

DETERMINATION OF RQD BY DIGITAL IMAGE ANALYSIS

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

BY

İ. TURGUT SARIÇAM

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

JANUARY 2018

Approval of the thesis:

DETERMINATION OF RQD BY DIGITAL IMAGE ANALYSIS

submitted by **İ. TURGUT SARIÇAM** in partial fulfillment of the requirements for
the degree of **Master of Science in Mining Engineering Department, Middle East
Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Celal Karpuz
Head of Department, **Mining Engineering**

Assoc. Prof. Dr. Hasan Öztürk
Supervisor, **Mining Engineering Dept., METU**

Examining Committee Members:

Asst. Prof. Dr. İ. Ferid Öge
Mining Engineering Dept., Muğla Sıtkı Koçman University

Assoc. Prof. Dr. Hasan Öztürk
Mining Engineering Dept., METU

Asst. Prof. Dr. Mustafa Erkayaoğlu
Mining Engineering Dept., METU

Date: 18.01.2018

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: İ. Turgut Sarıçam

Signature:

ABSTRACT

DETERMINATION OF RQD BY DIGITAL IMAGE ANALYSIS

SARIÇAM, İ. Turgut

M.Sc., Department of Mining Engineering

Supervisor: Assoc. Prof. Dr. Hasan Öztürk

January 2018, 92 pages

Digital image processing and analysis methods allow us to automate routine tasks. Rock Quality Designation (RQD) is a rock quality index used in rock mechanics and geotechnical designs of slopes and underground excavations. Manual logging of hundreds of meters of rock core samples with a tape measure is a very laboursome and tedious process. In this research study, a method is introduced for the segmentation of cores and the determination of RQD from digital images of rows of core samples in core boxes in order to compute RQD in an automatic way by finding and locating natural fractures in cores and measuring intact core lengths.

First, three digital true color images of a core box, with the same camera position but different light source positions, are taken using a high resolution camera. After the detection of the core box with color thresholding, the sections of the box are detected by using Hough transform and boundary tracing algorithms. Then, the cores are extracted from each section using color thresholding. After cleaning the shadows created by different light sources using various techniques, the segmentation part is finished by combining similar regions with each other. Later, non-cylindrical parts of the cores are detected by looking at the changes caused by two different light sources.

After completion of the fracture detection in the drill core, RQD is calculated by measuring the valid centerline lengths of each core. All coding routines are developed in MATLAB 2017a. Two different core boxes with 4 and 5 rows storing HQ and NQ diameter cores having various joint/bedding plane angles are photographed several times with different core placements.

It is shown that the method is capable of separating even tightly fit joint surface cores. Moreover, it can successfully detect non-cylindrical parts of the cores, and avoid small or irregularly shaped ones which should not be included in RQD calculation.

Keywords: RQD, Digital Image Processing, Geotechnical Logging

ÖZ

DİJİTAL İMAJ ANALİZİYLE RQD TAYİNİ

SARIÇAM, İ. Turgut

Yüksek Lisans, Maden Mühendisliği Bölümü

Tez Yöneticisi: Doçent Doktor Hasan Öztürk

Ocak 2018, 92 sayfa

Dijital görüntü işleme ve analiz yöntemleri, birçok olağan işin otomatik hale getirilebilmesine olanak sağlar. Kaya Kalite Göstergesi (RQD), şevlerin ve yeraltı açıklıklarının kaya mekaniği analizleri ve jeoteknik tasarımlarında kullanılan bir kaya kalite göstergesidir. Yüzlerce metre karotun el ile incelenip raporlanması bıkırtıcı ve uğraştırıcı bir süreçtir. Bu tez çalışmasında, RQD hesabının otomatik olarak yapılması, doğal çatlakların yerlerinin belirlenmesi ve el değmemiş karot uzunluklarının ölçülmesi için, karotları karot sandıklarının dijital görüntüleri üzerinden bölütleyen ve RQD tayini yapan bir algoritma sunulmaktadır.

İlk olarak, bir karot sandığının üç adet gerçek renk dijital görüntüsü, kamera pozisyonu aynı kalmak kaydıyla, pozisyonları farklı olan üç farklı ışık kaynağı altında yüksek çözünürlüklü kamera tarafından çekilir. Karot sandığının renk eşikleme yöntemiyle teşhis edilmesinin ardından, karot sandığının bölmeleri Hough dönüşümü ve sınır takip algoritmaları ile bulunur. Daha sonra, renk eşikleme yöntemi kullanılarak, her bir bölmedeki karotlar belirlenir. Farklı ışık kaynakları tarafından oluşturulan gölgeler çeşitli yöntemlerle temizlendikten sonra, benzer bölgelerin birbirleri ile birleştirilmesiyle bölütleme işlemi sonlandırılır. Ardından, iki farklı ışık kaynağından dolayı oluşan değişimler incelenerek, karotların silindirik olmayan

bölümleri teşhis edilir. Çatlak teşhisinin tamamlanmasının ardından, her bir karotun eksen uzunluğu ölçülerek RQD hesaplanır. Tüm kodlar MATLAB 2017a'da yazılmıştır. 4 ve 5 bölmeli, içerisinde çeşitli eklem ve katmanlaşmış düzlemler içeren NQ ve HQ çapında karotlar bulunan iki adet karot sandığı farklı karot yerleşimleri ile fotoğraflanmıştır.

Algoritmanın, yüzeyleri birbirine sıkı bir şekilde temas eden karotları bile doğru şekilde bölütleyebildiği gösterilmiştir. Ek olarak, algoritma, karotların silindirik olmayan yüzeylerini başarılı bir şekilde teşhis edebilir ve küçük olması veya düzgün bir şekle sahip olmaması sebebiyle RQD hesabına dahil edilmemesi gereken karotları da RQD hesabına dahil etmez.

Anahtar Kelimeler: RQD, Dijital Görüntü İşleme, Jeoteknik Loglama

ACKNOWLEDGEMENTS

I would like to express my deep and sincere appreciation to my supervisor Assoc. Prof. Dr. Hasan Öztürk for his guidance, support and encouragement, thanks to which this study has been completed successfully.

I owe my special thanks to the examining committee members Asst. Prof. Dr. Mustafa Erkayaoğlu and Asst. Prof. Dr. İ. Ferid Öge for their comments and suggestions and for serving on the M. Sc. thesis committee.

The author would also like to thank the Scientific and Technological Research Council of Turkey, TUBITAK, Grant No:116M692, for financial support.

To My Mother

TABLE OF CONTENTS

ABSTRACT.....	v
ÖZ	vii
ACKNOWLEDGEMENTS	ix
TABLE OF CONTENTS.....	xi
LIST OF TABLES.....	xiii
CHAPTERS	
1. INTRODUCTION.....	1
1.1 General Remarks	1
1.2 Problem Statement	1
1.3 Objectives of Study	2
1.4 Research Methodology.....	2
2. LITERATURE SURVEY.....	3
2.1 Introduction to Image Processing.....	3
2.1.1 Color Spaces	3
2.1.2 Edge Detection.....	11
2.1.3 Filtering.....	13
2.1.4 Labeling	15
2.1.5 Convex Hull.....	17
2.2 Algorithmic Background.....	19
2.2.1 Boundary Tracing	19
2.2.2 Canny Edge Detector	21

2.2.3	Dijkstra's Algorithm	24
2.2.4	Hough Transform.....	26
2.2.5	Image Differencing	28
2.2.6	RANSAC	29
2.3	Mining Background.....	29
2.3.1	Geotechnical Core Logging	30
2.3.2	RQD	40
2.4	Theoretical Background	44
2.4.1	Graph Theory	44
3.	DETERMINATION OF RQD BY DIGITAL IMAGE ANALYSIS	49
3.1	Material and Methods.....	50
3.2	Theory and Calculations.....	52
3.2.1	Box detection	54
3.2.2	Row Detection	55
3.2.3	Core Detection	58
3.2.4	Depth Image Creation	71
3.2.5	RQD Calculation.....	77
3.2.5.1	RQD Results.....	79
4.	CONCLUSIONS AND RECOMMENDATIONS	83
	REFERENCES	87
	APPENDIX.....	89

LIST OF TABLES

TABLES

Table 1. Wavelengths and the corresponding relative spectral power S of the CIE 1931 primaries (Koschan & Abidi, 2008).....	4
Table 2. Sample geotechnical core logging sheet.....	31
Table 3. Fracture types and their abbreviations	32
Table 4. Rock mass rating system (Bieniawski, 1989).....	34
Table 5. Classification of individual parameters used in the Tunnelling Quality Index Q (Barton et al., 1974)	35
Table 6. Weathering classification (Brown, 1981)	38
Table 7. Intact rock strength classification (Brown, 1981).....	40
Table 8. Comparison of RQD values computed manually and algorithmically.	81

LIST OF FIGURES

FIGURES

Figure 1. Research methodology	2
Figure 2. Representations of CIE color matching functions values r , g , and b for the 2° CIE standard observer (Koschan & Abidi, 2008)	5
Figure 3. CIE color matching functions $x(\lambda)$, $y(\lambda)$, and $z(\lambda)$ for 2° CIE standard observer (Koschan & Abidi, 2008).....	6
Figure 4. 1931 CIE chromaticity diagram for spectral colors in 400 – 700 nm range	7
Figure 5. 1931 CIE chromaticity diagram with IJK triangle	8
Figure 6. RGB color space.....	9
Figure 7. One-dimensional masks.....	12
Figure 8. Roberts edge detectors.....	12
Figure 9. Prewitt edge detectors. (a) Horizontal edge detector. (b) Vertical edge detector. (c) (d) Diagonal edge detectors.....	12
Figure 10. Sobel edge detectors. (a) Horizontal edge detector. (b) Vertical edge detector. (c) (d) Diagonal edge detectors.....	12
Figure 11. A section of an image f that is under a 3x3 filter mask w	13
Figure 12. Illustration of 1-D correlation and convolution. (Gonzalez & E., 2008, p. 169)	15
Figure 13. An example labeling operation.....	17
Figure 14. Process for finding the convex hull for an example region.....	19
Figure 15. An example boundary tracing operation. The operation starts from (a) and ends with (o)	20
Figure 16. Two possible horizontal edge orientations in a 3×3 region.....	22
Figure 17. The angle ranges of edge normals in a 3×3 region for the four orientations	22
Figure 18. The execution of Dijkstra's algorithm	26
Figure 19. A line in X-Y plane	26
Figure 20. Hough parameter space for a sample image.....	27

Figure 21. Correct and incorrect ways of measuring the length of a core sample (Deere & Deere, 1989, p. 14).....	41
Figure 22. RQD logging (Deere & Deere, 1989, p. 12).....	43
Figure 23. Diagrams of graphs G and H.....	45
Figure 24. Graph, G	46
Figure 25. Incidence and adjacency matrices of graph G	46
Figure 26. A sample graph.....	47
Figure 27. Equipment used in the thesis study	50
Figure 28. Equipment setup	51
Figure 29. Core boxes used in algorithm development process	52
Figure 30. Level-1 data flow diagram.....	53
Figure 31. Level-2 data flow diagram for 'detect box' process.....	54
Figure 32. Box detection using color thresholding.....	55
Figure 33. Level-2 data flow diagram for 'detect rows' process	56
Figure 34. Row detection.....	57
Figure 35. Level-2 data flow diagram for 'detect cores' process	58
Figure 36. Shadow images.....	60
Figure 37. Detected cracks.....	61
Figure 38. Detected shadows	62
Figure 39. Core mask.....	63
Figure 40. Contour connection graph	64
Figure 41. Convex hull	65
Figure 42. Connected concave points	66
Figure 43. Regions after separating touching cores.....	67
Figure 44. Merged labels	69
Figure 45. Detected cores	70
Figure 46. Detected cores in another core box	71
Figure 47. Level-2 data flow diagram for 'create depth image' process	72
Figure 48. Fracture paths	72
Figure 49. Row orientation	74
Figure 50. Depth image output for test core box 1.	75
Figure 51. Depth image output for test core box 2.	76

Figure 52. Level-2 data flow diagram for 'compute RQD' process	77
Figure 53. Detected centerline of a core	78
Figure 54. Camera view	89
Figure 55. Inside of the tent with all lights turned on and core box outside.....	89
Figure 56. Tent with core box outside	90
Figure 57. Ends of the rail of core box tray	91
Figure 58. Inside of the tent. Right light is turned on. The core box is inside.....	92

CHAPTER 1

INTRODUCTION

1.1 General Remarks

Digital image processing and analysis methods allow us to automate routine tasks. Core logging is one of the routine tasks done as part of a rock characterization procedure of geotechnical and rock mechanical designs of slopes and underground excavations, which is performed by using several tools, such as tape measure, geological pick, pocket knife, chemical solutions, hand lenses, and protractors. In addition to being exhaustive, core logging results are biased, because it depends on the experience of the person who performs the logging. Therefore, automation of core logging will help us decide RQD in a bias-free manner, in addition to providing a scalable approach to the problem. This study mainly focuses on the automation process of detecting naturally broken cores inside a core box properly, while trying to provide a low-cost solution.

1.2 Problem Statement

Determination of RQD is one of the routine jobs in a mine, which is manually done by geologists and mining engineers for thousands of meters of core. Doing this manually is a tedious, time consuming, laboursome and bias-prone process.

1.3 Objectives of Study

The objective of this study is to provide an accurate, fast, easy-to-setup, and relatively cheap method that can be used to determine RQD in an automated way that is free of human bias.

1.4 Research Methodology

The methodology shown in Figure 1 was followed in this research study.

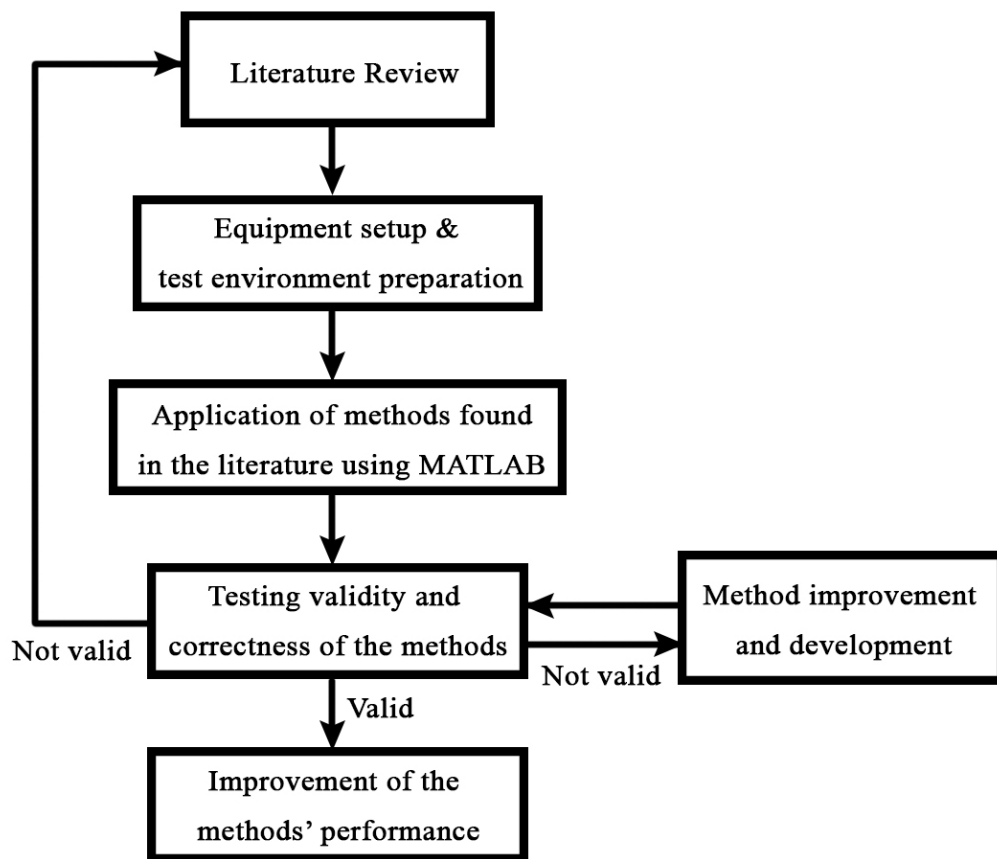


Figure 1. Research methodology

CHAPTER 2

LITERATURE SURVEY

Literature was comprehensively reviewed within the scope of this study and the details are given in this chapter.

This chapter is divided into four subsections, which are introduction to image processing, algorithmic background, mining background and theoretical background. Each subsection explains topics that are used in or related to this thesis study.

2.1 Introduction to Image Processing

In this subsection, basic image processing terms and operations that are related to this thesis study are explained, which are color spaces, edge detection, filtering, labeling, and convex hull.

2.1.1 Color Spaces

Colors are part of the electromagnetic spectrum, which means they are measurable as a physical quantity. However, identifying colors by their physical quantity is not useful when representing them, since, most of the time, they are measured and evaluated physiologically by humans and sensors. Colors must be represented considering mathematical demands, technical conditions, and human perception.

Image sensors and retina use *additive color mixture model* to perceive the color when different-wavelength lights hit the same place on the retina or on the sensor. These electromagnetic waves are mixed into a single color by using projectile overlapping. According to Grassmann's First Law of additive color mixture (Grassmann, 1853), any color stimulus can be created using a set of three color stimuli which are independent from each other. Grassmann's First Law can be written as:

$$\mathbf{M} = R \cdot \mathbf{R} + G \cdot \mathbf{G} + B \cdot \mathbf{B}$$

where primary colors \mathbf{R} , \mathbf{G} , and \mathbf{B} are indicated by R , G , and B in the mixed color \mathbf{M} . This law is also valid for a set \mathbf{M} of n colors:

$$\mathbf{M} = \{\mathbf{M}_i \mid i = 1, \dots, n\} \text{ with } \mathbf{M}_i = R_i \cdot \mathbf{R} + G_i \cdot \mathbf{G} + B_i \cdot \mathbf{B} \text{ and}$$

$$\mathbf{M}_{1\dots n} = \mathbf{R} \cdot \sum_{i=1}^n R_i + \mathbf{G} \cdot \sum_{i=1}^n G_i + \mathbf{B} \cdot \sum_{i=1}^n B_i$$

In 1931, International Lighting Commission CIE (Commission Internationale de L'Eclairage) announced the color matching functions and monochromatic primary values as the definition of the colormetric 2° CIE standard observer. Although the commission introduced 10° CIE standard observer in 1964, here, 2° is the size of the visual field that should guarantee color perception without stimulation of the rods, which are photoreceptors responsible for vision at low light levels. Table 1 shows wavelengths and corresponding spectral power of the primary values.

Table 1. Wavelengths and the corresponding relative spectral power S of the CIE 1931 primaries (Koschan & Abidi, 2008)

Primary	λ (in nm)	S
R	700.0	72.09
G	546.1	1.379
B	435.8	1.000

By the color stimuli standardized with $S(\lambda)$, the spectral tristimulus values $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, and $\bar{b}(\lambda)$ are produced. The following applies:

$$\bar{m}(\lambda) = \bar{r}(\lambda) \cdot \mathbf{R} + \bar{g}(\lambda) \cdot \mathbf{G} + \bar{b}(\lambda) \cdot \mathbf{B}$$

Color matching function of each stimulus in the spectral tristimulus values are shown in Figure 2.

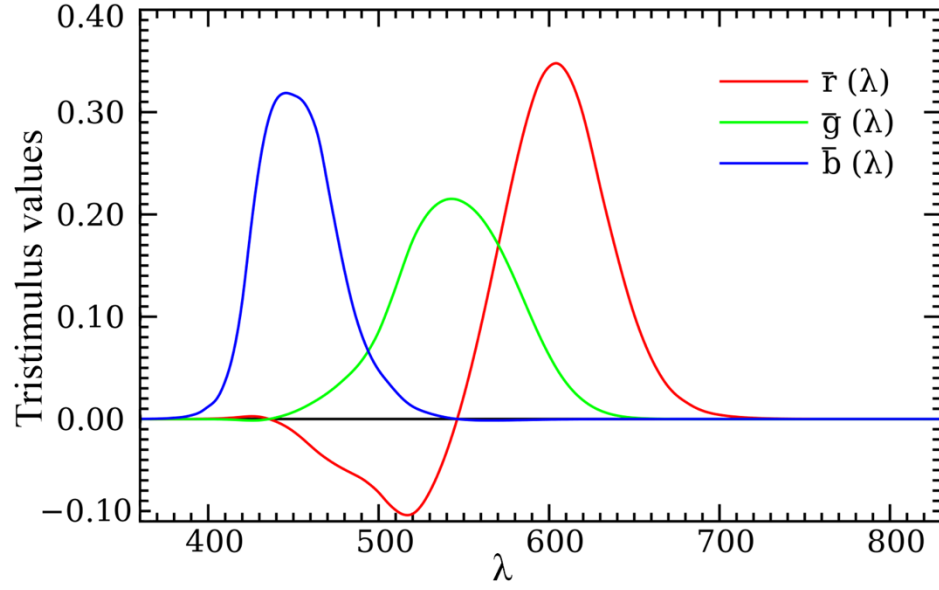


Figure 2. Representations of CIE color matching functions values \bar{r} , \bar{g} , and \bar{b} for the 2° CIE standard observer (Koschan & Abidi, 2008)

CIE developed virtual primary values X , Y , and Z to be able to create a color mixture which covers the entire visible wavelength area. There is not any corresponding physical spectral distribution for these virtual values. Following linear transformation is used to convert real spectral value curves into virtual color matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ by 2° CIE standard observer:

$$\begin{bmatrix} \bar{x}(\lambda) \\ \bar{y}(\lambda) \\ \bar{z}(\lambda) \end{bmatrix} = \begin{bmatrix} 0.49000 & 0.31000 & 0.20000 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00000 & 0.01000 & 0.99000 \end{bmatrix} \cdot \begin{bmatrix} \bar{r}(\lambda) \\ \bar{g}(\lambda) \\ \bar{b}(\lambda) \end{bmatrix}$$

Spectral values after the transformation can be seen in Figure 3.

In order to obtain the *standard color values* X , Y and Z , transformed spectral values $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ and a color stimulus function are used. A color stimulus function indicates radiation that causes a color stimulus in the eye. Color stimulus function is defined as:

$$\varphi(\lambda) = \begin{cases} S(\lambda) & \text{for luminous objects} \\ S(\lambda) \cdot R(\lambda) & \text{for body colors} \\ S(\lambda) \cdot R(\lambda) + S_F(\lambda) & \text{for fluorescent samples} \end{cases}$$

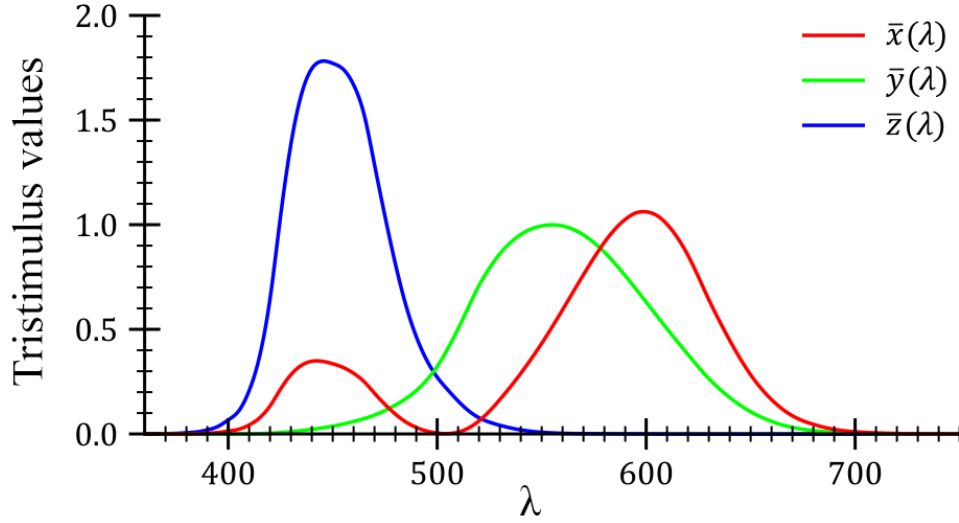


Figure 3. CIE color matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ for 2° CIE standard observer (Koschan & Abidi, 2008)

where $S(\lambda)$ is the spectral power, $R(\lambda)$ is the spectral reflection factor and $S_F(\lambda)$ is the fluorescence function. Then, the standard color values are obtained by following equations:

$$X = k \cdot \int \varphi(\lambda) \cdot \bar{x}(\lambda) d\lambda$$

$$Y = k \cdot \int \varphi(\lambda) \cdot \bar{y}(\lambda) d\lambda$$

$$Z = k \cdot \int \varphi(\lambda) \cdot \bar{z}(\lambda) d\lambda$$

where, the normalization factor k is defined as:

$$k = \frac{100}{\int S(\lambda) \cdot \bar{y}(\lambda) d\lambda}$$

Before moving onto color spaces, one last topic that needs to be described for a thorough understanding is *chromaticity diagrams*. *Chromaticity* is the color information that remain after eliminating brightness information. Hence, it is only defined by hue and saturation. A chromaticity diagram is a two-dimensional graphic

representation of an intersecting plane of a three-dimensional color space. For the *RGB cube*, this intersecting plane is the unit plane $R + G + B = 1$, which results in an equilateral triangle, also known as the *Maxwell color triangle* (Koschan & Abidi, 2008). The intersection between the unit plane and the observed color position is denoted as M . For a color M ,

$$r_M = \frac{R}{R + G + B}, g_M = \frac{G}{R + G + B}, \text{ and } b_M = \frac{B}{R + G + B}$$

relationship is found by using the Newtonian gravity formulation. A more commonly used representation is the Cartesian representation. In this representation, r is the abscissa, g is the ordinate and blue lies in the origin. The relationship above also applies to the Cartesian representation as well. An example of this representation can be observed in Figure 4.

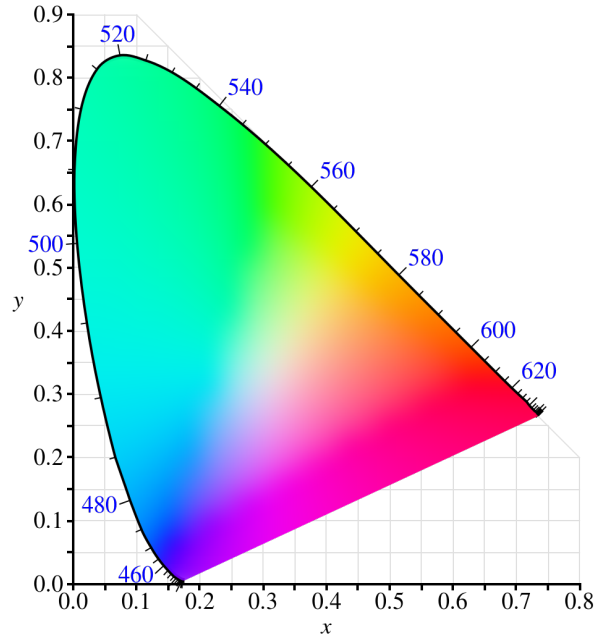


Figure 4. 1931 CIE chromaticity diagram for spectral colors in 400 – 700 nm range

Results of addition of two colors are defined as a *color gamut*. In Figure 5, by adding the color I and the color J together with different proportions, all values lying on line IJ can be created. Similarly, by adding the color J and the color K with different

proportions, all values lying on line JK can be produced. With an addition of a third color K, all colors within the triangle IJK can be generated.

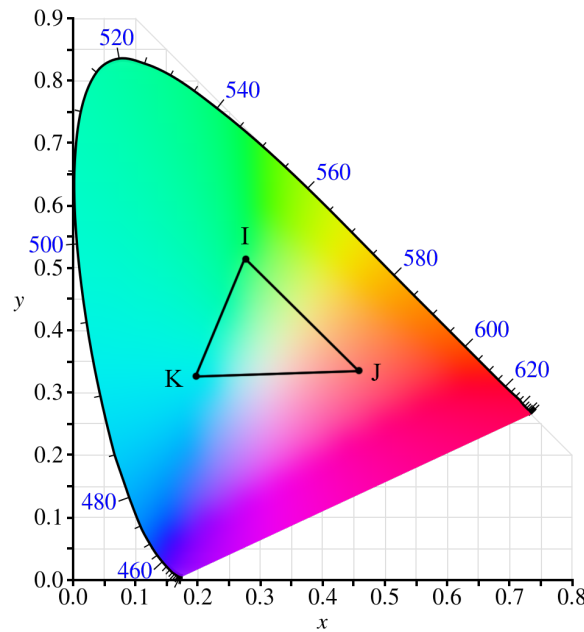


Figure 5. 1931 CIE chromaticity diagram with IJK triangle

Color spaces are *color coordinate systems* which represent the image values of a color image. *Color distance* is the difference of two image values in a color space. Different color distances in a color model are not identical to the color differences observed by humans.

Color spaces can be investigated under three different categories, namely physics and technics-based color spaces, uniform color spaces, and perception-based color spaces. Physics and technics-based color spaces are composed of RGB, CMY(K), YIQ, YUV, $YC_B C_R$, Kodak PhotoCD $YC_1 C_2$, and $I_1 I_2 I_3$. Uniform color spaces are CIELAB and CIELUV. Finally, perception-based color spaces are composed of HIS, HSV, and opponent color spaces.

In this research study, RGB and CIELAB color spaces are used. Therefore, only these two color spaces are explained below.

RGB color space is the most used color space in computer technology. It is formed by additive mixture of three primary colors, which are **R**, **G**, and **B** (Table 1). RGB color space is a 3-dimensional orthogonal color space whose base vectors are the primary colors. Therefore, any color can be obtained by combining the base vectors linearly. For a digital color image **C** with three channels, the three vector components are shown as

$$\mathbf{C}(x, y) = (R(x, y), G(x, y), B(x, y))^T = (R, G, B)^T$$

where (x, y) is an image pixel. These values are called as *tristimulus values*.

In RGB color space, each color vector created by integers R, G, and B characterize a single color, provided that $0 \leq R, G, B \leq G_{max}$ where G_{max} is the largest value of a color component. A representation of RGB color space can be observed in Figure 6.

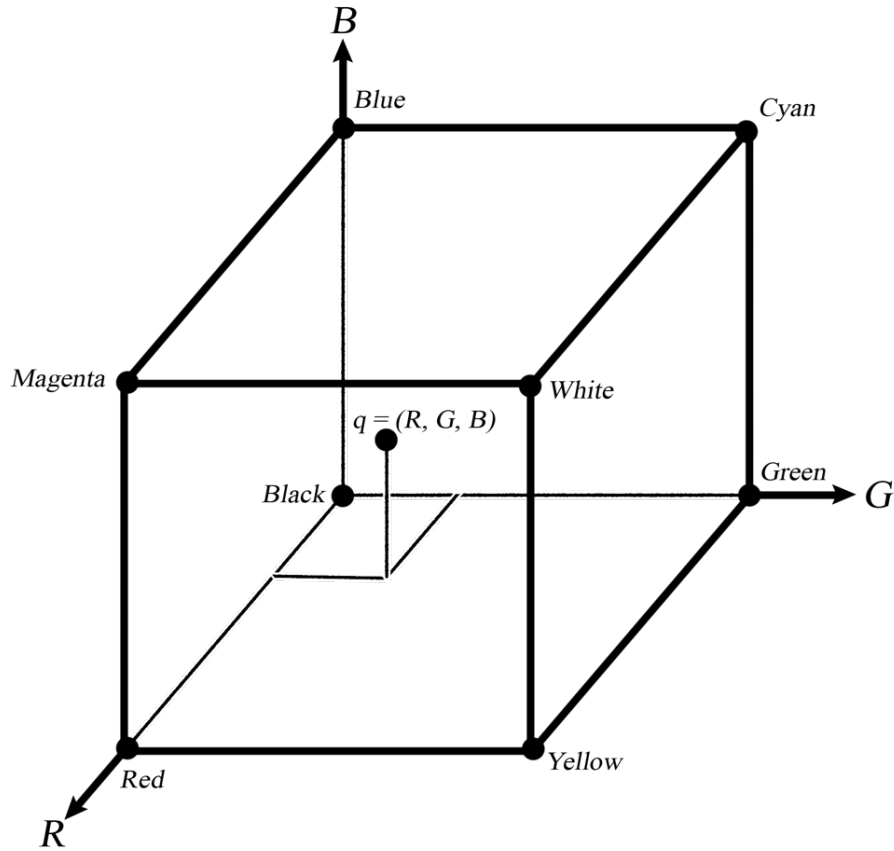


Figure 6. RGB color space

The cube's boundaries are formed by the primary colors, which are red, green, and blue, the complementary colors, which are yellow, magenta and cyan, and the achromatic colors, which are black and white. All achromatic colors lie on the diagonal created by combining the corners represented by black and white. With a linear combination of all three vectors, almost every color in the visible spectrum can be generated.

CIELAB is a uniform color space. In a uniform color space, the amount of change between color coordinates corresponds to the same amount of perceptible change in the visible color tones and color saturation. Although CIELAB is a uniform color space, there is no uniform color space that is perfectly uniform, which does not involve any biased or distorted representation. Although uniform color spaces are computationally expensive to convert from other color spaces, they are quite useful when comparing similar colors.

L*a*b* color space, less known name of CIELAB color space, is recommended by CIE in 1976. It is also adopted by the German institute DIN (Deutsches Institut für Normierung). The color space is produced from CIE XYZ primary system by using the following conversion:

$$\begin{aligned} X^* &= \sqrt[3]{X/X_n} & \text{for } X/X_n > 0.008856 \\ X^* &= 7.787 \cdot (X/X_n) + 0.138 & \text{for } X/X_n \leq 0.008856 \\ \\ Y^* &= \sqrt[3]{Y/Y_n} & \text{for } Y/Y_n > 0.008856 \\ Y^* &= 7.787 \cdot (Y/Y_n) + 0.138 & \text{for } Y/Y_n \leq 0.008856 \\ \\ Z^* &= \sqrt[3]{Z/Z_n} & \text{for } Z/Z_n > 0.008856 \\ Z^* &= 7.787 \cdot (Z/Z_n) + 0.138 & \text{for } Z/Z_n \leq 0.008856 \end{aligned}$$

Then,

$$L^* = 116 \cdot Y^* - 16,$$

$$a^* = 500 \cdot (X^* - Y^*),$$

$$b^* = 200 \cdot (Y^* - Z^*)$$

While L^* channel represents lightness, a^* represents green and red, and b^* represents blue and yellow colors.

2.1.2 Edge Detection

Edge direction and strength at location (x, y) of an image f is simply found by calculating the *gradient vector*, which is denoted by ∇f , defined as:

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

This vector points in the direction of the greatest rate of change of f at location (x, y) . The *magnitude* of the vector, denoted as $M(x, y)$, is calculated by:

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

Magnitude of the gradient is the value of the rate of change in the direction of the gradient vector. The direction of the gradient vector is the angle $\alpha(x, y)$ defined as:

$$\alpha(x, y) = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

The angle of the gradient vector at an arbitrary point (x, y) , $\alpha(x, y)$, is orthogonal to the direction of the edge at that point.

Here, g_x , g_y , $M(x, y)$ and $\alpha(x, y)$ are all images of the same size as the input image f .

To obtain the gradient of an image, partial derivatives of the image, $\partial f / \partial x$ and $\partial f / \partial y$, are required to be calculated at every pixel location. Since the partial derivatives are

$$g_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y)$$

and

$$g_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y)$$

they can be calculated for all pixels in the image by filtering $f(x, y)$ with the 1-D masks shown in Figure 7. In this figure, (a) is the mask for calculating g_x , while (b) is the mask for calculating g_y .

-1	-1
1	1

(a)
(b)

Figure 7. One-dimensional masks

The masks used for computing the gradient are also called *gradient operators*, *edge operators*, *edge detectors* or *difference operators*. There are several different commonly utilized gradient operators. Figure 8, Figure 9, and Figure 10 shows Roberts, Prewitt and Sobel edge detectors (Gonzalez & Woods, 2008), respectively.

-1	0	0	-1
0	1	1	0

(a)
(b)

Figure 8. Roberts edge detectors

-1	-1	-1	-1	0	1	1	-1	-1	0
0	0	0	-1	0	1	-1	0	1	-1
1	1	1	-1	0	1	-1	-1	0	0

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

Figure 9. Prewitt edge detectors. (a) Horizontal edge detector. (b) Vertical edge detector. (c) (d) Diagonal edge detectors

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

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

Figure 10. Sobel edge detectors. (a) Horizontal edge detector. (b) Vertical edge detector. (c) (d) Diagonal edge detectors

Filtering an image $f(x, y)$ with one of these edge detectors results in an image with gradient values computed for each pixel in the image, i.e. a *gradient image*.

2.1.3 Filtering

Filtering is used in image processing for several different purposes such as image enhancement and edge detection. A *spatial filter* consists of a *neighborhood*, and a *predefined operation*. The predefined operation is applied to the image pixels inside the neighborhood. The output of the filtering operation is a pixel with a new value at the same coordinates of the center of the neighborhood. If the filtering operation is linear, then the filter is called a *linear spatial filter*. Otherwise, it is a *nonlinear spatial filter*. Only linear filters are discussed within the scope of this research.

There are two important concepts that are related to linear spatial filtering, which are *correlation* and *convolution*. These two operations are very similar to each other.

Correlation is the process that a filter is moved over the image and sum of the products of overlapping values at each location is computed. Figure 11 illustrates a section of an image (a) that is under filter mask w that has a 3x3 neighborhood (b).

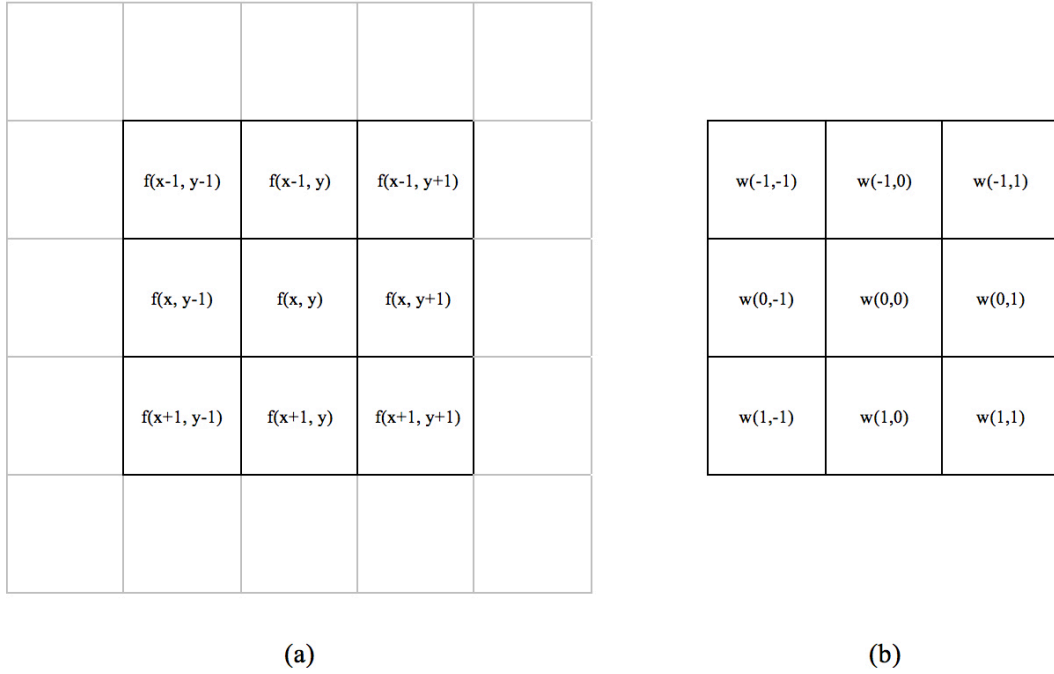


Figure 11. A section of an image f that is under a 3x3 filter mask w

The response of the filter at any point (x, y) in the image is, denoted as $g(x, y)$:

$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots \\ + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1)$$

Correlation filtering of an image $f(x, y)$ of size $M \times N$ with a filter $w(x, y)$ of size $m \times n$ is computed as follows:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x + s, y + t)$$

where, it is assumed that $m = 2a + 1$ and $n = 2b + 1$, a and b are positive integers. Although filters having an even size can be used, here, the focus is on odd-size filters.

Convolution is a very similar process to correlation. In convolution, the computation procedure is the same. However, the filter or the image is rotated 180° before the filter is applied. Note that rotating the image has the same effect as rotating the filter.

Convolution filtering of an image $f(x, y)$ of size $M \times N$ with a filter $w(x, y)$ of size $m \times n$ is computed as follows:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x - s, y - t)$$

Figure 12 illustrates correlation and convolution procedures applied to the same image f with the same spatial filter w in 1-dimension.

Figure 12(a) shows the image and the filter. In Figure 12(b), the filter is aligned at the starting position. Here, the filter does not completely overlap with the image. A solution for this is to zero-pad the image, i.e. adding 0s to either side of the image, which is illustrated in Figure 12(c). Figure 12(d) and Figure 12(e) shows the filter shifted one and four positions. The final position of the filter is shown in Figure 12(f). Figure 12(g) and Figure 12(h) demonstrates the final correlation results as fully and cropped, respectively. The right side of Figure 12 shows the same procedure, but for convolution.

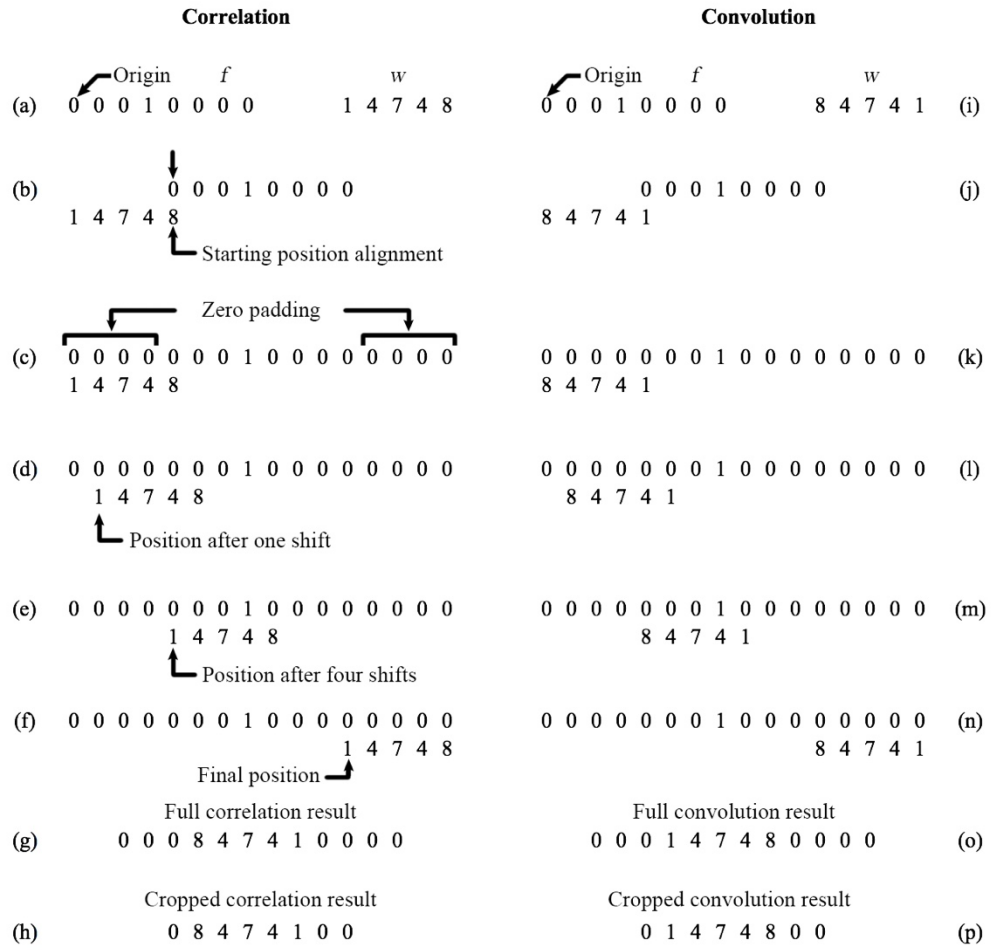


Figure 12. Illustration of 1-D correlation and convolution. (Gonzalez & E., 2008, p. 169)

Filtering in 2-D is the same as filtering in 1-D, which is illustrated in Figure 12. In 2-D, a 2-D filter mask is used to filter a 2-D image. The mask is moved over on the image and filtering computation is done for each pixel in the image.

2.1.4 Labeling

A pixel p whose coordinates are (x, y) has four *horizontal* and *vertical* neighbors. These neighbors are

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

The set formed by these pixels are called the *4-neighbors* of p , and is denoted by $N_4(p)$. Adding the *four diagonal* neighbors of p , which are

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

denoted by $N_D(p)$, to $N_4(p)$, creates 8-neighbors of p , denoted as $N_8(p)$. Now, let V be the set containing the values used to define adjacency. In a binary image, $V = \{1\}$. In a gray-scale image, V contains all 256 values from 0 to 255. There are three types of adjacency:

- a) *4-adjacency*. Pixels p and q whose values are in set V are 4-adjacent if q is in the set $N_4(p)$.
- b) *8-adjacency*. Pixels p and q whose values are in set V are 8-adjacent if q is in the set $N_8(p)$.
- c) *m-adjacency* (mixed adjacency). Pixels p and q whose values are in set V are m-adjacent if
 - i. q is in $N_4(p)$, or
 - ii. q is in $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ does not have any pixels whose values are from V .

Pixels p and q that are in set S , a subset of pixels in an image, are *connected* in S if there is a path between these pixels which is formed entirely by the pixels in S . The set of pixels which are connected to a pixel p in S is called a *connected component* of S . If there is only one connected component in S , S is called a *connected set* or *region*.

Extraction of connected components, or *labeling*, is accomplished by the following iterative procedure:

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1, 2, 3, \dots$$

where B is a structuring element. The iteration is finalized when $X_k = X_{k-1}$. When finding regions, it is important to define the adjacency of pixels, since an 8-connected region might be composed of several 4-connected regions. Therefore, a suitable B needs to be provided.

An example labeling operation is shown in Figure 13. The structuring element B is used to find the 8-connected component in the image. Note that, with the same initialization, the left-most pixel would not be included if structuring element B is chosen to be 4-connected.

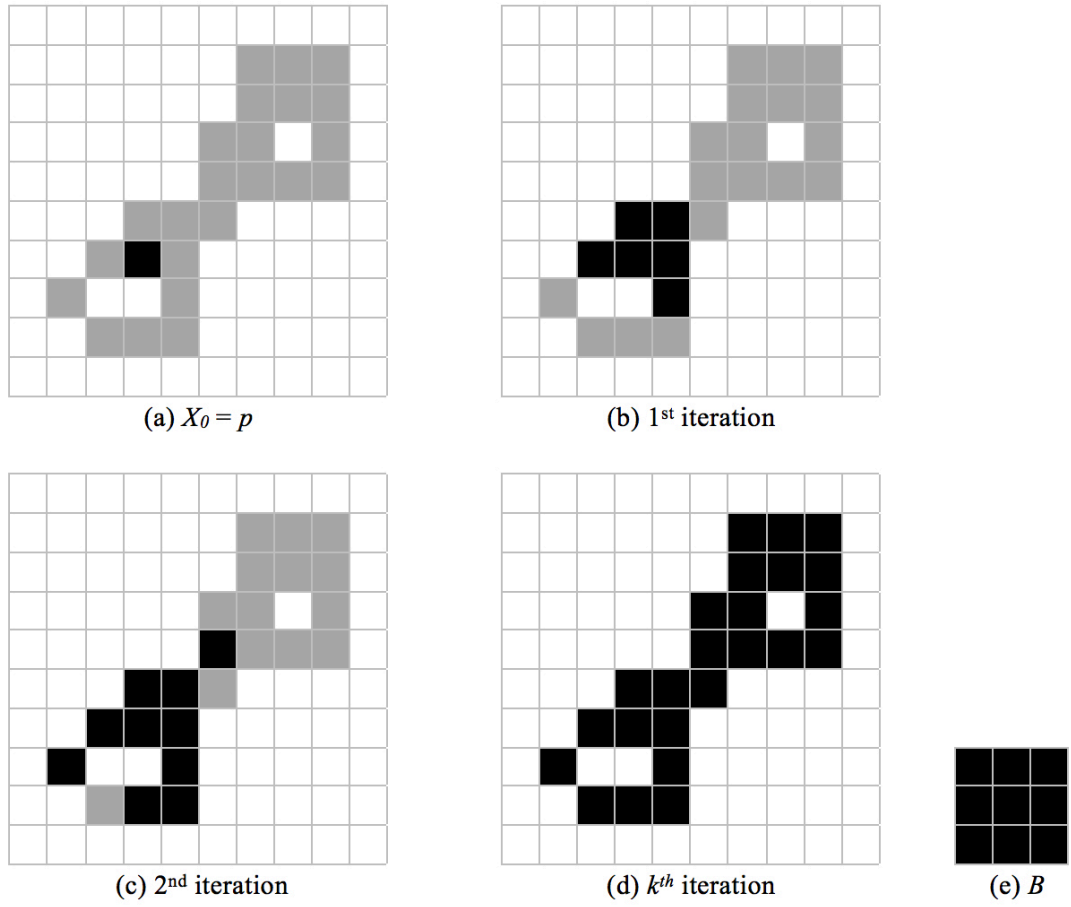


Figure 13. An example labeling operation

2.1.5 Convex Hull

A *convex set* is a set such that, the straight line connecting any two points lies entirely within that set. The smallest convex set containing an arbitrary set S is the *convex hull* H of set S .

A convex set of an arbitrary set can be found by morphological operations.

Let the structuring elements in Figure 14(a) be represented with B^i , where $i = 1, 2, 3, 4$. Finding the convex hull implements the following equation:

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1, 2, 3, 4 \quad \text{and} \quad k = 1, 2, 3, \dots$$

with $X_0^i = A$. The iterative process is stopped when $X_k^i = X_{k-1}^i$, D^i is set as X_k^i . Then, the convex hull of set A , denoted as $C(A)$, is:

$$C(A) = \bigcup_{i=1}^4 D^i$$

This procedure is based on applying the hit-or-miss transform to A with the four structuring elements B_1 , B_2 , B_3 and B_4 one by one. The results of the hit-or-miss transforms are then merged to create the convex hull.

An example process is shown in Figure 14. At the top of the figure, there are four 3x3 structuring elements defined. The origin of each structuring element is at the center. The positions marked with an “x” indicate that values of those pixels do not matter when applying the hit-or-miss transform. Figure 14(b) shows the set whose convex hull is sought. Figure 14(c) shows the result of the application of the above equation by using the structuring element B_1 . X_4^1 indicates that the convergence is achieved at 4th iteration. Similarly, Figure 14(d), Figure 14(e) and Figure 14(f) shows the results D^2 , D^3 and D^4 . Figure 14(g) illustrates the union of D^1 , D^2 , D^3 and D^4 , which is the convex hull of set A , $C(A)$. Finally, in Figure 14(h), the contributions of each structuring element to the convex hull is demonstrated.

As illustrated in Figure 14, with this procedure, the convex hull may grow beyond the minimum dimensions required to guarantee convexity. Although it adds computational complexity to the solution, this can be fixed by limiting the convex hull with the horizontal and vertical limits of the set.

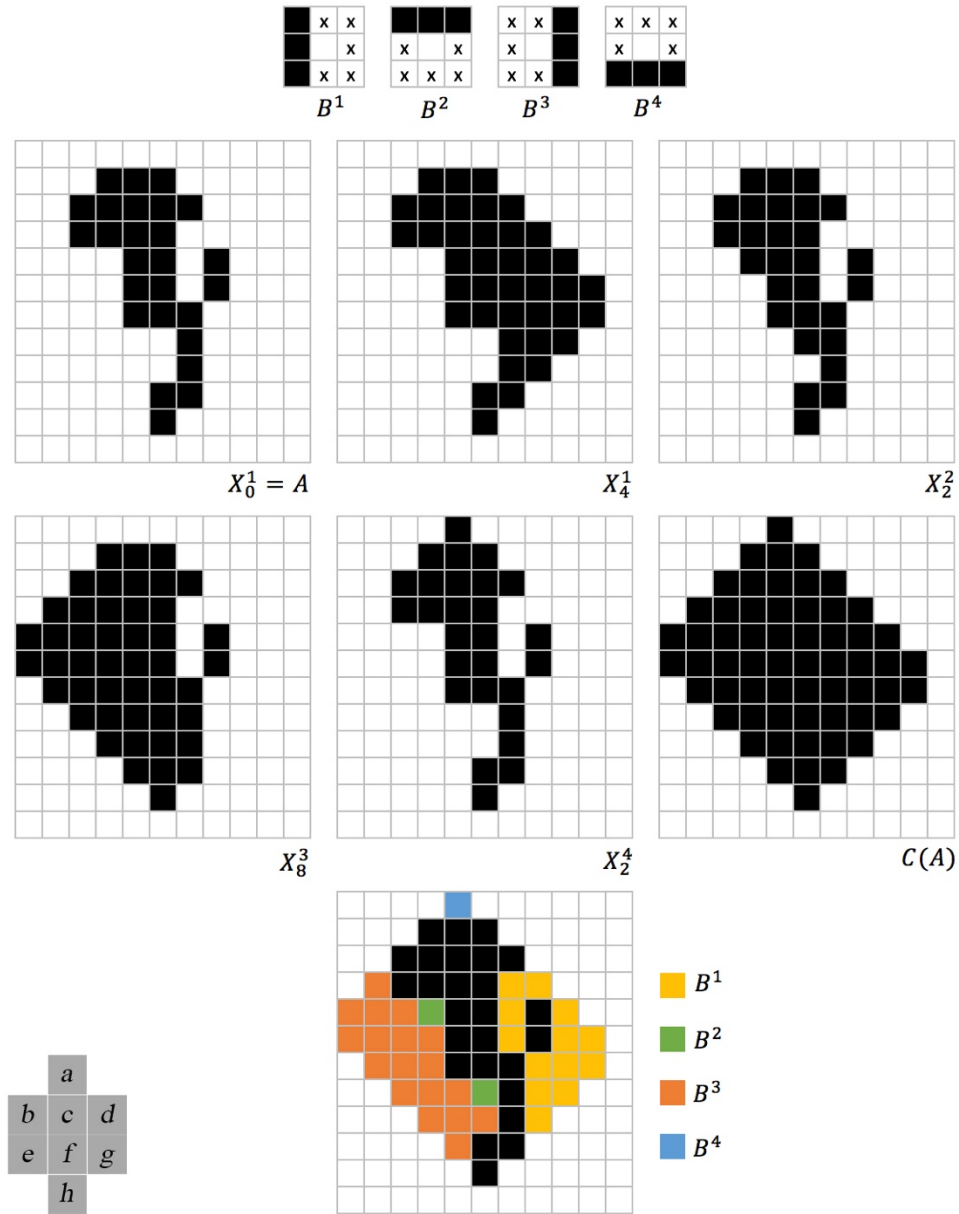


Figure 14. Process for finding the convex hull for an example region

2.2 Algorithmic Background

In this section, algorithms used in this study are reviewed.

2.2.1 Boundary Tracing

The boundary tracing algorithm used in this study is based on *Moore neighborhood* of a pixel, which consists of 8 neighboring pixels of a pixel. Given that the boundary is composed of pixels labeled as 1 and the background pixels are 0, the following steps apply:

1. Let starting pixel b_0 the upper-most and left-most pixel labeled as 1, and c_0 be the left neighboring pixel of b_0 . Let $i = 0$, $b = b_i$, and $c = c_i$.
2. Increase i by 1. Examine 8-neighbors of b , starting at c and going clockwise. Let b_i be the pixel with value 1 that is hit first and c_i be the preceding 0. Let $b = b_i$, and $c = c_i$.
3. Repeat process of updating b and c .
4. Continue until $b = b_0$ and the next boundary point found as b_1 .
5. The boundary is sequence of found bs .

An example boundary tracing operation is shown in Figure 15.

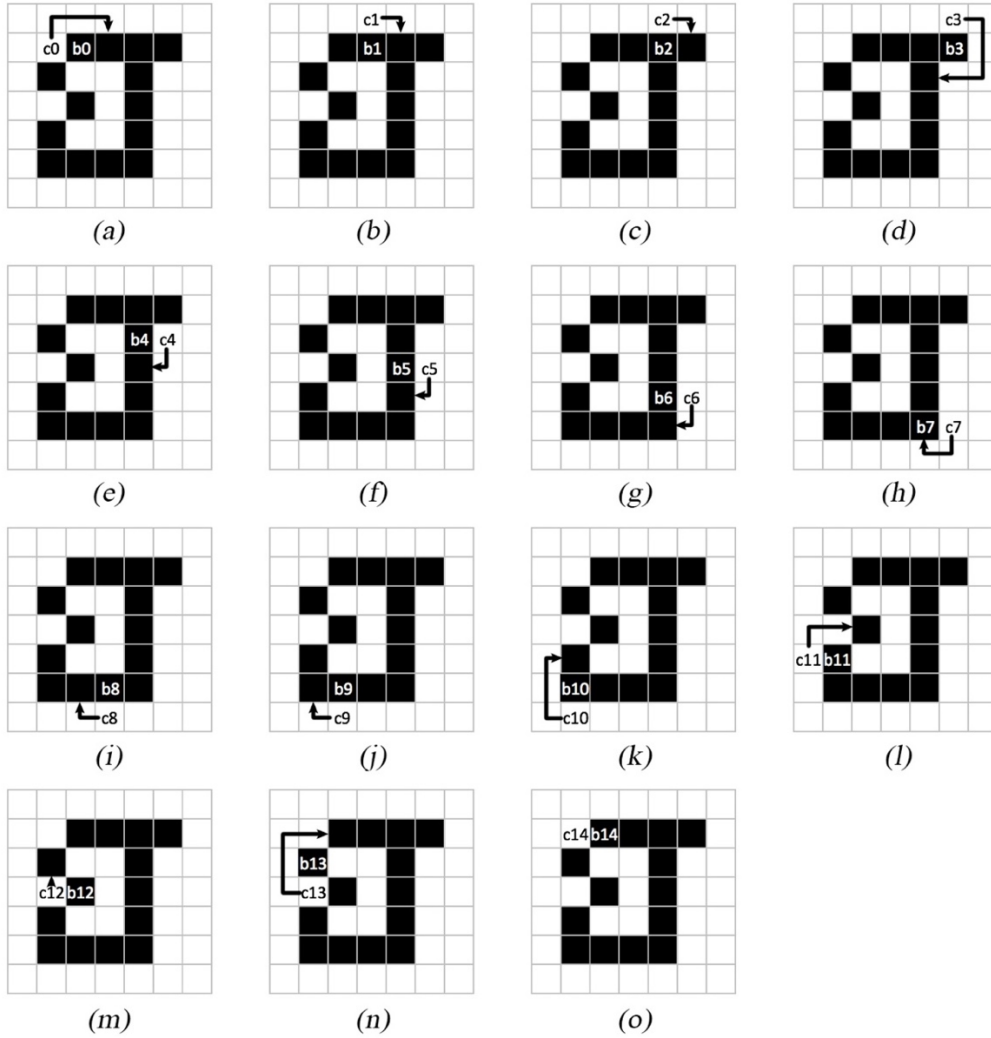


Figure 15. An example boundary tracing operation. The operation starts from (a) and ends with (o)

The boundary is sequence of found bs , which is $\{b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, b12, b13\}$. $b14$ is not included in the boundary because it corresponds to $b1$, which is the stopping case of the algorithm.

2.2.2 Canny Edge Detector

Canny edge detector (Canny, 1986) is an edge detector which detects edges using the following set of procedures:

1. Apply a Gaussian filter to the image for smoothing.
2. Compute the gradient angle and magnitude images.
3. Apply nonmaxima suppression to the gradient magnitude image.
4. Apply hysteresis thresholding.

Let Gaussian function, denoted as $G(x, y)$, be:

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

For an image $f(x, y)$, smoothed image $f_s(x, y)$ is formed by convolution of G and f :

$$f_s(x, y) = G(x, y) \star f(x, y)$$

Next, the gradient magnitude, $M(x, y)$, and angle (direction), $\sigma(x, y)$ are:

$$M(x, y) = \sqrt{g_x^2 + g_y^2}$$

$$\sigma(x, y) = \tan^{-1} \left[\frac{g_y}{g_x} \right]$$

where $g_x = \frac{\partial f_s}{\partial x}$ and $g_y = \frac{\partial f_s}{\partial y}$. g_x and g_y can be obtained by using any edge detectors such as Roberts, Prewitt or Sobel.

The magnitude image $M(x, y)$, because it is generated using the gradient, generally contains wide ridges around local maxima. As a next step, Canny edge detector thins those ridges by a procedure called *nonmaxima suppression*. One of the ways to do this is to specify a number of orientations of the gradient vector (edge normal). For example, for a 3×3 region, four different orientations which pass through the center

point of the region can be defined. These are horizontal, vertical, $+45^\circ$ and -45° edges. Note that each edge has two possible orientations, the second one being the 180° -rotated version of the first one. For example, edge normals of two possible orientations of a horizontal edge in a 3×3 region can be observed in Figure 16.

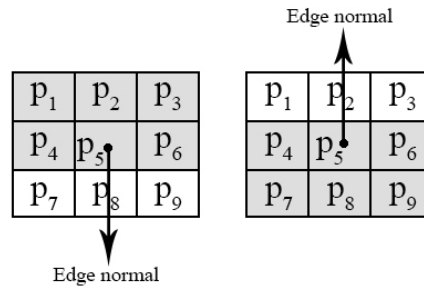


Figure 16. Two possible horizontal edge orientations in a 3×3 region

Since the edges are classified into four different orientations, edge normal ranges for which an edge is considered to be horizontal, vertical, $+45^\circ$ or -45° need to be defined as well. The ranges for the four orientations in a 3×3 region can be observed in Figure 17. Ranges indicating the same orientation are represented with the same color.

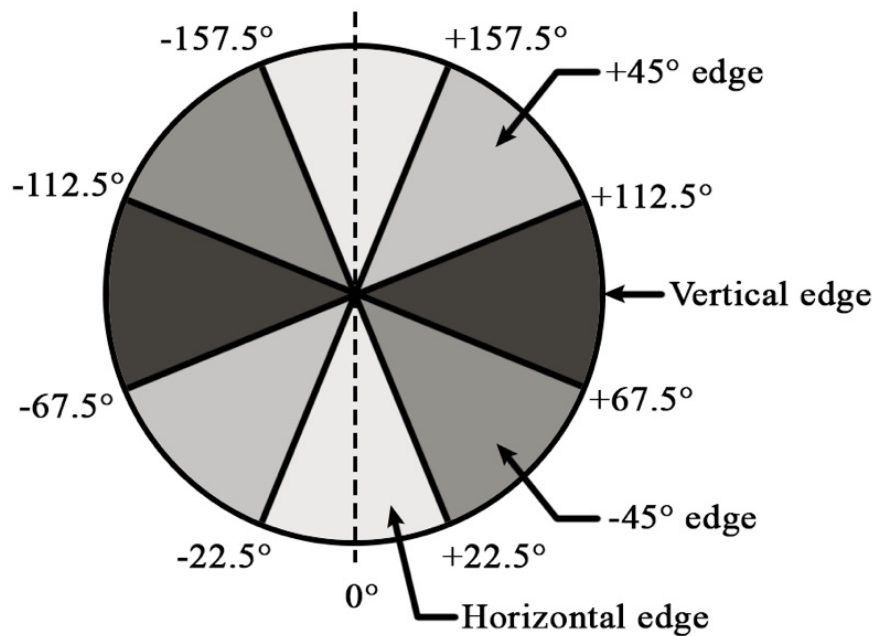


Figure 17. The angle ranges of edge normals in a 3×3 region for the four orientations

Let the four basic edge directions be denoted as d_1 , d_2 , d_3 and d_4 , for the 3×3 region, which are horizontal, vertical, $+45^\circ$ and -45° . Nonmaxima suppression procedure which is applied at every point (x, y) in $\sigma(x, y)$ is as follows:

1. Find the direction d_k which is closest to the angle $\sigma(x, y)$
2. Along d_k , if the magnitude $M(x, y)$ is less than at least one of its two neighbors, then suppress the value by letting $g_N(x, y) = 0$. Otherwise, let $g_N(x, y) = M(x, y)$.

where $g_N(x, y)$ is the image that is nonmaxima suppressed.

Finally, after nonmaxima suppression, hysteresis thresholding is applied to nonmaxima-suppressed image $g_N(x, y)$. The aim is to reduce false edge points while maintaining strong edges. The operation is done by thresholding the image with two different thresholds, called high and low threshold, denoted as T_H and T_L respectively. Canny (1986) suggested that the ratio of the high to low threshold should be two or three to one. Although the hysteresis thresholding can be applied on the same image, for the sake of simplicity, two additional images can be used to clearly visualize the procedure:

$$g_{NH}(x, y) = g(x, y) \geq T_H$$

and

$$g_{NL}(x, y) = g(x, y) \geq T_L$$

where $g_{NH}(x, y)$ and $g_{NL}(x, y)$ are set to 0 initially. After thresholding, $g_{NL}(x, y)$ will have more non-zero pixels compared to $g_{NH}(x, y)$ in general. However, all of the non-zero pixels in $g_{NH}(x, y)$ will be contained in $g_{NL}(x, y)$, which is created with a lower threshold. The non-zero pixels coming from $g_{NH}(x, y)$ are eliminated from $g_{NL}(x, y)$ by letting

$$g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$$

From now on, $g_{NL}(x, y)$ can be considered as it contains only “weak” edge pixels, while $g_{NH}(x, y)$ can be considered as it contains only “strong” edge pixels.

After thresholding, because $g_{NH}(x, y)$ contains the strong edge pixels, they are marked immediately. The edges in $g_{NL}(x, y)$ generally have gaps, depending on the threshold T_H . To create longer edges, following procedure is used:

1. Locate the next unvisited pixel, p , in $g_{NH}(x, y)$
2. Mark all the weak pixels in $g_{NL}(x, y)$ that are connected to p as valid edge pixels using, e.g., 8-connectivity
3. If there are unvisited pixels in $g_{NH}(x, y)$, return to step 1. Otherwise, continue with step 4.
4. Set all pixels that were not marked as valid edge pixels in $g_{NL}(x, y)$ to zero.

Finally, the output of the algorithm is created by appending $g_{NH}(x, y)$ to the all non-zero pixels in $g_{NL}(x, y)$. At this point, there might remain edges that are wider than 1 px. To make all edges 1-px wide, edge-thinning algorithm is typically applied to the final output.

2.2.3 Dijkstra's Algorithm

In this thesis study, Dijkstra's algorithm is used to find the shortest paths in contour connection graphs, which are graphs created by using detected edges. Contour connection graphs are explained in section 3.2.3.

Dijkstra's algorithm (Dijkstra, 1959) finds the shortest path starting from a predefined vertex on a weighted, directed graph $G = (V, E)$, where weight of each edge $w(u, v) \geq 0$ for each edge $(u, v) \in E$.

A set S is used to store vertices whose final shortest-path weights have already been determined. The shortest-paths are calculated from the predefined source vertex s . The algorithm initializes S as an empty set. At the same time, a queue Q is used to store vertices whose final shortest-path weights have not already been determined. In the beginning, $Q = V$. The final shortest-path weight of s is selected as 0, and others as infinity.

The algorithm keeps extracting the vertex with the smallest shortest-path estimate, u , from Q and adding it to S , which results in $Q = V - S$. Then, for each vertex v adjacent

to s , it calculates the shortest-path by adding the estimated shortest-path weight of u to the weight of the smallest-weight edge that connects u and v . If one of the calculated weights is smaller than the current shortest-path weight of v , then the shortest-path weight of v is updated. This is repeated until $Q = \emptyset$ and, hence, $S = V$. Pseudocode of the algorithm is given below (Cormen et al, 2009, p. 658):

```

Dijkstra( $G, w, s$ )
1   Initialize-Single-Source( $G, s$ )
2    $S = \emptyset$ 
3    $Q = G.V$ 
4   while  $Q \neq \emptyset$ 
5        $u = \text{Extract-Min}(Q)$ 
6        $S = S \cup \{u\}$ 
7       for each vertex  $v \in G.Adj[u]$ 
8           Relax( $u, v, w$ )

```

Line 1 initializes the the weight of s as 0. Line 2 sets S as an empty set, while line 3 sets Q equal to V . **while** loop of lines 4–8 is repeated until there is no vertex in Q . Line 5 extracts the vertex with the minimum shortest-path estimate and assigns it to u . For the first time through the **while** loop, $u = s$, since the weight of s is 0 while that of others is infinity. Line 6 adds vertex u to set S . Next, lines 7–8 relax each edge (u, v) leaving u , and, by this way, update the estimate shortest-path weight of v .

Figure 18 shows the execution of Dijkstra’s algorithm over an example. In the figure, the estimates of the shortest paths are shown within vertices. Shaded edges indicate predecessor values. The vertices in the set S are shown with white text over black background. The vertices in the queue Q are shown with black text over white background. In the figure, (a) shows the initial situation where $u = s$, (b)-(f) are the situations after each successive repetition of the **while** loop where the vertices with gray background are the vertices selected as u , (f) shows the final situation where Q is an empty set. (Cormen et al, 2009, p. 659)

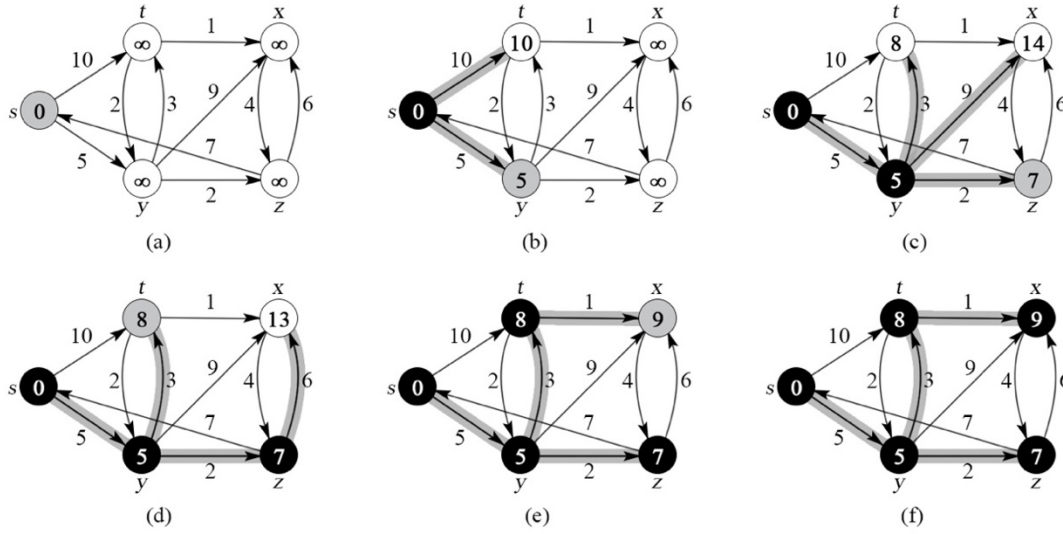


Figure 18. The execution of Dijkstra's algorithm

2.2.4 Hough Transform

Hough transform can be used to detect several different features such as lines and circles. In this section, only straight line detection is explained.

Hough line transform detects straight lines in an edge image by expressing lines in polar coordinates. Figure 19 shows a line in x-y plane, whose normal makes an angle θ with the horizontal axis and its distance to the origin is ρ . Then, this line can be expressed as a parametric equation:

$$\rho = x \cos(\theta) + y \sin(\theta)$$

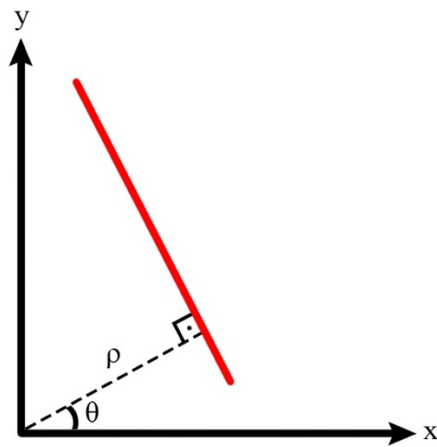


Figure 19. A line in X-Y plane

In an edge image, position (x, y) of every pixel is known. Therefore, x and y are constants in the parametric equation, while ρ and θ are unknowns. For each point in the edge image, Hough transform finds all lines that pass through that point by finding (ρ, θ) pairs that satisfy the parametric equation for that point. In *Hough parameter space*, or $\rho - \theta$ space, lines passing through a point form a sinusoidal curve. If a line passes through two different points, then, the curves of these points in Hough parameter space intersect. In *Hough parameter space*, lines passing through a point form a sinusoidal curve. If a line passes through two different points, then, the curves of these points in Hough parameter space intersect.

Figure 20(b) shows Hough parameter space for the edge image in Figure 20(a). The Hough parameter space contains all (ρ, θ) pairs for each pixel in the edge image. For each pixel, a sinusoidal curve is created in the parameter space. Intersection points of the curves are ρ and θ values for the lines found in the image. The points where the most number of curves intersect indicate a strong straight line. These points are marked with green frames in Figure 20(b). As a result, three strong lines are found in the edge image with ρ and θ values of approximately $(-4, -45)$, $(15, 0)$ and $(3, 90)$, which correspond to the three lines in Figure 20(a).

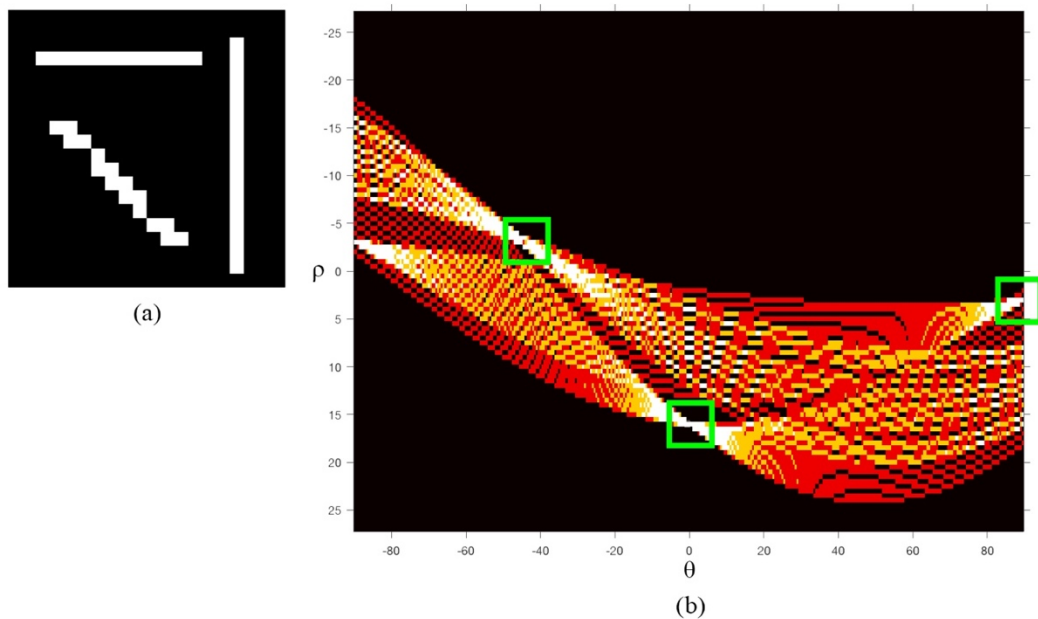


Figure 20. Hough parameter space for a sample image

Hough transform uses the fact that, for every point (x, y) on a line, the parametric equation is satisfied with the same (ρ, θ) pair. In a different point of view, each point

(x, y) in the edge image votes for several (ρ, θ) pairs. As a result, (ρ, θ) pairs with the highest number of votes indicate a straight line, since a line consists of several points.

2.2.5 Image Differencing

Image differencing is used to find differences or changes between two images. Let $I_1(x, y)$ and $I_2(x, y)$ be the two gray-scale images. Then, the difference image, denoted as $I_d(x, y)$, is defined as

$$I_d(x, y) = I_1(x, y) - I_2(x, y)$$

In this research study, image differencing is utilized to detect shadows. According to İlsever and Ünsalan (2012): “Intensity differences due to land cover change resides at the tails of the difference distribution of the image. Assuming that changes due to land cover are less than changes by other factors, we expect that most of the difference is distributed around the mean.”. This approach is adopted for the shadow detection as well. For a zero mean difference, I_2 can be normalized as

$$\tilde{I}_2(x, y) = \frac{\sigma_1}{\sigma_2} (I_2(x, y) - \mu_2) + \mu_1$$

where \tilde{I}_2 is the normalized form of I_2 . μ_1, σ_1 and μ_2, σ_2 are the mean and the standard deviation of I_1 and I_2 , respectively. After normalization, the mean and standard deviation of the two images are equalized. Hence, the difference image will have zero mean (İlsever & Ünsalan, 2012). Now, the difference equation can be updated as

$$I_d(x, y) = I_1(x, y) - \tilde{I}_2(x, y)$$

The values that are less than zero represent the pixels that are brighter in the shadow image. It is expected that the values representing the shadows in the difference image will be greater than a certain threshold and greater than zero. Therefore, shadow mask is

$$S(x, y) = \begin{cases} 1, & I_d(x, y) \geq \tau \\ 0, & \text{otherwise} \end{cases}$$

where the threshold τ is determined empirically.

2.2.6 RANSAC

RANSAC, which is an abbreviation of *Random Sample Consensus*, is a general parameter estimation approach which is designed to deal with large number of outliers in the input data. While conventional sampling techniques use as much data points as possible, RANSAC generates candidate solutions by using minimum number of data points.

The algorithm uses the smallest set possible and then proceeds with enlarging the set by using consistent data points. The algorithm works as follows:

1. Randomly select the minimum number of data points to determine the parameters of the model.
2. Solve for parameters of the model.
3. From the set of all data points, determine how many of them fit with a predefined tolerance ϵ .
4. If the ratio of the number of inliers to the total number of data points in the set exceeds a predefined threshold τ , make another estimation for the model parameters using all the identified inliers and terminate. Otherwise, proceed with step 5.
5. Repeat steps 1 through 4 until predefined iteration limit N is reached.

The number of iterations, N , is

$$N = \frac{\log(1 - P)}{\log(1 - (1 - \epsilon)^k)}$$

where ϵ is the independent probability of any point being an outlier, k is the number of points inside one random sampling containing no outliers, and P is the probability that at least one of the sets of random samples does not include an outlier.

2.3 Mining Background

This section covers the mining background related to this thesis study. In this section, geotechnical core logging and RQD are reviewed.

2.3.1 Geotechnical Core Logging

Geotechnical core logging is performed in order to determine several parameters about the rock mass. In a standard geotechnical core logging process, information about structural region, RQD, condition of discontinuities, Q and RMR parameters, and intact rock strength parameters are recorded. In addition, comments are added for the examined cores. The format of the log should be structured to contain all required data that are required for rock mass classification schemes, such as RMR (Bieniawski, 1976), GSI (Marinos & Hoek, 2000) and Q (Barton et al., 1974). A sample geotechnical core logging sheet can be observed in Table 2.

Structural region parameters

Structural region parameters consist of *rock type*, *depth*, and *total core recovery* (TCR).

Rock type is the name of the rock that is logged. The defined rock type should be consistent throughout the project so that it does not cause confusion with identification of geological units.

Depth information is recorded as an interval. *From* is the length from the collar of the hole to the beginning of the logged interval, while *to* is the length from the collar of the hole to the end of the logged interval. For example, an interval from 32.6 meters to 54.8 meters would be recorded as “From: 32.6 m ... To: 54.8 m”. For consistency, the measurements should be made over uniform intervals of drill core, such as core runs.

Total core recovery, or *TCR*, represented as percentage, is the total amount of core recovered over the measured length drilled for each core run. Because core losses are encountered in highly fractured or weak zones, which may be crucial for determining rock mass properties, they are important indicators of potentially poor geotechnical conditions. When determining TCR, rubble, redrill, or slough recovered at the top of a core lift which was not in place is not considered as recovered core. In addition, cores that slip through the core lifter and drop out of its core tube should be included in TCR. This only indicates a worn or unsuitable core lifter. TCR must not exceed 100% on any logged interval.

Table 2. Sample geotechnical core logging sheet

[illegible]

Condition of discontinuities

Condition of discontinuities are composed of *fracture type*, *fracture frequency*, *dip with respect to core axis*, *fracture condition*, and *weathering index*.

Fracture type is the type of the fractures present in the core interval. Table 3 shows the types of the fractures and their abbreviations.

Table 3. Fracture types and their abbreviations

Fracture (FR):	A discontinuity of uncertain origin (e.g., possible mechanical break)
Joint (J):	A discontinuity with no infilling and no evidence of previous movement
Fault (F):	A discontinuity across which there has been substantial movement. Infill will be relatively thick.
Shear (S):	A discontinuity across which there has been limited movement. Infill will be relatively thin, generally associated with polished or slicken-sided surfaces.
Bedding (B):	A discontinuity associated with sedimentary processes (e.g. mud seam in sandstone)
Foliation (FO):	A preferential direction of structural weakness in the rock due to alignment of weak minerals, caused by metamorphism.
Vein (V):	A discontinuity infilled or healed by another mineral (e.g. quartz). Veins are generally of limited interest unless material is particularly weak or broken.

Fracture frequency is the number of natural fractures found over the examined core interval. It can also be recorded for each fracture type. Drilling-induced fractures and fractures whose length of persistence is less than the core diameter are not included in the count. The fracture frequency is determined by dividing the number of fractures to the length of the examined interval. It is used to determine the spacing between

fractures within a rock mass, which is called *fracture spacing*. Strength and behavior of the rock mass is directly influenced by the fracture spacing. This parameter is important for mine design studies, because it is harder to maintain the stability of the excavations when the fracture spacing is low. Moreover, since RQD is not sensitive enough for fracture spacing values greater than 10 cm, this parameter can be used as a complement to RQD.

Dip with respect to core axis is the dip angle or range of dip angles for the fractures found in the examined interval. The angle is measured relative to the core axis, where dipping perpendicular to core axis is recorded as 90° and dipping parallel to the core axis is recorded as 0°. If there are more than one joint set, average dip angle for each joint set is recorded.

Fracture condition is estimated using the definitions in RMR (Bieniawski, 1976) or Q (Barton et al., 1974) rock mass classification systems. The values can be obtained from Table 4 and Table 5.

Table 4. Rock mass rating system (Bieniawski, 1989)

A. CLASSIFICATION PARAMETERS AND THEIR RATINGS									
Parameter			Range of values						
1	Strength of intact rock material	Point-load strength index	>10 MPa	4 - 10 MPa	2 - 4 MPa	1 - 2 MPa	For this low range - uniaxial compressive test is preferred		
		Uniaxial comp. strength	>250 MPa	100 - 250 MPa	50 - 100 MPa	25 - 50 MPa	5 - 25 MPa	1 - 5 MPa	< 1 MPa
	Rating		15	12	7	4	2	1	0
2	Drill core Quality RQD		90% - 100%	75% - 90%	50% - 75%	25% - 50%	< 25%		
	Rating		20	17	13	8	3		
3	Spacing of		> 2 m	0.6 - 2 . m	200 - 600 mm	60 - 200 mm	< 60 mm		
	Rating		20	15	10	8	5		
4	Condition of discontinuities (See E)		Very rough surfaces Not continuous No separation Unweathered wall rock	Slightly rough surfaces Separation < 1 mm Slightly weathered walls	Slightly rough surfaces Separation < 1 mm Highly weathered walls	Slickensided surfaces or Gouge < 5 mm thick or Separation 1-5 mm Continuous	Soft gouge >5 mm thick or Separation > 5 mm Continuous		
	Rating		30	25	20	10	0		
5	Groundwater	Inflow per 10 m tunnel length (l/m)	None	< 10	10 - 25	25 - 125	> 125		
		(Joint water press)/ (Major principal σ)	0	< 0.1	0.1, - 0.2	0.2 - 0.5	> 0.5		
		General conditions	Completely dry	Damp	Wet	Dripping	Flowing		
		Rating	15	10	7	4	0		
B. RATING ADJUSTMENT FOR DISCONTINUITY ORIENTATIONS (See F)									
Strike and dip orientations			Very favourable	Favourable	Fair	Unfavourable	Very Unfavourable		
Ratings	Tunnels & mines		0	-2	-5	-10	-12		
	Foundations		0	-2	-7	-15	-25		
	Slopes		0	-5	-25	-50			
C. ROCK MASS CLASSES DETERMINED FROM TOTAL RATINGS									
Rating			100 ← 81	80 ← 61	60 ← 41	40 ← 21	< 21		
Class number			I	II	III	IV	V		
Description			Very good rock	Good rock	Fair rock	Poor rock	Very poor rock		
D. MEANING OF ROCK CLASSES									
Class number			I	II	III	IV	V		
Average stand-up time			20 yrs for 15 m span	1 year for 10 m span	1 week for 5 m span	10 hrs for 2.5 m span	30 min for 1 m span		
Cohesion of rock mass (kPa)			> 400	300 - 400	200 - 300	100 - 200	< 100		
Friction angle of rock mass (deg)			> 45	35 - 45	25 - 35	15 - 25	< 15		
E. GUIDELINES FOR CLASSIFICATION OF DISCONTINUITY conditions									
Discontinuity length (persistence)			< 1 m	1 - 3 m	3 - 10 m	10 - 20 m	> 20 m		
Rating			6	4	2	1	0		
Separation (aperture)			None	< 0.1 mm	0.1 - 1.0 mm	1 - 5 mm	> 5 mm		
Rating			6	5	4	1	0		
Roughness			Very rough	Rough	Slightly rough	Smooth	Slickensided		
Rating			6	5	3	1	0		
Infilling (gouge)			None	Hard filling < 5 mm	Hard filling > 5 mm	Soft filling < 5 mm	Soft filling > 5 mm		
Rating			6	4	2	2	0		
Weathering			Unweathered	Slightly weathered	Moderately weathered	Highly weathered	Decomposed		
Ratings			6	5	3	1	0		
F. EFFECT OF DISCONTINUITY STRIKE AND DIP ORIENTATION IN TUNNELLING**									
Strike perpendicular to tunnel axis					Strike parallel to tunnel axis				
Drive with dip - Dip 45 - 90°			Drive with dip - Dip 20 - 45°		Dip 45 - 90°			Dip 20 - 45°	
Very favourable			Favourable		Very unfavourable			Fair	
Drive against dip - Dip 45-90°			Drive against dip - Dip 20-45°		Dip 0-20 - Irrespective of strike°				
Fair			Unfavourable		Fair				

Table 5. Classification of individual parameters used in the Tunnelling Quality Index Q (Barton et al., 1974)

DESCRIPTION	VALUE	NOTES
1. ROCK QUALITY DESIGNATION	RQD	
A. Very poor	0 - 25	1. Where RQD is reported or measured as ≤ 10 (including 0), a nominal value of 10 is used to evaluate Q .
B. Poor	25 - 50	
C. Fair	50 - 75	
D. Good	75 - 90	2. RQD intervals of 5, i.e. 100, 95, 90 etc. are sufficiently accurate.
E. Excellent	90 - 100	
2. JOINT SET NUMBER	J_n	
A. Massive, no or few joints	0.5 - 1.0	
B. One joint set	2	
C. One joint set plus random	3	
D. Two joint sets	4	
E. Two joint sets plus random	6	
F. Three joint sets	9	1. For intersections use $(3.0 \times J_n)$
G. Three joint sets plus random	12	
H. Four or more joint sets, random, heavily jointed, 'sugar cube', etc.	15	2. For portals use $(2.0 \times J_n)$
J. Crushed rock, earthlike	20	
3. JOINT ROUGHNESS NUMBER	J_r	
a. Rock wall contact		
b. Rock wall contact before 10 cm shear		
A. Discontinuous joints	4	
B. Rough and irregular, undulating	3	
C. Smooth undulating	2	
D. Slickensided undulating	1.5	1. Add 1.0 if the mean spacing of the relevant joint set is greater than 3 m.
E. Rough or irregular, planar	1.5	
F. Smooth, planar	1.0	
G. Slickensided, planar	0.5	2. $J_r = 0.5$ can be used for planar, slickensided joints having lineations, provided that the lineations are oriented for minimum strength.
c. No rock wall contact when sheared		
H. Zones containing clay minerals thick enough to prevent rock wall contact	1.0 (nominal)	
J. Sandy, gravelly or crushed zone thick enough to prevent rock wall contact	1.0 (nominal)	
4. JOINT ALTERATION NUMBER	J_a	ϕ_r degrees (approx.)
a. Rock wall contact		
A. Tightly healed, hard, non-softening, impermeable filling	0.75	1. Values of ϕ_r , the residual friction angle, are intended as an approximate guide to the mineralogical properties of the alteration products, if present.
B. Unaltered joint walls, surface staining only	1.0	
C. Slightly altered joint walls, non-softening mineral coatings, sandy particles, clay-free disintegrated rock, etc.	2.0	
D. Silty-, or sandy-clay coatings, small clay-fraction (non-softening)	3.0	
E. Softening or low-friction clay mineral coatings, i.e. kaolinite, mica. Also chlorite, talc, gypsum and graphite etc., and small quantities of swelling clays. (Discontinuous coatings, 1 - 2 mm or less)	4.0	

Table 5. (cont'd) Classification of individual parameters used in the Tunnelling
Quality Index Q (Barton et al., 1974)

4. JOINT ALTERATION NUMBER	J_a	ϕ degrees (approx.)	
b. Rock wall contact before 10 cm shear			
F. Sandy particles, clay-free, disintegrating rock etc.	4.0	25 - 30	
G. Strongly over-consolidated, non-softening clay mineral fillings (continuous < 5 mm thick)	6.0	16 - 24	
H. Medium or low over-consolidation, softening clay mineral fillings (continuous < 5 mm thick)	8.0	12 - 16	
J. Swelling clay fillings, i.e. montmorillonite, (continuous < 5 mm thick). Values of J_a depend on percent of swelling clay-size particles, and access to water.	8.0 - 12.0	6 - 12	
c. No rock wall contact when sheared			
K. Zones or bands of disintegrated or crushed	6.0		
L. rock and clay (see G, H and J for clay	8.0		
M. conditions)	8.0 - 12.0	6 - 24	
N. Zones or bands of silty- or sandy-clay, small clay fraction, non-softening	5.0		
O. Thick continuous zones or bands of clay	10.0 - 13.0		
P. & R. (see G.H and J for clay conditions)	6.0 - 24.0		
5. JOINT WATER REDUCTION	J_w	approx. water pressure (kgf/cm ²)	
A. Dry excavation or minor inflow i.e. < 5 l/m locally	1.0	< 1.0	
B. Medium inflow or pressure, occasional outwash of joint fillings	0.66	1.0 - 2.5	
C. Large inflow or high pressure in competent rock with unfilled joints	0.5	2.5 - 10.0	1. Factors C to F are crude estimates; increase J_w if drainage installed.
D. Large inflow or high pressure	0.33	2.5 - 10.0	
E. Exceptionally high inflow or pressure at blasting, decaying with time	0.2 - 0.1	> 10	2. Special problems caused by ice formation are not considered.
F. Exceptionally high inflow or pressure	0.1 - 0.05	> 10	
6. STRESS REDUCTION FACTOR		SRF	
a. Weakness zones intersecting excavation, which may cause loosening of rock mass when tunnel is excavated			
A. Multiple occurrences of weakness zones containing clay or chemically disintegrated rock, very loose surrounding rock any depth)	10.0		1. Reduce these values of SRF by 25 - 50% but only if the relevant shear zones influence do not intersect the excavation
B. Single weakness zones containing clay, or chemically disintegrated rock (excavation depth < 50 m)	5.0		
C. Single weakness zones containing clay, or chemically disintegrated rock (excavation depth > 50 m)	2.5		
D. Multiple shear zones in competent rock (clay free), loose surrounding rock (any depth)	7.5		
E. Single shear zone in competent rock (clay free). (depth of excavation < 50 m)	5.0		
F. Single shear zone in competent rock (clay free). (depth of excavation > 50 m)	2.5		
G. Loose open joints, heavily jointed or 'sugar cube', (any depth)	5.0		

Table 5. (cont'd) Classification of individual parameters used in the Tunnelling Quality Index Q (Barton et al., 1974)

DESCRIPTION	VALUE		NOTES
6. STRESS REDUCTION FACTOR			SRF
b. Competent rock, rock stress problems			
	σ_c/σ_1	$\alpha_t\sigma_1$	2. For strongly anisotropic virgin stress field
H. Low stress, near surface	> 200	> 13	(if measured): when $5 \leq \sigma_1/\sigma_3 \leq 10$, reduce σ_c
J. Medium stress	200 - 10	13 - 0.66	to $0.8\sigma_c$ and α_t to $0.8\alpha_t$. When $\sigma_1/\sigma_3 > 10$,
K. High stress, very tight structure (usually favourable to stability, may be unfavourable to wall stability)	10 - 5	0.66 - 0.33	reduce σ_c and α_t to $0.6\sigma_c$ and $0.6\alpha_t$, where σ_c = unconfined compressive strength, and α_t = tensile strength (point load) and σ_1 and σ_3 are the major and minor principal stresses.
L. Mild rockburst (massive rock)	5 - 2.5	0.33 - 0.16	5 - 10
M. Heavy rockburst (massive rock)	< 2.5	< 0.16	10 - 20
c. Squeezing rock, plastic flow of incompetent rock under influence of high rock pressure			3. Few case records available where depth of crown below surface is less than span width. Suggest SRF increase from 2.5 to 5 for such cases (see H).
N. Mild squeezing rock pressure			5 - 10
O. Heavy squeezing rock pressure			10 - 20
d. Swelling rock, chemical swelling activity depending on presence of water			
P. Mild swelling rock pressure			5 - 10
R. Heavy swelling rock pressure			10 - 15
ADDITIONAL NOTES ON THE USE OF THESE TABLES			
When making estimates of the rock mass Quality (Q), the following guidelines should be followed in addition to the notes listed in the tables:			
1. When borehole core is unavailable, RQD can be estimated from the number of joints per unit volume, in which the number of joints per metre for each joint set are added. A simple relationship can be used to convert this number to RQD for the case of clay free rock masses: $RQD = 115 - 3.3 J_v$ (approx.), where J_v = total number of joints per m^3 ($0 < RQD < 100$ for $35 > J_v > 4.5$).			
2. The parameter J_n representing the number of joint sets will often be affected by foliation, schistosity, slaty cleavage or bedding etc. If strongly developed, these parallel 'joints' should obviously be counted as a complete joint set. However, if there are few 'joints' visible, or if only occasional breaks in the core are due to these features, then it will be more appropriate to count them as 'random' joints when evaluating J_n .			
3. The parameters J_r and J_a (representing shear strength) should be relevant to the weakest significant joint set or clay filled discontinuity in the given zone. However, if the joint set or discontinuity with the minimum value of J_r/J_a is favourably oriented for stability, then a second, less favourably oriented joint set or discontinuity may sometimes be more significant, and its higher value of J_r/J_a should be used when evaluating Q. The value of J_r/J_a should in fact relate to the surface most likely to allow failure to initiate.			
4. When a rock mass contains clay, the factor SRF appropriate to loosening loads should be evaluated. In such cases the strength of the intact rock is of little interest. However, when jointing is minimal and clay is completely absent, the strength of the intact rock may become the weakest link, and the stability will then depend on the ratio rock-stress/rock-strength. A strongly anisotropic stress field is unfavourable for stability and is roughly accounted for as in note 2 in the table for stress reduction factor evaluation.			
5. The compressive and tensile strengths (σ_c and α_t) of the intact rock should be evaluated in the saturated condition if this is appropriate to the present and future in situ conditions. A very conservative estimate of the strength should be made for those rocks that deteriorate when exposed to moist or saturated conditions.			

Weathering index, or *alteration index*, is the degree of weathering. This parameter is decided using Table 6. By this way, the degree of weathering for the original rock material is defined qualitatively. The intensity of the alteration is recorded along with the weathering, as well. Because alteration of the rock material has a negative effect on the strength of the rock, logging the weathering is crucial.

Table 6. Weathering classification (Brown, 1981)

Term	Symbol	Description	Discoloration Extent	Fracture Condition	Surface Characteristics
Fresh	W1	No visible sign of rock material weatherings.	None	Closed or Discolored	Unchanged
Slightly Weathered	W2	Discoloration indicates weathering of rock material on discontinuity surfaces. Less than 5% of rock mass altered.	<20% of fracture spacing on both sides of fracture	Discolored, may contain thin filling	Partial discoloration
Moderately Weathered	W3	Less than 50% of the rock material is decomposed and/or disintegrated to a soil. Fresh or discolored rock is present either as a discontinuous framework or as corestones.	>20% of fracture spacing on both sides of fracture	Discolored, may contain thick filling	Partial to complete discoloration, not friable except poorly cemented rocks
Highly Weathered	W4	More than 50% of the rock material is decomposed and/or disintegrated to a soil. Fresh or discolored rock is present either as a discontinuous framework or as corestones.	Throughout	Filled with alteration minerals	Friable and possibly pitted
Completely Weathered	W5	100% of rock material is decomposed and/or disintegrated to soil. The original mass structure is still largely intact.	Throughout	Filled with alteration minerals	Resembles soil
Residual Soil	W6	All rock material is converted to soil. The mass structure and material fabric are destroyed. There is a large change in volume, but the soil has not been significantly transported.	Throughout	N/A	Resembles soil

Q Parameters

Q parameters are the parameters which are required as an input to *Q* (Barton et al., 1974) rock mass classification system. These parameters are *RQD*, *joint set number* (J_n), *joint roughness number* (J_r), *joint alteration number* (J_a), *joint water reduction*

(J_w), and *stress reduction factor (SRF)*. The values of these parameters can be found using Table 5.

RMR Parameters

RMR parameters are the input parameters required to find the value of RMR (Bieniawski, 1976) rock mass classification system. These parameters are *strength of intact rock material*, *RQD*, *spacing of discontinuities*, *condition of discontinuities*, *groundwater*, and *strike and dip orientations*. The values of these parameters can be found using Table 4.

Intact rock strength

The strength of the intact core pieces can be determined using Table 7, with the help of a pocket knife and a rock hammer, in a cost-effective way. In addition to this, point load tests should be carried out in the field on the projects for which intact rock strength is an important factor. When selecting samples to be tested, it is important to concentrate on representing rock mass rather than selecting the samples that are easiest to test. To provide a measure of anisotropy, both axial and diametric tests should be carried out.

Comments

The *comments* section in the log should be filled with a short description of principal geological characteristics of the interval that is examined. These characteristics include distinguishing features such as fabric, grain size, color, major minerals, descriptions of particularly weak zones and comments related to overall competency of the core.

Table 7. Intact rock strength classification (Brown, 1981)

GRADE	DESCRIPTION	FIELD IDENTIFICATION	APPROX. RANGE OF UNIAXIAL COMPRESSIVE STRENGTH (Mpa)
R0	Extremely weak rock	Indented by thumbnail.	0.25 - 1.0
R1	Very weak rock	Crumbles under firm blows with point of geological hammer, can be peeled by a pocket knife.	1.0 - 5.0
R2	Weak rock	Can be peeled by a pocket knife with difficulty, shallow indentations made by firm blow with point of geological hammer.	5.0 - 25
R3	Medium strong rock	Cannot be scraped or peeled with a pocket knife, specimen can be fractured with single firm blow of geological hammer.	25 - 50
R4	Strong rock	Specimen requires more than one blow of geological hammer to fracture it.	50 - 100
R5	Very strong rock	Specimen requires many blows of geological hammer to fracture it.	100 - 250
R6	Extremely strong rock	Specimen can only be chipped with geological hammer.	>250

2.3.2 RQD

Rock Quality Designation, RQD, which was first used on a design and construction project in 1964 (Deere & Deere, 1989), was developed as a rock quality index. The index is also included as an input parameter in rock classification systems developed by Bieniawski (1973) and Barton et al. (1974).

RQD is obtained by counting only sound core pieces that are at least 100 mm (4-in.) in length, and whose diameter is at least NQ-size. Because of the potential core

breakage and loss, inclusion of BQ and BX size cores in RQD is discouraged. Core pieces that are slightly and moderately weathered can be included in the calculation as long as they cannot be broken by hand. RQD is calculated by:

$$RQD = \frac{\sum(\text{Length of sound core pieces} > 100 \text{ mm})}{\text{Total core run length}}$$

Length of core pieces are measured through their centerline. Correct and incorrect length measurements are shown in Figure 21.

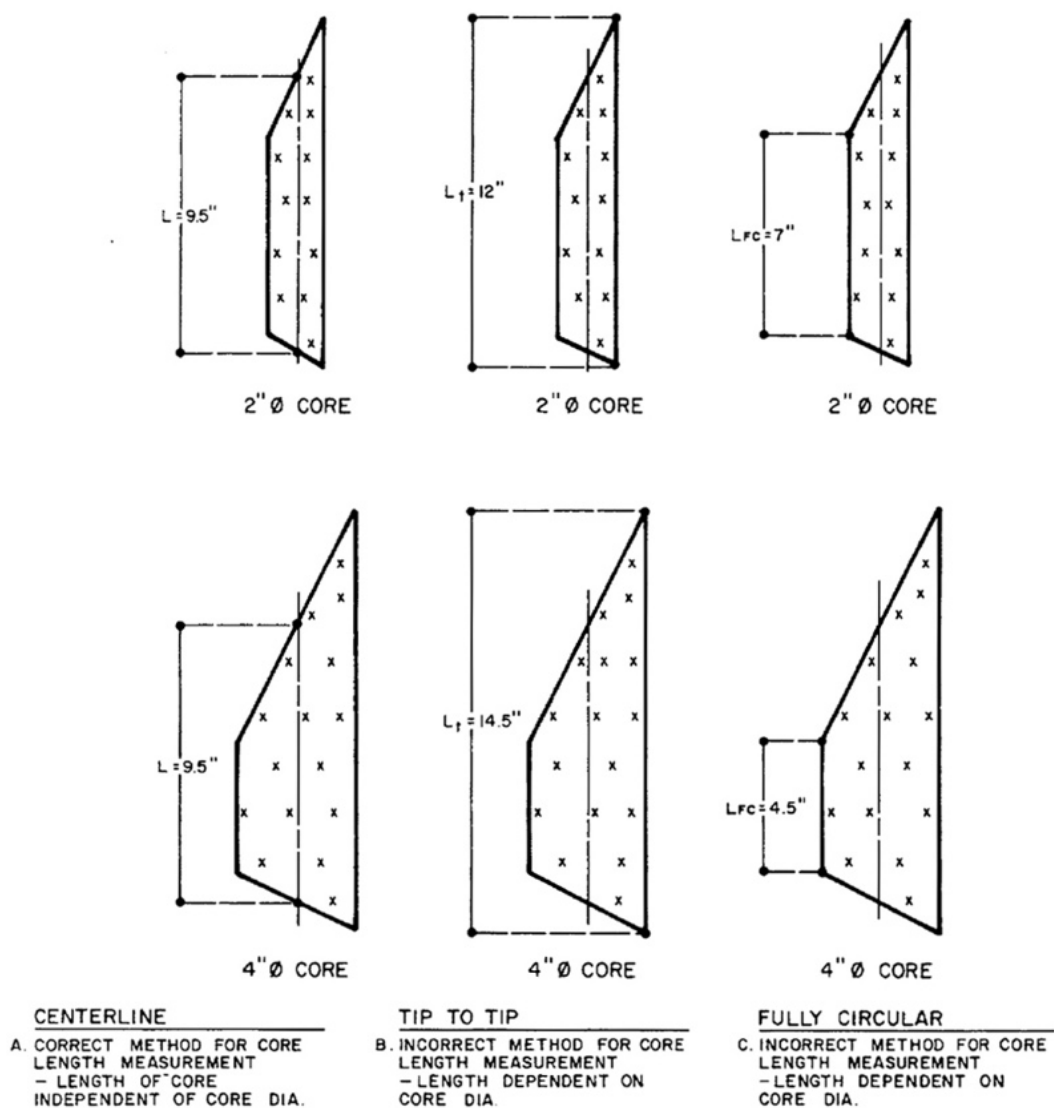


Figure 21. Correct and incorrect ways of measuring the length of a core sample
(Deere & Deere, 1989, p. 14)

Measuring the length along the centerline is the recommended procedure, which is advocated by the International Society for Rock Mechanics (ISRM), because of the fact that the measurement is independent of the core diameter, providing a standardized RQD, and that improperly penalizing the rock mass quality for cases where fractures are parallel to the borehole and are cut by a second set.

When measuring the core length, core breaks caused by the drilling process, i.e. man-made or mechanical breaks, must be considered as non-existent. In other words, core pieces created by mechanical breaks must be fitted together and counted as one single core. If the type of the break cannot be decided exactly, the type should be considered as natural in order to be conservative in the calculation.

Core pieces that are not *hard* and *sound* should be included in the RQD calculation even if their measured length is greater than 100 mm (4-in.). The objective of this requirement is to downgrade the rock quality where the rock has been altered and weakened. If the core being a sound one or not cannot be decided, to be conservative in the calculation, it should be considered as not fulfilling the soundness requirement. While cores whose weathering type is classified as *fresh* or *slightly weathered* are included in the RQD count, and *moderately weathered* cores are included with the asterisk qualifier (*) next to them, *highly weathered*, *completely weathered*, and *residual soil* are excluded from the RQD count (Deere & Deere, 1989).

The other input parameter of RQD is length of the core run. When the length of the core run decreases, RQD becomes more sensitive. For example, RQD value for a highly fractured 300-mm-long zone within a massive rock would be 90%, 80% and 40% respectively for run lengths of 3 m, 1.5 m and 0.5 m. Deere & Deere (1989) recommend that the calculation of the RQD be based on the actual drilling-run length used in the field, which should not be greater than 1.5 m and certainly not more than 3 m. “The ISRM Commission on Standardization of Laboratory and Field Tests (ISRM, 1978) recommends RQD logging using variable *run lengths* to separate individual beds, structural domains, weakness zones, etc., so as to indicate any inherent variability, and to provide a more accurate picture of the location and width of zones with anomalously low or high RQD values.” (Deere & Deere, 1989, p. 20).

An example RQD measurement can be observed in Figure 22.

RQD is correlated with tunnel support/reinforcement design, prediction of in-situ modulus, foundation settlement and fracture frequency. It is also utilized in rock classification systems such as *Rock Mass Rating* (RMR) (Bieniawski, 1976) and *Q* (Barton et al., 1974) systems.

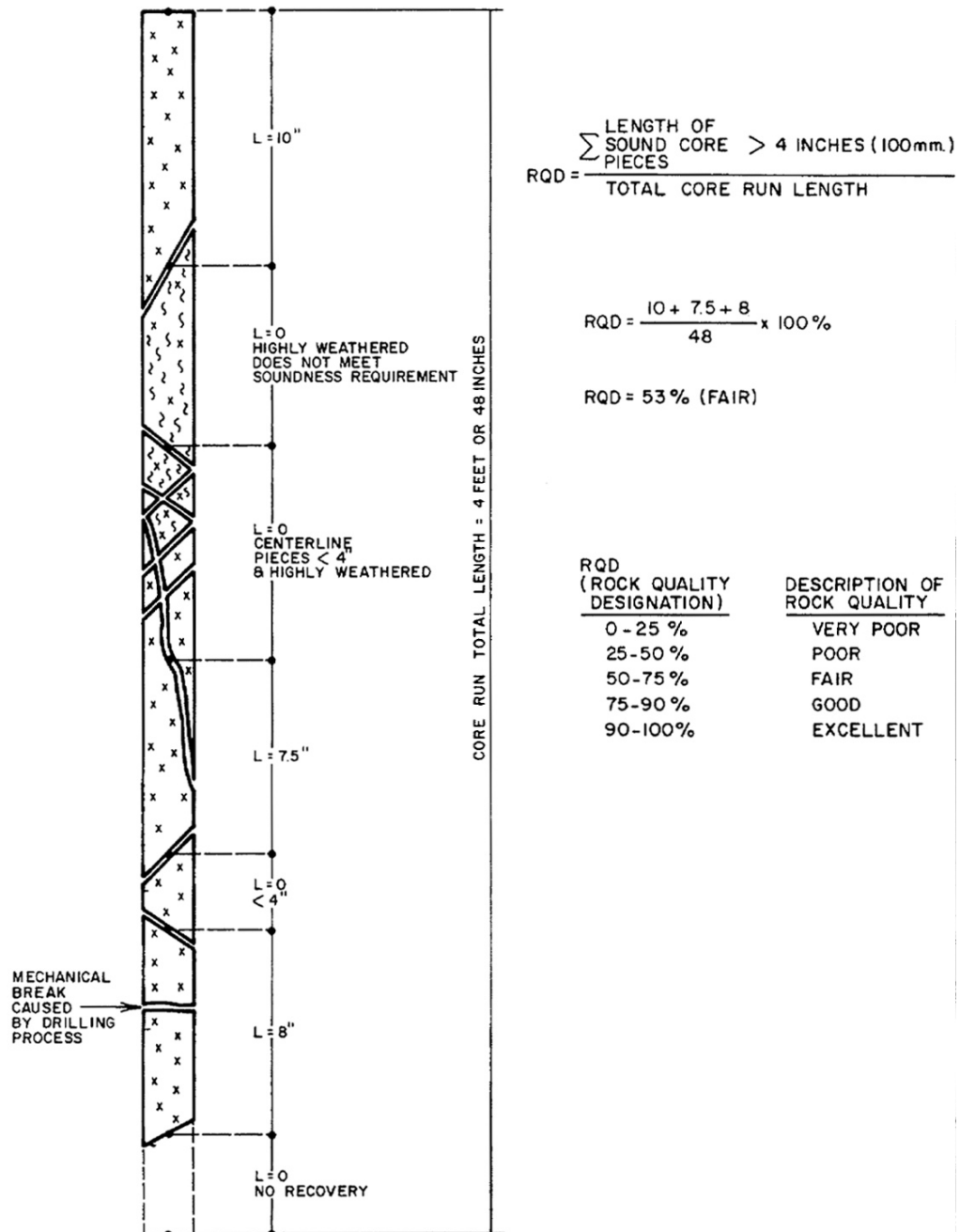


Figure 22. RQD logging (Deere & Deere, 1989, p. 12)

2.4 Theoretical Background

This chapter contains the theories that are utilized in this thesis study.

2.4.1 Graph Theory

Graph theory has several applications. In this thesis study, graph theory is used to create contour connection graphs to find the shortest paths that connect detected edges. The shortest paths are then used in the segmentation processes and in finding core fracture paths.

A *graph* G is an ordered triple $(V(G), E(G), \psi_G)$ where $V(G)$ is a non-empty set of *vertices*, $E(G)$ is a set, disjoint from $V(G)$, of *edges* and ψ_G is an *incidence* function. ψ_G associates each *edge* of G with an *unordered* pair of *vertices* of G . Let $e \in E(G)$ and $u, v \in V(G)$ such that $\psi_G(e) = uv$. Then, u and v are said to be joined by e , and called the *ends* of e .

As an example, a graph G is defined as

$$G = (V(G), E(G), \psi_G)$$

where

$$V(G) = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$$

and

$$\psi_G(e_1) = v_1 v_2, \quad \psi_G(e_2) = v_2 v_3, \quad \psi_G(e_3) = v_3 v_3, \quad \psi_G(e_4) = v_3 v_4$$

$$\psi_G(e_5) = v_2 v_4, \quad \psi_G(e_6) = v_4 v_5, \quad \psi_G(e_7) = v_2 v_5, \quad \psi_G(e_8) = v_2 v_5$$

A graphic representation of graph G is shown in Figure 23. Although the names of the vertices and edges do not have any meaning in the graph, they are shown for clarity. In the diagram of the graph G , the vertices are shown as small circles while edges are shown as lines.

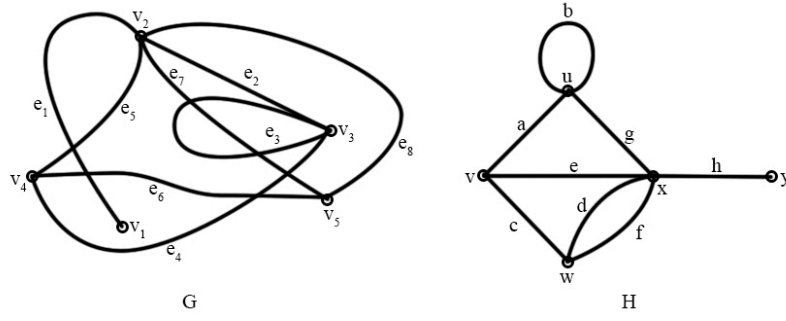


Figure 23. Diagrams of graphs G and H

The vertices of an edge are said to be *incident* with that edge. Also, the edges of a vertex are said to be *incident* with that vertex. *Adjacent* vertices are the vertices that are incident with a common edge. Likewise, *adjacent* edges are the edges that are incident with a common vertex. A *loop* is an edge whose ends are identical. For example, *b* of graph *H* in Figure 23 is a loop. Edges that are not a loop are called *links*.

A *finite* graph is a graph whose set of vertices and edges are both finite. A *simple graph* is composed only of links, i.e. without loops, and the same pair of vertices are connected by only one edge.

Identical graphs are the graphs whose vertices, edges and incident functions are the same. Identical graphs can be represented by the same diagram. However, two graphs that are not identical can also be represented by the same diagram as well. For example, if the labels of vertices and the edges of graph *H* in Figure 23 are different, then it is a different graph which has the same representation as graph *H*. In this case, these graphs are called *isomorphic*. Let *G* and *H* be two graphs. If there are bijections $\theta: V(G) \rightarrow V(H)$ and $\phi: E(G) \rightarrow E(H)$ such that $\psi_G(e) = uv$ if and only if $\psi_H(\phi(e)) = \theta(u)\theta(v)$. In this case, the graphs *G* and *H* are called *isomorphic*, denoted as $G \cong H$ and pairs of mappings (θ, ϕ) are called *isomorphism* between *G* and *H*.

Incidence matrix is the matrix $v \times e$. Let the vertices of a graph *G* be v_1, v_2, \dots, v_v and the edges be e_1, e_2, \dots, e_e . Then, the *incidence matrix* of *G* is $\mathbf{M}(G) = [m_{ij}]$, where $[m_{ij}]$ is the number of times v_i and e_j are incident, which can only be 0, 1, or 2. A graph can be specified by an incident matrix. *Adjacency matrix* of graph *G*, denoted

as $\mathbf{A}(G)$, is $v \times v$ matrix $\mathbf{A}(G) = [a_{ij}]$, where a_{ij} is the number of edges that connects v_i and v_j . Incidence and adjacency matrices of graph G in Figure 24 is shown in Figure 25.

A graph H is a subgraph of G , denoted as $H \subseteq G$, if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$, and ψ_H is the restriction of ψ_G to $E(H)$. If $H \subset G$, H is called a *proper subgraph*. In these cases, G is called a supergraph of H . A subgraph H with $V(H) = V(G)$ is called a *spanning subgraph* (Bondy & Murty, 1976). The graph created after deleting all loops and duplicate links in G is called the *underlying simple graph* of G .

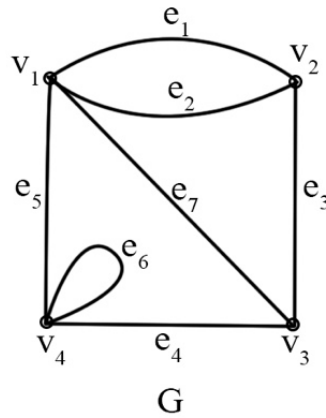


Figure 24. Graph, G

	e_1	e_2	e_3	e_4	e_5	e_6	e_7		v_1	v_2	v_3	v_4
v_1	1	1	0	0	1	0	1	v_1	0	2	1	1
v_2	1	1	1	0	0	0	0	v_2	2	0	1	0
v_3	0	0	1	1	0	0	1	v_3	1	1	0	1
v_4	0	0	0	1	1	2	0	v_4	1	0	1	1

Figure 25. Incidence and adjacency matrices of graph G

The *degree of a vertex* v in graph G , denoted as $d_G(v)$, is the number of edges that are incident with v , where each loops is counted as two edges.

A *walk* in graph G is a finite non-null sequence $W = v_0 e_1 v_1 e_2 v_2 \dots e_k v_k$, where $v_{0..k}$ denotes a vertex and $e_{1..k}$ denotes an edge. A walk starts with a vertex, called the *origin*, and ends with a vertex, called the *terminus*. In a walk, there comes an edge

after a vertex, and vice versa. The vertices v_1, v_2, \dots, v_{k-1} are *internal vertices*, where k is the length of W . A walk in a simple graph can also be represented by its vertex sequence, without including the edges. If a walk W is composed of distinct edges, W is also called a *trail*. If a trail's vertices are distinct, it is also called a *path*. In the graph shown in Figure 26, $uavfyfvgyhwbv$ is a walk, $wcxdyhw bvg y$ is a trail, and $xcwhy euav$ is a path.

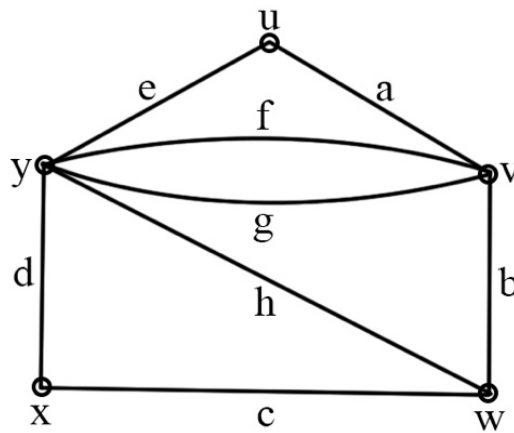


Figure 26. A sample graph

Two vertices u and v of graph G are said to be *connected* when there is a path that connects u and v in G . Subgraphs $G[V_1], G[V_2], \dots, G[V_\infty]$ of graph G are called the *components* of G . If G is composed of a single component, then it is a *connected graph*. Otherwise, it is a *disconnected graph*.

A *weighted graph* is a graph where there is a real number associated with each edge. A *directed graph* D is a graph where each edge has an *ordered* pair of vertices such that $\psi_D(e) = (u, v)$, where e is an edge, and u and v are ordered vertices, called *tail* and *head* of e .

CHAPTER 3

DETERMINATION OF RQD BY DIGITAL IMAGE ANALYSIS

Rock Quality Designation (RQD) (Deere et al., 1967) is a rock quality index used in tunnel support/reinforcement design, prediction of in-situ modulus, finding allowable foundation contact pressures and fracture frequency, and as a parameter of RMR (Bieniawski, 1976), Q (Barton et al., 1974), and Geological Strength Index (GSI) (Marinos & Hoek, 2000) rock mass classification systems. Hence, determination of RQD by geotechnical logging is one of the routine tasks done before making a judgment about the rock. However, manual logging of hundreds of meters of core with a tape measure is a very laboursome and tedious process. A systematic approach can be a better alternative for RQD logging.

In literature, two approaches were used for calculation of RQD using photography and scanning techniques. Lemy et al. (2001) used an edge-detection-based approach. In their study, after detecting the edges with Steger algorithm (Steger, 1989), they followed an edge reconstruction process which calculates a cost for pairs of line segments and connects the line segments presenting the lowest cost to find the breaks. Finally, they calculated RQD after eliminating spurious breaks. In their approach, they were not able to detect mechanical breaks. Moreover, the algorithm considers the core box full of cores, i.e. no core-free regions (lost or washed-out cores), which might cause including core-free regions as cores in RQD calculation. Since the algorithm relies on edges only, it might consider thick fillings as breaks as well. Olson et al. (2015) used a 3D non-contact laser digitizer to generate 3D point cloud data of the entire core box. They detected the breaks and location of each core by analyzing the 3D point cloud. They also created a fracture characterization algorithm which finds out whether a fracture is formed due to a mechanical break or not. While using a 3D

laser digitizer provides an excellent accuracy in core segmentation, it is an expensive device and the process is slow, tedious and hard to apply in the field.

A shadow-based method for segmentation of cores and determination of RQD from digital images of rows of core in core boxes in order to compute RQD in an automatic way by finding and locating natural fractures in cores and measuring intact core lengths is introduced. The method is fast, cheap, and easily applicable in field conditions.

3.1 Material and Methods

In this study, a digital camera (Canon EOS 7D Mark II with Canon EF-S 18-135mm IS STM), a wireless remote controller (Canon RC-6), a flash (Canon Speedlite 600EX II-RT), a flash diffuser, a tripod with horizontal arm, a tripod head, a 5-cm-long pink marker, and a few black plastic trash bags (Figure 27 and Figure 28) are used to process core boxes with 4 and 5 rows each having 100 cm length, containing several different cores with mechanical and natural cracks, and having different core/lump lengths (Figure 29).

Figure 27 shows the equipment used in this study, where 1) is the flash (Canon Speedlite 600EX II-RT), 2) is the digital camera (Canon EOS 7D Mark II with Canon EF-S 18-135mm IS STM), 3) is the flash diffuser, 4) is the wireless remote controller (Canon RC-6) and 5) is a 5-cm-long pink marker.



Figure 27. Equipment used in the thesis study

To be able to take pictures of the core boxes from above, the camera is mounted to the horizontal arm of the tripod with a tripod head. The arm is extended so that the camera is positioned above the core box. The black plastic bags are used to create a black background and prevent unwanted light sources from illuminating the core box. The pink marker is used to measure a pixel's physical length, as well as an indicator of the starting location of the shallowest depth in the core boxes. The setup can be observed in Figure 28 and in Appendix.



Figure 28. Equipment setup

This research study is conducted by using a single flash which is manually moved to three different positions to capture three different images. The tent setup with three different light sources is designed to create a dark-room environment in the field.

Algorithm development has been performed in MATLAB 2017a (The MathWorks, 2017), which comes with a bundle of toolboxes, such as image processing and statistics toolboxes, that contain a lot of well-written and efficient algorithms used in common image processing and statistics operations.



Figure 29. Core boxes used in algorithm development process

3.2 Theory and Calculations

In a typical logging procedure, the rock mass is divided into a number of structural domains and each domain is classified separately. The boundaries of the structural domains usually coincide with a major structural feature such as a fault or with a change in rock type. In some cases, significant changes in discontinuity spacing or characteristics, within the same rock type, may necessitate the division of the rock

mass into a number of small structural domains. In this study, each drilling maneuver is assumed to be a domain.

Figure 30 shows level-1 data flow diagram for the algorithm used to compute RQD.

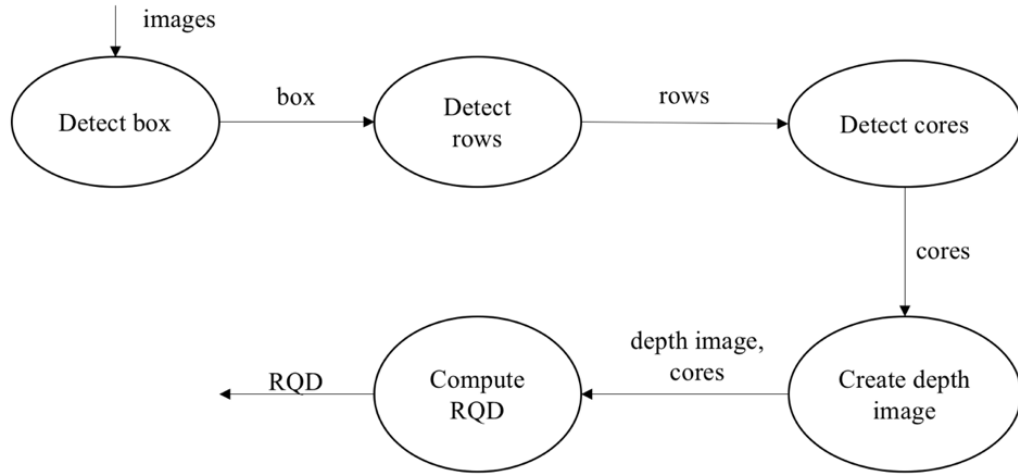


Figure 30. Level-1 data flow diagram

After the images are retrieved, box detection, row detection, core detection, depth image creation, and RQD computation operations are performed, respectively.

First, after placing a 5-cm-long pink marker at the beginning of the core retrieved from the shallowest depth, three digital true color images of the core box, with the same camera position but different light source positions (left, right, and top) are taken using a high-resolution camera. Next, after detection of the core box with color thresholding, the rows of the box are detected by using Hough transform and boundary tracing algorithms. Then, cracks and shadows, which are detected by using the images captured under different light source positions, and concave points in core mask are used to separate touching cores from each other. After core segmentation, each core's cylindrical and non-cylindrical parts, along with fracture paths separating these parts, are found by, again, using the shadows. Following this, by using the pink marker's position, cores in which rows should be flipped upside-down are found. Benefiting from this information, a depth image is created. After finding the centerlines of the cores whether the fracture paths are mechanical or not is decided; RQD is calculated by considering the length of valid sections of the centerlines of the cores.

3.2.1 Box detection

Core box detection consists of detection of the core box and the marker by color thresholding, and finding physical length of a pixel. Figure 31 shows the data flow diagram for this process.

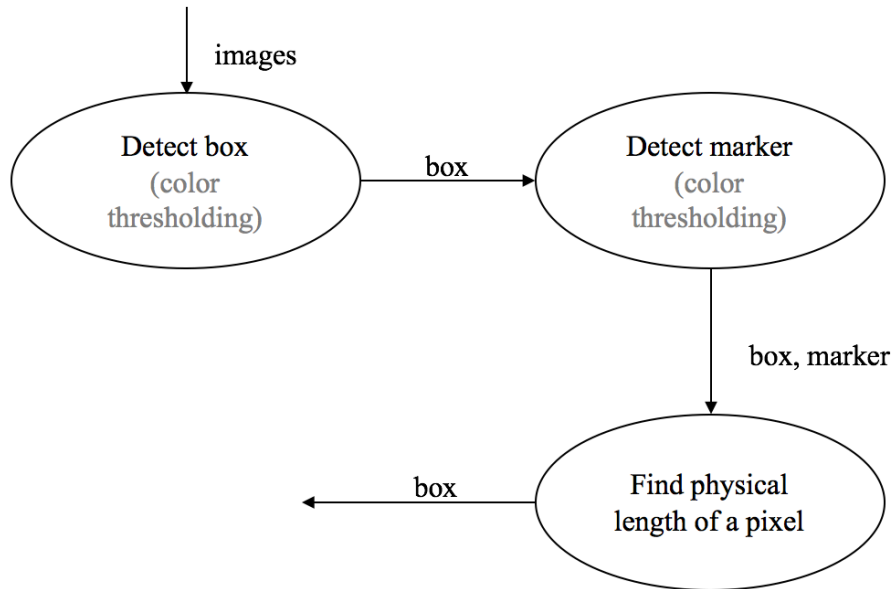


Figure 31. Level-2 data flow diagram for 'detect box' process

Detecting the Box

Blue plastic core boxes are widely used in the mining industry. The core boxes used in this study are also blue. Therefore, color thresholding approach is used to detect the core box. To improve the performance of color thresholding, black trash bags are laid under the core box so that the only blue object in the image is the core box.

It is easier to separate blue from other colors in Lab color space compared to RGB, because Lab color space has lightness values separately in L* channel. Therefore, the image is converted from RGB to Lab using MATLAB's `rgb2lab` function. [-100, 0] range of Lab's b* channel represents the blues in the image. First, the histogram of b* channel is calculated. The threshold value is then decided as the value at the local minima closest to the most frequent blue value and whose value is less than the most frequent value.

An example is shown in Figure 32. The image on the left is the main image without color thresholding, and histogram of b^* channel of the image below it. The image on the right is the main image with color thresholding. The histogram below, which shows the values in b^* channel, shows the threshold value (the value at where left handle is) that removes the core box.

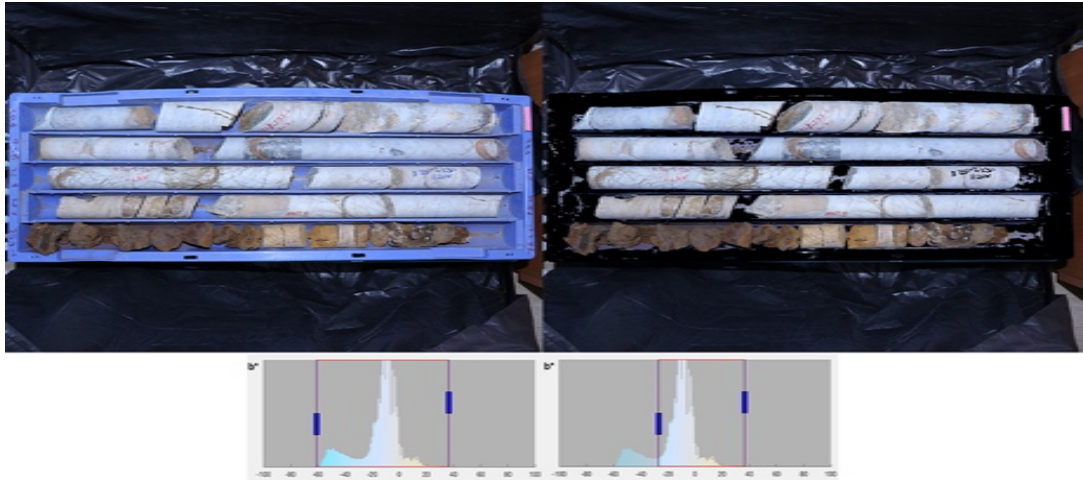


Figure 32. Box detection using color thresholding

To extract the box from the main image, the mask found by thresholding is inverted. After that, coordinates of the non-zero pixels that are the closest to left, right, top and bottom limits of the inverted mask are used to crop the image.

Detecting the Marker and Finding Physical Length of a Pixel

A 5-cm-long pink marker is used to find physical length of a pixel. Before taking pictures of a core box, a pink marker having an exact length of 5 cm is placed on the core box, parallel to one of the short edges of the core box. This marker is detected by color thresholding. The found rectangular object's length is measured in pixels. Finally, a pixel's length in mm is found by dividing 50 mm to the length of the object in pixels.

3.2.2 Row Detection

After box detection, the rows are detected. Figure 33 shows the data flow diagram for this process.

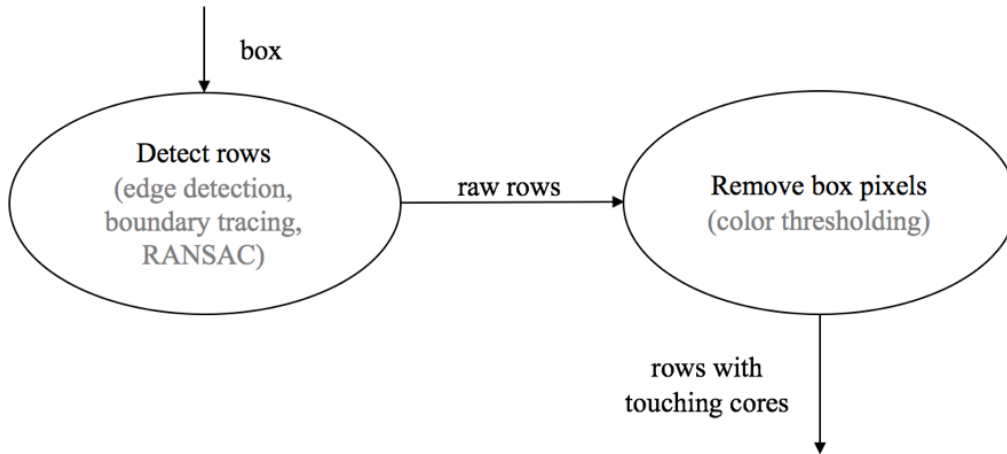


Figure 33. Level-2 data flow diagram for 'detect rows' process

First, the rows are detected by using edge detection, boundary tracing and RANSAC approaches, respectively. Then, the box pixels are removed by color thresholding to get the cores.

Detecting Rows

Each row of a core box is separated from each other with a separator. When the core box is viewed from above, these separators look like vertical lines. Row detection is done by exploiting this.

First, strong edges in the box image is found by Canny edge detector (Canny, 1986). Next, by using Hough transform, vertical lines in the edge image are detected. After that, the contours in the edge image that overlap one of the detected vertical lines are traced by using a slightly modified boundary tracing algorithm which stops tracing the contour when it detects a non-vertical part of the contour. In other words, non-vertical parts of the contours are trimmed. The traced and trimmed contours can be observed in Figure 34, where, from left to right, the first image shows the strong edges in the box image, detected by Canny edge detector, the second image shows Hough lines (green) and vertical parts of the traced contours (magenta) and the third image shows vertical lines fit to traced vertical contours by RANSAC.

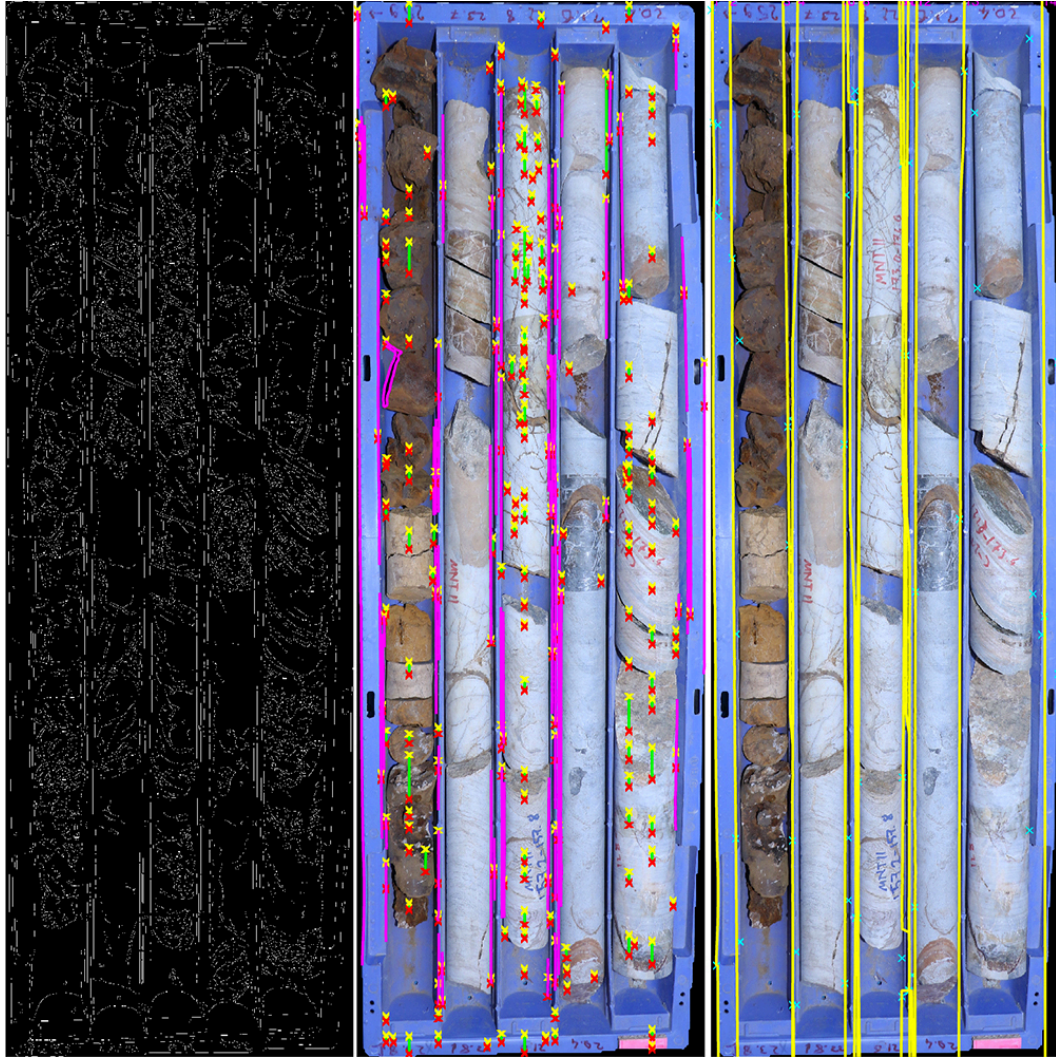


Figure 34. Row detection

Then, vertical contours that are close to each other in X and that do not overlap each other are connected to each other with a straight line. Next, by using RANSAC (random sample consensus), a vertical line is fit to the connected vertical contours. At this point, there might be more than one vertical line found for each row separator. In that case, the vertical line whose slope angle is the closest to 90° is selected. Finally, the regions between the vertical lines are extracted by eliminating the regions whose area is less than a threshold value. These regions are the rows of the core box.

Removing Box Pixels

After the rows are detected, the box pixels are removed from each row region by using color thresholding, which is described in section 3.2.1.

3.2.3 Core Detection

Row detection process is followed by core detection process. Level-2 data flow diagram for core detection can be observed in Figure 35.

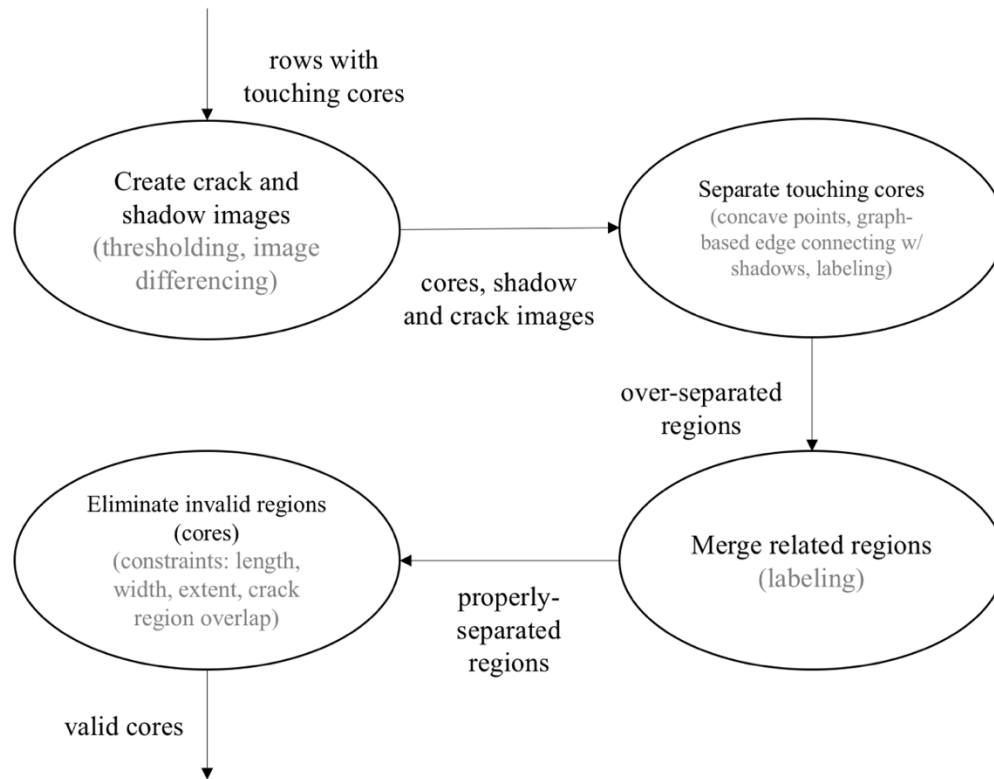


Figure 35. Level-2 data flow diagram for 'detect cores' process

The rows with touching cores that are output by row detection process are input to core detection process. First, crack and shadow images are created by thresholding and image differencing methods. Then, touching cores are separated from each other. After this, over-separated regions are created. Next, the related regions are merged with each other to obtain regions that better represent the cores. Finally, invalid regions are eliminated to retrieve valid cores.

In a row, cores are placed adjacent to each other. Most of the times, the adjacent cores touch each other, or they look like they touch each other when the core box is viewed from the top. Hence, cores that touch each other needs to be detected separately for a proper segmentation.

Core segmentation can be done by using several methods, such as using edge detection, pattern recognition, stereo photogrammetry, depth estimation using defocus, depth estimation from a single image using deep learning, a 3D laser scanner, and shadows. Using the edges alone does not produce good results because cores might have too many different textures on them. In this case, whether an edge is formed due to a touching point or not is ambiguous. Pattern recognition might be a good choice. However, it might not be too precise and quickly become a tedious task because of the hard-to-predict nature of textures on the cores. Stereo photogrammetry can be used to get a 3D point cloud of the entire core box. However, it requires at least two cameras or a camera placed on top of a slider, dealing with which might be tiresome. Depth estimation using defocus produces good results when the subject is close to the camera. However, 3D point cloud of an object as big as a core box that is far away from the camera cannot be precisely retrieved using defocus method. In addition, several photographs or a high quality video must be taken by changing the focus. Estimating the depth from a single image with deep learning requires a lot of training, and hence, collecting too much data. In addition, training a deep neural network might take too much time and, since it requires a lot of GPU power, it might be quite expensive. With a 3D laser scanner, 3D point cloud of the entire core box along with the cores can be detected with high precision. However, it is an expensive device. Using shadows require controlled lighting conditions and more than one photograph of the core box to be taken, in this study, three photographs are taken. However, when used with edge detection algorithms, this method can produce good results. Moreover, it is cheaper than a 3D laser scanner and easier to implement when compared to pattern recognition and the deep learning methods. Also, taking photographs can be automated. Hence, in this study, core segmentation task is done by using shadows because it is cheaper and easier to implement.

The cores are placed in rows inside a core box, and a typical core fills the width of a row. Hence, a light source placed at the start or the end of a row, which beams light parallel to the row, causes creation of shadows that can be used to detect cylindrical and non-cylindrical parts of the cores. Considering this, in addition to the photograph taken with a top light source, two additional photographs are taken by placing the light

source to the left and right sides of the core box so that shadows can be created for each end of each core. The images can be observed in Figure 36. In this figure, from left to right, the first image is the gray core box image illuminated by a light source at the top, the second image is the gray core box image illuminated by a light source on the left, and the third image is the gray core box image illuminated by a light source on the right.

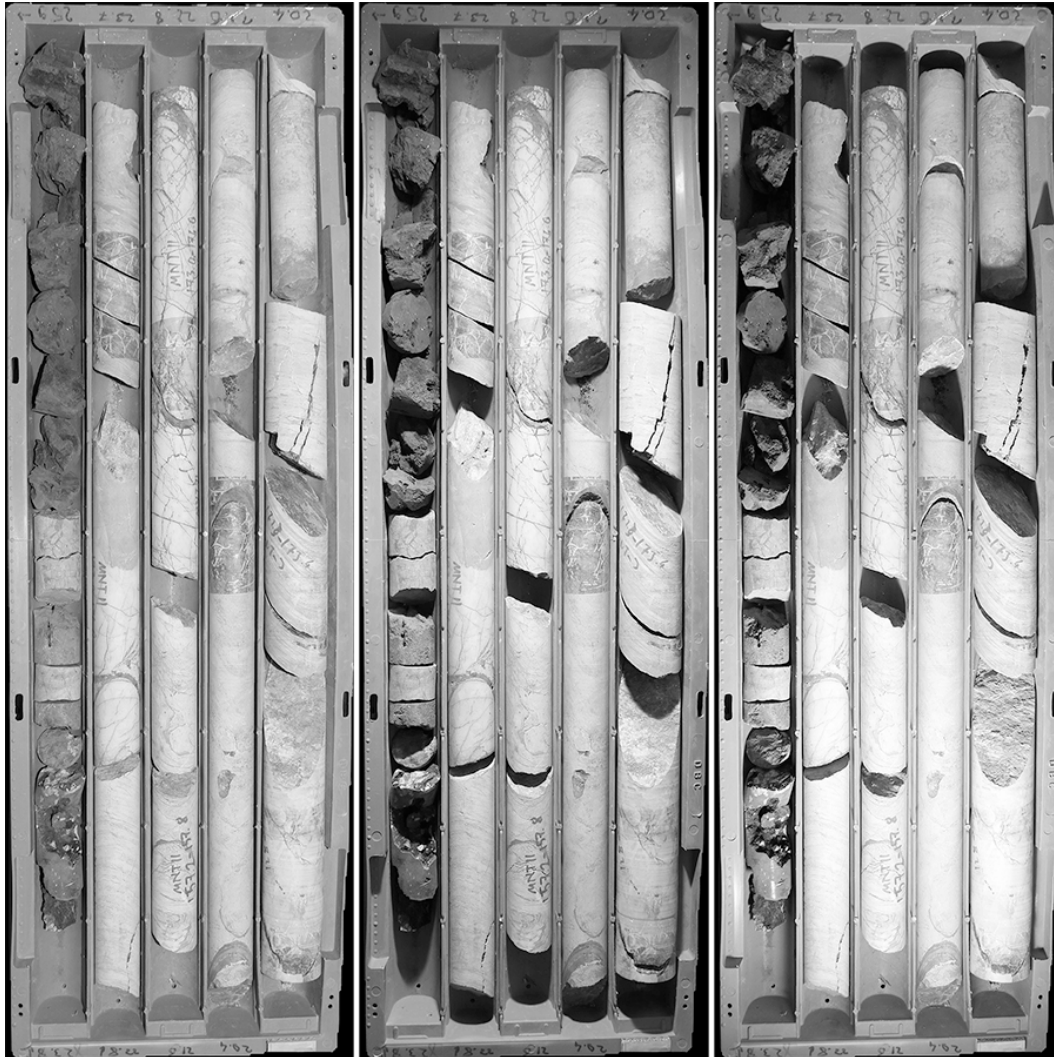


Figure 36. Shadow images

Detecting Cracks

Cracks are actually the regions having too low intensity. They are found by simply thresholding the gray versions of three core box images that were taken under different lighting conditions.

First, the pixels whose intensity is lower than some threshold value are found in all three images of the core box. Then, the found low-intensity regions are combined to create a single mask. Next, the pixels that are too close to row separators are removed, because they might be created by the shadows of row separators. After that, the image is prepared by some morphological operations such as cleaning and closing. Crack regions whose area is less than some threshold area are removed to make sure small low-intensity regions are not included, since they might have been created by rough surfaces on the cores, which actually do not indicate a crack. Finally, the mask is dilated and the regions outside of the core mask, i.e. regions that overlap the core box mask, are removed. The crack image will be used to separate touching cores. Therefore, the regions in this image should overlap contours that might separate the touching cores. A dilated crack region will overlap more contours, which increases the chance of finding the right contours that separate the touching cores. A typical final crack mask can be observed in Figure 37. In this figure, the image on the left is the shadow-free RGB image of the box, while the image on the right is the crack mask.

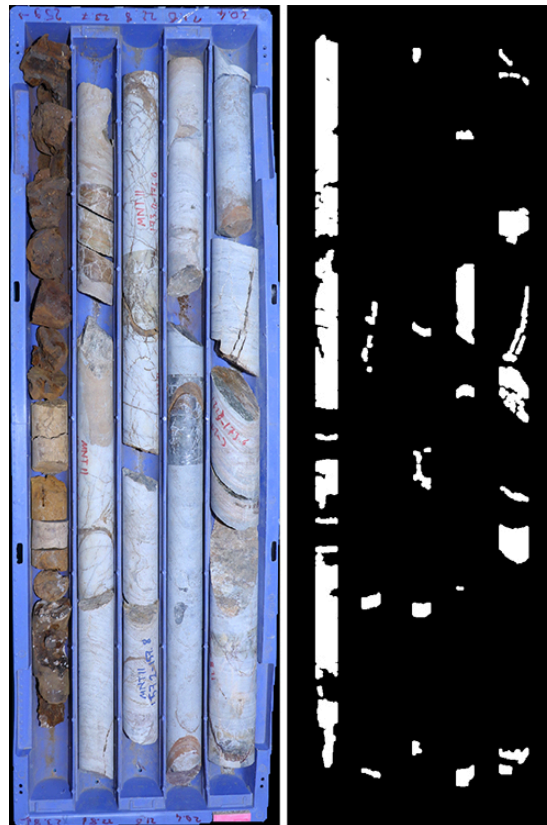


Figure 37. Detected cracks

Detecting Shadows

Shadows are found by comparing the shadow images to the shadow-free image. Since the differences between these images are the shadows, a pixel-based change detection approach is adopted. For this purpose, image differencing method is used.

Found shadow masks are prepared by morphological operations, such as closing, filling holes and cleaning, and by removing the regions that fall outside of the core mask. Final shadow masks can be observed in Figure 38. In this figure, from left to right: 1) shadow-free image of the box, 2) shadows detected from the image of the box illuminated by a light source on the left side, and 3) shadows detected from the image of the box illuminated by a light source on the right side.

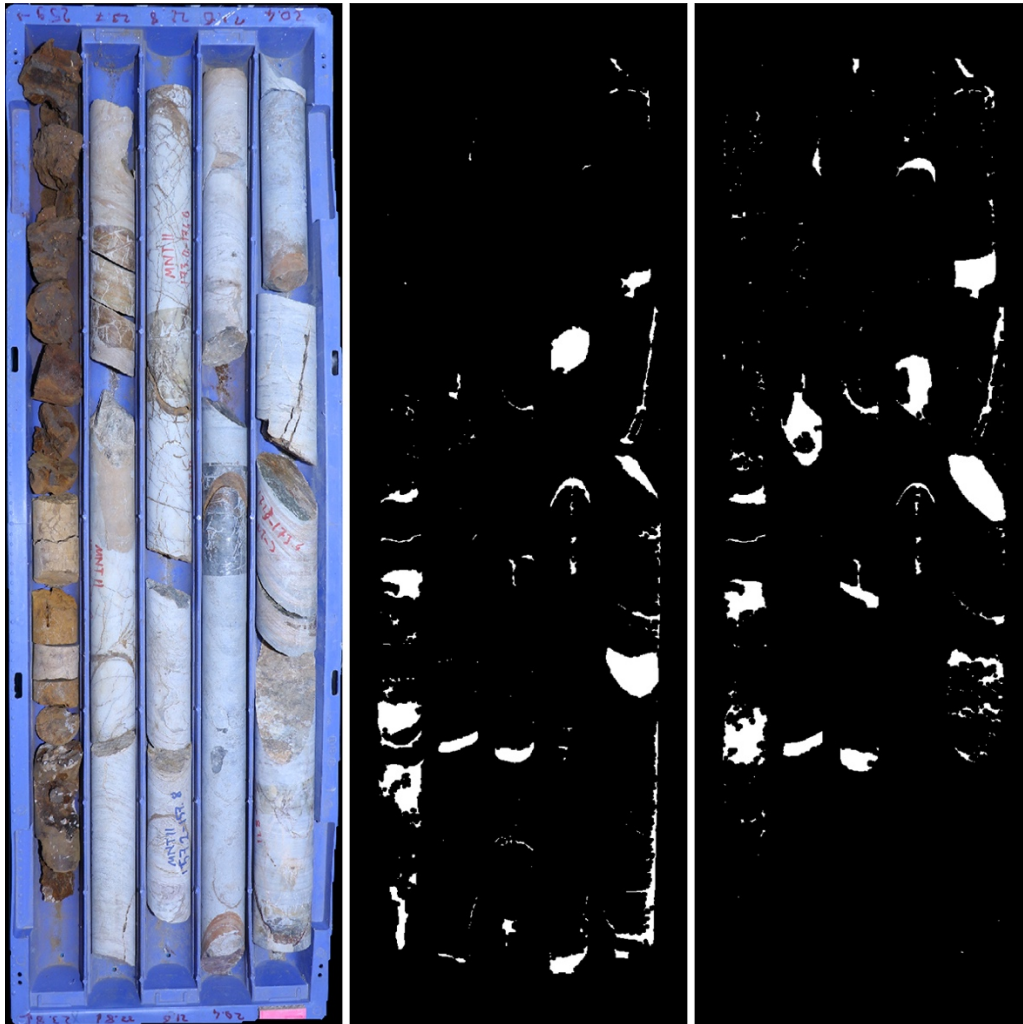


Figure 38. Detected shadows

Detecting Touching Cores

Cores are detected using the crack mask (Figure 37), the shadow masks (Figure 38) and core mask (Figure 39), i.e. the mask that removes the pixels belonging to the box itself, which is retrieved by color thresholding.

Figure 39 shows the found core mask. In this figure, from left to right: 1) gray version of box image, 2) core mask, and 3) gray core image.

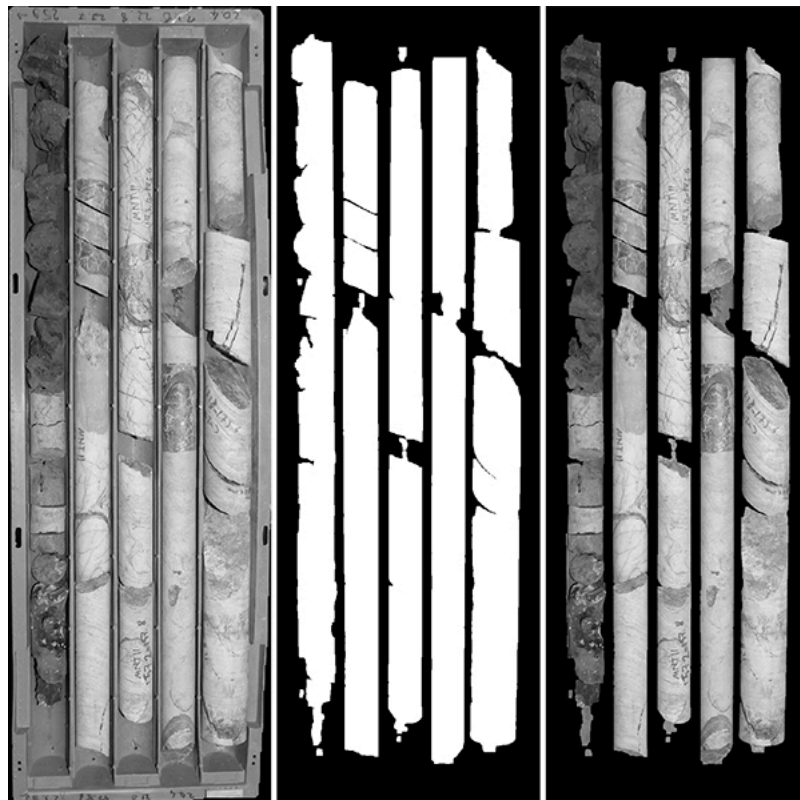


Figure 39. Core mask

Separating Touching Cores

First, strong edges in gray core box image are found by Canny edge detector. The edges that come from the box itself are removed by applying the core mask to the edge image. Next, a contour connection graph is created.

A *contour connection graph* is a graph which is formed by considering contours (detected edges) as graph nodes and the smallest distance between the contours as graph edges. Visualization of a sample graph can be found in Figure 40. In this figure,

from left to right: 1) image showing two cores that touch each other. Blue frame shows the region of interest (ROI), 2) edges detected in the ROI, and 3) contour connection graph created from the edges in the ROI.

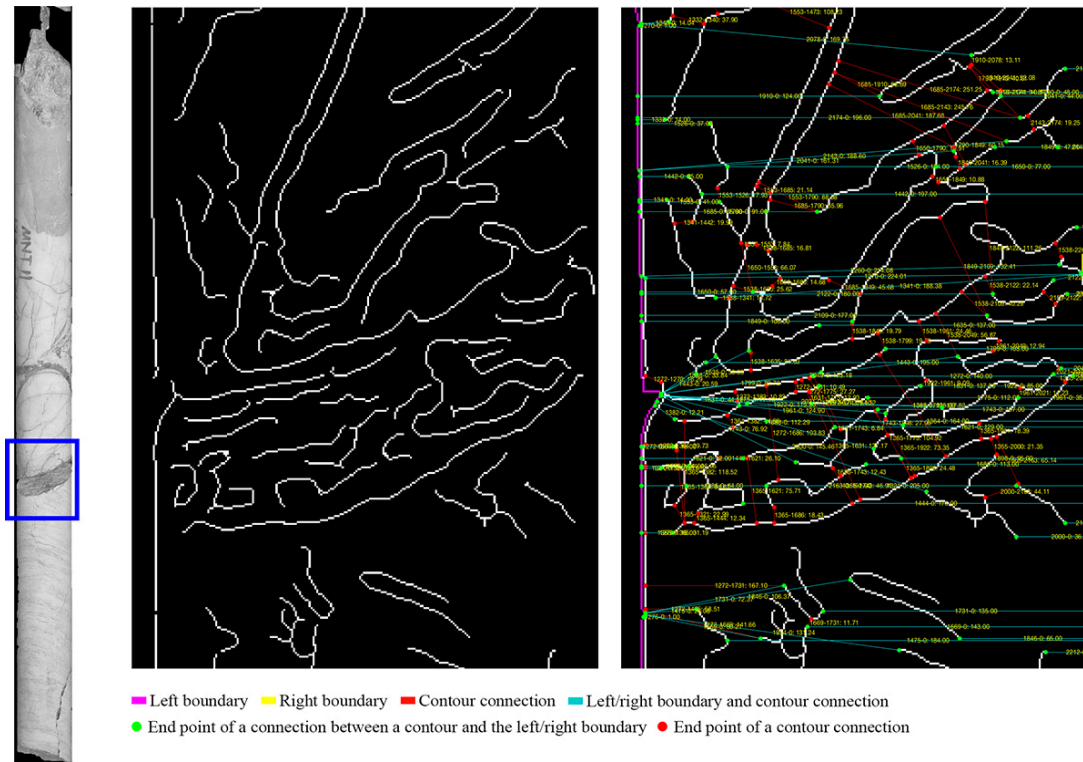


Figure 40. Contour connection graph

To create the graph, first, contours in the close vicinity of each contour is found. Then, the smallest distance between the contours that are close to each other is found. Next, a connection is created between the two contours using the two points that create the smallest distance between them. By following this procedure, each contour is virtually connected to the contours in their close vicinity.

As the next step in detecting cores, the boundary of the regions that store the cores are separated into two parts as left and right boundaries. The left boundary of a core region is the part of its boundary that is closer to the row separator on its left. Likewise, the right boundary is the part of its boundary closer to the row separator on its right. The theory is that if the left boundary can be connected to the right boundary by following a short path passing through the detected contours, that path is the path that separates two touching cores. To implement this logic, the left boundary and the right boundary

are assigned as sink and source nodes, respectively, in the contour connection graph, and every contour in the graph is virtually connected to both sink and source nodes.

After contour connection graph creation, all shortest paths that connect the left and the right boundaries and that pass through one of the contours overlapping a crack region are found by using Dijkstra's algorithm (Dijkstra, 1959). The contours forming the shortest paths are connected to each other with a straight line to make sure the paths separate the core region into at least two regions. Note that the final path is not cleaned from possible branches to allow natural separations to happen. Hence, most of the time, the crack region is separated into smaller regions by the branches of the shortest paths as well. These small regions will be merged together in the following parts of this section.

The core regions are further divided into regions by using concave points as well. The theory behind this idea is that when there is a concave point in a core region, it is probably created due to two adjacent cores. This is because a typical core is cylindrical and it does not have concave points on it. The concave points are found by calculating Euclidean distances from the boundary points of the core region to the boundary points of the region's convex hull. Convex hull of a region is the smallest convex region that contains the region. An example of convex hull can be observed in Figure 41. In this figure, the image at the top is the core region mask, while the one at the bottom is the convex hull of the core region mask.

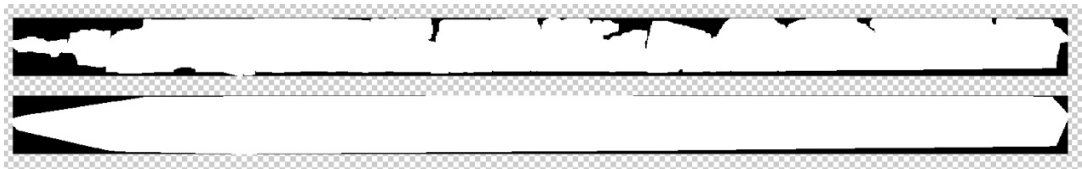


Figure 41. Convex hull

After the distances from each point of convex hull's boundary to the points of the region's boundary are calculated, the points whose distance to the hull's boundary is greater than a threshold are selected as concave points. Then, the concave points that are in the close vicinity of each other are connected by following low-intensity pixels, if the distance to be travelled is lower than a threshold value. In addition, for two

concave points to be connected, while one is on the left boundary, the other one should be on the right boundary. In other words, if two points that are close to each other are on the same side of the core, they are not connected. This is because the path created by connecting the two points should divide the region.

The connection path from one concave point to another is found by using MATLAB's `graydist` function. First, gray distance from first point to every point is calculated. Next, gray distance from second point to every point is calculated. Then, the distance images are summed and the minimum value is found. The pixels having the minimum value create the shortest gray path between the two concave points. After the skeleton of the path is found, the path is cleared from its branches following the same procedure on the path image by using MATLAB's `bwdistgeodesic` function this time, instead of `graydist`. Found concave points and connections between them can be observed in Figure 42. In this figure, from left to right: 1) core mask, 2) core mask with concave points represented by red dots, 3) RGB of the box with concave points. Yellow frames show the locations of the zoomed-in regions. Red lines in the zoomed-in regions are the valid connections between concave points.

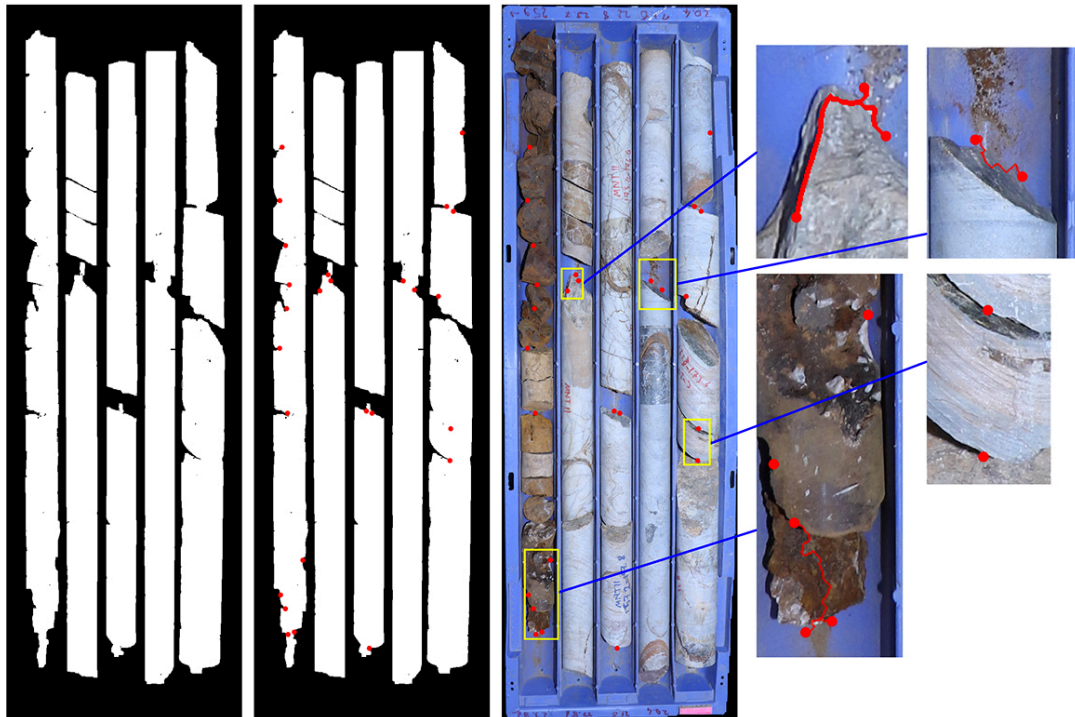


Figure 42. Connected concave points

The paths created between concave points are used to divide the core mask into smaller regions. Separation results can be seen in Figure 43. In this figure, the image on the left is the label image showing the regions created after separating touching cores. Each color represents a different region. Regions that have the same color but are not adjacent to each other are different regions. Red dots are the concave points. The second image is the RGB image of the core box. Again, red dots are the concave points.

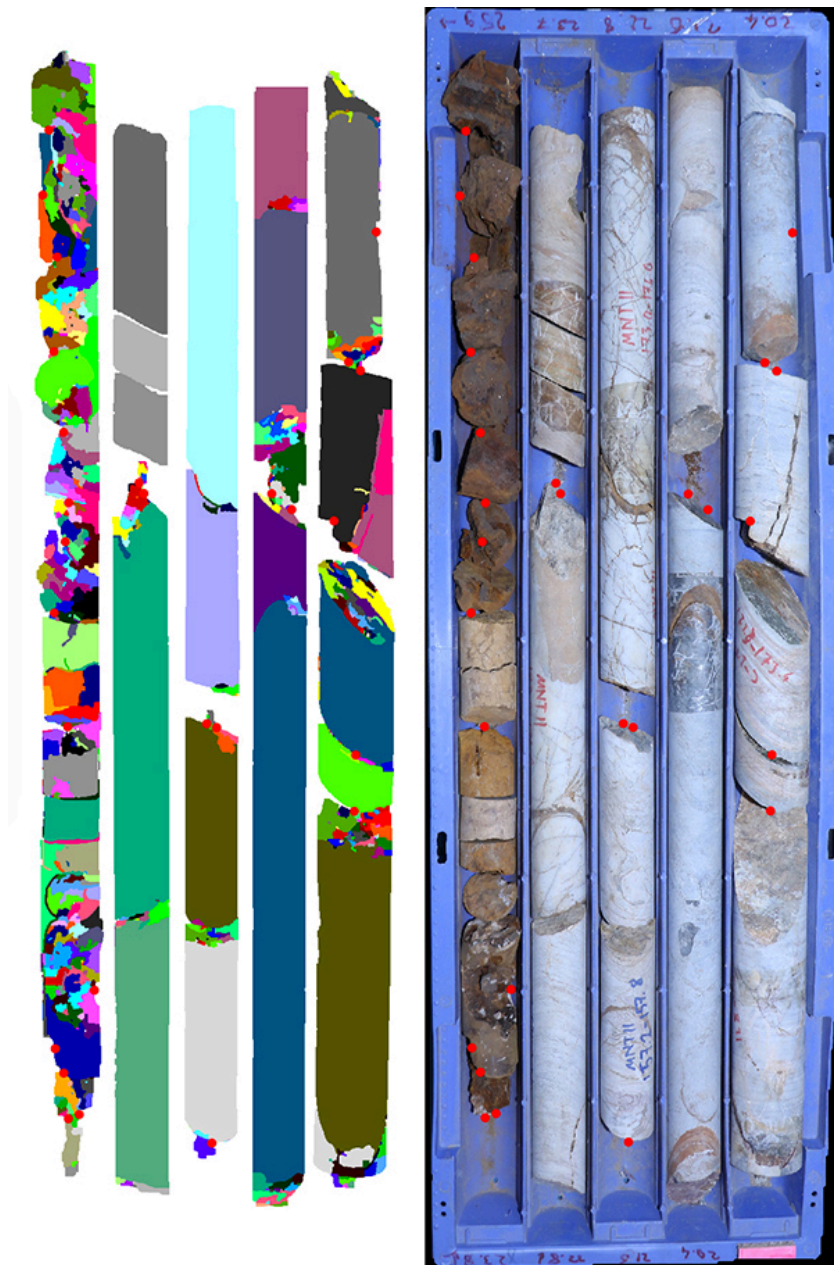


Figure 43. Regions after separating touching cores

Merging Related Regions

Because of the procedure followed to separate cores, there are several small and big regions created after the touching cores are separated. For a more precise segmentation, the regions that are found to be related to each other are merged with each other. Whether two regions are related or not is decided by using concave points and shadows.

First, before doing any merging, the regions that might represent a core are found. To decide this, there are several thresholds defined, which are minimum length, minimum width, minimum extent and maximum crack region overlap. Minimum length threshold is a value that defines the minimum vertical length a region must have so that it can be considered as a core region. Likewise, minimum width threshold defines the smallest width of a core region. Minimum extent constraint looks at the ratio of the number of pixels in the region to the number of pixels in its bounding box. Maximum crack region overlap is the threshold that sets a limit to how much of a core region overlaps a crack region at most.

According to the assumption about the concave points, a concave point only exists around where two cores touch each other. Therefore, the regions staying between a core region and a concave point are considered as they are related to that core region.

After the regions are merged by using the concave points, another merge operation is done on the created label image. This time, shadows are used to find related regions. Regions that overlap the same shadow region are considered as related. First, the shadow regions are found in the left and the right shadow masks. Next, areas that the two shadow masks overlap are found. Among these areas, smaller ones are discarded. Then, the remaining shadow areas are used to group the labels of non-core regions. After merging related non-core regions with each other, which core region they belong to is decided by looking at the shadow regions and concave points. When a non-core region is adjacent to a core region or they overlap the same shadow area, and there are no concave points between them, they are merged with each other. Results of merging can be seen in Figure 44. In this figure, red dots show the concave points. For the first and the second image from the left, each color represents a different region. Regions

that have the same color but are not adjacent to each other are different regions. In this figure, from left to right: 1) labels after separating touching cores, 2) labels after merging related regions, 3) left shadow image, 4) right shadow image, and 5) RGB of the core box.

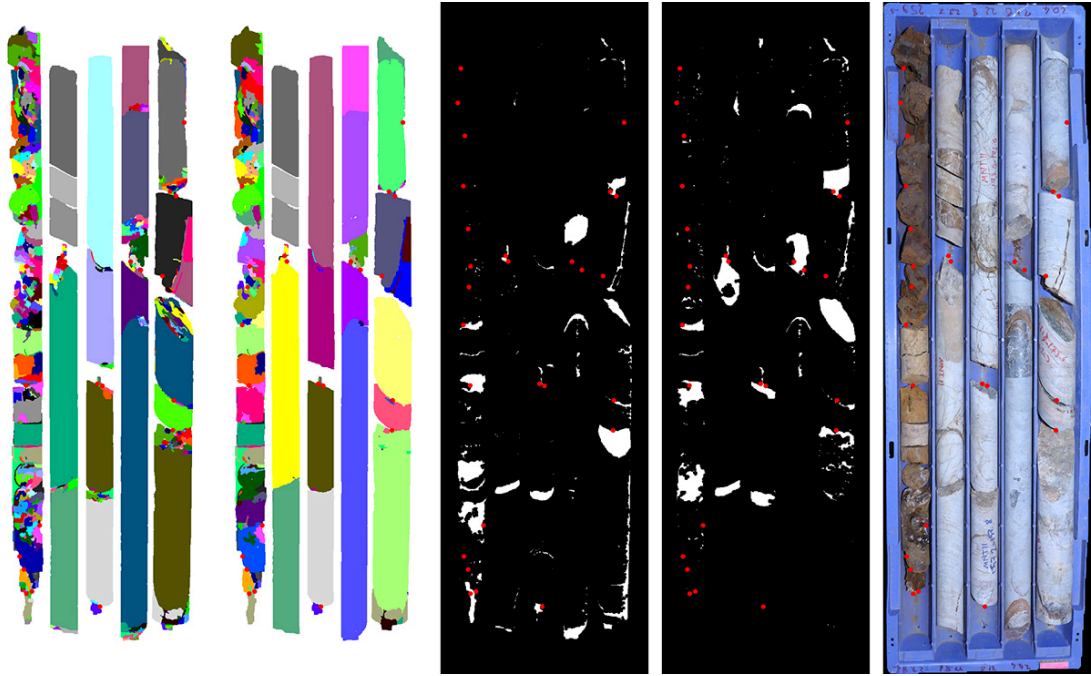


Figure 44. Merged labels

Eliminating Invalid Regions

After separation of touching cores and merging of related regions, the regions are subjected to an elimination process. Core regions are found by applying the thresholds described previously. Regions that are not classified as cores are eliminated. Figure 45 and Figure 46 show the detected cores. In these figures, from left to right, the first image is RGB of the core box. The second image is the labels decided as core regions. Each color represents a different region. Regions that have the same color but are not adjacent to each other are different regions. The third one shows the cores. Red frames are bounding boxes of each core. An ID is given to each core. The IDs are shown in green.

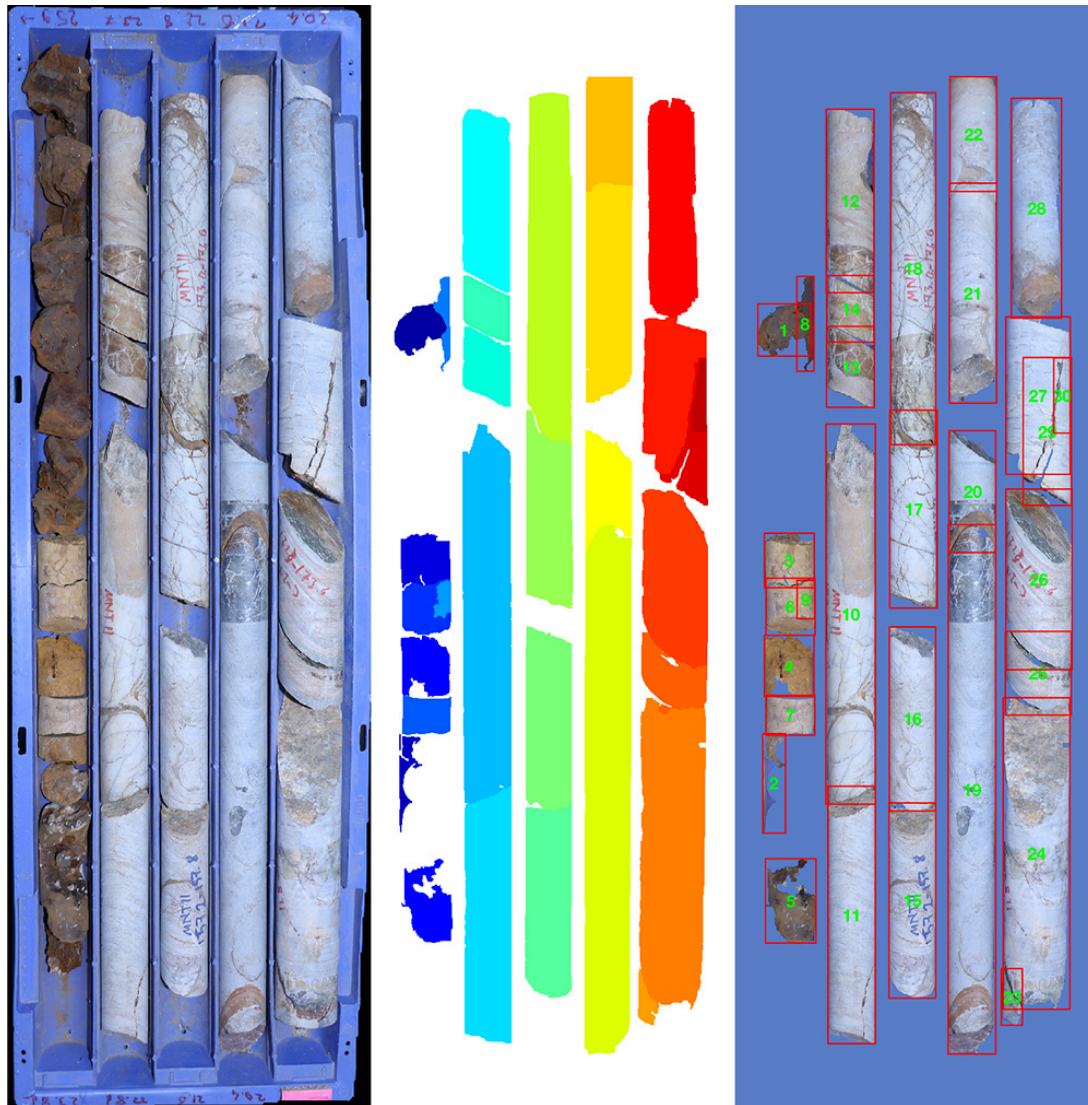


Figure 45. Detected cores

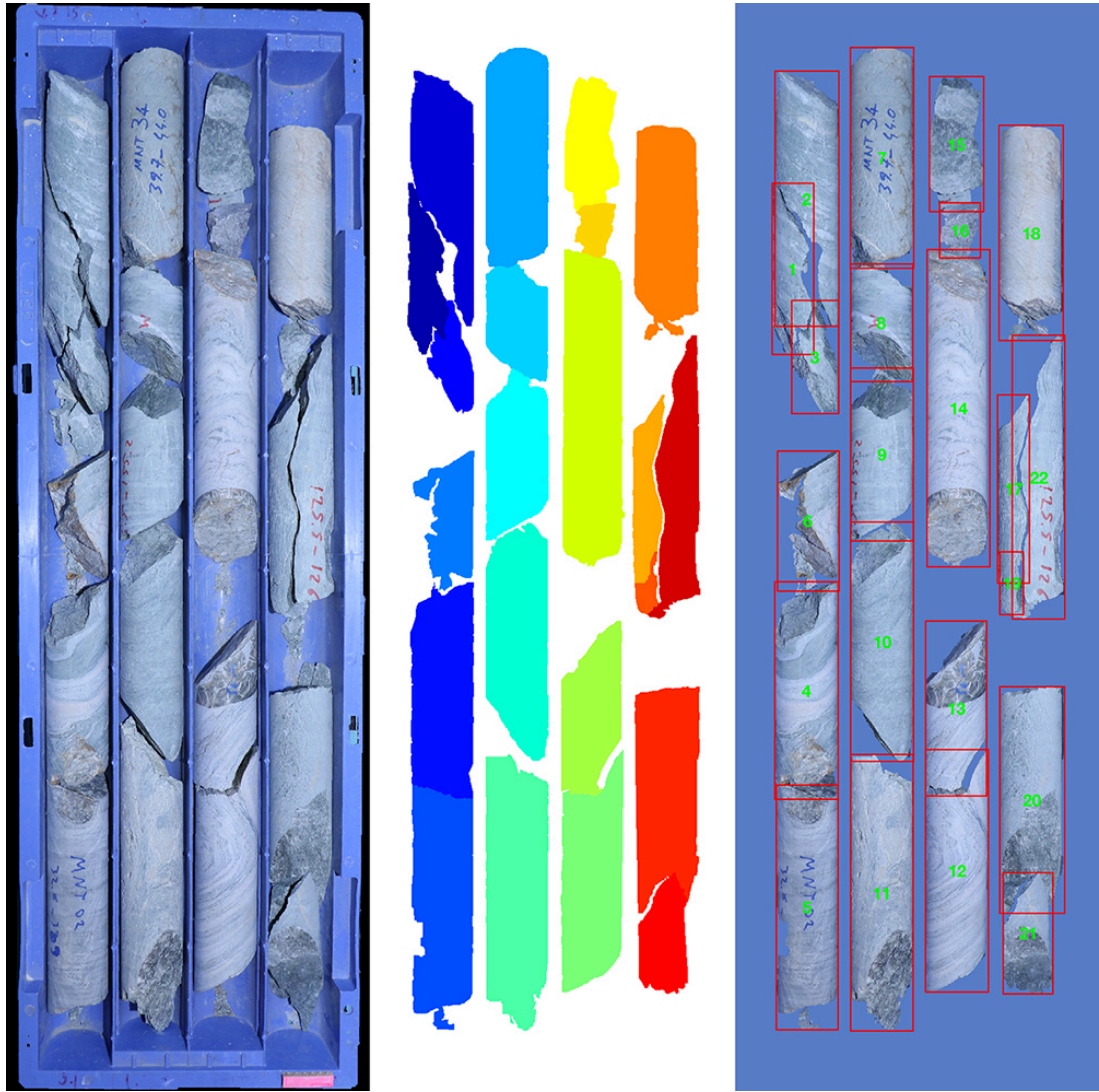


Figure 46. Detected cores in another core box

3.2.4 Depth Image Creation

After the cores are detected, the depth image is created. Creating the depth image consists of finding break paths of each core, detecting the depth of each core, and combining the cores into an image that starts from the shallowest core and ends with the deepest core. The process can be observed in the data flow diagram shown in Figure 47.

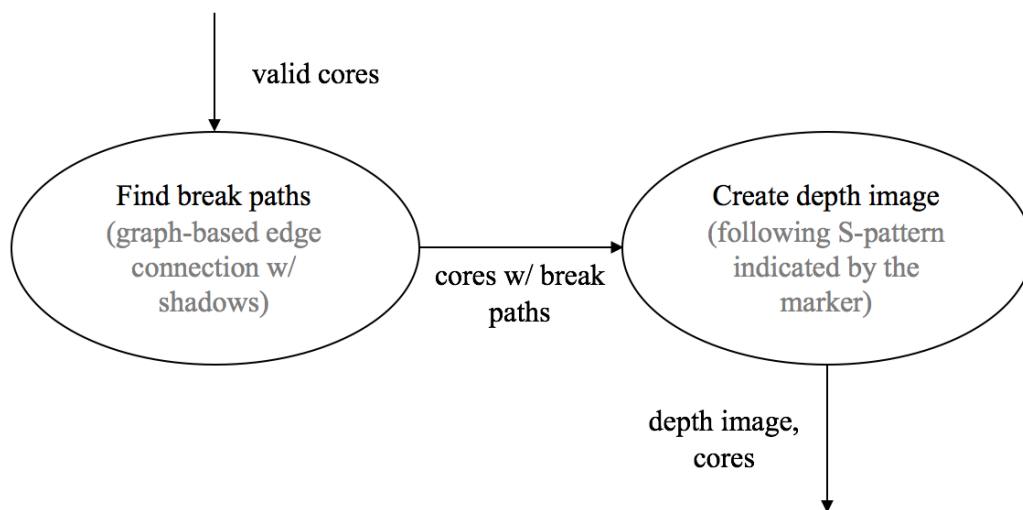


Figure 47. Level-2 data flow diagram for 'create depth image' process

Finding Break Paths

Fracture paths are the paths that show where cylindrical part of the core ends and its fractured region starts. The fracture paths will be used to decide if two adjacent cores are mechanically-cracked, which should be included in RQD calculation as intact core (Deere et al., 1967).

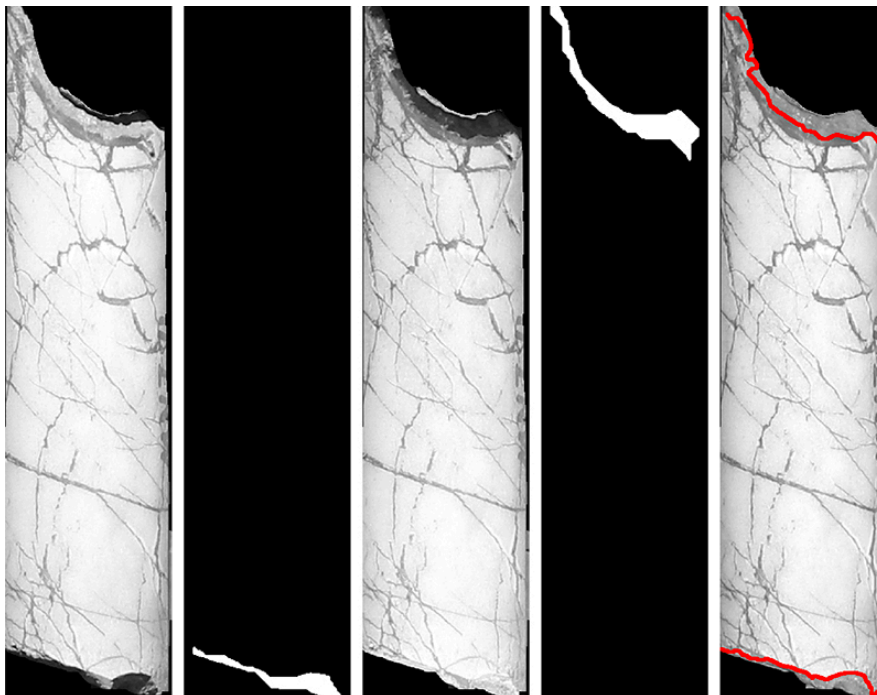


Figure 48. Fracture paths

The paths are detected using the shadow masks. For each detected core, related shadows are retrieved from the left and the right shadow masks. When a core piece is illuminated from its left side, shadows are created on its right side. Likewise, when the light source is placed on the right side of the core, the left side of the core will be under shadows. This can be seen in Figure 48. In this figure, from left to right: 1) core illuminated from its left (top), 2) shadows detected in 1), 3) core illuminated from its right (bottom), 4) shadows detected in 3), 5) shadow-free core image. Detected fracture paths are shown in red.

After the shadows are detected, boundaries of the shadow regions are retrieved. The boundaries are trimmed so that only the points closest to the core center remain. Next, a contour connection graph is created by using the edges detected by Canny edge detector (Canny, 1986). To find the fracture path at the top side of the core, edges of the image of the core illuminated from its bottom are detected, because the fracture path is more obvious in that image due to shadows. In the same way, the core image illuminated from its top side is used to detect edges for the fracture path at the bottom. After the edges are detected, a contour connection graph is created for both edge masks. Then, for each edge that overlaps the found shadow boundaries, the shortest path that connects the left-most end and the right-most end of the image is found. Finally, the best path is found by looking at how many edges passing through the shadow boundary a path contains. For example, if a path contains 20 edges that pass through the shadow boundary while another does 10, the first one is chosen as the best path. When there are multiple paths containing the same number of edges passing through the shadow boundary, the path with the smallest distance is selected as the best one. In the cases where there are no shadows, the boundary of the core mask is considered as the fracture path.

Creating the Depth Image

Depth image is an image showing the cores as they are retrieved from the drilling rod, before they are put into the core box.

First, the 5-cm pink marker's position is retrieved by color thresholding. The marker's position indicates the position of the core retrieved from the shallowest depth

compared to the other cores inside the core box. Since the placement of the cores in the box follow a snake-like pattern with respect to their depth, which rows store the cores upside-down can be detected using the marker's position.

Figure 49 shows the orientations of each row.

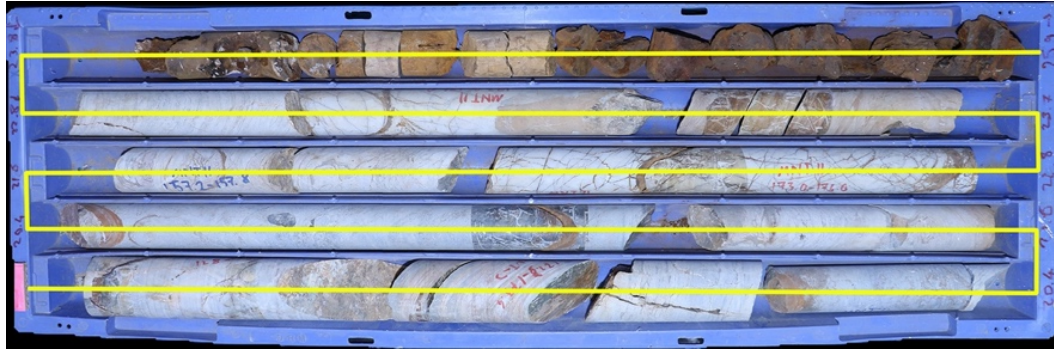


Figure 49. Row orientation

The depth gets deeper along yellow snake-like path. The pink marker is placed at the beginning of the core having the shallowest depth.

Finally, the cores placed upside down are flipped in XY, and then each core is placed to their right depth in the depth image.

Depth images of the two core boxes can be observed in Figure 50 and Figure 51. In these figures, the depth increases from top to bottom, and left to right. Red vertical lines show the centerline of each core. Green vertical lines indicate that the core is included in RQD calculation. Red paths on each end of the cores indicate the break paths.

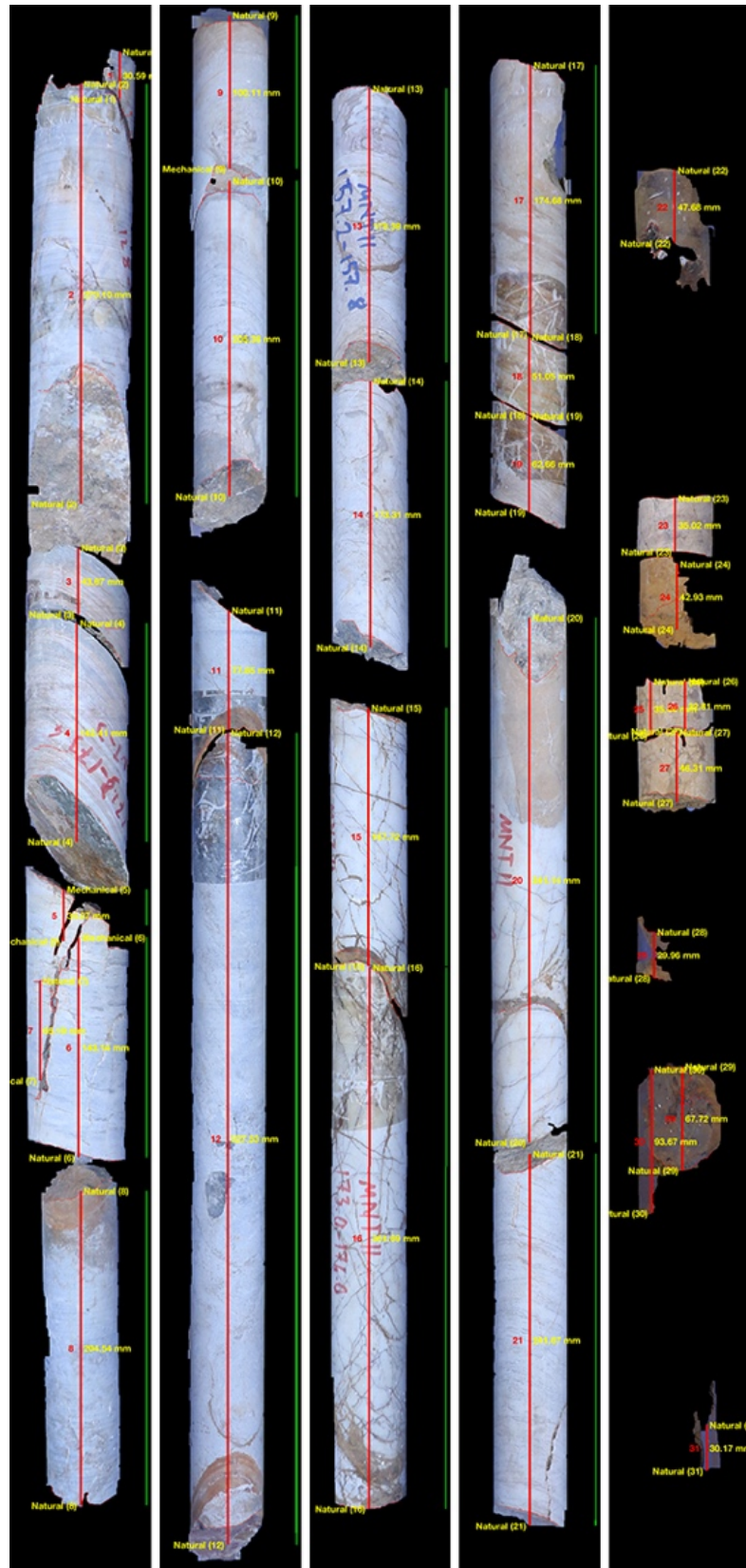


Figure 50. Depth image output for test core box 1.

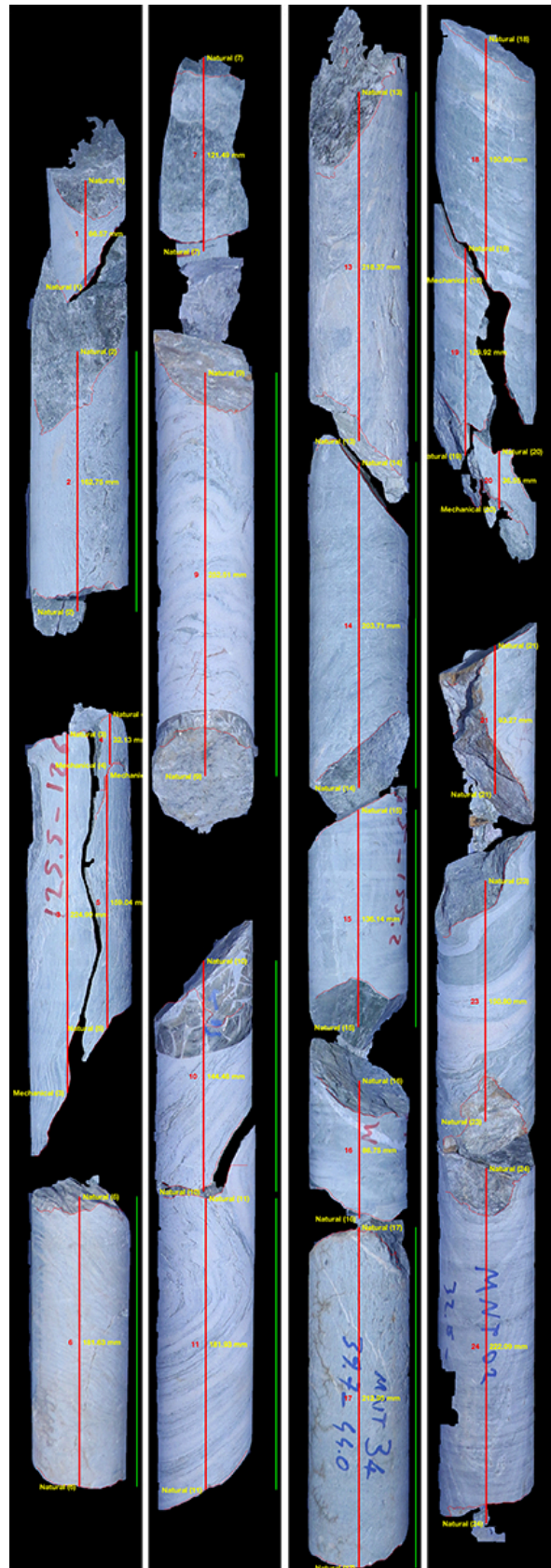


Figure 51. Depth image output for test core box 2.

3.2.5 RQD Calculation

After the depth image is created, RQD is calculated. Figure 52 shows the data flow diagram for RQD calculation process.

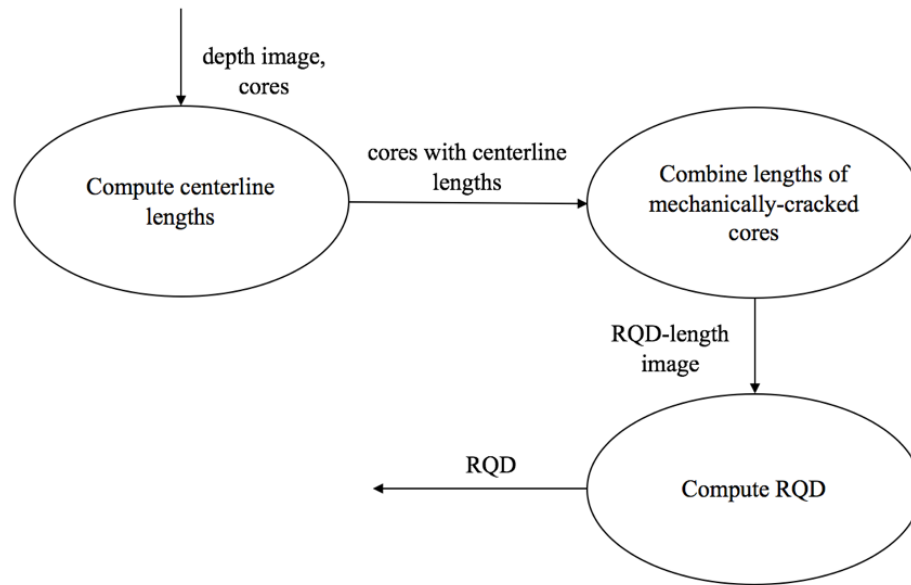


Figure 52. Level-2 data flow diagram for 'compute RQD' process

In this study, total core run length is taken as the length of a row. To find the lengths of the core pieces, first, their centerline length is calculated. Next, fracture types of the two ends of each core is decided. Finally, mechanically-cracked cores are detected and the centerline lengths of the mechanically-cracked core pieces are added to each other by excluding the centerline parts overlapping the non-cylindrical parts created due to the mechanical crack. When calculating the centerline length, while length of the cylindrical part of the core is included fully, only half of the length of the centerline parts overlapping non-cylindrical parts of the cores are included.

Finding Centerline Lengths

Centerline of a core is the line that passes through the center of the core, parallel to long axis of the rows of the core box. When calculating the length of the centerline, in addition to the cylindrical part of the core, the non-cylindrical parts are considered as well. Cylindrical and non-cylindrical parts of a core are found by utilizing the shadow masks. Figure 53 shows a core mask with the detected centerline. In this figure, blue

line shows the part of the centerline overlapping the cylindrical part of the core, while the red line shows the part that overlaps the non-cylindrical part at the bottom. Core length is calculated by adding half of the red line's length to the blue line's length. Green paths are the fracture paths.



Figure 53. Detected centerline of a core

Fracture Characterization

Fractures are characterized as mechanical and natural by using Olson et al. (2015)'s fracture characterization algorithm. The algorithm examines the path in two ways. The first one is to fit a line to the path and measure its angle with respect to the long axis of the core. The second way is to examine the roughness of the path. The roughness is handled by using two different methods. One of them is to measure the residuals – the normal difference between the points on the path and the best-fit line. The other one is to compute the angle between each path point and the core axis. In the end, there are total of three decisions made by three different measurements. In all decisions, the most frequent one is selected as the type of the fracture path.

Detecting and Combining Mechanically-Cracked Cores

After fracture type of the top and bottom parts of the cores are determined, adjacent cores are examined to find mechanically-cracked cores. When the fracture types of the

adjacent sides of two adjacent cores are both mechanical, these two cores are considered as mechanically-cracked. In the cases where there are mechanical cracks, centerline lengths of the cores must be recalculated by removing centerline parts that overlap the non-cylindrical parts formed due to the man-made crack.

Computing RQD

After collecting all of the necessary data, RQD is calculated using the following equation (Deere et al., 1967):

$$RQD = \frac{\sum(\text{Length of sound core pieces} > 100 \text{ mm})}{\text{Total core run length}}$$

3.2.5.1 RQD Results

RQD values for the two test core boxes are computed by assuming that total core run length is equal to length of a row, which is 1 meter for both cases. The measurements below are in millimeters.

RQD values computed by the algorithm for each row of core box 1 are:

$$RQD_{row1}^1 \frac{273.1 + 142.41 + 34.97 + 143.14 + 204.54}{1000} = 79.82\%$$

$$RQD_{row2}^1 \frac{100.11 + 205.38 + 527.53}{1000} = 83.3\%$$

$$RQD_{row3}^1 \frac{178.38 + 173.31 + 167.72 + 351.69}{1000} = 87.11\%$$

$$RQD_{row4}^1 \frac{174.68 + 341.14 + 241.67}{1000} = 75.74\%$$

$$RQD_{row5}^1 \frac{0}{1000} = 0\%$$

RQD values computed manually for each row of core box 1 are:

$$RQD_{row1}^1 \frac{270 + 142 + 165 + 206}{1000} = 78.3\%$$

$$RQD_{row2}^1 \frac{100 + 190 + 530}{1000} = 82\%$$

$$RQD_{row3}^1 \frac{165 + 170 + 160 + 350}{1000} = 84.5\%$$

$$RQD_{row4}^1 \frac{175 + 340 + 230}{1000} = 74.5\%$$

$$RQD_{row5}^1 \frac{0}{1000} = 0\%$$

RQD values computed by the algorithm for each row of core box 2 are:

$$RQD_{row1}^2 \frac{162.75 + 181.63}{1000} = 34.44\%$$

$$RQD_{row2}^2 \frac{252.51 + 144.48 + 181.93}{1000} = 57.89\%$$

$$RQD_{row3}^2 \frac{218.37 + 203.71 + 136.14 + 212.95}{1000} = 77.11\%$$

$$RQD_{row4}^2 \frac{150.9 + 150.8 + 222.09}{1000} = 52.38\%$$

RQD values computed manually for each row of core box 2 are:

$$RQD_{row1}^2 \frac{180 + 180}{1000} = 36\%$$

$$RQD_{row2}^2 \frac{260 + 130 + 180}{1000} = 57\%$$

$$RQD_{row3}^2 \frac{225 + 200 + 125 + 200}{1000} = 75\%$$

$$RQD_{row4}^2 \frac{140 + 165 + 210}{1000} = 51.5\%$$

Comparison of the RQD values computed by the algorithm and by manual logging can be observed in Table 8.

Table 8. Comparison of RQD values computed manually and algorithmically.

	Core Box 1			Core Box 2		
	Manual	Algorithm	Error (%)	Manual	Algorithm	Error (%)
Row 1	78.3	79.82	1.94	36	34.44	-4.33
Row 2	82	83.3	1.59	57	57.89	1.56
Row 3	84.5	87.11	3.09	75	77.11	2.81
Row 4	74.5	75.74	1.66	51.5	52.38	1.71
Row 5	0	0	0.00	-	-	-

Considering manual measurement correct, the algorithm computes RQD within an error margin of 5% as it can be seen in Table 8.

CHAPTER 4

CONCLUSIONS AND RECOMMENDATIONS

The objective of this thesis is to introduce a shadow-based approach to automate RQD calculation using digital image processing and analysis methods.

Main conclusions from this thesis are as follows:

- i. The study has been conducted using two different core boxes with 4 and 5 rows, having 4 and 5 meters of core respectively.
- ii. The algorithm successfully detects the core box, its rows, cores inside each row, and each core's cylindrical/non-cylindrical parts, mechanical and natural fractures. In addition, it can create a depth image.
- iii. The algorithm calculates RQD with an error rate of around 5%.
- iv. The algorithm takes 90 to 120 seconds (with 2.5 GHz Intel Core i7 processor, 16 GB 1600 MHz DDR3 RAM) to complete depending on the number of cores, shadow areas and edges in the image, and available processing power.
- v. Since shadows are utilized, edges created by the patterns on the cores do not cause a misbehavior in separation of touching cores.
- vi. Because creation of the shadows requires only a light source, the automation can be done at a low cost.

Although the method works well for most of the time, some variables might affect the segmentation negatively and some improvements can be made.

- i. The method cannot properly differentiate two cores whose ends fit each other perfectly such that no shadow is created.

- ii. Rows with dirt causes a change in core box color. Since the method separates the box from the cores by using color-thresholding, the dirt is considered as core as well. Although the dirty areas will be eliminated because of their size and shape, they might be considered as a part of a core, which causes inaccuracy in the final results.
- iii. The light sources causing the creation of shadows should be placed such that the illumination creates enough shadows to properly segment the touching cores as well as to decide where the non-cylindrical parts of the cores are. In addition, it is important that the light is diffused so that it does not create a light burst. Hence, lighting has a vital role for the method to succeed, which requires a controlled lighting environment that might be hard to create.
- iv. Cores having deep wears on their cylindrical parts which causes shadow creation may mislead the segmentation, if these shadows are not cleared properly.
- v. Too dark cores might be considered as cracks because the cracks are detected by thresholding the gray versions of the photographs.
- vi. A machine learning approach might be more suitable for fracture characterization.
- vii. Camera calibration might increase the accuracy of the measurements.

For the algorithm to work properly, the following conditions are recommended to be provided:

- i. The core box should be blue. If blue is not an option, then a color which is distinct from any color that can be seen on a core should be selected, and the algorithm should be modified so that it will consider the selected color as the box color.
- ii. The color of the core box should be different from the colors of the cores in the core box.
- iii. The core box should be free of dirt and dust. It should not contain anything other than the cores themselves.

- iv. Before taking the photographs, the cores that are too close to each other such that no shadow can be created between them should be separated from each other to allow shadow creation.
- v. The cores should not be marked.
- vi. The cores should be cleaned such that no unwanted shadows are created on them.
- vii. The cores should not be sprayed with water because it causes unwanted glitters to occur.

For the future work, more core boxes containing cores of various rock types are suggested to be evaluated.

REFERENCES

- Barton, N., Lien, R., & Lunde, J. (1974). Engineering classification of rock masses for the design of tunnel support. *Rock mechanics*, 6(4), 189-236.
- Bieniawski, Z. T. (1973). Engineering classification of jointed rock masses. *Civil Engineer in South Africa*, 15(12).
- Bieniawski, Z. T. (1976). Rock mass classification in rock engineering. *Exploration for rock engineering, proc. of the symp.*, (ed. Z.T. Bieniawski) 1, 97-106. Cape Town, Balkema.
- Bieniawski, Z. T. (1989). *Engineering rock mass classifications: a complete manual for engineers and geologists in mining, civil, and petroleum engineering*. John Wiley & Sons.
- Bondy, J. A., & Murty, U. S. R. (1976). *Graph Theory with Applications*. New York, NY, USA: Elsevier Science Publishing Co., Inc.
- Brown, E. T. (1981). *Rock characterization, testing & monitoring: ISRM suggested methods*.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Clifford, S. (2009). *Introduction to Algorithms (Third Edition)*. Cambridge, Massachusetts, USA: The MIT Press.
- Deere, D. U., & Deere, D. W. (1989). *Rock Quality Designation (RQD) After Twenty Years*. Springfield: U.S. Department of Commerce.
- Deere, D.U., Hendron, A.J., Patton, F.D., Cording, E.J. 1967. "Design of surface and near surface construction in rock", *Failure and breakage of rock*, proc. 8th U.S. symp. rock mech., C. Fairhurst, 237-302.
- Derpanis, K. G. (2010, May 13). Overview of the RANSAC Algorithm.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische matematik*, 1(1), 269-271.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395.

Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd Edition). Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Grassmann, H. (1853). Zur theorie der farbenmischung. *Annalen der Physik*, 165(5), 69-84.

Hough, P. V. C. (1962). Method and means for recognizing complex patterns (No. US 3069654).

ISRM (1978). The ISRM Commission on Standardization of Laboratory and Field Tests. *Int. J. Rock Mech. Min. Sci. & Geomech. Abst.* Vol. 15. Pp. 319-368. Pergamon Press Ltd.

İlsever, M., & Unsalan, C. (2012). Two-dimensional change detection methods: Remote sensing applications. Springer Science & Business Media.

Koschan, A., & Abidi, M. (2008). Digital Color Image Processing. Hoboken, New Jersey, USA: John Wiley & Sons, Inc.

Lemy, F., Hadjigeorgiou, J., Côté, P., & Maldague, X. (2001). Image analysis of drill core. *Mining Technology*, 110(3), 172-177.

Marinos, P., & Hoek, E. (2000). GSI: a geologically friendly tool for rock mass strength estimation. In *ISRM International Symposium*. International Society for Rock Mechanics.

Olson, L., Samson, C., & McKinnon, S. D. (2015). 3-D laser imaging of drill core for fracture detection and Rock Quality Designation. *International Journal of Rock Mechanics and Mining Sciences*, 73, 156-164.

Radke, R. J. (2013). Computer Vision for Visual Effects. New York, NY, USA: Cambridge University Press.

Steger, C. (1998). An unbiased detector of curvilinear structures. *IEEE Transactions on pattern analysis and machine intelligence*, 20(2), 113-125.

The MathWorks. (2017). MATLAB and Statistics and Image Processing Toolbox Release 2017a, Natick, Massachusetts, United States.

APPENDIX



Figure 54. Camera view



Figure 55. Inside of the tent with all lights turned on and core box outside



Figure 56. Tent with core box outside



Figure 57. Ends of the rail of core box tray

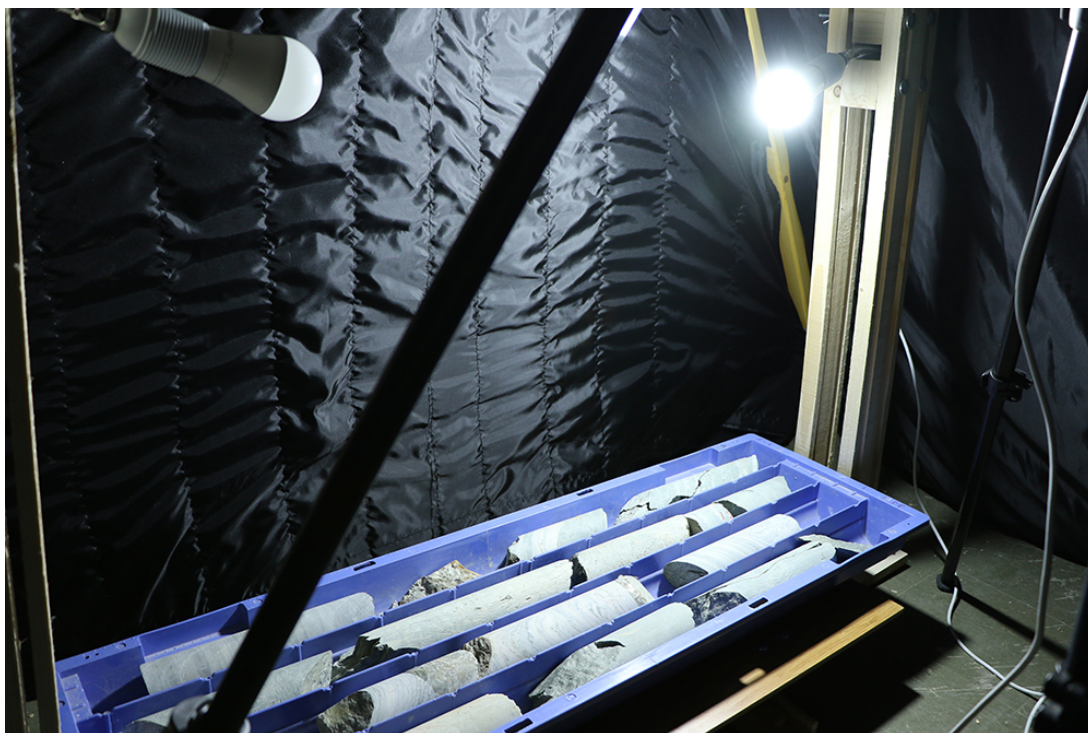


Figure 58. Inside of the tent. Right light is turned on. The core box is inside