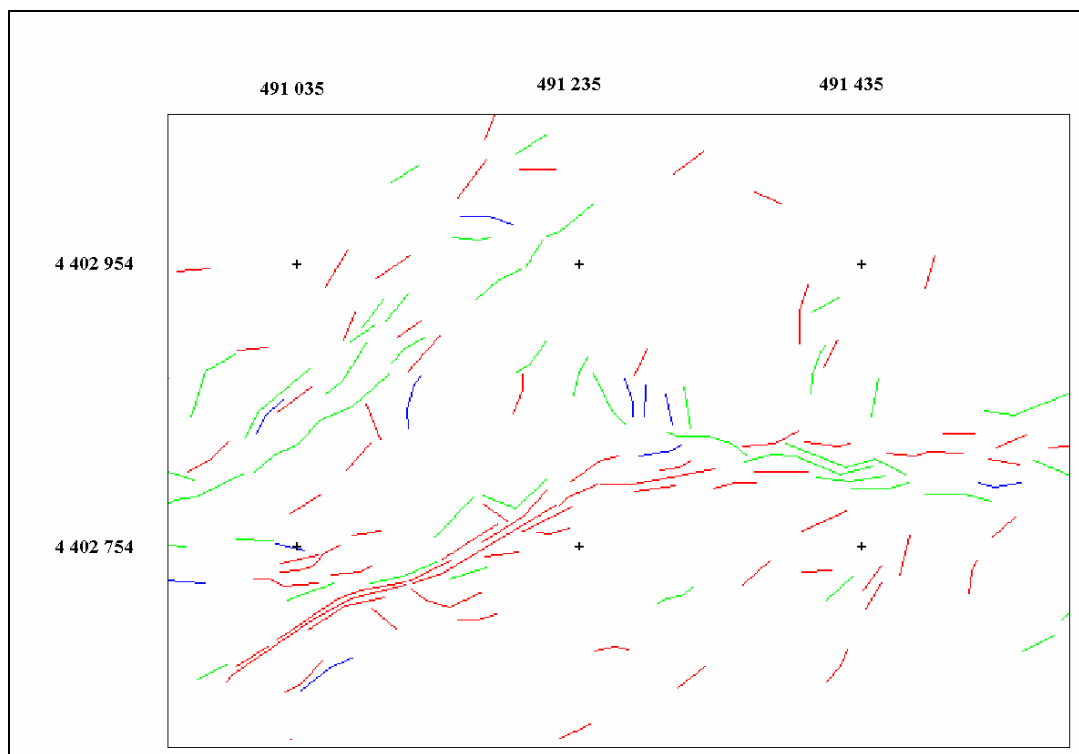


Figure D5. Classified Image of LINE 2 in Zone C with Tolerance Value = 4

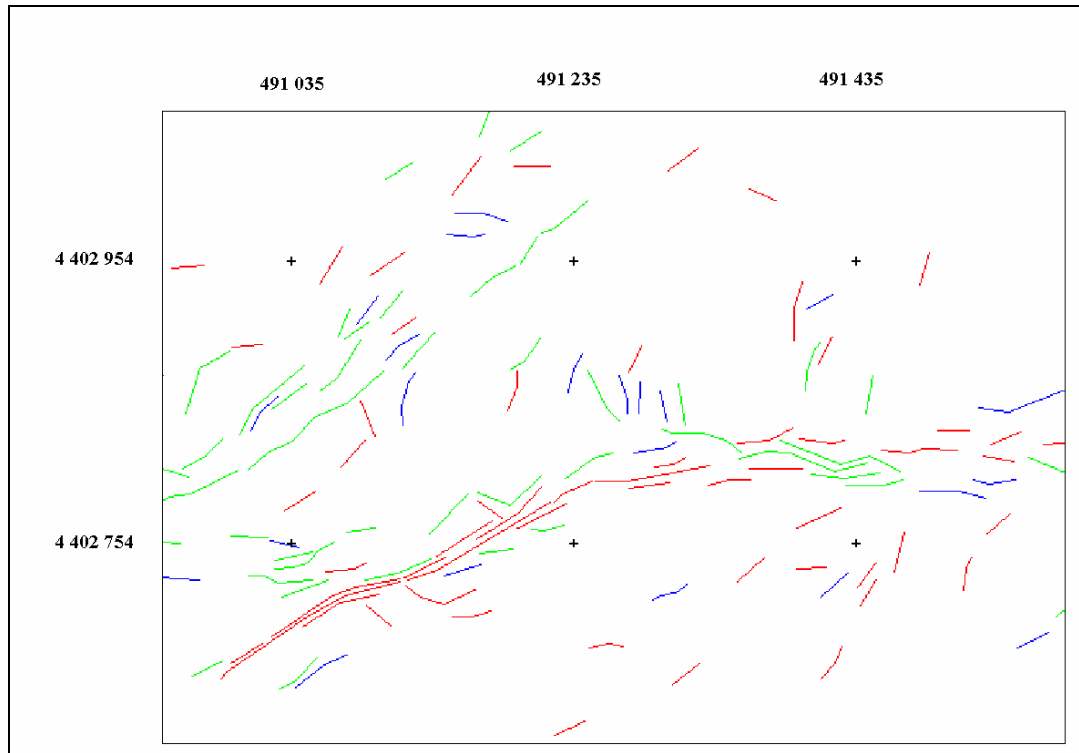


Figure D6. Classified Image of LINE 2 in Zone C with Tolerance Value = 5

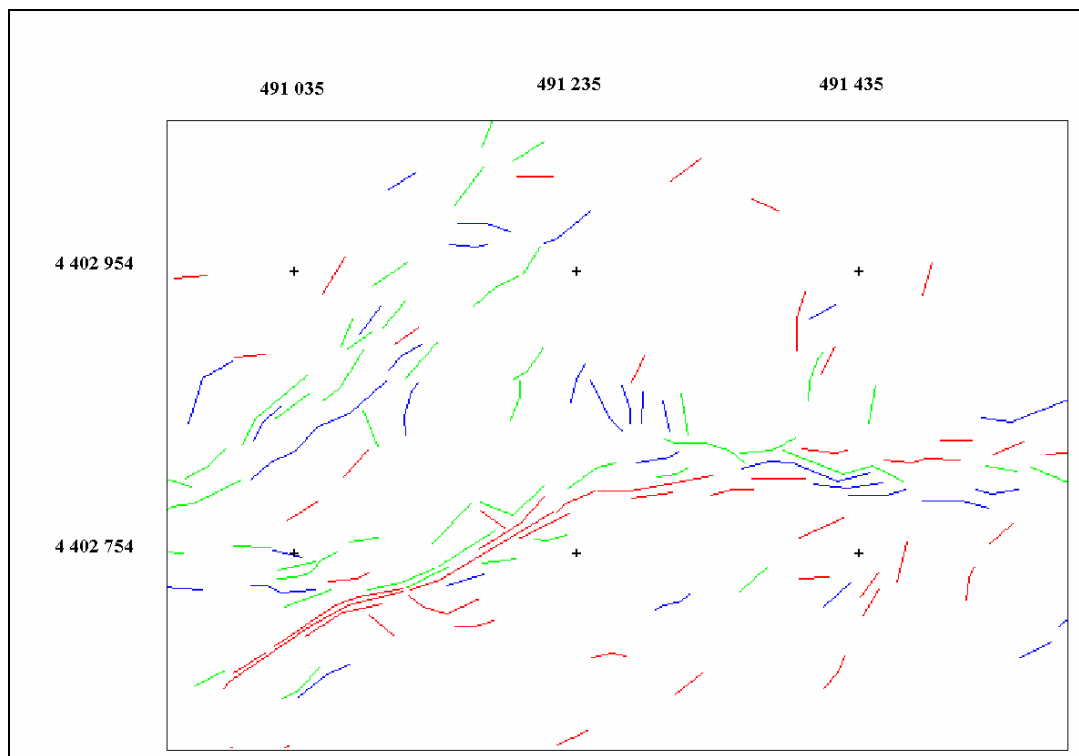


Figure D7. Classified Image of LINE 2 in Zone C with Tolerance Value = 6

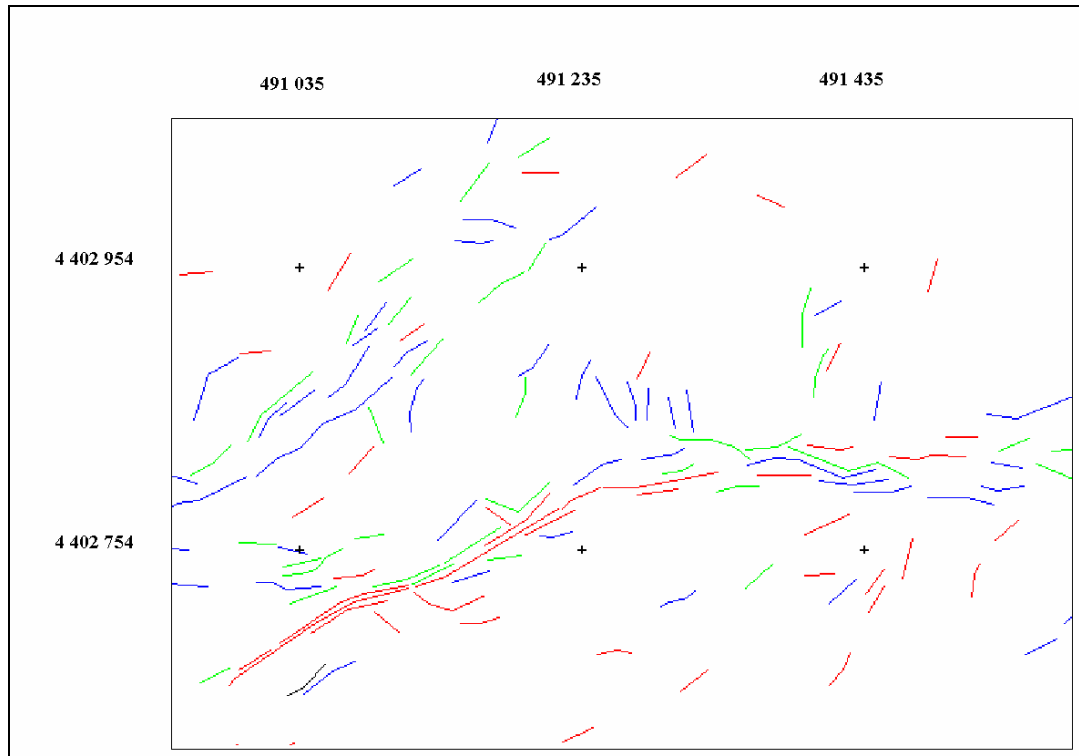


Figure D8. Classified Image of LINE 2 in Zone C with Tolerance Value = 7

APPENDIX E

In this appendix the java code for LINE ADDRESS is presented

```
import java.util.*;
import java.lang.Math;
/**
 * Write a description of class Matrix here.
 *
 * @author MINE
 * @version LINE ADDRESS
 */
public class SearchVector
{
    static int ROW = 3685;
    static int COLUMN = 4104;
    static int MAX_EXPECTED_VECTOR_NUMBER= 10000;

    static boolean[][] matrix = new boolean[ROW][COLUMN];
    static boolean[][] visited = new boolean[ROW][COLUMN];
    static int vectorId = 0;
    static Vector[] vectors = new Vector[MAX_EXPECTED_VECTOR_NUMBER];

    public static void visit(int i, int j) {
        if (visited[i][j])
            if (matrix[i][j] == true) {
                //System.out.print "[" + i + ", " + j + "]: " + vectorId + " ";
                for (int k=i-1; k<=i+1; k++)
                    for (int l=j-1; l<=j+1; l++)
                        if (((k>=0) && (k<ROW)) && ((l>=0) && (l<COLUMN)))
                            if ((matrix[k][l] == true) && (visited[k][l] == false)) {
                                //System.out.print "[" + k + ", " + l + "]: " + vectorId + " ";
                                visited[k][l] = true;
                                (vectors[vectorId]).add(new Index(k,l,true));
                                visit(k,l);
                            }
            }
        return;
    }
}
```

```

        else
            return;
    else {
        visited[i][j] = true;
        if (matrix[i][j] == true) {
            vectors[vectorId] = new Vector();
            (vectors[vectorId]).add(new Index(i,j,true));
            //System.out.print("[ "+ i + ", " + j + "]: " + vectorId + " ");
            for (int k=i-1; k<=i+1; k++)
                for (int l=j-1; l<=j+1; l++)
                    if ((k>=0) && (k<ROW) && (l>=0) && (l<COLUMN))
                        if ((matrix[k][l] == true) && (visited[k][l] == false)){
                            visited[k][l] = true;
                            //System.out.print("[ "+ k + ", " + l + "]: " + vectorId + " ");
                            (vectors[vectorId]).add(new Index(k,l,true));
                            visit(k,l);
                        }
                    vectorId++;
            System.out.println(vectorId);
            return;
        }
    }
    else return;
}
}

```

```

public static void main(String args[])
{

    LineReader reader = new LineReader("Select the txt matrix file");
    String line = "";
    Vector birler = new Vector();

    line = reader.readLine();
    int i,j; i = j = 0;
    while (line!=null) {
        if (line.equals("") == false) {
            StringTokenizer st = new StringTokenizer(line);
            while (st.hasMoreTokens()) {
                String temp = st.nextToken();
                if (temp.equals("1") || temp.equals("10")) {
                    matrix[i][j] = true;
                }
            }
        }
    }
}

```

```

        else {
            matrix[i][j] = false;
        }
        j++;
    }
    line = reader.readLine();
}
else {
    line = reader.readLine();
    i++; j=0;
}
}
reader.close();

for (int k=0; k<ROW; k++)
    for (int l=0; l<COLUMN; l++)
        visited[k][l] = false;

for (int k=0; k<ROW; k++)
    for (int l=0; l<COLUMN; l++)
        visit(k,l);

LineWriter writer = new LineWriter("Select a file to write the detected vectors");
writer.writeLine("Detected Vectors:");
writer.writeLine("-----");
for (int k=0; k<vectorId; k++) {
    String s = "vector "+k+": ";
    for (int l=0; l<vectors[k].size(); l++) {
        Index ne = (Index)vectors[k].elementAt(l);
        s = s + "[" + ne.row + "," + ne.column + "]";
    }
    writer.writeLine(s);
}
writer.close();
}
}

```

APPENDIX F

In this appendix the java code for LINE COMPARE is presented

```
import java.util.*;
/**
 * Write a description of class Main here.
 *
 * @author MINE
 * @version LINE COMPARE
 */
public class Main
{
    static int NUMBER_OF_VECTORS_TO_BE_COMPARED= 3056;

    static int NUMBER_OF_MANUAL_VECTORS = 3065;

    static int TOLERANCE = 1;

    public static void main(String Args[])
    {
        LineReader reader = new LineReader("Select the files of vectors to be
        compared");
        String line = reader.readLine();
        line = reader.readLine();
        line = reader.readLine();
        Vector[] manual = new
        Vector[NUMBER_OF_VECTORS_TO_BE_COMPARED +1];
        int i = 0;
        while (line!=null) {
            StringTokenizer st = new StringTokenizer(line, "[,]t ",false);
            st.nextToken();
            st.nextToken();
            manual[i] = new Vector();//System.out.println(i);
            while (st.hasMoreTokens()) {
                Integer s = new Integer(Integer.parseInt(st.nextToken()));
                //System.out.println(s.intValue());
                manual[i].addElement(s);
            }
        }
    }
}
```

```

    }
    line = reader.readLine(); i++; //System.out.println("\n ctr="+ctr/2);
}
// for (int i=0; i<manual.size()-1; i++)
//     System.out.print("[ "+ ((Integer)manual.elementAt(i)).intValue()+", "+
//     ((Integer)manual.elementAt(i+1)).intValue()+"] ");
reader.close();
reader = new LineReader("Select the manual vectors file");
line = reader.readLine();
line = reader.readLine();
line = reader.readLine();
Vector[] referans = new Vector[NUMBER_OF_MANUAL_VECTORS +1];
i=0;
while (line!=null) {
    StringTokenizer st = new StringTokenizer(line, "[,]\t ",false);
    st.nextToken();
    st.nextToken();
    //int ctr = 0;
    referans[i] = new Vector(); //System.out.println(i);
    while (st.hasMoreTokens()) {
        Integer s = new Integer(Integer.parseInt(st.nextToken()));
        //System.out.println(s.intValue());
        referans[i].addElement(s);
        //s = ((Integer) st.nextElement());
        //manual.addElement(s);
        //System.out.print(s+ " "); ctr++;
    }
    line = reader.readLine(); i++; //System.out.println("\n ctr="+ctr/2);
}
reader.close();
//System.out.println("manual:"+manual[0].size()+" "+manual[1].size());
//System.out.println("ref:"+referans[0].size()+" "+referans[1].size());

int[] ctr = new int[NUMBER_OF_MANUAL_VECTORS +1];
for (int l=0; l< NUMBER_OF_MANUAL_VECTORS +1; l++) ctr[l] = 0;

int[] max = new int[NUMBER_OF_VECTORS_TO_BE_COMPARED +1];
for (int l=0; l< NUMBER_OF_VECTORS_TO_BE_COMPARED +1; l++)
max[l] = 0;
int[] maxEnt = new int[NUMBER_OF_VECTORS_TO_BE_COMPARED +1];
for (int l=0; l< NUMBER_OF_VECTORS_TO_BE_COMPARED +1; l++)
maxEnt[l] = 0;

for (int j=0; j< NUMBER_OF_VECTORS_TO_BE_COMPARED +1; j++){

```

```

for (int k=0; k< NUMBER_OF_MANUAL_VECTORS +1 ;k++)
    for (int l=0; l<manual[j].size(); l++) {
        for (int m=0; m<referans[k].size(); m++) {
            int m1,m2,r1,r2;
            m1 = ((Integer) manual[j].elementAt(l)).intValue();
            m2 = ((Integer) manual[j].elementAt(l+1)).intValue();
            r1 = ((Integer) referans[k].elementAt(m)).intValue();
            r2 = ((Integer) referans[k].elementAt(m+1)).intValue();
            if ( ((m1 >= r1- TOLERANCE) && (m1 <= r1 + TOLERANCE))
                && ((m2 >= r2- TOLERANCE) && (m2 <= r2 + TOLERANCE)) ) {
                ctr[k]++;
                if (ctr[k]>max[j])
                    {max[j]=ctr[k]; maxEnt[j] = k;}
                //System.out.println("[ " + m1 + " , " + m2 + " ] : [ " +r1+ " : " + r2 + " ]");
            }
            m++;
            if (m>=referans[k].size()) break;
        }
        l++;
        if (l>=manual[0].size()) break;
    }
for (int l=0; l< NUMBER_OF_MANUAL_VECTORS +1; l++) ctr[l] = 0;
System.out.println("Vector "+j+" icin karsilastirma islemi bitti");
}
//for (int l=0; l< NUMBER_OF_MANUAL_VECTORS +1; l++)
//    System.out.println("ctr["+l+"]: "+ctr[l]);
//System.out.println("Referanstaki vektor 0 a gore diger dosyadan "+maxEnt+"
vectoru toplam "+max
//    +" ortusme yapmis");

System.out.println("Printing out");
LineWriter writer = new LineWriter("Select an .xls file for output");

writer.writeLine("Vec# \t Pixel# \t Pixel# from other file \t matching number\t
percent");
for (int j=0; j< NUMBER_OF_VECTORS_TO_BE_COMPARED +1; j++){
    double yuzde = (max[j]*100)/manual[j].size();
    writer.writeLine(j + " \t " + manual[j].size() + " \t " +maxEnt[j]+ " \t " +
max[j]+ " \t " + yuzde);
}
writer.close();
}
}

```

In this appendix the java code for LINE CLASSIFICATION is presented.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.InputStreamReader;
import java.util.Enumeration;
import java.util.Hashtable;
import java.util.StringTokenizer;
import java.util.zip.ZipEntry;
import java.util.zip.ZipFile;

import javax.swing.JDialog;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.UIManager;

/*
 * LINE CLASSIFICATION
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */

/**
 * @author MINE
```



```

*
* To change the template for this generated type comment go to
* Window>Preferences>Java>Code Generation>Code and Comments
*/
public class Changer extends JFrame {

    private javax.swing.JPanel jContentPane = null;
    private int[][] originalMatrix = null;
    private File outFile = null;
    private File TXT_File = null;
    private ZipFile ZIP_File = null;
    private Hashtable ZIP_Content = null;

    private javax.swing.JButton btnBrowse = null;
    private javax.swing.JButton btnBrowse1TXT = null;
    private javax.swing.JTextArea jTextArea = null;
    private javax.swing.JButton btnCalculate = null;
    /**
     * This is the default constructor
     */
    public Changer() {
        super();
        initialize();
    }
    /**
     * This method initializes this
     *
     * @return void
     */
    private void initialize() {

```

```

        this.originalMatrix = new int[3685][4104]; //lets form matrix
        for (int i = 0; i < 3685; i++) {
            for (int j = 0; j < 4104; j++) {
                originalMatrix[i][j] = 0;
            }
        }
        this.ZIP_Content = new Hashtable();
        this.outFile = new File("C:/output.txt"); //lets form output file
        this.setSize(496, 314);
        this.setContentPane(getJContentPane());
        this.setTitle("Matrix");
        this.setResizable(false);
        this.addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
    }
    /**
     * This method initializes jContentPane
     *
     * @return javax.swing.JPanel
     */
    private javax.swing.JPanel getJContentPane() {
        if (jContentPane == null) {
            jContentPane = new javax.swing.JPanel();
            jContentPane.setLayout(null);
            jContentPane.add(getBtnBrowse(), null);
            jContentPane.add(getBtnBrowse1TXT(), null);
            jContentPane.add(getJTextArea(), null);
        }
    }

```



```

InputStreamReader(ZIP_File.getInputStream(zipEntry)));
new BufferedReader(new

String gecici = null;
while ((gecici = zipReader.readLine()) != null) {
    if (x > 1) {
        StringTokenizer token = new StringTokenizer(gecici,
        ":"");
        while (token.hasMoreElements()) {
            token.nextToken();
            ZIP_Content.put("" + rowNum,
            token.nextToken());
        }
        rowNum++;
    }
    x++;
}

} catch (Exception ex) {
    System.out.println("Error looking ZIP file:" + ex.getMessage());
}

});
}

return btnBrowse;
}

public static void main(String args[]) throws Exception {
    UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

```

```

        new Changer().show();
    }
    /**
     * This method initializes btnBrowse1TXT
     *
     * @return javax.swing.JButton
     */
    private javax.swing.JButton getBtnBrowse1TXT() {
        if (btnBrowse1TXT == null) {
            btnBrowse1TXT = new javax.swing.JButton();
            btnBrowse1TXT.setBounds(129, 136, 218, 25);
            btnBrowse1TXT.setText("Select .txt file");
            btnBrowse1TXT.addMouseListener(new java.awt.event.MouseAdapter() {
                public void mouseClicked(java.awt.event.MouseEvent e) {
                    //Look for txt file
                    try {
                        JFileChooser choose = new JFileChooser();
                        File f = new File("c:/");
                        choose.setCurrentDirectory(f);
                        int returnVal = choose.showOpenDialog(Changer.this);
                        if (returnVal == JFileChooser.APPROVE_OPTION) {
                            String fileName = choose.getSelectedFile().getCanonicalPath();
                            TXT_File = new File(fileName);
                        }
                    } catch (Exception ex) {
                        System.out.println("Error looking TXT file:" + ex.getMessage());
                    }
                }
            });
        }
    }
}

```

```

        return btnBrowse1TXT;
    }
    /**
     * This method initializes jTextArea
     *
     * @return javax.swing.JTextArea
     */
    private javax.swing.JTextArea getJTextArea() {
        if (jTextArea == null) {
            jTextArea = new javax.swing.JTextArea();
            jTextArea.setBounds(42, 25, 402, 100);
            jTextArea.setText("First change the .xls file that contain the vector # and matching percentage into .txt
file. Select the .txt file and the ZIP file that stores the pixel addresses. Then press Compute. The output will be stored .n
C:output.txt");
            jTextArea.setLineWrap(true);
            jTextArea.setEditable(false);
            jTextArea.setEnabled(true);
        }
        return jTextArea;
    }
    /**
     * This method initializes btnCalculate
     *
     * @return javax.swing.JButton
     */
    private javax.swing.JButton getBtnCalculate() {
        if (btnCalculate == null) {
            btnCalculate = new javax.swing.JButton();
            btnCalculate.setBounds(192, 230, 96, 30);
            btnCalculate.setText("Compute");
        }
    }
}

```

```

btnCalculate.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent e) {
        try {
            if (ZIP_File != null && TXT_File != null) {
                //txt dosyasına bakıp işlemlere başlayalım
                FileReader TXT_FileReader = new FileReader(TXT_File);
                BufferedReader TXT_Reader = new BufferedReader(TXT_FileReader);
                String A_B = null;
                while ((A_B = TXT_Reader.readLine()) != null) {
                    StringTokenizer TXT_TOKEN = new StringTokenizer(A_B, "\t");
                    while (TXT_TOKEN.hasMoreElements()) {
                        int A = Integer.parseInt((String)
120         TXT_TOKEN.nextToken());
                        double B = Double.parseDouble((String)
120         TXT_TOKEN.nextToken());

                        //B sıfır ise-->1
                        if (B == 0)
                            changeMatrix(A, 1);
                        //B 100 ise-->2
                        if (B == 100)
                            changeMatrix(A, 2);
                        //B 1 ile 99 arasında ise-->3
                        if (B >= 1 && B < 100)
                            changeMatrix(A, 3);
                        //B 101'den büyük ise-->4
                        if (B > 100)
                            changeMatrix(A, 4);
                    }
                }
            }
            //FileOutputStream fileWriter = new FileOutputStream(outFile);

```

```

        FileWriter fileWriter = new FileWriter(outFile);
        //String data = "";

        for(int i=0;i<3685;i++){
            //data = "";
            for(int j=0;j<4104;j++){
                fileWriter.write(""+originalMatrix[i][j]);
                fileWriter.write(",");
                /*data += ""+ originalMatrix[i][j];
                data += ",";    */

            }
            fileWriter.write("\n");
            //data += "\n";
        }
        //fileWriter.write(data);
        fileWriter.close();
        JOptionPane.showMessageDialog(Changer.this, "Matrix Formed.");

        } else {
            JOptionPane.showMessageDialog(Changer.this, "You don't show the txt
and Zip file");
        }
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}

});
}

```



```

        return btnCalculate;
    }
    private void changeMatrix(int rowNumber, int value) {
        try {
            String rowContent = (String) ZIP_Content.get("" + rowNumber);
            StringTokenizer token = new StringTokenizer(rowContent, "[");
            while (token.hasMoreElements()) {
                int xCoor = 0;
                int yCoor = 0;
                String gecici2 = (String) token.nextToken();
                StringTokenizer tokenizer2 = new StringTokenizer(gecici2, ",");
                while (tokenizer2.hasMoreElements()) {
                    try {
                        xCoor = Integer.parseInt((tokenizer2.nextToken()));
                        yCoor = Integer.parseInt((tokenizer2.nextToken()));
                        this.originalMatrix[xCoor][yCoor] = value;
                    } catch (Exception exp2) {
                        //System.out.println(exp2.getMessage());
                    }
                }
            }
        } catch (Exception e) {
            System.out.println("Exception in cahnging matrix:" + e.getMessage());
        }
    }
} // @jve:visual-info decl-index=0 visual-constraint="10,10"

```