

**AUTOMATED IMAGE ANALYSIS FOR PETROGRAPHIC
IMAGE ASSESSMENTS**

CENTRE FOR NEWFOUNDLAND STUDIES

**TOTAL OF 10 PAGES ONLY
MAY BE XEROXED**

(Without Author's Permission)

XIANGHONG ZHAO



INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

Our file *Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-62444-7

Canada

Automated Image Analysis for Petrographic Image Assessments

By

©Xianghong Zhao, B.Eng

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment
of the Requirements of the Degree of
Master of Engineering

Faculty of Engineering and Applied Science
Memorial University of Newfoundland
November, 2000

St. John's

Newfoundland

Canada

Abstract

In this thesis, the algorithms developed for an automated image analysis toolkit called PetrograFX for petrographic image assessments, particularly thin section images, are presented. These algorithms perform two main functions, porosity determination and quartz grain measurements. For porosity determination, the pore space is segmented using a seeded region growing scheme in color space where the seeds are generated automatically based on the absolute R – B differential image. The porosity is then derived by pixel-counting to identify the pore space regions. For quartz grain measurements, adaptive thresholding is applied to make the system robust to the color variations in the entire image for the segmentation of the quartz grains. Median filtering and blob analysis are used to remove lines of fluid inclusions, which appear as black speckles and spots, on the quartz grains before the subsequent measurement operations are performed. The distance transformation and watershed transformation are then performed to separate connected objects. A modified watershed transformation is developed to eliminate false watersheds based on the physical nature of quartz grains. Finally, the grain are characterized in terms of NSD, which is the nominal sectional diameter, NSD distribution and sorting.

Acknowledgement

I am deeply indebted to my supervisor, Dr. Raymond G. Gosine, without whose encouragement, advice, assistance and financial support, this thesis could not have been completed. He has been very generous with his ideas and time. I also thank him for his kindness and patience throughout my studying.

I am grateful to C-CORE in particular for providing me with a friendly environment and the necessary facilities to complete my programme.

I especially thank Ms. Patricia LeFeuvre who helped me developing the automatic seed input algorithms and suggested useful implementation methods related to Matrox Imaging Library, Dr. Peter McGuire, Dr. George Mann and Ms. Stephanie Davis who developed the GUI for PetrograFX and other members of the Intelligent Systems at C-CORE for the valuable discussions, suggestions and comments.

I thank Dr. Iain Sinclair who provided significant helps related to the understanding of reservoir rocks, and manually segmented thin section images.

I also thank the Faculty of Engineering and Applied Science and the School of Graduate Studies at Memorial University of Newfoundland for the Teaching Assistantships, Fellowships and support.

Finally, I would like to sincerely thank my husband, Dr. Yongxin Zhou, and my daughter, Lily, for their understanding and support.

Table of Contents

| | |
|---|------------|
| Abstract..... | ii |
| Acknowledgement..... | iii |
| Table of Contents..... | iv |
| List of Figures..... | vi |
| Chapter 1 Introduction..... | 1 |
| 1.1 The Reservoir Development in the Petroleum Industry..... | 1 |
| 1.2 Manual Thin Section Analysis..... | 2 |
| 1.3 Objective..... | 3 |
| 1.4 The Automated Image Analysis System..... | 3 |
| Chapter 2 Background..... | 6 |
| 2.1 Segmentation Techniques..... | 6 |
| 2.1.1 Thresholding..... | 7 |
| 2.1.2 Edge-based Segmentation..... | 11 |
| 2.1.3 Region-oriented Segmentation..... | 12 |
| 2.1.4 Texture Segmentation..... | 14 |
| 2.1.5 Watershed Segmentation..... | 17 |
| 2.1.6 Color Image Segmentation..... | 21 |
| 2.1.7 Conclusions..... | 22 |
| 2.2 Shape Representation and Description..... | 23 |
| 2.2.1 Contour-based Shape Analysis..... | 24 |
| 2.2.2 Region-based Shape Analysis..... | 28 |
| 2.2.3 Conclusions..... | 30 |
| 2.3 Image Preprocessing..... | 31 |
| 2.3.1 Neighborhood Averaging..... | 31 |
| 2.3.2 Median Filtering..... | 31 |
| 2.4 Blob Analysis..... | 32 |
| 2.5 Conclusion..... | 32 |
| Chapter 3 Pore Space Segmentation: Algorithm Overview and Feature Extraction | 34 |
| 3.1 Algorithm Overview..... | 35 |
| 3.1.1 Segmentation Scheme Selection..... | 36 |
| 3.1.2 System Flow Chart..... | 37 |
| 3.1.3 Color Space Selection..... | 39 |
| 3.2 Feature Selection and Extraction..... | 40 |
| 3.2.1 Criteria for the Feature Selection and Extraction..... | 41 |
| 3.2.2 Feature Extraction: R – B Difference Image..... | 42 |
| Chapter 4 Porosity Determination Based on the Pore Space Segmentation..... | 47 |
| 4.1 Feature Image Thresholding..... | 47 |

| | | |
|------------------------|--|------------|
| 4.1.1 | Predefined Thresholds..... | 48 |
| 4.1.2 | Automatic Thresholds Selection..... | 48 |
| 4.1.3 | Feature Image Thresholding..... | 49 |
| 4.2 | Seeds Selection for Region Growing..... | 50 |
| 4.2.1 | Blob Generation..... | 51 |
| 4.2.2 | Seed Extraction from Blobs..... | 51 |
| 4.3 | Seeded Region Growing in Color Space..... | 54 |
| 4.3.1 | Region Growing by Pixel Aggregation in the Red Channel..... | 55 |
| 4.3.2 | Implementation of the Seeded Region Growing..... | 55 |
| 4.3.3 | Region Growing in the Color Space..... | 57 |
| 4.3.4 | Three-Step Seed Input..... | 59 |
| 4.4 | Porosity Derivation..... | 61 |
| 4.5 | Pore Space distribution..... | 61 |
| Chapter 5 | Quartz Grain Segmentation | 64 |
| 5.1 | Sandstone Classification | 64 |
| 5.2 | Quartz Grain Segmentation..... | 65 |
| 5.2.1 | Characteristics of Quartz Grains..... | 65 |
| 5.2.2 | System Flow Chart..... | 66 |
| 5.2.3 | Quartz Grain Obtaining..... | 67 |
| 5.2.4 | Median Filter for Speckles Removal..... | 70 |
| 5.2.5 | Blob Analysis for the Small Hole Filling..... | 73 |
| 5.3 | Watershed Transformation and Adjoining Quartz Grain Separation..... | 74 |
| Chapter 6 | Modified Watershed Transformation and Quartz Grain Measurements | 77 |
| 6.1 | Modified Watershed Transformation..... | 77 |
| 6.1.1 | Physical Nature of Quartz Grains..... | 78 |
| 6.1.2 | Error Analysis..... | 79 |
| 6.1.3 | Introduction to the Modified Watershed Transformation..... | 80 |
| 6.1.4 | Algorithms for Modified Watershed Transformation..... | 82 |
| 6.1.5 | The Line Drawing Algorithm..... | 83 |
| 6.1.6 | Special Cases in Implementation of the Line Drawing Algorithm..... | 86 |
| 6.2 | Quartz Grain Measurement..... | 88 |
| 6.2.1 | NSD of the Grain..... | 88 |
| 6.2.2 | NSD Distribution Histogram..... | 89 |
| 6.2.3 | Sorting..... | 89 |
| Chapter 7 | Results, Integration into Windows Tool and Discussions..... | 93 |
| 7.1 | Results and Discussion..... | 93 |
| 7.1.1 | Porosity Determination..... | 94 |
| 7.1.2 | Quartz Grain Measurements..... | 95 |
| 7.2 | Integration into Windows Tool..... | 97 |
| Chapter 8 | Conclusions and Recommendations..... | 109 |
| 8.1 | Conclusions..... | 109 |
| 8.2 | Recommendations..... | 110 |
| References..... | | 112 |
| Appendices..... | | 116 |

List of Figures

| | |
|---|--------|
| Figure 1.1 Core sample image..... | 1 |
| Figure 1.2 Thin section image..... | 1 |
| Figure 1.3 PetrograFX system diagram..... | 4 |
| Figure 2.1 Catchment basins and watershed lines..... | 18 |
| Figure 2.2 Geodesic influence zone of connected components B_1 inside set A..... | 20 |
| Figure 2.3 Illustration of the watershed segmentation..... | 21 |
| Figure 3.1 Original JPEG thin section image..... | 35 |
| Figure 3.2 Different original images..... | 36 |
| Figure 3.3 Flow chart of porosity determination..... | 38 |
| Figure 3.4 (a) Hue, (b) Saturation channels of the image shown in Figure 3.1..... | 41 |
| Figure 3.5 (a) Red, (b) Green and (c) Blue channels of Figure 3.1..... | 44 |
| Figure 3.6 ABS(R-B) differential Image..... | 45 |
| Figure 3.7 Differential Images of Original Images Shown in Figure 3.2..... | 46 |
| Figure 4.1 Thresholded Image of the Differential Image..... | 46 |
| Figure 4.2 (a) Binary Image After One time Opening Operation, (b) Three times Operation, (c) Six Times Operation, (d) six times opening operation..... | 52 |
| Figure 4.3 Illustration of the seeded region growing..... | 56 |
| Figure 4.4 (a) Thin section color image with clay, (b) the red channel..... | 58 |
| Figure 4.5 Comparison chart for visual porosity estimation..... | 62 |
| Figure 4.6 Segmented pore space..... | 63 |
| Figure 4.7 Pore space distribution..... | 63 |
| Figure 5.1 Thin section color image..... | 65 |
| Figure 5.2 Flow chart for quartz grain segmentation and measurement..... | 67 |
| Figure 5.3 (a) Red channel, (b) thresholded red channel..... | 68 |
| Figure 5.4 Decomposition of the original image..... | 69 |
| Figure 5.5 Binary image after adaptive thresholding..... | 70 |
| Figure 5.6 (a) Binary image after applying median filter. (b) All holes extracted from blobs in (a). (c) Small holes. (d) Binary image after small hole filling..... | 71-72 |
| Figure 5.7 Hole filling in blob analysis. (a) An original blob. (b) The blob after hole Filling..... | 74 |
| Figure 5.8 Watershed lines added quartz grains..... | 75 |
| Figure 5.9 Arrow marked false watersheds..... | 76 |
| Figure 6.1 Comparison chart for roundness..... | 79 |
| Figure 6.2 Illustration of the watershed line examination..... | 82 |
| Figure 6.3 A typical blob..... | 84 |
| Figure 6.4 An irregular grain..... | 87 |
| Figure 6.5 Resulting image after the watershed line examination..... | 88 |
| Figure 6.6 Quartz grain distribution histogram..... | 91 |
| Figure 6.7 Sorting images..... | 91 |
| Figure 7.1 X125 original and pore space resulting images..... | 98 |
| Figure 7.2 Resulting images for the pore space segmentation..... | 99-102 |

Figure 7.3 Defect resulting images for the pore space segmentation.....103-104
Figure 7.4 Auto and manually segmented quartz grains.....105-107
Figure 7.5 PetrograFX GUI window.....108

Chapter 1

Introduction

1.1 The Reservoir Development in the Petroleum Industry

For most of the last century, the oil and gas industry has relied on extracting geological information from sediment samples in order to make decisions about exploration and production. Mineralogy, pore geometry and grain size of petroleum reservoir rocks are essential information for reservoir scientists and engineers since these parameters will help to determine whether the project can make favorable economics. This information is gathered from the analysis of core samples as illustrated in Figure 1.1. It is important and desirable to access core samples, which are obtained from as far as 1.6 – 3.2 miles down in the earth. for determining mineral composition and porosity type. However, obtaining, storing and processing cores from petroleum reservoirs are expensive and difficult procedures.

Thin sections of core samples are one source of information derived from a core sample. These sections, as illustrated in Figure 1.2, are cut perpendicular to bedding. After digitization, the thin section image provides important information, such as the form and origin of porosity, grain orientation and packing, related to petroleum recovery. [Biederman, 1986]



Figure 1.1 Core Sample Image [Hibernia]

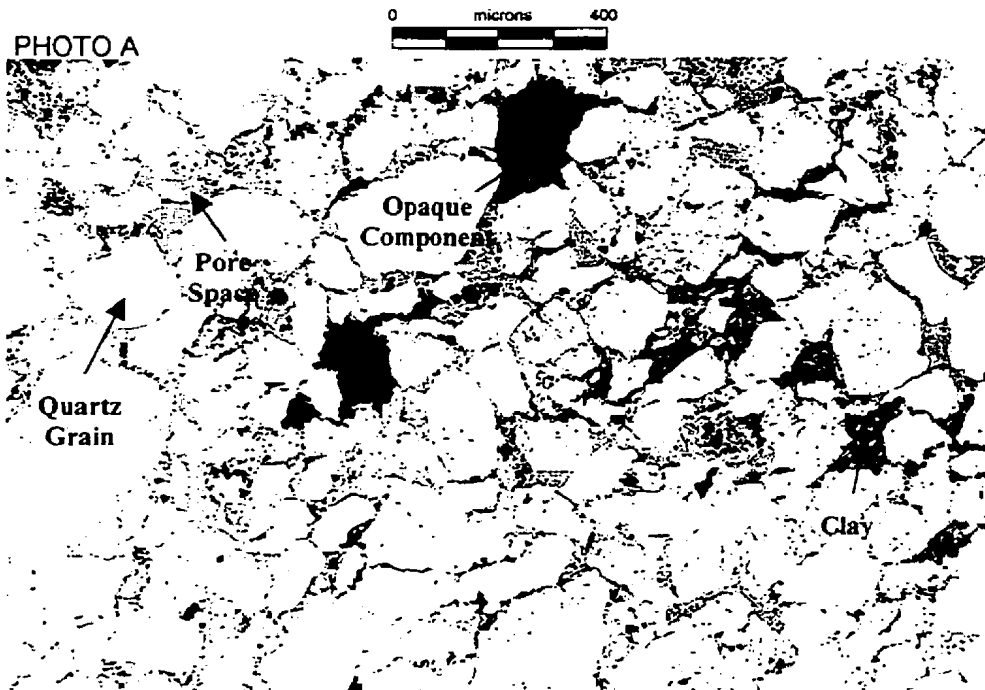


Figure 1.2 Thin Section Image

1.2 Manual Thin Section Analysis

In Figure 1.2, the blue areas correspond to the pore space. There are three types of minerals, quartz grain, opaque components and clay, in this image.

Porosity and grain size are two key parameters, which can be obtained by thin section analysis [Biederman, 1986 and Kennedy and Mazzullo, 1991]. A manual thin section analysis is a necessary but time consuming activity. Usually, conducting thin section analysis on a typical sequential analysis of a slide for containing 200-500 grains

by a highly qualified person can take more than two hours to complete. 7000 images are required by a single well, therefore, a complete manual assessment is not practical. [Gosine and Burden, 1999]

1.3 Objectives

The objective of this work is to develop the necessary pore space and quartz grain segmentation and measurement algorithms that will enable automated, fast and accurate thin section image analysis to replace or partially replace the human analysis. Developments in computer vision and image analysis techniques makes this possible. Kennedy and Mazzullo [1991] describe a grain-by-grain technique for determining grain size that involve using the length of one or two axes, perimeter, projection area and Nominal Sectional Diameter (NSD). NSD is defined as “the diameter of a circle with the same area as the maximum projection profile of a single particle”. Ehrlich [1991] introduces an objective classification method of porosity based on color images that are appropriate for pore identification at or very near the surface of the section. Ford [1998] is developing the image analysis software for a small test image in petrographic image analysis (PIA). In his work, the porosity was estimated using unsupervised classification methods in such small test images, and the software has not been released yet. According to Goins (2000), the new image analysis technologies “not only speeds traditional microscopical methods, such as point counting, but also improves accuracy”. An automated, fast and accurate thin section image analysis tool, however, is not available at present.

1.4 The Automated Image Analysis System

In this thesis, the work has been done as a part of a project called PetroGraFX. PetroGraFX is an automatic thin section image analysis toolkit developed in a collaboration with Hibernia Management and Development Company Ltd. where provides the thin section image database, the Centre for Earth Resources Research and C-CORE, Memorial University of Newfoundland. Figure 1.3 shows the system diagram for PetroGraFX. When an image is loaded from the image database into the PetrograFX window, the user can select from three functions: pore space identification, quartz grain size measurement or search for a selected structure in the entire image or in a database of images. The work in this thesis focuses on two key steps of the analysis: porosity determination and grain size measurement.

The first part of this thesis deals with the development of techniques to segment the pore space from other image components. Since pore space measurement is very important to the application, accuracy becomes the primary criterion for this step. In this thesis, a seeded region growing scheme is developed for pore space segmentation.

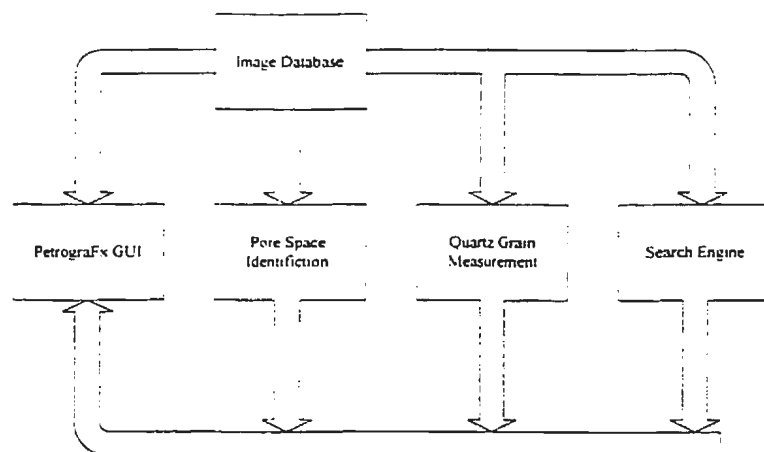


Figure 1.3 PetrograFX system diagram

The second part of this thesis details the measurement of the quartz grain size. In the first stage of this part, individual grains are identified using adaptive thresholding, median filter and blob analysis methods. A modified watershed transformation is then performed for disconnecting cemented quartz grains to form the individual grains. In the second stage, the grain sizes are measured in terms of the individual NSD, NSD distribution and sorting class for the whole image. The results between automatic mode and manual mode segmentation and measurement are compared and displayed.

Chapter 2 gives the background of image analysis and image processing techniques, while Chapter 3 presents an overview of the pore space segmentation, as well as describes the feature extraction based on the color space. Chapter 4 elaborates on the pore space segmentation and porosity derivation. Chapter 5 presents the individual quartz grain segmentation, and Chapter 6 presents a modified watershed for disconnecting cemented grains. Also, Chapter 6 shows the details for grain size measurement. Chapter 7 presents the results in a small set of test image and discusses the performances of the algorithm and system integration. Finally, the conclusions and recommendations are given in Chapter 8.

Chapter 2

Background

In this chapter, the fundamental techniques for image segmentation, shape representation and description, image preprocessing and blob analysis are reviewed. In addition, some related morphological operations related to the work are also described.

2.1 Segmentation Techniques

Image segmentation is one of the most important steps in translating raw data into information. There are a wide variety of image segmentation techniques including thresholding, edge-based, region-based, and morphological operations. The main purpose of segmentation is to partition an image into meaningful regions. Generally, segmentation algorithms are based on one of two basic properties of gray-level images: discontinuity or similarity. Edge-based segmentation belongs to the first category, and edge-based methods focus on the detection of isolated points, lines and edges in the image. The techniques in the second category include thresholding, region growing, and region splitting and merging. [Gonzalez and Wintz, 1987] Many different features can be used in edge detection and region growing, mainly gray level, color and texture.

2.1.1 Thresholding

Thresholding is one of the simplest and most frequently used methods for image segmentation. The thresholded image translates an intensity (or other images) into a binary image. The most important and fundamental step in this segmentation scheme is the selection of an appropriate threshold value to achieve the desired segments. Generally, threshold is defined as one or a range of gray-level value(s). Based on the gray-level of the image, the threshold can be selected:

$$\begin{aligned}g(i, j) &= 1 \quad \text{for } f(i, j) \geq T \\ &= 0 \quad \text{for } f(i, j) < T\end{aligned}\tag{2.1}$$

where T is the threshold, $g(i, j) = 1$ represents objects and $g(i, j) = 0$ is the background. [Sonka et al. 1998].

Thresholding may also be carried out on a color image by selecting a threshold based on the Euclidean distance between two colors or the difference between two parameters. When an image contains textural properties, it cannot be satisfactorily thresholded using single gray-level values. This will be discussed in later sections.

1. Global Threshold Selection Based on Histograms

If objects have unique gray level range in an image, the image histogram can be used for automatic threshold selection. A technique called the mode method utilized the valleys, also known as automodes, of the histogram for choosing appropriate thresholds. The procedure for automatic threshold selection includes histogram data smoothing, modes searching, and threshold selection between the modes that should meet the minimum segmentation error requirements. This technique analyzes the shape of the

histogram, the peaks and valleys corresponding to the regions of objects and the background. [Weszka, 1978]

2. Adaptive thresholding

Due to the non-uniform lighting, input devices and noise, which is introduced in the image generation, transmission, and receiving processes, the results of basic thresholding are rarely satisfactory and more sophisticated approaches, such as variable thresholds should be used. Variable threshold approaches utilize a function of the local features of the image. One approach is to divide the image into subimages and select thresholds locally [Sonka et al, 1998]:

$$T = T(f, f_c) \quad (2.2)$$

where f is the whole image, and f_c is a subimage, and T is the threshold. This method is particularly useful when there are gray-level variations across the entire image, although objects are distinguishable from the image background.

3. Thresholding based on local properties

Spatially local properties can be added to the threshold selection based on an analysis of the histogram. In other words, the histogram is computed locally. Instead of computing the local histogram for each pixel's neighborhood, the image is divided into exclusive blocks, and the local threshold is selected based on computing the histogram for each block [Haralick and Shapiro, 1985]

In [Haralick and Shapiro, 1985], it is also suggested that a histogram for pixels which have large Laplacian magnitude should be generated. These pixels represent the shoulder of the gray-level intensity function on each side of the boundary. Another

suggestion is to choose a threshold value which maximizes the sum of gradients over all pixels whose gray-level equals the threshold value.

4. *Minimum error thresholding*

Kittler and Illingworth [1986] propose a thresholding method in which thresholding is considered as a classification problem. First they assume the respective population is distributed normally with distinct means and standard deviations. Then the gray-level histogram is used to represent the population parameters

$$P_1 P(g|1) \begin{cases} > P_2 P(g|2) & g \leq \tau \\ < P_2 P(g|2) & g > \tau \end{cases} \quad (2.3)$$

where g is the gray-level value, and τ is the Bayes minimum error threshold.

Ramesh et al [1995] present two automatic threshold-selection schemes based on a functional approximation of the image histogram. One scheme is based on minimum square errors, while the other is determined by the minimum variance of the histogram. These two methods focus on the nonbimodal distribution images in which it is usually difficult to determine the threshold values. Instead of viewing the histogram as a mixture density function and treating the problem of thresholding as a classification problem, they view the histogram as a 1-D function and lead to multilevel thresholding.

5. *Entropic thresholding*

The use of entropic-based thresholding schemes has been proposed by Shannon. The Shannon Index is used in paleontology to assess species diversity, a form of entropy. His work is described by Beghdadi [1995]. The Shannon entropy is defined by

$$H = - \sum_{k=1}^{256} P_k \log P_k \quad (2.4)$$

where P_k is the probability of a certain gray level.

Beghdadi [1995] introduces some methods and gives a new approach for entropy thresholding. One method is to maximize the posteriori entropy of the gray-level in order to optimize the gray-level threshold using first order entropy. Since the histogram does not give any spatial information about the image, the problem of this method is that it can not deal with two different images when they have the same histogram. One solution for this problem is to use high order entropy, such as second-order entropy, while the other is to use a two-dimensional entropy for gray-level thresholding. Beghdadi exploits the spatial properties among neighbor pixels in an image without using high order entropy. The main idea is to find another symbol source which is not related to the gray-level distribution and can be used to obtain the first order entropy. The method defines a window that corresponds to the gray-level value of pixels enclosed; therefore, it is considered as another symbol source. In Beghdadi's work, two different approaches, fixed block and moving block, are proposed, although the moving block gives better results.

Brink [1995] gives a different solution for the spatial problem using spatial entropy. The idea is to define a bivariate entropy measure to exploit the interrelationships among neighbor pixels in an image. Consider a random variable $x(\mathbf{k})$ which is the gray-level of each pixel at $\mathbf{k} = (i, j)$, the spatial entropy is decided by the spatial distance (lag λ)

$$H(\lambda) = -\sum_{g=0}^{n-1} \sum_{g'=0}^{n-1} p_{gg'}(\lambda) \log p_{gg'}(\lambda) \geq 0 \quad (2.5)$$

$$p_{gg'} = \Pr\{x(k) \in g, x(k + \lambda) \in g'\}, g, g' = 0, \dots, n-1 \quad (2.6)$$

where $p_{gg'}$ is the probability of gray level g occurs a distance λ from a gray level g' .

2.1.2 Edge-based Segmentation

Segmentation algorithms based on the gray-level discontinuity focus on abrupt changes in gray-level, color or texture. Techniques that detect points, lines, and edges to achieve the segmentation goal belong to this category. Edge detection is the most common method for identifying meaningful image discontinuities. An edge is the boundary between an object and the background or the boundary between two overlapping objects. The edge information is very useful for segmentation since it can be used to obtain other image properties such as area and shape. Edges can be obtained by applying various edge detectors to the target image. Usually, this information cannot be used immediately as the segmentation result because it contains noises. Generally, edges are combined into edge chains, which correspond to the boundaries of objects in an image.

An edge usually occurs where the gray-level or other features have an abrupt change, and a derivative operator is sensitive to this change. An image function has two variables, the co-ordinates, in the image plane. This leads to describing the edges using partial derivatives. An edge is a vector variable with two properties, magnitude and direction. This operator is also called the gradient [Parker, 1997]. The Laplacian operator, Laplacian of Gaussian (LoG), difference of Gaussians (DoG), Sobel operator are popular tools for the edge detection. Canny [1986] gave an advanced approach to edge detection. There are three criteria defined in Canny's edge detector, no missing edges, the minimum distance between the actual edge and the detected edge and the minimum multiple edge response.

2.1.3 Region-Oriented Segmentation

Region-oriented segmentation techniques partition an image directly into regions [Gonzalez and Wirtz, 1987]. When an image is noisy, region-based segmentation is more suitable because the edge is difficult to detect in such an image. Generally, region growing schemes are based on homogeneity of regions. Many features can be used as criteria for homogeneity, such as gray-level, color, shape and texture. We denote R to represent the whole image region. R is partitioned into n subregions, R_1, R_2, \dots, R_n , such that

$$(a) \bigcup_{i=1}^n R_i = R.$$

(b) R_i is a connected region, $i = 1, 2, \dots, n$.

(c) $R_i \cap R_j = \Phi$, for all i and j , $i \neq j$.

(d) $P(R_i) = \text{True}$, for $i = 1, 2, \dots, n$.

(e) $P(R_i \cup R_j) = \text{False}$, for $i \neq j$. (2.7)

where $P(R_i)$ is a logical predicate defined over the points in set R_i . [Gonzalez, 1987]

1. Region merging and splitting

The simplest region merging method considers each pixel as a region, and merges regions according to some criteria. There are diverse approaches that vary in starting condition chosen and merge criteria selection. One method is to begin merging by dividing an image into small regions, for example, 2×2 or 4×4 , then merge regions based on the local histogram. Usually the result of segmentation depends on the order of starting subregions and merging order. [Sonka et al, 1998 and Gonzalez, 1987]

Hojjatoleslami and Kitter [1998] give a novel region growing approach. In their work, both the similarity and discontinuity measures are used for pixel aggregation. In addition, two discontinuity criteria, average contrast and peripheral contrast, are employed for controlling the region growing process.

Region splitting is the opposite operation of region merging. Considering the image as a whole region, then use the same criteria of homogeneity on the region merging. Let R represent the whole image and P is a predicate value. The simple approach is to subdivide the image into smaller quadrant regions when $P(R)$ is False. If P is False for any quadrant, we subdivide that quadrant into subquadrants, and so on. This is also called quadtree. [Sonka et al, 1998 and Gonzalez and Wirtz, 1987]

2. *Adaptive image region-growing*

Chang and Li [1994] present a different region-growing method, adaptive image region-growing. The main idea behind this method is to adjust the threshold value taking over the whole process of the region growing, rather than use a fixed threshold value. Generally, threshold selection is a difficult task, and even if the suitable threshold exist, it is hard to find. Chang and Li's algorithm uses position and time-varied thresholds that are dynamically and automatically computed in the region-growing process.

The region-growing segmentation commonly starts from dividing an image into many equal size subimages, called primitive regions. Then these primitive regions are merged to form larger regions if the merging criteria are satisfied. The repeat process will stop when no more merge happens. Most region-growing methods use a fixed threshold T

$$FLX(R_i \cup R_j) = \begin{cases} true & \text{if } |\bar{X} - \bar{Y}| < T, \\ false & \text{otherwise} \end{cases} \quad (2.8)$$

where R_i and R_j are two regions consist of n and m primitive regions, $R_i = \{X_1, \dots, X_n\}$ and $R_j = \{Y_1, \dots, Y_m\}$, \bar{X} and \bar{Y} are feature means of R_i and R_j .

In Chang and Li's work, they estimate a range based on the feature histogram of a region, called the adaptive range (l_1, l_2) . The central value is λ ($0 \leq \lambda \leq 1$), called the adaptive parameter. The adaptive regions

$$ADAP(R_i \cup R_j) = \begin{cases} true & \text{if } l_{j1} < \bar{X} < l_{j2} \text{ and } l_{i2} < \bar{Y} < l_{i1} \\ false & \text{otherwise.} \end{cases} \quad (2.9)$$

The results show that the adaptive methods are superior when compared comparing to the conventional region growing schemes.

3. Seeded region growing

The basic idea of seeded region growing is to aggregate pixels or subregions into a larger region starting from an individual pixel or a small region, known as a seed. Adams and Bischof [1994] give an approach for this technique to be applied to gray-level images. In their work, the seeds are input manually. First, they group n set of seeds, A_1, A_2, \dots, A_n . Seeds may be consist of single points or small regions to avoid embedded noise. Based on seeds, a tessellation is formed in the image. The tiles, areas that contain only one seed in each area, are formed in which pixels are all similar to the seed.

2.1.4 Texture Segmentation

Many images contain a variation in gray-level, known as texture. Texture is defined from one pixel to the next or within a small region, also called a texture primitive or texture element. Texture is widely used as a measure of the surface roughness. For textural images, segmentation can not be done accurately in gray-level space. Texture can

be used as a good feature in thresholding, edge detection, region-growing and other segmentation techniques for such images. Texture can be estimated and described in many ways.

1. Variance:

Variance is a measure of dispersion. It is related to the gray-level of the image.

$$\sigma^2 = \frac{\sum(x-m)^2}{N} \quad (2.10)$$

where σ is the variance, x is the gray-level value of a certain pixel, and m is the local mean value and N the number of the pixels in local area.

Variance responds to operator: the variation of pixels in the region.

2. Gradient

As described in the early section, gradient has two attributes, the magnitude and direction. We denote

$$G_x = \frac{\partial f}{\partial x}, \quad G_y = \frac{\partial f}{\partial y} \quad (2.11)$$

The magnitude of gradient can be approximated by absolute values: [Gonzalez and Wirtz, 1987]

$$G[f(x, y)] \approx |G_x| + |G_y| \quad (2.12)$$

and the direction of gradient:

$$\alpha[f(x, y)] = \text{atan}(G_y/G_x) \quad (2.13)$$

The first partial derivatives can be obtained by applying various operators, such as Sobel, involving pixels in a small local area.

3. *Gabor filter*

Gabor filter is a multi-channel filtering approach to the texture analysis. It is the product of a sinusoid with a Gaussian function, and becomes one of the most popular textural methods. Jain and Farrokhnia [1991] point out that multi-channel filtering can exploit the magnitudes and orientations for different textures, as well as the simple statistics of gray levels can be used. Clausi and Jernigan [Waterloo] used the Gabor filter to improve the performance of a common texture feature, the gray-level cooccurrence matrix (GLCM), in SAR imagery segmentation. Vision systems are tuned to recognize different orientations and frequency sensitivity. In human vision system there are three aspects are involved in texture identification: frequency, orientation, and complexity. Gabor filter is the output of a sinusoid with a Gaussian function and can be tuned to specific orientation (θ) and frequencies (F). A 2-D Gabor filter is defined as:

$$h(x, y) = g(x, y) \exp(2\pi j(Ux + Vy)) \quad (2.14)$$

where $g(x, y)$ is a two-dimensional Gaussian. $F = \sqrt{U^2 + V^2}$, $\theta = \text{atan}(U/V)$. The input signal to the Gabor filter should be pure sinusoid in order to obtain the optimal results. The only prior information for Gabor filter is the correct frequency. Since the textural properties cannot be expressed by pure sinusoids, in practice, multiple filters are used for texture capture.

4. *Discrete Cosine Transform (DCT)*

Discrete Cosine Transform (DCT) represents a local linear transform technique for texture segmentation. Ng, Tan and Kittler [1992] present a texture segmentation approach using the orthogonal masks derived from the DCT. A $N \times 1$ DCT basis vector u_m is denoted as

$$u_m(k) = \begin{cases} 1/\sqrt{N}; & m = 1 \\ \sqrt{\frac{2}{N}} \cos\left\{\frac{(2k-1)(m-1)\pi}{2N}\right\}; & m = 2, \dots, N. \end{cases} \quad (2.15)$$

The 2D filters of N^2 entities are generated by multiplying the column basis vectors with the row vector of identical length.

$$d_l(i, j) = u_m(i)u_n(j) \quad (2.16)$$

where $d_l(i, j)$ is the l -th entity in this filter bank, $l = m + (n-1)N$ for $1 \leq m, n \leq N$.

Randen and Husoy [1999] include this method in the comparative study for texture classification. The results show that the DCT filter gives good results in some types of textures.

2.1.5 Watershed Segmentation

Morphological segmentation methods are very useful for segmenting touching particles. Vincent [1991, 1993] present binary and grayscale image segmentation using the watershed transformation, as well as efficient algorithms for their implementation. Consider an image as a topographic surface with a gray scale of each pixel representing the height in a 3D landscape view. The lowest height corresponds to the pixel that has the lowest gray scale, while the highest height represents the pixel that has the highest gray scale. The sea and the lakes are region minima in landscapes, and regional minima in the image are pixels whose gray scales are lower than their neighbors. A whole landscape can be divided into regions that are called catchment basins, and watershed lines separate these catchment basins. Figure 2.1 shows the catchment basins and watershed lines. [Vincent and Soille, 1991]. In order to understand watershed transformation, it is necessary to introduce some basic definitions.

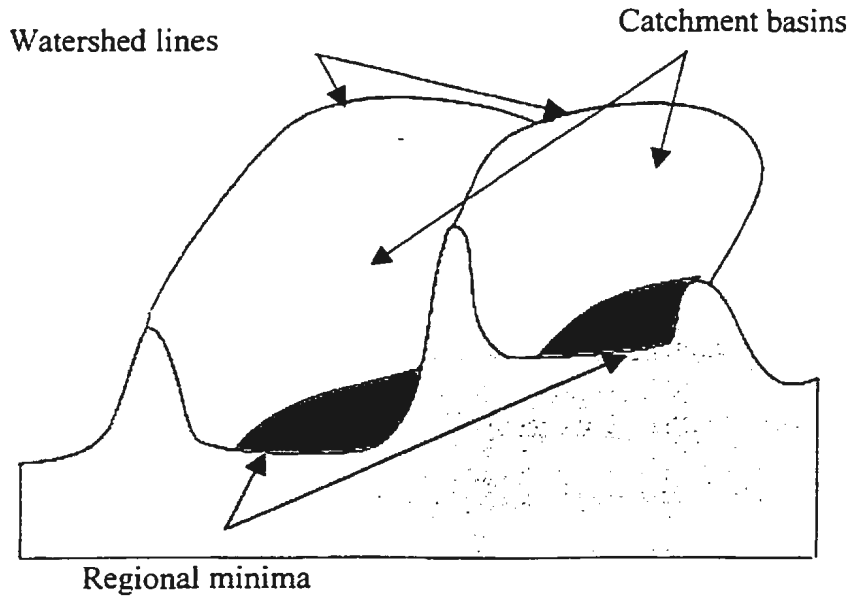


Figure 2.1 Catchment basins and watershed lines [Vincent and Soille, 1991]

1. Basic Definitions

The formal definitions of catchment basins and watersheds depend on the following concepts and descriptions.

The grayscale image I is defined to be a subset of Z^2 , where Z is the integer domain. We assume that the grayscale values for the image I are in a range of $[0, N]$, where N is an integer.

Let G denote the digital grid with any type, such as a square grid in four or eight connectivity. Then we can define a path P between two pixels p and q with a length l in the image I to be a $(l + 1)$ -vector of pixels (p_0, p_1, \dots, p_l) where $(p_{i-1}, p_i) \in G, \forall i \in [1, l]$.

Now we can define the minimum $M_I(h)$ of the image I at the gray level h to be the subset of I satisfying the following conditions:

for any p in $M_I(h)$ and any q not in $M_I(h)$ with $I_q \leq I_p$, where I_q and I_p are gray level values of p and q , and for any path $P = (p_0, p_1, \dots, p_l)$, where $p_0 = p$ and $p_l = q$, we can always find an $i \in [1, l]$ such that $I_{p_i} > I_p$.

From another perspective, a minimum is an region in an image whose gray level is darker than all neighbor pixels.

Let h_{\min} be the smallest gray level value of the image I , and h_{\max} be the maximal. For any h with $h_{\min} \leq h \leq h_{\max}$, we can define the threshold of I at gray level h .

$$T_h(I) = \{p \in D_I, I_p \leq h\} \quad (2.17)$$

where D_I is the domain of I in Z^2 .

Let A be a single connected region in I . The geodesic distance $d_A(x, y)$ between two pixels x and y in A is infimum of the length of the path between x and y inside A . By using this definition, we can define the geodesic influence zone. Figure 2.2 illustrates this definition.

Suppose B is the set of B_1, B_2, \dots, B_k which are several connected subregions in A . Then the geodesic influence zone $iz_A(B_i)$ of a connected component B_i of B in A is

$$iz_A(B_i) = \{p \in A \mid \forall j \in [1, k] / \{i\}, d_A(p, B_i) < d_A(p, B_j)\} \quad (2.18)$$

$$\text{The influence zone of } B \text{ in } A \text{ is defined by } IZ_A(B) = \bigcup_{i \in [1, k]} iz_A(B_i). \quad (2.19)$$

2. Catchment Basins and Watershed Lines

Finally, we can give the formal definitions of the catchment basin and watershed line.

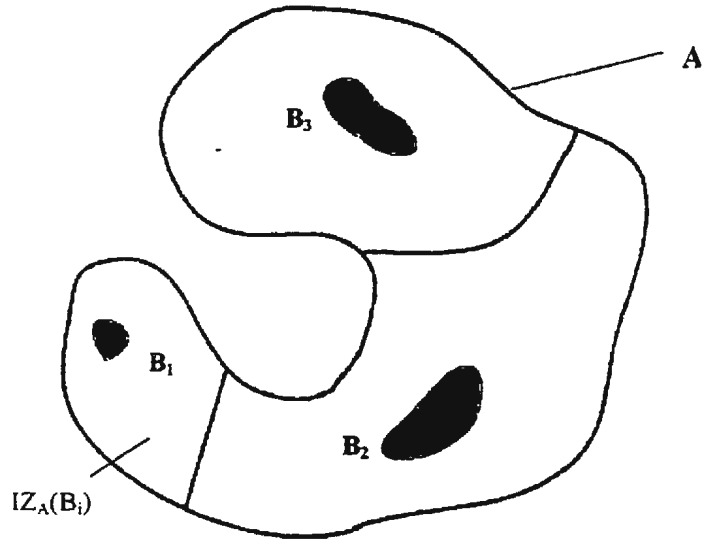


Figure 2.2 Geodesic influence zone of connected components B_1 inside set A . [Vincent and Soille, 1991]

For any $h \in [h_{\min}, h_{\max}]$ let X_h be a subset of D_I and define $X_{h_{\min}} = T_{h_{\min}}(I)$. Then we can recursively define all X_h by

$$X_{h+1} = M_I(h+1) \cup IZ_{T_{h+1}(I)} \quad (2.20)$$

Definition: The set of catchment basin of the image I is $X_{h_{\max}}$. Watersheds of I is $D_I / X_{h_{\max}}$, i.e. the complement of catchment basin in the domain D_I .

3. Image Segmentation Using Watersheds

The watershed transformation is a powerful tool for separating overlapping particles in order to measure them individually in a binary image. The first step is to mark all objects correctly using a marking function in which the catchment basins correspond to the individual objects. This function is determined by the distance function which is described in Section 2.3.2. The watershed transformation then is applied for

disconnecting adjoining objects in the image. [Vincent, 1993] Figure 2.3 shows this process.

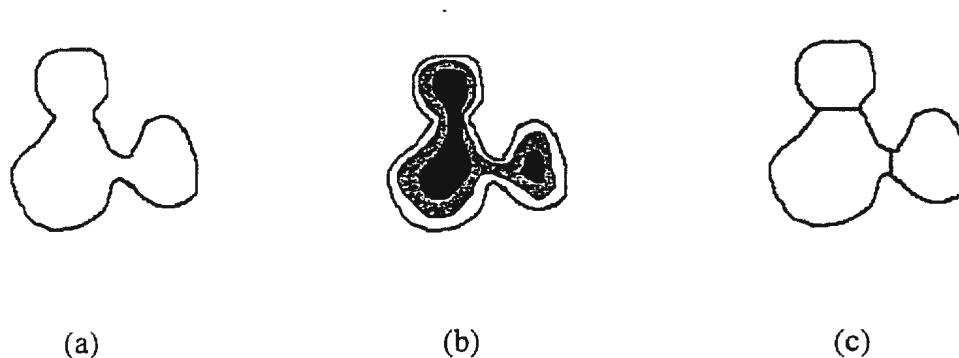


Figure 2.3 Illustration of the watershed segmentation. (a) A region contains connected objects. (b) after applying the $\frac{3}{4}$ chamfer distance transformation. (c) the watersheds added region.

2.1.6 Color Image Segmentation

In the literature, a large fraction of segmentation techniques deals only with monochrome images. Some of them can be extended to the color image. With the rapid technology development, there is an increased use of color images. It is important to understand the mechanisms of color vision [Sharma and Trussell, 1997]. Two basic types of methods are employed for the color segmentation: statistical and spatial methods. Schettini [1993] presents a color image segmentation algorithm. The procedure starts with color mapping. Since RGB is the basic color space, the color value is converted into a device-independent uniform color space to avoid RGB feature limitation. Second, clustering is employed to divide the image into many homogeneous regions. This is based on recursive 1-D histogram. The final step is to merge adjacent regions using color similarity and spatial proximity criteria.

Ohta, Kanade and Sakai [1980] propose a set of effective color features for color image segmentation. The eigenvectors of the covariance matrix of a whole image are calculated and found that the eigenvectors W_1 , W_2 and W_3 are about $\left(\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3}\right)'$, $\left(\frac{1}{2} \quad 0 \quad -\frac{1}{2}\right)'$, $\left(-\frac{1}{4} \quad \frac{1}{2} \quad -\frac{1}{4}\right)'$, respectively. They test eight different images, and W_1 , W_2 and W_3 are about the same for every image. They conclude that the three orthogonal color features, $I1 = (R + G + B)/3$, $I2 = (R - B)/2$, and $I3 = (2G - R - B)/4$ are useful features for the color representation.

2.1.7 Conclusions

This completes briefly review of the common segmentation techniques of thresholding, edge-based, region-oriented and texture. Threshold value selection is essential to thresholding methods, and statistical properties, such as image histogram, are important means for obtaining this value. The limitation of these statistical methods is that the spatial properties are lost after applying these methods. Although there are approaches for improving this limitation, such as adaptive methods, it is very difficult to achieve accurate segmentation results only using thresholding even for those simple images. Consequently, it is almost always used as combinations with other types of techniques.

In edge-based segmentation techniques, most of them are application dependent, and no general edge detector exists. Another significant aspect of this type of methods is the noise. It degrades the qualities of the edge operator outputs, and causes many false edges; therefore, edge-based methods are not appropriate for noisy images. The crucial

aspect in region-based methods is the choice of the merging or splitting criteria. Generally, region-oriented techniques exploit the internal properties of a region, and form the region directly. Compared to thresholding and edge-based methods, the speed of the performance for these techniques is relatively slow, and they also require more space to store the information, therefore, the segmentation accuracy is higher.

Morphological segmentation methods, such as watershed transformation, provides the best way to separate touching particles using topographics. The watershed transformation is usually combined with other operations, such as the distance transformation.

For many textural images, such as SAR and MRI, texture is a good feature for segmentation. There are many papers published in this area, and many new technologies, such as wavelet and Gabor functions, are employed to achieve desired results. The main disadvantage of this type of technique is that it is very difficult to obtain the precise object boundary. The reason is that each pixel is re-represented by a value generated by its local neighbors that reflect its texture property.

For color image segmentation, the results rely on not only the color reference system, but also the algorithms. The color space can be represented by some useful features, such as $(R - B) / 2$.

2.2 Shape Representation and Description

Object shape representation and description are important issues in image analysis. After an image is segmented into regions by various segmentation methods, these regions need to be represented and described for further image analysis. Usually,

the shape of the object is represented as a binary image, and this representation is non-numeric. Shape description methods generally define a feature vector corresponding to the given shape. There already exist a variety of shape analysis methods, and most methods are application-oriented. Generally, these methods are divided into two classes. One method uses boundary-based representation in which the external characteristics are utilized to represent the object, while the other chooses the internal characteristics to represent it. This is also known as the region-oriented representation or the global operation. Moreover, mathematical morphology provides a useful tool for shape-related analysis tasks in image processing.

2.2.1 Contour-Based Shape Analysis

Generally, contour-based methods are suitable for applications that focus on morphological features. Before describing contour-based methods, it is necessary to introduce some basic issues in mathematical morphology.

1. Morphological operations

Morphological operations are related to the shape of the object, and are used to identify objects or boundaries in an image. The three major morphological functions are erosion, dilation and hit-or-miss. [Burdick, 1997]. Since the input image is usually binary, we can simply apply the logical operations of Boolean algebra.

1). Binary erosion and dilation

Erosion and dilation are a pair of binary convolutions. They combine a local neighborhood of pixels with a mask. Dilation fills the holes and cracks, and makes the boundaries smoother. The following function performs the dilation. [Jahne, 1991]

$$G'_{mn} = \bigvee_{k=-R}^R \bigvee_{l=-R}^R M_{k,l} \wedge G_{m-k,n-l} \quad (2.21)$$

The \wedge and \vee denote the logical *and* and *or* operation, respectively. G is a binary image convolving with a symmetric mask M .

The dilation is defined as:

$$G \oplus M = \{p : M_p \cap G \neq \emptyset\} \quad (2.22)$$

where G is the set of all the pixels of the matrix which are not zero. M is the set of the non-zero mask pixels. M_p is the mask and shifts to the pixel p .

The erosion has the opposite effect. After erosion operation is performed, the objects are eroded. Those connected only by a small bridge will be disconnected. The following operation can perform erosion:

$$G'_{mn} = \bigvee_{k=-R}^R \bigvee_{l=-R}^R M_{k,l} \wedge \overline{G_{m-k,n-l}} \quad (2.23)$$

leaving erosion as:

$$G \ominus M = \{p : M_p \subseteq G\} \quad (2.24)$$

2). Opening and closing

By using basic erosion and dilation other operations may be performed on objects. The erosion operation can remove the small objects (usually noise), which isolated from the big objects. The main side effect of erosion is that all objects shrink in size. Usually, the dilation is used following the erosion to avoid this. This method is also known as an opening operation [Jahne, 1991]

$$G \circ M = (G \ominus M) \oplus M \quad (2.25)$$

Dilation followed by erosion is called a closing operation. Dilation fills small holds and smoothes the boundaries, but enlarges the objects. The following erosion operation can reverse this effect.

$$G \bullet M = (G \oplus M) \ominus M \quad (2.26)$$

The mathematical proofs of properties of erosion and dilation operations were given in Haralick et al [1987], where more properties were discussed, such as the inclusion relationship of sets holds under the opening and closing operations.

2. Polygonal approximations

Polygonal approximations are popular schemes in boundary approximation. In these schemes, the polygonal line is used to describe the shape boundary. The approximation criteria include the maximal internal polygon area, minimal external polygon area, minimal error and the minimal polygon perimeter.

The split-and-merge algorithm is one of the most often used methods. The main idea behind this approach is to split a curve into segments until the desired error criteria are reached. The area of a polygon can be computed by the following equation [Wu and Leou, 1993]:

$$AREA(P^{(n)}) = \frac{1}{2} \sum_{j=1}^n (x_j y_{j+1} - y_j x_{j+1}) \quad (2.27)$$

where $p^{(n)}$ is the initial polygon, n is the number of vertices and $x(x_i, y_i)$, $i = 1, 2, \dots, n$, are the vertex coordinates.

Wu and Leou [1993] developed new algorithms for the polygonal approximation schemes, internal maximum area polygonal approximation (IMAPA), External minimum area polygon approximation (EMAPA), and Minimum area deviation polygonal approximation (MADPA).

3. Fourier Descriptors

The Fourier descriptor is a very useful tool for the boundary-based methods. Since the algorithm is based on the well-developed Fourier theories and only uses the boundary to describe the object, this method is very fast and fairly efficient.

Zahn and Roskies [1972] present a method using Fourier descriptors for plane closed curves. For a closed curve γ with the length L , the cumulative angular function is denoted as $\Phi(l)$ where l is the arc length between the starting point and the point l . The normalized function in $[0, 2\pi]$ is defined as

$$\Phi^*(t) = \Phi\left(\frac{Lt}{2\pi}\right) + t \quad (2.28)$$

The Fourier series of $\Phi^*(t)$ is

$$\begin{aligned} \Phi^*(t) &= \mu_0 + \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt) \\ \text{or} & \\ \Phi^*(t) &= \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k) \end{aligned} \quad (2.29)$$

If the curve γ is polygonal with m vertices V_0, \dots, V_{m-1} and the edge length is Δl_i , then

$$\Phi^*(t) = \mu_0 + \sum_{n=1}^{\infty} (a_n \cos nt + b_n \sin nt) \quad (2.30)$$

where

$$\begin{aligned}
\mu_0 &= -\pi - \frac{1}{L} \sum_{k=1}^m l_k \Delta\Phi_k \\
a_n &= \frac{-1}{n\pi} \sum_{k=1}^m \Delta\Phi_k \sin \frac{2\pi n l_k}{L} \\
b_n &= \frac{-1}{n\pi} \sum_{k=1}^m \Delta\Phi_k \cos \frac{2\pi n l_k}{L} \\
l_k &= \sum_{i=1}^k \Delta l_i
\end{aligned} \tag{2.31}$$

Since all coefficients are determined by Δl_i and $\Delta\phi_i$, they are invariant under translations, rotation and scale.

2.2.2 Region-based Shape Analysis

While contour-based shape methods have the advantage of computational efficiency, the region-oriented techniques exploit internal properties, such as color and texture. Some simple region descriptors, such as region area and Euler's number, are suitable for small and simple region descriptions. The complex shape is usually segmented into a set of simple shapes; however, there are a number of approaches in this area.

1. Shape matrices

This method uses an $M \times N$ matrix to describe the region shape [Taza and Suen, 1989 and Goshtasby, 1985]. The 2-D shape is first converted into 1-D function, which is composed of radii. Then the maximum radius r is divided into $(n-1)$ equal lengths, and $n-1$ circles with the centroid O are drawn. The circles are divided into m equal arcs. Finally, all pixels that belong to the object and fall in the subarea (m, n) are assigned to form the shape matrix.

After obtaining the shape matrices of the objects, the shape descriptor is achieved. By comparing the shape matrices, the shape discrimination is also achieved. The two shapes can be compared in terms of the redundancy R , which is defined as follows:

$$R = \frac{\text{number of sample points}}{\text{number of pixels}} \quad (2.32)$$

In Taza and Suen's work, this algorithm is applied to alphabet letters, numerals, geometrical figures, and physical objects. The criterion used in this work is the similarity in percentage.

2. Distance transformation

Borgefors [1986] develops a different approach for shape description, named distance transformations (DT). In this method, all pixels in a binary image are divided into two classes, feature pixels and non-feature pixels. This image is converted into a grayscale image where every feature pixel takes value 0, and every non-feature pixel takes value according to the distance to its nearest non-feature pixel. This is also known as chamfer $\frac{3}{4}$ distance transformation. Since the globe operation is usually computationally expensive, this operation is restricted to the local neighborhood. In his work, 3×3 , 5×5 and 7×7 sizes are used and also the relationship between DT and the Euclidean distance transformation are discussed. The results showed that this algorithm has a maximal error of about 2%.

2.2.3 Conclusions

Generally speaking, external operations, also known as boundary (or contour)-based methods, are fast and efficient, such as polygonal approximation and Fourier

descriptors, because they require less data to represent the object. The main disadvantage is that these methods ignore all internal properties of the object and have difficulty describing the symmetrical shapes. Internal operations exploit the information inside the region, but they are generally computationally expensive.

Morphological operations, such as dilation, erosion, opening and closing, are often used before the shape representation and description. They are useful for small holes' removal and narrow bridges' disconnection.

In boundary-based techniques, polygonal approximation or piece-wise linear approximation is one of most popular methods in boundary approximation. Wu and Leou [1993], compared their algorithm to other existing methods by using the criterion of area deviation under the certain uniform error and the corresponding approximating lines. Furthermore, their methods are in both sequential and parallel manners instead of only sequential in order to increase the computational efficiency. Fourier descriptors are very useful tools for shape description because their high performance and efficiency. In addition, Fourier descriptors are very easy to implement.

For region-based methods, the most important criterion is the cost since these operations are global, and heavy computation is usually involved. Since almost all boundary-based methods are shape approximation schemes, i.e. some data are ignored, the region-based schemes are usually more accurate especially for the local properties.

2.3 Image Preprocessing

Image preprocessing is also called image enhancement. These operations include smoothing, sharpening images, or improving the contrast of images etc.. Image

enhancement does not increase the image information content in terms of entropy. This section gives two methods of image smoothing.

2.3.1 Neighborhood Averaging

Image smoothing belongs to the local pre-processing for the purposes of removing the noise in the image. A common method for the image smoothing is neighborhood averaging. One approach is to replace the gray level of each pixel with the average value of its neighbors

$$P^*_{x,y} = \frac{\sum_{i,j=-m}^{+m} W_{i,j} \cdot P_{x+1,y+j}}{\sum_{i,j=-m}^{+m} W_{i,j}} \quad (2.33)$$

where $W_{i,j}$ is the set of integer weights. This is also known as kernel operation. [Sonka et al. 1998]

2.3.2 Median Filtering

The main difficulty of the neighborhood averaging operation is that it blurs the edges that are crucial details in some applications. The median filter provides a different approach. Median filter replaced the gray level of each pixel by the median of the gray levels of its neighbors. The median m of a set values is determined such that half of the values in the set are less than m , while another half are greater than m . The advantage of median filter is that it attenuates noise, while the edge sharpness is preserved. [Sonka et al. 1998]

2.4 Blob Analysis

In an image, regions of connected pixels that are in the same logical pixel state are called blobs. Usually, the foreground that consists of these connected pixels is set to 255 or some values other than zero, and the background is set to zero. Blob analysis provides a tool for identifying and measuring these regions.

Matrox Imaging Library [1999] contains a comprehensive blob analysis package. By using this, one can count the total blob number, extract holes from blobs, reconstruct blobs according to some criteria, and measure the area, length and many other features of a blob. A typical blob analysis procedure is as follows:

- Load an image that is under good qualities to minimize the preprocessing.
- Reduce the noise in the image.
- Generate a blob image in which the foreground is non-zero and the background is zero.
- Select a feature list according to the application.
- Calculate the selected features and retrieve them from the result buffers.

2.5 Conclusion

This chapter provides a review of techniques for image segmentation, shape representation and description, image preprocessing and blob analysis.

The image preprocessing improves the image appearance and removes the noise. For various image smoothing methods, median filtering eliminates image noise and keeps the edge sharpness. So that it is appropriate for the thin section images processing since the shape analysis of such images requires accurate boundaries.

As the thin section image is relatively noisy especially for the quartz grain component, the region-based segmentation is more appropriate. Watershed transformation is a powerful tool for disconnecting adjoining objects, and ensures the individual mineral to be segmented. Matrox Imaging Library [1999] provides a large number of blob analysis functions and can be used for measuring the size of the mineral and pore space, fill holes in blobs and include or exclude blobs under certain conditions.

Chapter 3

Pore Space Segmentation: Algorithm Overview and Feature Extraction

One of the main tasks of PetroGraFX is to determine the porosity in a certain image or slide. Porosity calculated visually is also known as point-count porosity. In thin section analysis, this parameter is estimated as a percentage of the slide or photograph area corresponding to the pore space that was filled by a colored epoxy during initial thin section preparation. Therefore, the first stage of the porosity determination is the pore space segmentation. Based on the obtained regions of pores, the porosity can be derived by pixel-counting over such regions.

Figure 3.1 is an original 880×649 JPEG color image, which is chosen from a set of images from the Hibernia oilfield. The objects in the thin section image include pore space and minerals, such as quartz grains, opaque components, rock fragments and clay. In Figure 3.1, the pore space is dyed bright blue, so that it may be distinguished from rock and mineral fragments. Quartz grains form the dominant component in an image, and usually occupy at least 50% of the image area. They appear white in this test image set; opaque components and clay show black and dark brown, respectively.

3.1 Algorithm Overview

In the particular image shown in Figure 3.1, one can find much of pore area by using a thresholding method in a selected channel of the original image. In the test database, images vary, both within images and between images, in color of pore-filling epoxy and matrix minerals, as well as the lighting conditions and digitizing parameters. Figure 3.2 gives four different images chosen from the test image set.

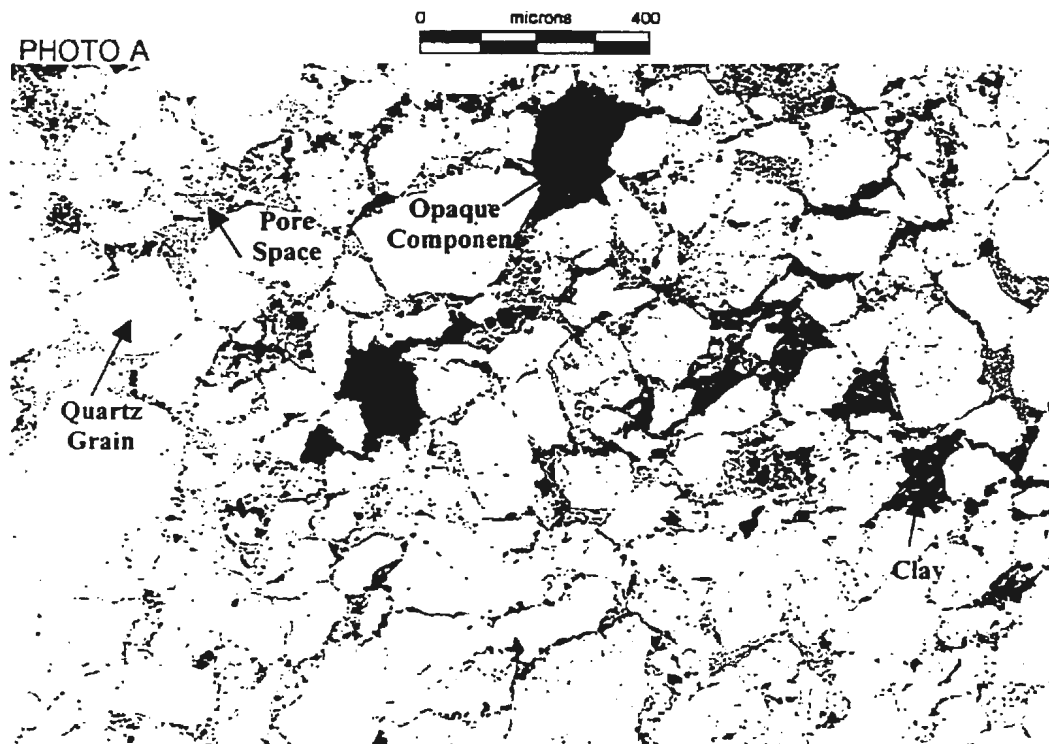


Figure 3.1 Original JPEG thin section image

In Figure 3.2 (a), (c), and (d), the color of the pores is dim, while it is very bright in (b). Furthermore, quartz grains are much darker than that in other images in (c). From this point of view, the simple thresholding technique can not be applied alone for the pore space segmentation since the predetermined threshold value cannot be appropriate to all images. In addition, porosity is a very important parameter for resource assessments and

reservoir discovery, and accuracy is the most important criterion for this stage. Therefore, algorithms for pore space segmentation should be adaptive to these variations. Moreover, the segmentation scheme selection should focus on the achievement of high accuracy.

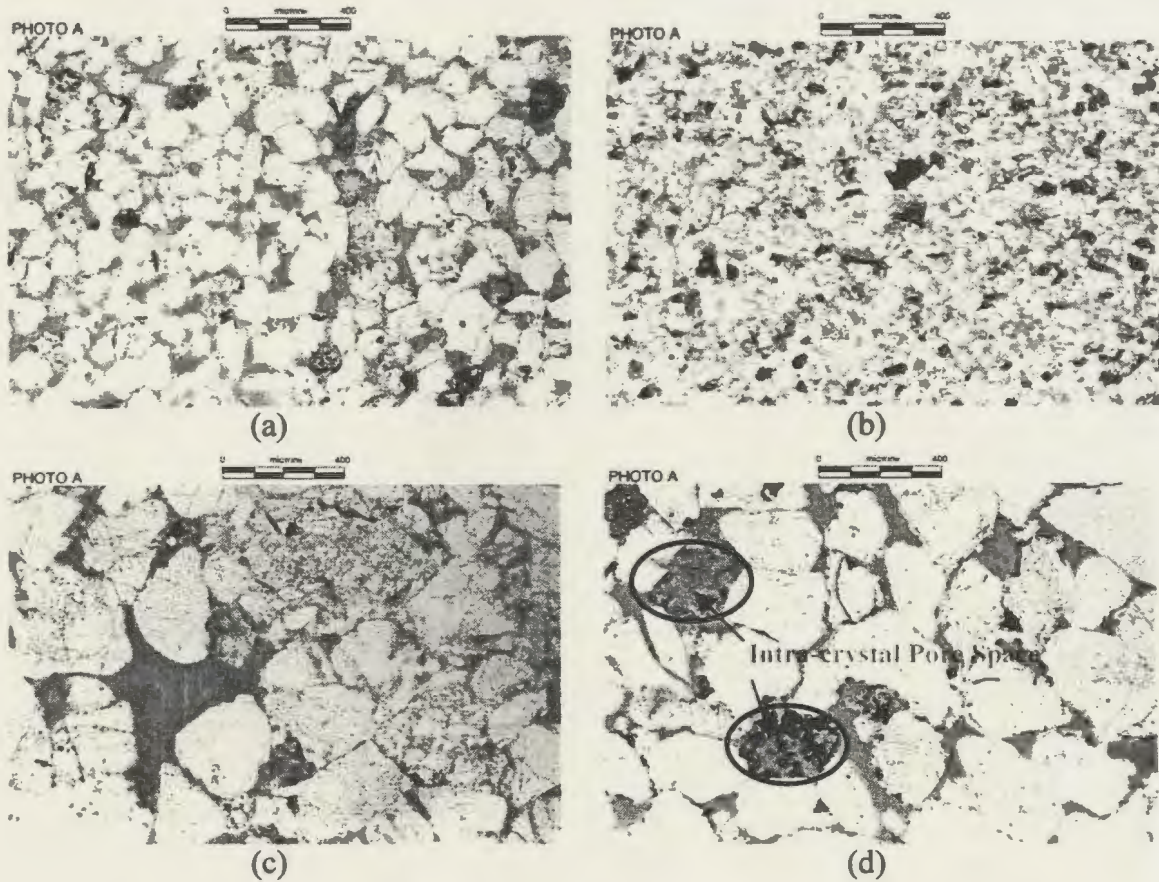


Figure 3.2 Different original images

3.1.1 Segmentation Scheme Selection

There are a variety of segmentation techniques in the literature. These techniques are mainly in two major categories: edge-based and region-oriented segmentation methods. In Chapter 2, it is concluded that edge-based methods are not appropriate for noisy images since the noise also has high frequency properties as edges do. In the image shown in Figure 3.1 and Figure 3.2, there is plenty of “noise”, speckles and black spots,

on the quartz grains. In fact, this is a common texture feature of quartz grains, which makes edge-based methods perform poorly in extracting real boundaries of the quartz grain. In addition, there are different types of pores, such as inter-granular, intra-granular, inter-crystal and intra-crystal. For the intra-type pores seen in (d), it is found that they do not have clear boundaries.

Region-oriented methods focus on the internal properties of a region, and exploit the homogeneity of spatially localized features. Seeded region growing methods [Adams and Bischof, 1994 and Hojjatoleslami and Kittler, 1998] are typical of region merging techniques, which start with a region from a given pixel or a very small region, known as a seed, and performs a pixel aggregation operation by merging the seed's neighbors into the region under certain merging conditions. Since this technique is performed locally, it usually yields accurate results. In the literature, most segmentation techniques are developed based on monochrome images. In PetroGraFX, the original images are full color. As mentioned in the early section, the color image is appropriate for the pore space identification. Consequently, the seeded region growing scheme has been extended to the color space.

3.1.2 System Flow Chart

In order to manipulate the color image, a proper color representation system should be chosen for further segmentation purposes in the first step. The second step is the extraction of the most appropriate features for highlighting the pore space against its background. Once the best features are obtained, an automatic seed selection operation is performed for extracting seeds from all possible regions of pores. Then a seeded region

growing scheme in color space is applied to achieve more accurate results. Finally, the porosity is derived from the percentage of pores in the whole image. Figure 3.3 illustrates the system flow chart of the porosity derivation.

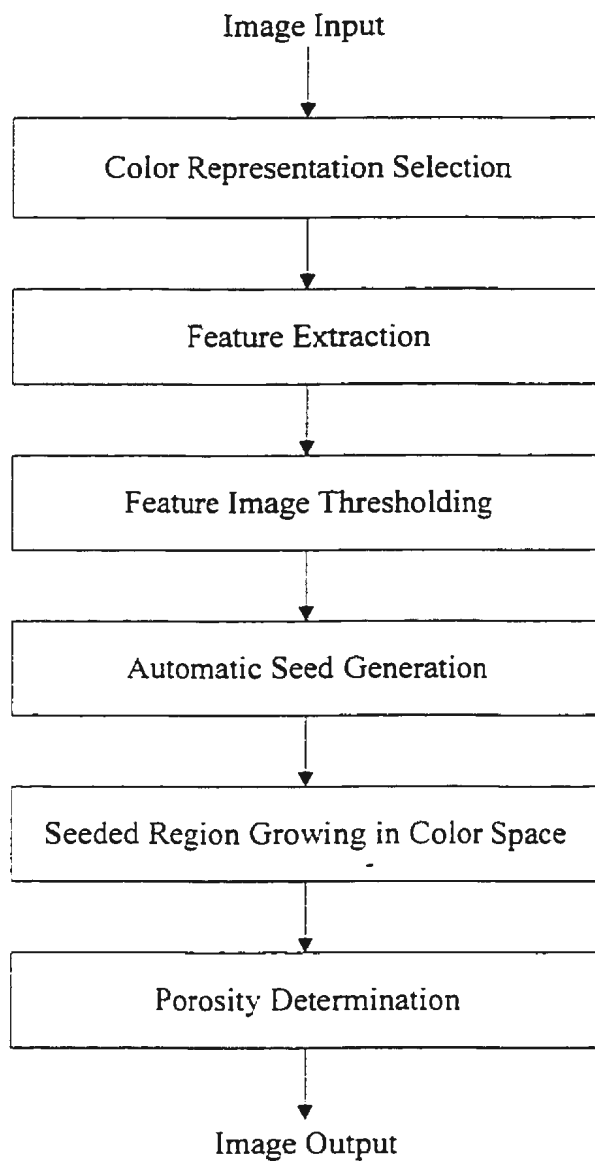


Figure 3.3 Flow chart of porosity determination

3.1.3 Color Space Selection

With the increasing use of color image acquisition technology, such as color scanners, monitors and printers, the color imaging is being developed rapidly. For users of this technology, color is sometimes thought of as a combination of the primary colors, red, green and blue (**RGB**). This is also considered as the basic color space.

Based on RGB, cyan, yellow, magenta, and black (**CYMK**), also known as subtractive primaries, is particularly useful in the publishing industry since it provides the ease of mixing inks. [Wilson, 1988]

The **YIQ**, in which Y is luminance and I and Q are chrominance, is a color model that has been widely used in television broadcast:

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (3.1)$$

Another important color model is hue, saturation, and intensity (**HSI**). Hue refers to a pure color, such as yellow or red whereas saturation describes a measure of its dilution by white light. [Sonka et al, 98] This model is very useful for image processing algorithms. We assume that

$$0 \leq r, g, b \leq 1 \quad (3.2)$$

Then, hue, saturation and intensity can be derived from RGB:

$$\begin{aligned}
i &= \frac{r+g+b}{3} \\
h &= \cos^{-1} \left\{ \frac{\frac{1}{2}[(r-g)+(r-b)]}{[(r-g)^2+(r-b)(g-b)]^{1/2}} \right\} \\
s &= 1 - \frac{3}{r+g+b} \min(r, g, b)
\end{aligned} \tag{3.3}$$

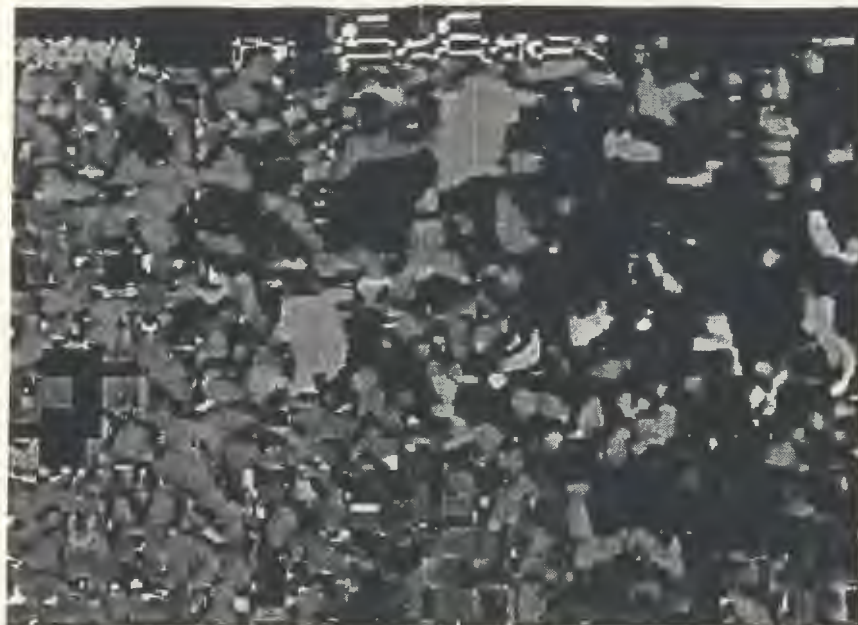
if $b/i > g/i$, then $h:=2\pi-h$. [Sonka, 1998]

According to Ohta et al [1980], hue is the most useful feature for color image segmentation. However, in our test data set, most images do not show the pore space clearly in HSI color space. Figure 3.4 gives hue and saturation channels of the original image shown in Figure 3.1.

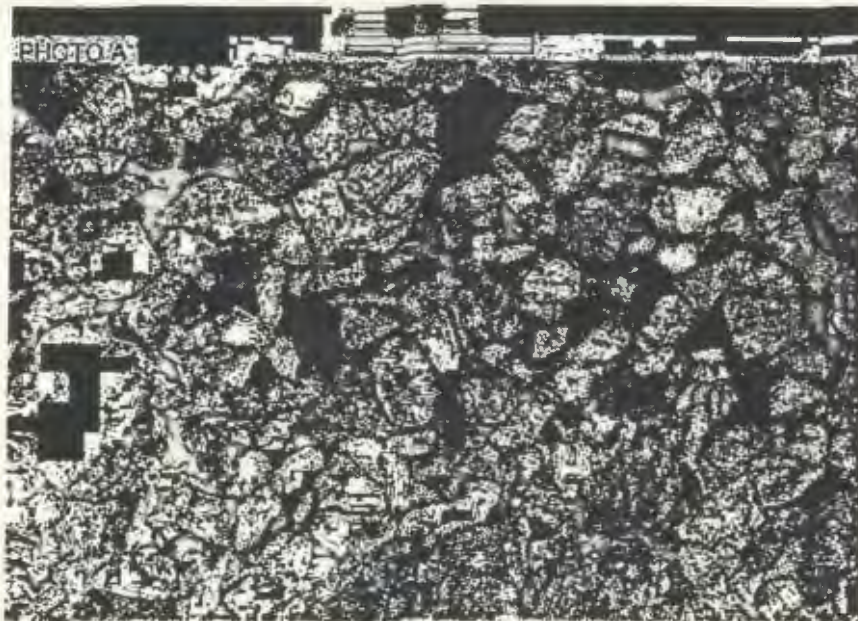
As a result, the RGB color space is chosen since it is the basic color space, and many features can be extracted from r, g and b channels.

3.2. Feature Selection and Extraction

The pore space segmentation is viewed as a classification problem in this research work. All the pixels in an image are classified into one of two classes, pores and non-pores. In classification schemes, feature selection and extraction are crucial steps. Features can be pixel intensities, edges and/or texture features.



(a)



(b)

Figure 3.4 (a) Hue, (b) Saturation channels of the image shown in Figure 3.1.

3.2.1 Criteria for the Feature Selection and Extraction

The objective of feature extraction is to find a set of features which provide greatest separation between the classes of data. In other words, feature extraction is to

form a new smaller dimensional space Y from a larger dimensional feature space X . All the elements in Y are called features, and Y refers to the feature vector,

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (3.4)$$

Feature extraction reduces computational complexity for pattern classification, and produces fairly accurate estimates and reliable decision rule for a certain number of training samples. [Therrien, 1991]

The criteria for feature selection and extraction are [Chen, 1978]:

- Features should properly describe the pattern, which means there is no increase in the minimum probability of error.
- Features are manipulated without difficulty.
- Features should be invariant to translation and rotation of the pattern.

In practice, it is very difficult to satisfy all three criteria at the same time. One should decide which measure is more important in a particular application.

3.2.2 Feature Extraction: R – B Difference Image

As mentioned in the beginning of this chapter, the pore space is distinguished from its background in its color, which is bright blue although the intensity can vary within an image and between images. Generally speaking, color perception is very important since human eye is more sensitive to color than to brightness. In this work,

feature extraction is performed in the color space. The original image is split into red, green and blue channels as shown in Figure 3.5.

Observation of the split r, g and b channels indicates that red and green channels are nearly identical except the red displays higher contrast than the green does, while there are significant differences between the blue and the other two channels.

In chapter 2, Haralick [1985] suggested that it is not necessary to compute histograms in red, green and blue channels individually, but uses $(R + G + B) / 3$, $(R - B) / 2$ and $(2G - R - B) / 4$ in color images.

Recall the predictive coding schemes in image encoding techniques:

$$e_m = x_m - \hat{x}_m \quad (3.5)$$

where e_m is the difference or the error signal, x_m is the actual pixel value, while \hat{x}_m is the prediction. [Appendix I] The purpose of the predictive coding is to reduce the variance inside the image, so that the difference image has less correlated data. This will produce high compression ratio. The idea of this technique is the same as that of feature extraction to a certain extent.

From Figure 3.5, the pore space shows higher intensity in blue channel and lower intensity in red channel. Quartz grains are very bright, while the opaque components and the clay appear very dark in all channels. From this point of view, the differences between the red and the blue, or the green and the blue will highlight the pore space. Since the red channel has higher contrast, the absolute difference image is generated between the red and the blue. Figure 3.6 illustrates the generation of this difference image.

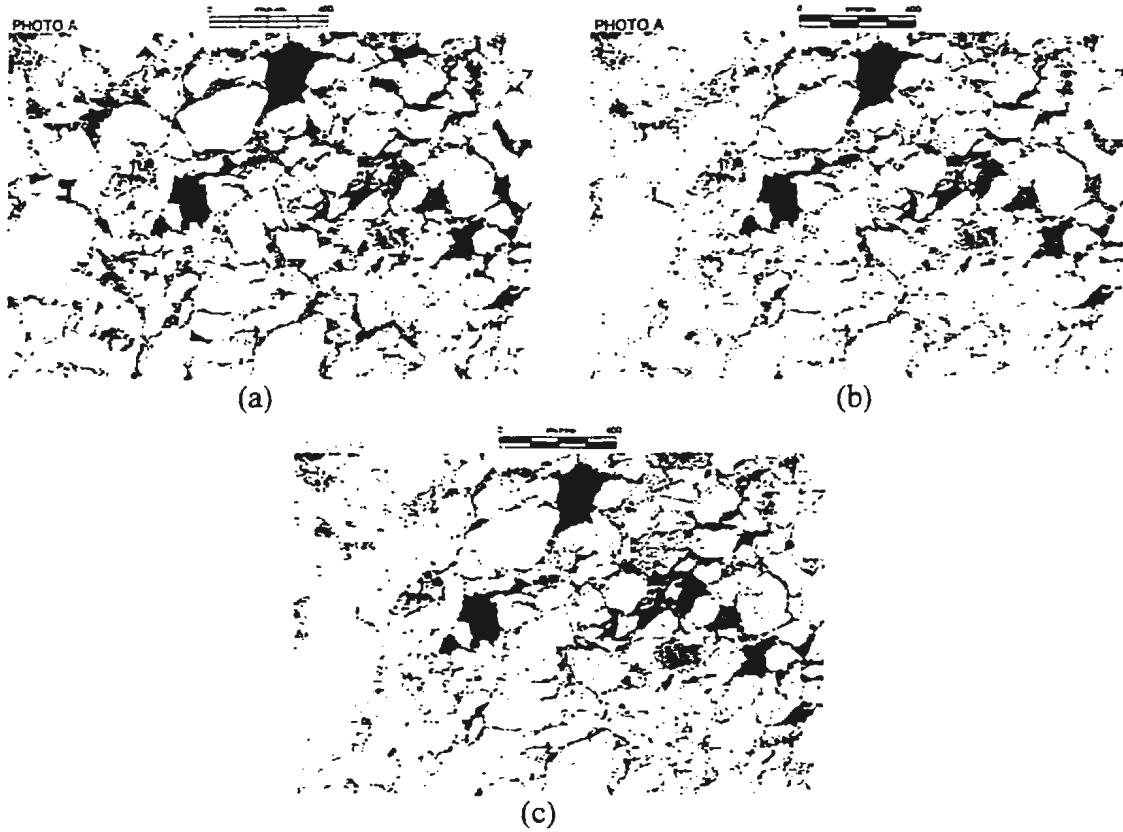
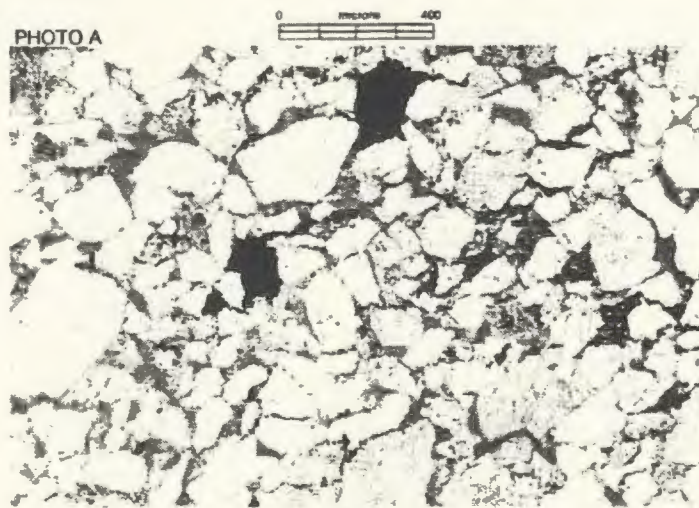
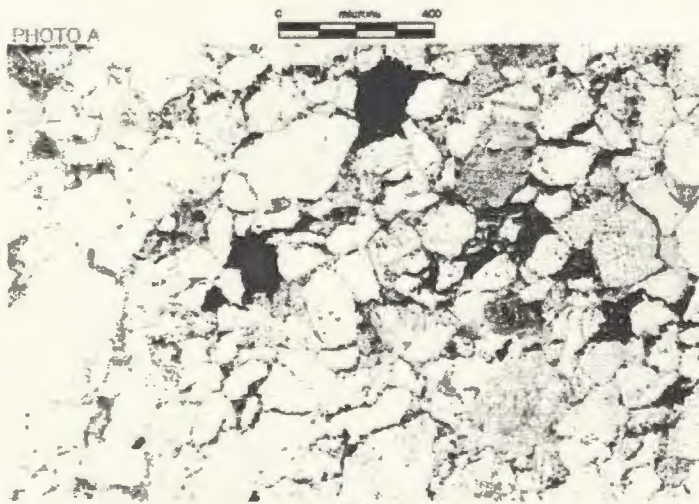


Figure 3.5 (a) Red, (b) Green and (c) Blue channels of Figure 3.1.

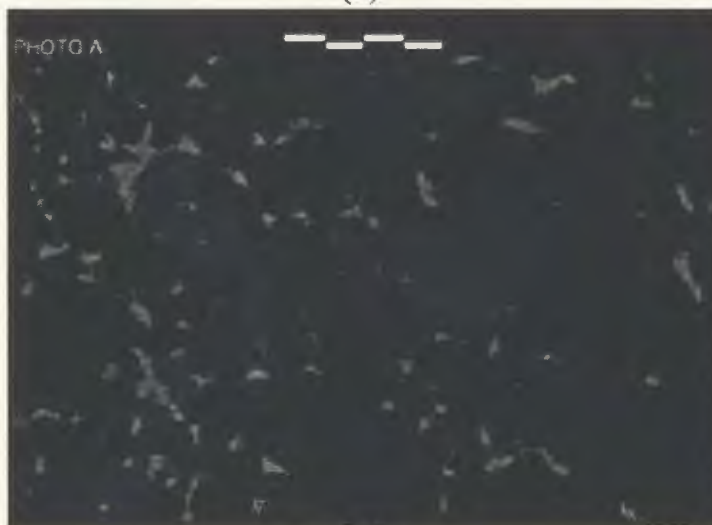
Obviously, only the pore space is highlighted in the absolute difference image of $R - B$. This difference image also has another benefit: the variation of the lighting condition and the color of quartz grains are suppressed since this image is a subtraction of two channels of the same image. This factor will be particularly useful during the threshold value selection stage. The difference image operation is applied to all images in the test data set, and the pore space is fairly clear in the resulting images. Figure 3.7 shows the absolute difference images corresponding to the original images shown in Figure 3.2.



(a)



(b)



(c)

Figure 3.6 (a) the red channel, (b) the blue channel, (c) ABS(R-B) difference image

Now, the color representation is reduced from three channels to one, and the desired image region corresponding to the pore space, is highlighted.

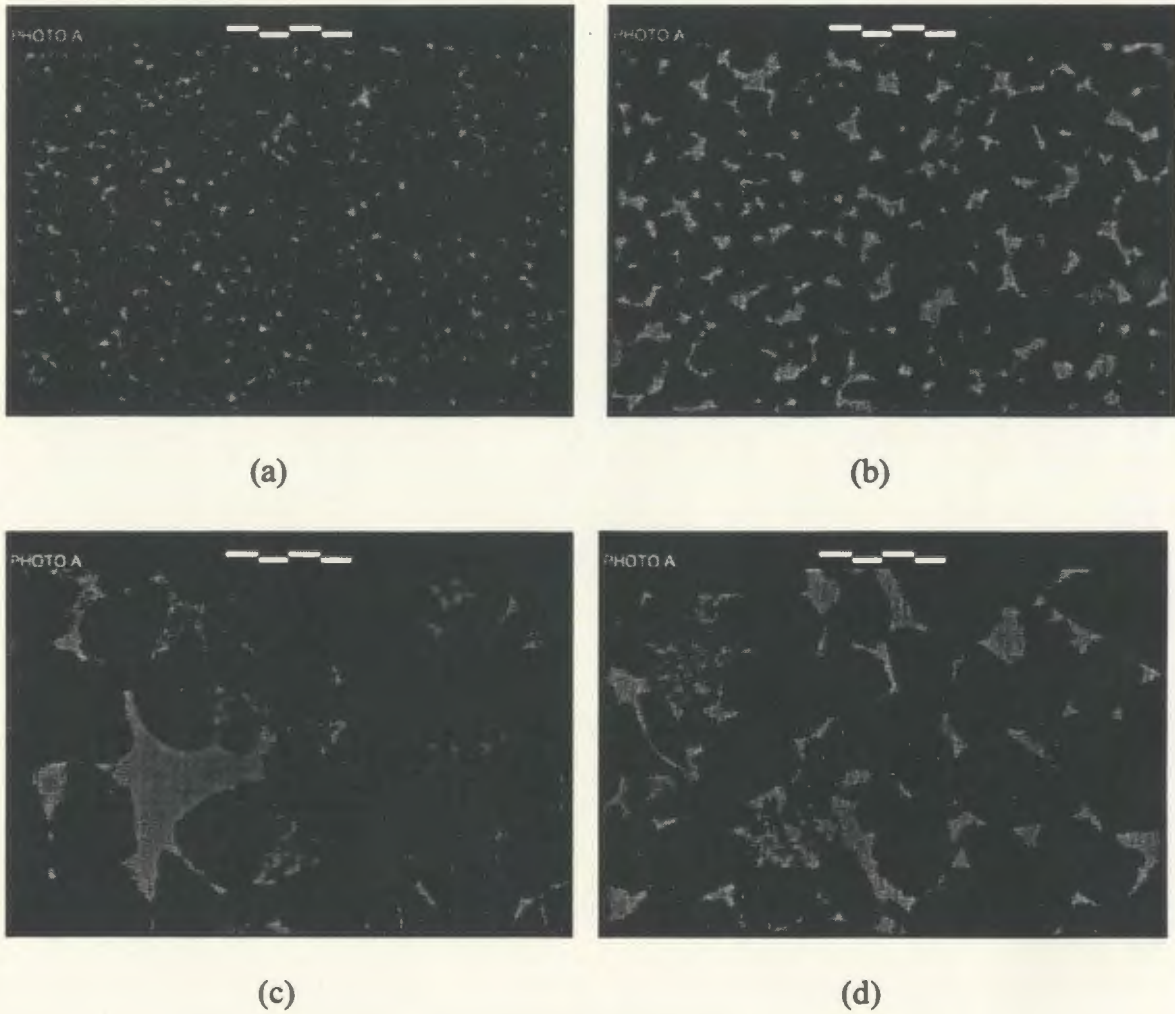


Figure 3.7 Difference images of original images shown in Figure 3.2.

Chapter 4

Porosity Determination based on the Pore Space

Segmentation

In this chapter, the pore space segmentation will be detailed. After the feature space is obtained, the appropriate segmentation techniques are employed to yield satisfactory results.

4.1 Feature Image Thresholding

Chapter 3 illustrated that the absolute difference image of R- B highlights the pore space, as well as suppresses variations of lighting conditions and the color of quartz grains. This provides for an option of presetting the threshold values. As mentioned in Chapter 2, thresholding is a global operation, and local properties will be subsequently lost. Therefore, thresholding the feature image can only yield approximate pore regions and it cannot be used alone for producing final accurate results. As described in Section 2.5 and 3.1.1, the seeded region growing scheme is presented for the pore space segmentation since it exploits the homogeneity in the seed's neighborhood, and a more accurate segmentation result is yielded. Since the goal of PetroGraFX is to provide for automated image analysis, the seeds that are required by the local region growing technique should be generated automatically. These seeds can be extracted from all

regions of pores after the feature image thresholding as long as these regions are real pores.

4.1.1 Predefined Thresholds

In order to find all pore regions and get rid of all other components, an investigation of the color distribution for various components was made over all images in the test image set. It was found that there are two kinds of components that should be excluded when thresholding the difference image:

a. Clay

Clay component has similar values to the pore space in the difference image. This component has the positive difference of R and B, while the pore space has negative values. A condition for removing clay was set to $B > R$.

b. Rock fragments

Usually, rock fragments in an image appear black, and some of these also have similar $(R - B)$ values to the pore space. These components, however, have low gray levels in both the red and blue channels. Experimentally, the threshold values, T_{red} and T_{blue} , were selected as 60 and 160 for the red and blue channels, and these values were tested over the image data set.

Another predefined threshold is for the difference image itself. This value, T_{diff} , was experimentally set to 28 which is very low in order to avoid light blue areas missing.

4.1.2 Automatic Threshold Selection

From Figure 3.2, it is observed that the color of quartz grains varies in a great deal in different images, so that the maximum value of quartz grains cannot be predefined. As

indicated in Chapter 2, the image histogram can be used for automatic threshold selection if the object is distinguished from the background in its gray level. This type of methods is based on monochrome images. In this application, this operation was extended to color images by choosing one representative channel. The red channel is a good candidate since it is most similar to the intensity of the original.

The red channel histogram is generated, and the maximum occurrence of the gray level in the range, 180 – 252 which corresponding to the quartz grain intensity range, is calculated. Based on this value, the maximum gray level of quartz grains in the red channel, T_{quartz} , was obtained.

4.1.3 Feature Image Thresholding

As described in Section 4.1.1 and Section 4.1.2, the rough pore regions can be obtained by thresholding the difference image based on a threshold T_{diff} . These regions include some false pore regions, such as clay and quartz grains, and should be eliminated by comparing the gray level for each pixel in the red and blue channels to the corresponding thresholds T_{red} , T_{blue} and T_{quartz} . Figure 4.1 shows the thresholded difference image according to both predefined and automatic generated threshold values.

According to Figure 4.1, all pore regions in the original image shown in Figure 3.1 are found, while pixels near some pore boundaries are lost as predicted. Notice that some very thin regions that correspond to the boundaries of the clay are also included in this image. These rough regions are used for the automatic seed generation purpose.

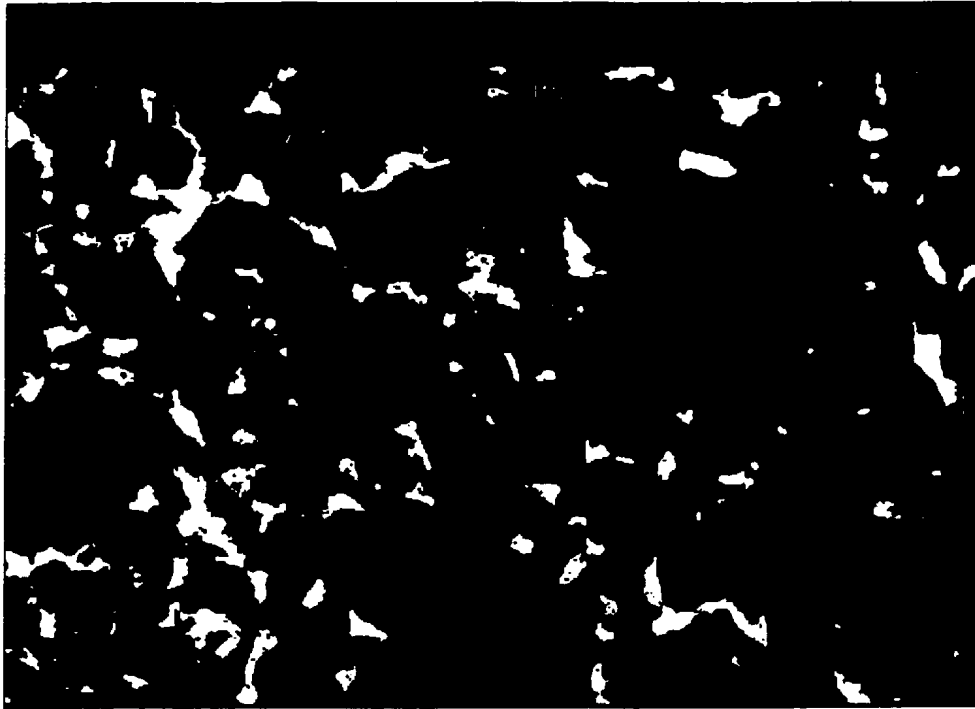


Figure 4.1 Thresholded image of the difference image

4.2 Seed Selection for Region Growing

In the literature, the seeded region growing techniques usually require the input of seeds manually. In order to reach the goal of the automatic image analysis, these seeds should be generated automatically. Since the seed selection is crucial to the entire algorithm, and seeds missing will directly affect the accuracy of the final results, these seeds should satisfy following conditions:

- Cover all regions of the pore space. This means that there is no missing pore space.
- Only corresponding to a desired region (i.e. the pore space).

In order to reach these goals, all possible regions, or blobs, need to be generated first as described in Section 4.1.

4.2.1 Blob Generation

As mentioned in Section 4.1.1, the threshold in the difference image is predefined to a very low value, 28, so that all possible pores are obtained including those pixels nearby the pore space boundaries after applying a threshold operation. This ensures the first goal for the seed extraction is reached.

The second condition focuses on the removal of false blobs that are not the pore space. In Figure 4.1, there exist some very thin blobs, whose widths are usually one or two pixels, which correspond to the boundaries of the minerals in the original image. These blobs are actually minerals that have similar values as the pore space in the difference image, and will not yield real seeds; therefore, they should be removed from the feature image.

Since one significant characteristic of these blobs is that their widths are very thin, only one or two pixels, the opening operation is an effective for the removal of these blobs. Meantime, other blobs will not be affected by this operation except for the slight shape variations, and this will have no influence in the seed generation. Upon this stage, all blobs in the feature image represent only the pore space. Figure 4.2(b) illustrates the binary image after the one time opening operation. Comparing to Figure 4.2(a), it was found that those thin regions corresponding to mineral boundaries are eliminated.

4.2.2 Seed Extraction from Blobs

A problem arises when considering shape analysis of the blob. Given that there are small holes inside some blobs. These holes normally correspond to tiny mineral chips

dropped into the pore space during the thin section preparation, or small pieces of minerals nearby the pore space in the original. They are formed during the thresholding operation. since the pixels inside these holes have different colors. If a seed falls in a hole. the color of this seed will not be blue, but either white or black. Another problem for the second criterion is that the seed should be chosen close to the central part of a blob. This is because pixels along the boundaries are not representative, and may result in an inaccurate region growing.



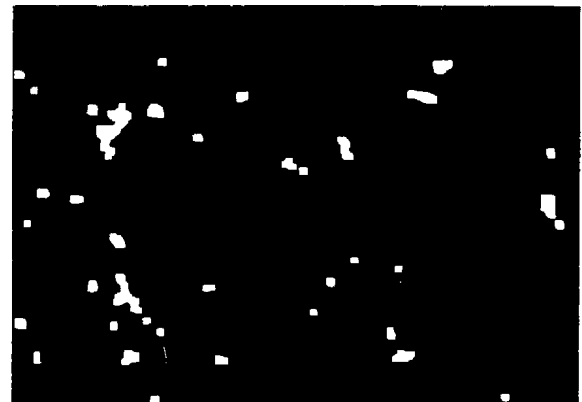
(a)



(b)



(c)



(d)

Figure 4.2 (a) Original binary image, (b) binary image after one time opening operation, (c) three times opening operation, (d) six times opening operation

In order to find all true pore regions, each blob in Figure 4.2(b) is considered as a seed-carrier. The seed should be a pixel in the central part of a blob, and not in a hole. The blob analysis is employed to generate seeds from blobs, and the pixels in a hole can be excluded from a blob. The procedure of seed extraction is as follows:

1. Choose a blob Figure 4.2 (b), (c) and (d) are binary images with all pores in white, and the rest in black. The white region forms blobs, and these are labeled, and processed in a *for* loop.

2. Generate the distance transformation The purpose for this step is to find a pixel that is in the central part of a blob. The chamfer $\frac{1}{4}$ distance transformation is used to produce a distance image in which the value of a pixel is the distance from this pixel to the nearest edge of the blob.

3. Find a pixel that has maximum distance In the distance image, the pixels with the highest value within a blob should be near the center of the blob in most cases. The only exception is when a seed is located in a blob that touches the image border and is not included entirely in the image. According to the distance, a seed may be exactly on the image border in fade blue. The solution is to correct the image border, typically 2 – 3 pixels, to black before this operation, so that the blob, which is the white part, will exclude the pixels on the border and will ensure that the seed be blue. Notice that the centroid of the blob cannot be used since some blobs are not convex, and the centroids of these blobs will be outside their blobs. Finally, the pixel selected from these which has the maximum value yields a real seed.

4.3 Seeded Region Growing in Color Space

The objective of the seeded region growing operation is to achieve more accurate results for the pore space segmentation. It is found that most pores are obtained after thresholding the feature image. Since this is a global operation, some local properties especially near the pore space boundaries are missing. Based on seeds obtained from thresholding the feature image, the local seeded region growing operation exploits the similarities in the neighborhood of these seeds, and the pixels are aggregated under certain merging criteria, which are crucial to this operation, until the operation reaches the region boundaries.

The first step of the region merging is the base image selection. Although the feature image highlights the pore space, it ignores some important information. For example, the sign of the difference between gray levels in the red and blue channels is lost since the gray levels can not take negative values. As mentioned before, the clay has similar intensity values as the pore space, and the sign is a flag to distinguish between them. In RGB space, there is no single channel that can be used for this operation due to the overlap of gray-level distribution for different components. Region growing, therefore, should be performed in RGB color space. In chapter 2, it was concluded that most segmentation methods are based on monochrome images and can be extended to color images. Recall that the red channel is similar to the intensity channel, and most components show different gray levels in this channel. As a result, the red channel is chosen as the base image for the region growing operation.

4.3.1 Region Growing by Pixel Aggregation in the Red Channel

This operation starts from the seed and appends the region to the seed's neighbor pixels that have the similar properties, in this case, the color.

Let R be a set of a region, such that for any x in R

$$\delta(x) < T$$

$$\delta(x) = \left| g(x) - \underset{x \in R}{\text{mean}}[g(x)] \right| \quad (4.1)$$

where $\delta(x)$ is a measure of difference of x from the region, $g(x)$ is the gray-level in red channel of x and T is the threshold in the red channel. In the entire operation, the mean of the region is updated continually. Region growing proceeds by examining all pixels' eight neighbors in the red channel starting from the seed. When $\delta(x)$ is bigger, the operation stops. All pixels which satisfied the condition are marked after examining; therefore, they are counted only once in order to increase the algorithm efficiency. Figure 4.3 illustrates the seeded region growing operation.

In Figure 4.3, the threshold T , is predetermined to 30, which produces good region growing results.

4.3.2 Implementation of the Seeded Region Growing

One of the disadvantages of the region based segmentation techniques is the efficiency of the performance because the local properties, which usually mean heavy computations, are involved. As a commercial product, however, PetroGraFX requires high speed. So considerable effort was spent in dealing with finding ways to speed up the region growing performance.

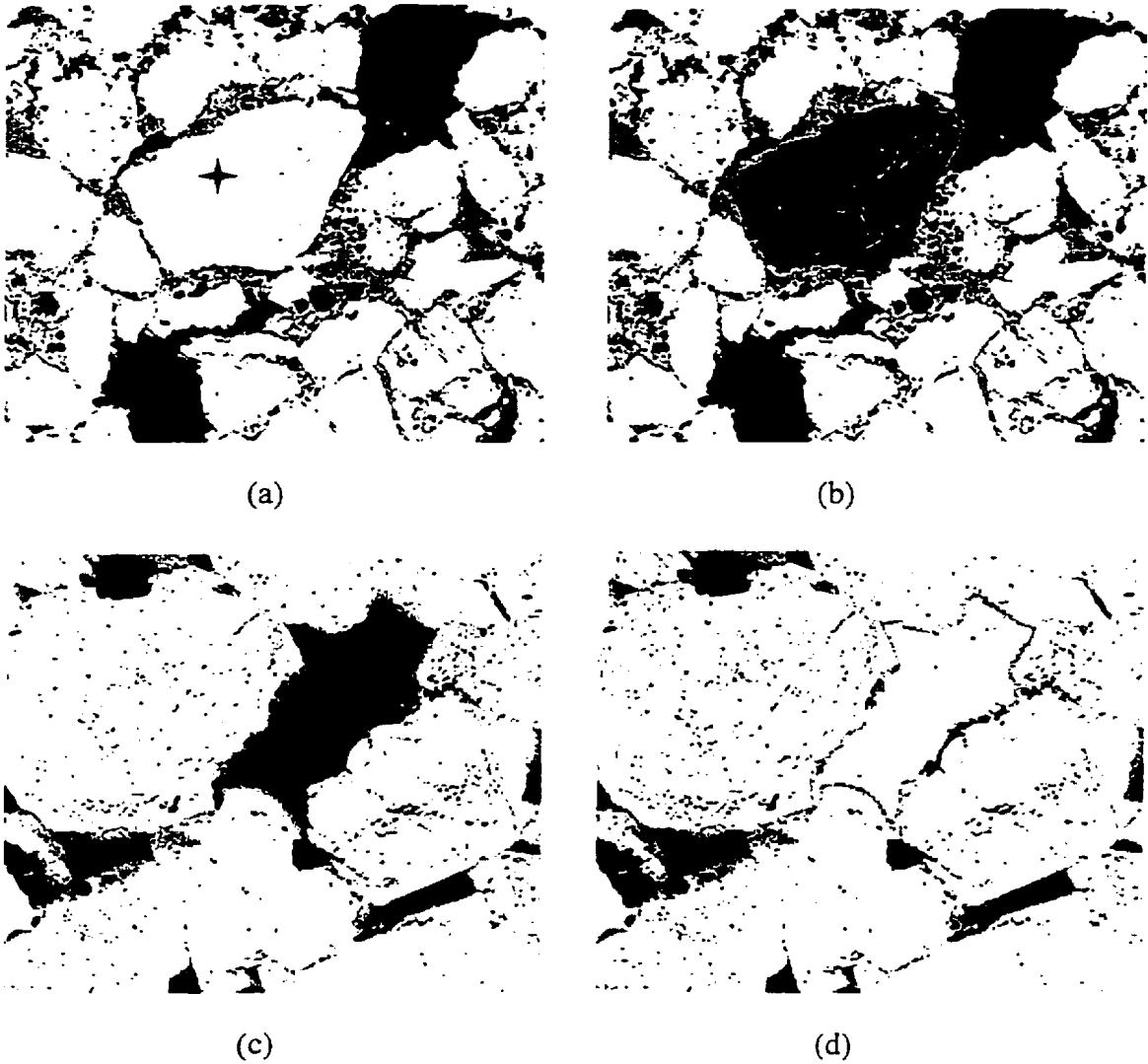


Figure 4.3 Illustration of the seeded region growing

There are two approaches for the implementation of this algorithm:

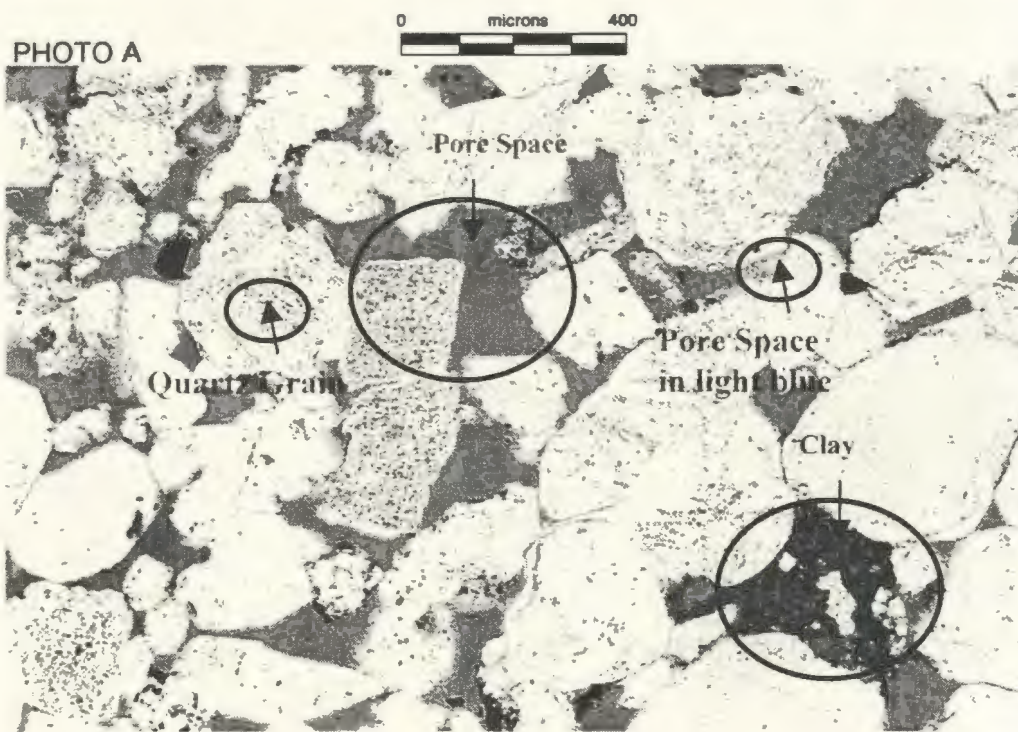
1. Recursion The algorithm for the recursion is very simple with the 8 neighbors of the seed being examined, and then the neighbor's neighbors, and so on. The implementation is straightforward, but the problems arise when the region becomes big or the image size is large because the recursion requires a significant of stack space. When a large amount of data are involved, the stack dumps.

2. Quick Sorting The main idea of this operation is to start a region from a seed, and to extend the 8 neighbors of this seed. Only neighbor pixels satisfied merging criteria are sorted into the region to form a set. Then the neighbors of all pixels in the set are examined, and their neighbors are sorted into the set when merging criteria are met, and so on. Finally, the set contains all pixels that are similar to the seed. The implementation using the linked list ensures fast operation.

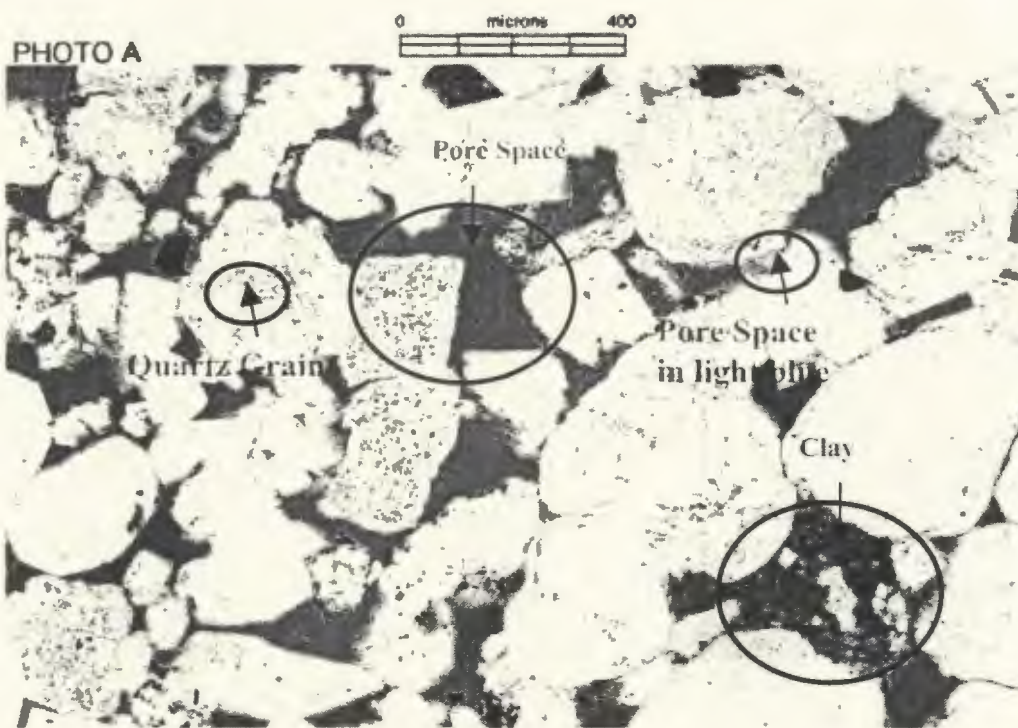
It was approved that quick sorting is superior in practice and used in this work.

4.3.3 Region Growing in the Color Space

In practice, region growing is more complicated. For example, in Figure 4.4 (a), there is a piece of clay near the right bottom corner, and it has almost the same gray level as the pore space in the red channel shown in Figure 4.4 (b). Furthermore, some pore regions have higher gray levels than usual, such as those light blue regions shown in Figure 4.4 that are close to quartz grains, in the red channel as shown in Figure 4.4 (b). As a result, the threshold T in Equation 4.1 should be relatively high to enable all pixels near edges that appear light blue to be selected. A T value of predetermined 60 was selected experimentally. This introduces a new problem: some non-pore regions will also be included; therefore, other information should be used for excluding these regions. In order to exclude the non-pore components, mainly rock fragments and clay, the region growing should be extended to the color space instead of only the red channel.



(a)



(b)

Figure 4.4 (a) Thin section color image with clay (b) the red channel

As discussed early in this chapter, the red and blue channels are efficient for recognizing different components. Like the work in thresholding the feature image, five criteria are given for restricting the growth of a region:

- Intensity in the difference image $T_{diff} > 28$
- Intensity in blue channel $T_{blue} > 160$ for getting rid of the clay
- Intensity in red channel $T_{red} > 60$ for avoiding rock fragments, and $< T_{quartz}$ which is determined by running the histogram in red channel for excluding the quartz grains.
- The difference $R - B$ should be negative, or $R < B$ for deleting the clay.

4.3.4 Three-Step Seed Input

There is a problem when applying the seeded region growing operation from seeds which are obtained from the thresholded difference image. Although they are all real seeds, some pore regions are missing in the region growing resulting image. This requires the mapping between the seed selection and the seeded region growing. Recall that the threshold for the difference image is set to a very low value to ensure all pores are included. This has the side effect that a blob in the feature image probably represents more than one region of the pore space. The low threshold value results in some separated regions being connected together. If only one seed is extracted from such a blob, a smaller region will be produced from this seed in the region growing operation. In other words, more seeds need to be generated to represent one region in the feature image.

In order to solve this problem, a three-step seed extraction is performed. The main idea behind this operation is to separate connected blobs by applying an opening operation. Since small blobs will vanish during the opening, no single feature image can be used to represent all blobs. The seed selection operation should be performed multiple times to ensure that seeds from all blobs are extracted. The sequence of operations follows:

- Perform opening once to remove the very small false objects (already shown in Figure 4.2(b)) – Seeds A.
- Opening three times to break the connected region and keep most small blobs shown in Figure 4.2 (c) – Seeds B.
- Opening six times to break widely connected regions shown in Figure 4.2 (d) – Seeds C.

Three images are generated, and used in a sequence for the seed selection. The seeds are selected automatically and stored in three individual groups, Seeds A, Seeds B and Seeds C in which Seeds A contains the maximal number of seeds, while Seeds B medium and Seeds C the minimum. The region growing starts from Seeds C. In this step, all seeds from large regions are input, and all pixels inside these regions are counted in and marked. In second and third steps, Seeds B and A are input, and the seeds are checked before the region growing function is called. If a seed is already marked, it means that region growing is applied to its neighbors and is removed from the corresponding seed group since the region it represents is already included in a region. This ensures every pixel that is belonging to a pore space region is only counted once during this three-step seed input, so that the system has high efficiency.

4.4 Porosity Derivation

In manual thin section analysis, the porosity of a given slide is measured visually or by point-counting while the former is less accurate and the latter is time consuming. This is straightforward in image analysis after all regions of pores are obtained. The porosity is derived from the pore space percentage calculation: the number of all marked pixels divided by the pixel number of in the whole image. The percentage is approximated to the nearest integer and printed onto the output image. Figure 4.5 gives a comparison chart for visual porosity estimation [Scholle, 1979], while Figure 4.6 is the resulting image for porosity determination. All pixels of pores are changed into pink in this image.

4.5 Pore Space Distribution

The purpose of the generating the pore space distribution is that it is related to another very important parameter, permeability. Given two samples with the same porosity, and image of equal magnification, the image containing the less number of pores will correspond to the well from which it will be easier to recover oil.

The area of every individual pore is calculated, and the histogram of the pore space, that is the occurrences via area (μm^2), is generated and plotted as shown in Figure 4.6. From this figure, it is clear that the size distribution is not normally distributed, and about 80% of the pores are small and within a very narrow range of area. Only a few is over $5000\mu\text{m}^2$.

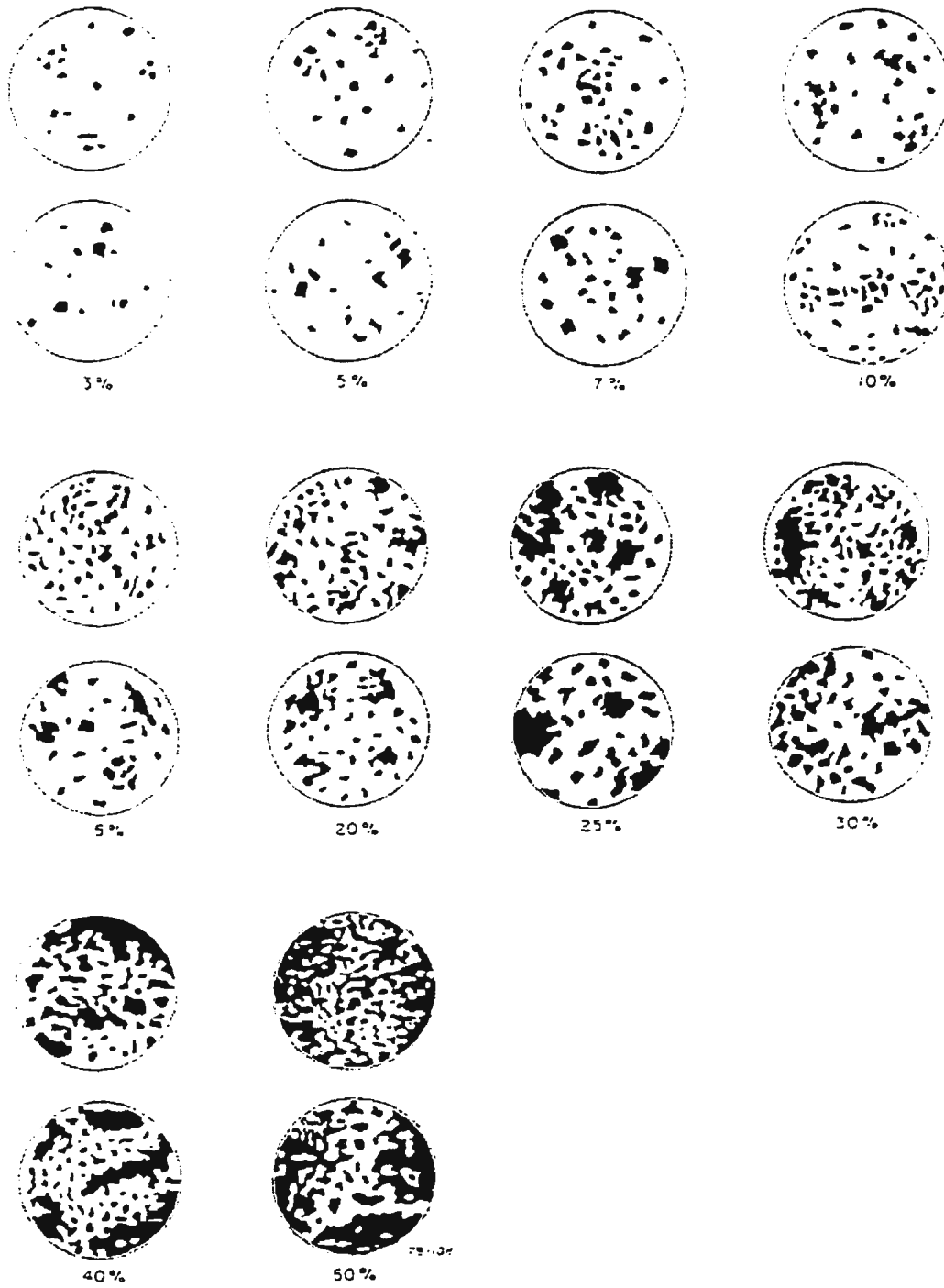


Figure 4.5 Comparison chart for visual porosity estimation estimation [Scholle, 1979]

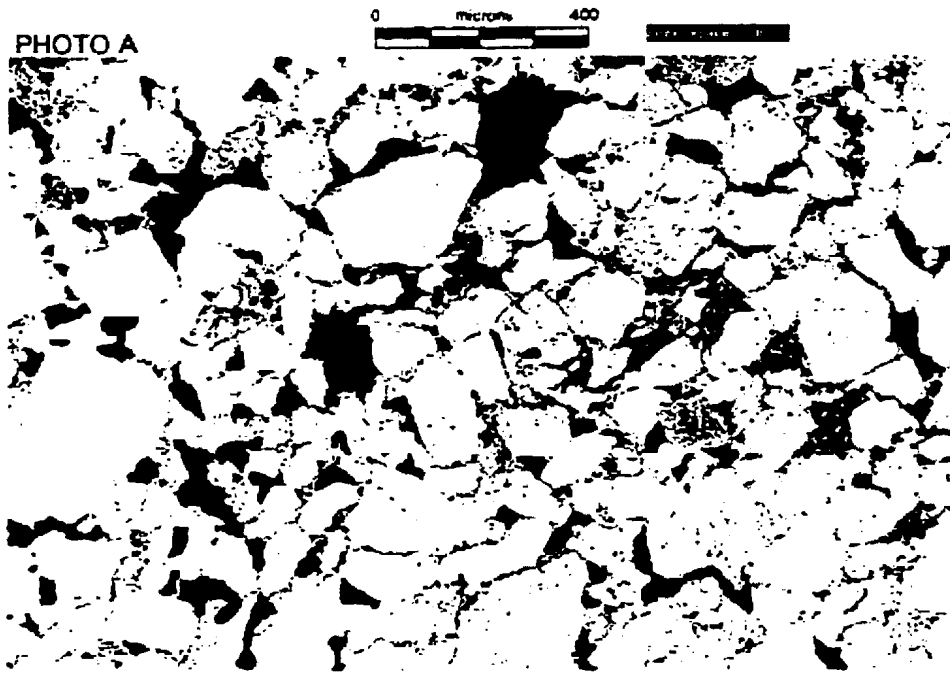


Figure 4.6 Segmented pore space

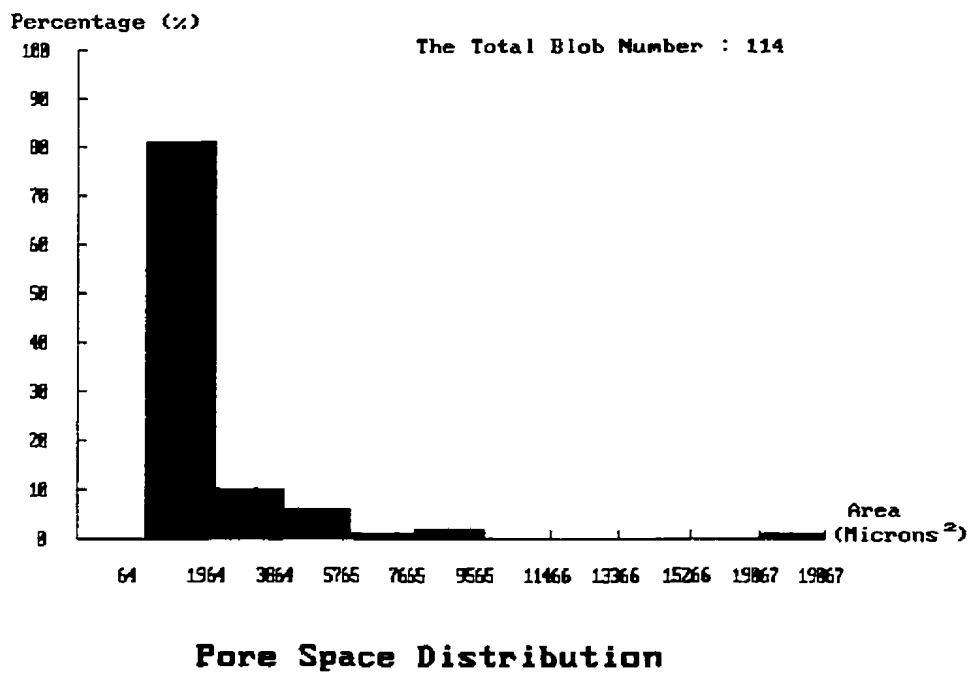


Figure 4.7 Pore space distribution

Chapter 5

Quartz Grain Segmentation

The minerals, which belong to the sandstone category, are the other predominant objects in the thin section image. In analyzing thin section images, reservoir engineers are typically interested in materials surrounding the pore space in the core sample.

5.1 Sandstone Classification

According to Williams et al [1983], “Sandstones are the rocks in which detrital grains of sand predominate”. In the image shown in Figure 5.1, a sandstone consists of several different types of minerals, such as quartz grains, clay and opaque components, which corresponds to coal in this image. These minerals vary in many respects, including size, shape, orientation and texture.

According to some reservoir geologists, quartz is the most interesting mineral in this type of images, while all other minerals are secondary. Therefore, quartz measurement becomes the major task for the mineral analysis in this work. Opaque components are visually different from all other materials. They appear as black and texture free, and are easy to distinguish. Clay usually contains rich textural properties and appears as brown. The remainder of the work discussed in this thesis restricts the mineral measurements on quartz grains.

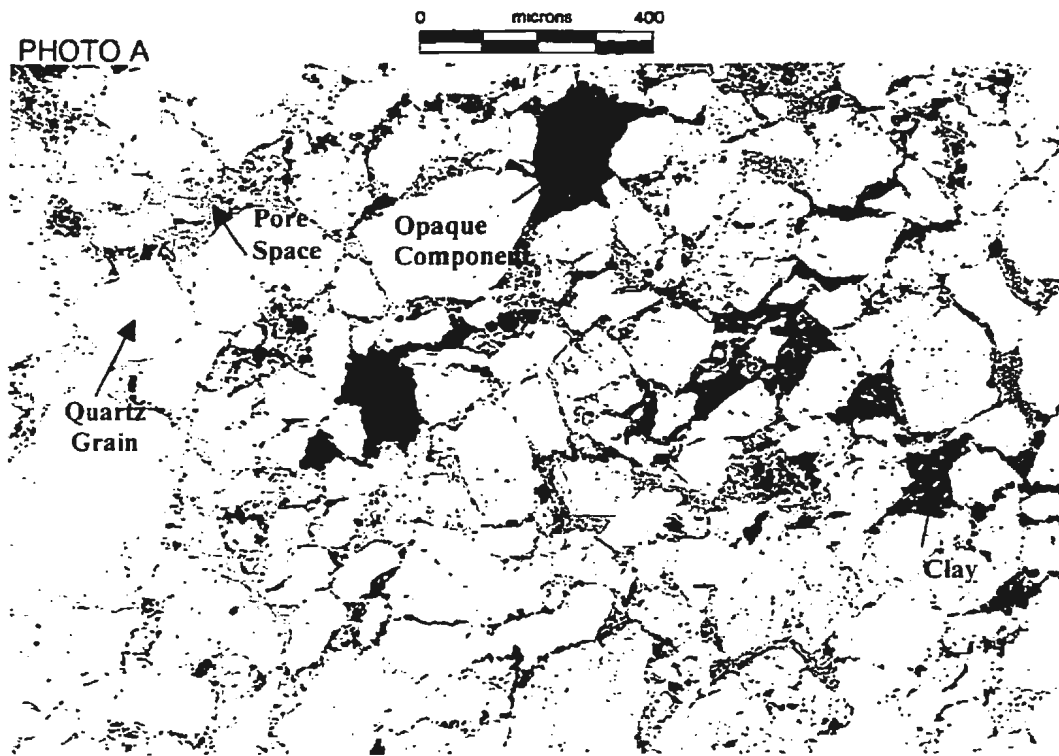


Figure 5.1 Thin section color image

5.2 Quartz Grain Segmentation

Grain size assessment is an important parameter in thin section analysis of core samples. In order to measure the quartz grain size individually or statistically, all quartz grains should be obtained using segmentation techniques. In addition, all adjoining grains should be disconnected before the size measurement. Performing this operation automatically and reliably is perhaps the most challenging problem in developing PetroGraFX.

5.2.1 Characteristics of Quartz Grains

From the image shown in Figure 5.1, quartz grains are commonly bright except for few “dirty” grains. They usually have high contrast to other image objects in all three

channels. Therefore, a single channel can be used to segment them. Another significant characteristic is that most quartz grains are nearly texture free and only a few speckles and black spots which correspond to lines of fluid inclusions are usually found on these grains.

Since most texture-based segmentation methods use a window for taking measurement in the local neighborhoods, the boundaries of objects are degraded in the resulting image. In taking grain size measurements, grain shape is important and a clear boundary description, such as angularity and roundness, is required. On the other hand, the texture properties for most quartz grains are limited to a few speckles and small black spots that can be removed through image preprocessing and blob analysis techniques. As a result, the conventional segmentation techniques are employed for obtaining the quartz grains. For the small partition of quartz grains that contain rich texture properties, texture-based methods will be appropriate, and would be the future work for PetroGraFX.

5.2.2 System Flow Chart

In order to measure the grain size, all individual grains should be obtained in advance. According to reservoir geologists, the speckles and black spots which correspond to lines of fluid inclusions and often contain water and oil inside quartz grains should be included in the corresponding quartz grain. Therefore, they should be removed. Since there are many cemented grains in the image due to the quartz grain overgrowth, the watershed transformation was investigated to separate adjoining grains. Due to the characteristics of the quartz grain, some false watersheds are obtained inside some long

and large grains. A modified watershed transformation was developed for removing these false watersheds. The system flow chart is illustrated in Figure 5.2.

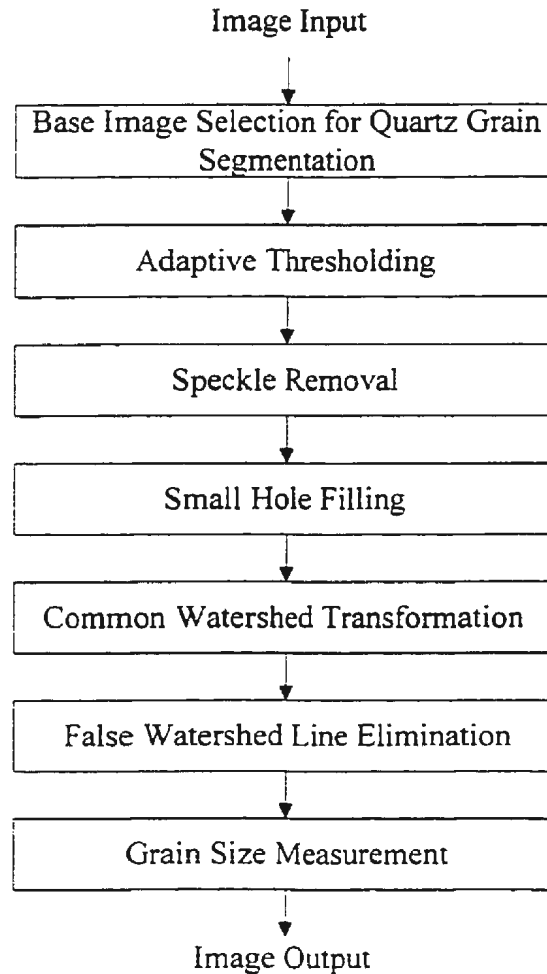
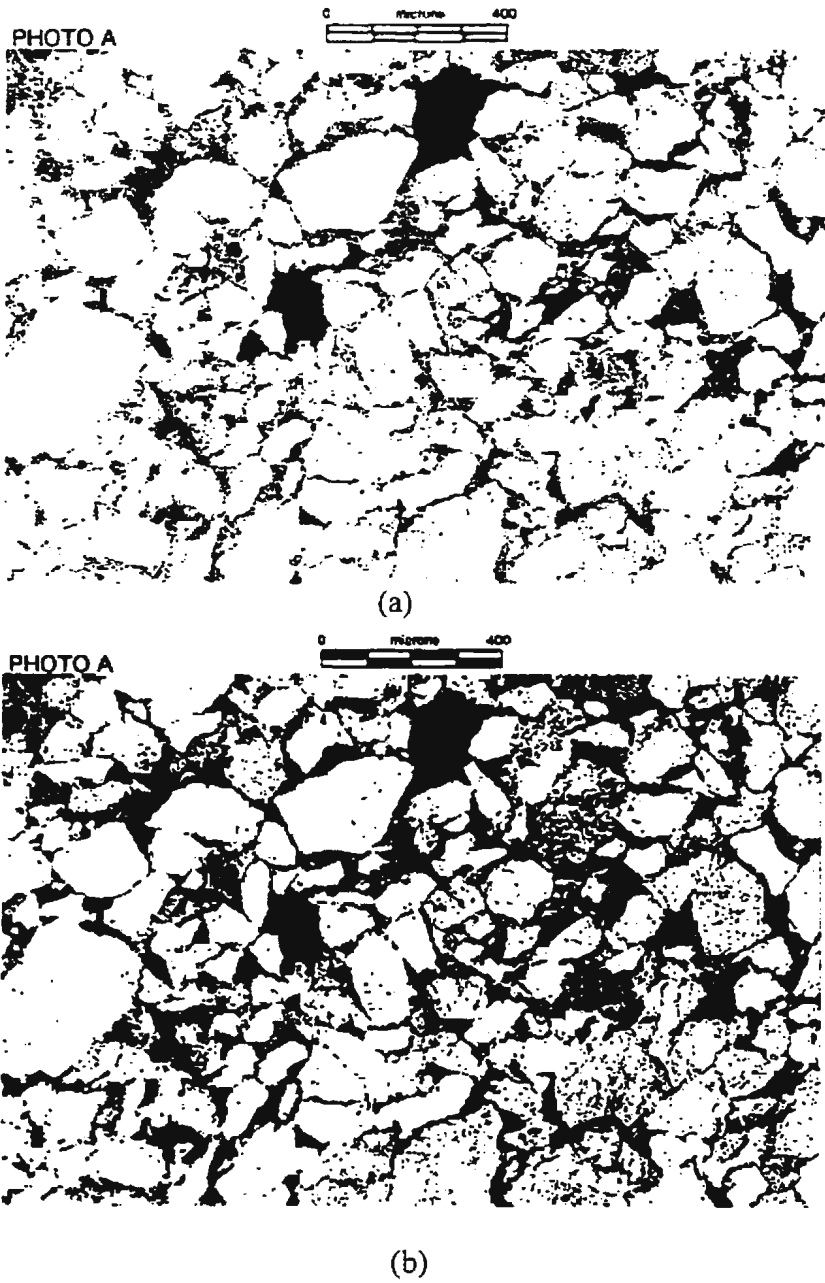


Figure 5.2 Flow chart for quartz grain segmentation and measurement

5.2.3 Quartz Grain Obtaining

As mentioned in Section 3.2.2, the red channel provide highest contrast for recognizing different objects, and has been chosen as the base image for the quartz grain segmentation. Since the shape is an important parameter for thin section images,

considerable effort has been made to extract precise boundaries of grains. From the image shown in Figure 5.3(a), most quartz grains have very high image intensity. Thresholding this image based on the image histogram finds most of quartz grains as illustrated in Figure 5.3(b).



(b)
Figure 5.3 (a) Red channel, (b) Thresholded red channel

Comparing Figure 5.3 (a) and (b), it was found that most quartz grains in the left part of the image are captured successfully, while some regions of them in the right part are omitted. This is due to the non-uniform lighting condition, as well as the global thresholding operation. In Chapter 2, it was suggested that an adaptive thresholding technique could be used to solve this problem. In this work, the whole image is divided into subimages. Figure 5.4, in which the image f is divided into four subimages f_1 , f_2 , f_3 and f_4 , illustrates this procedure.

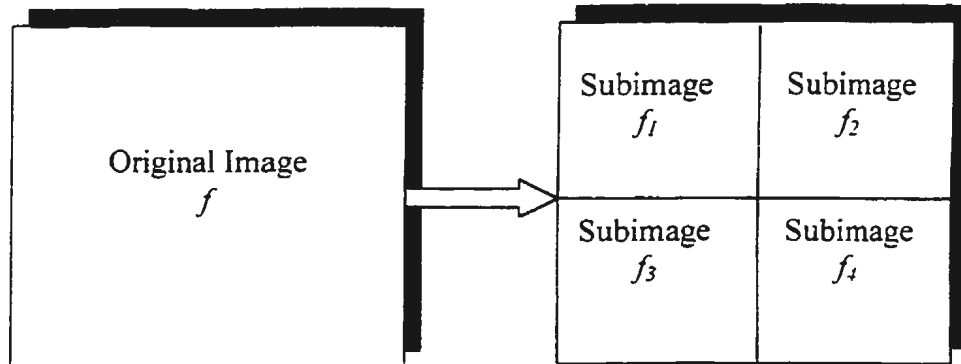


Figure 5.4 Decomposition of the original image

The thresholds are selected locally,

$$T = T(f, f_c)$$

where f is the whole image, f_c is a subimage, and T is the threshold obtained by generating the subimage histogram. The thresholded subimages are combined together to form a whole image. Figure 5.5 shows the adaptively thresholded image. It is clear that there is an improvement in the segmentation of quartz in the left part of the image.

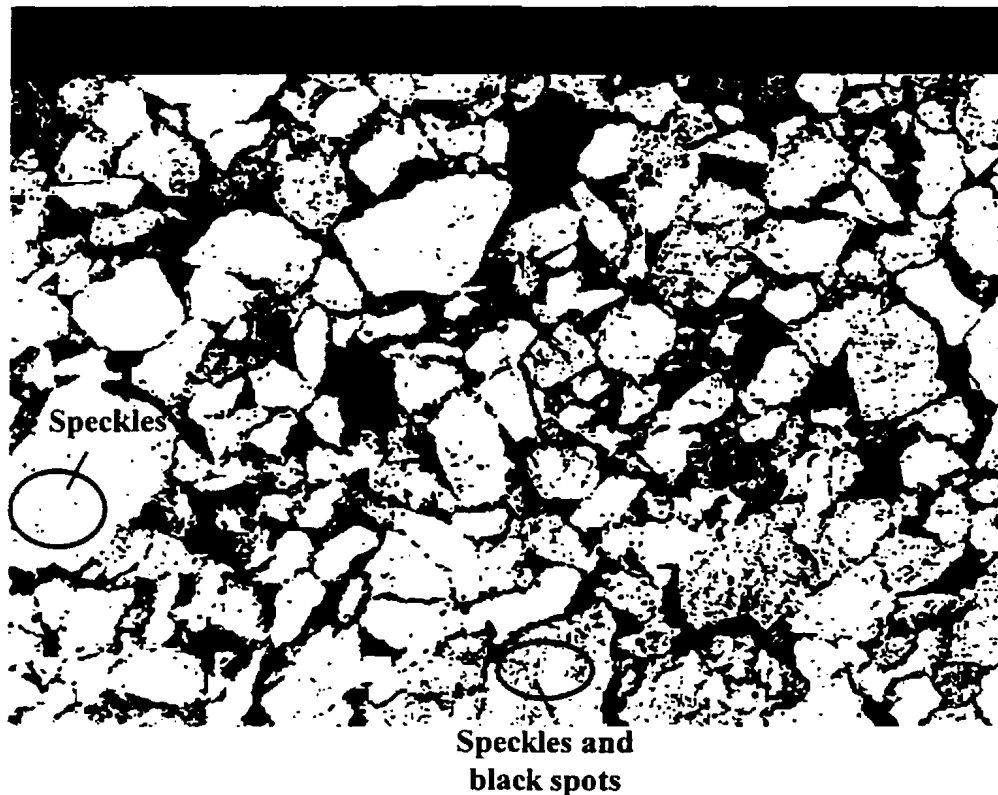
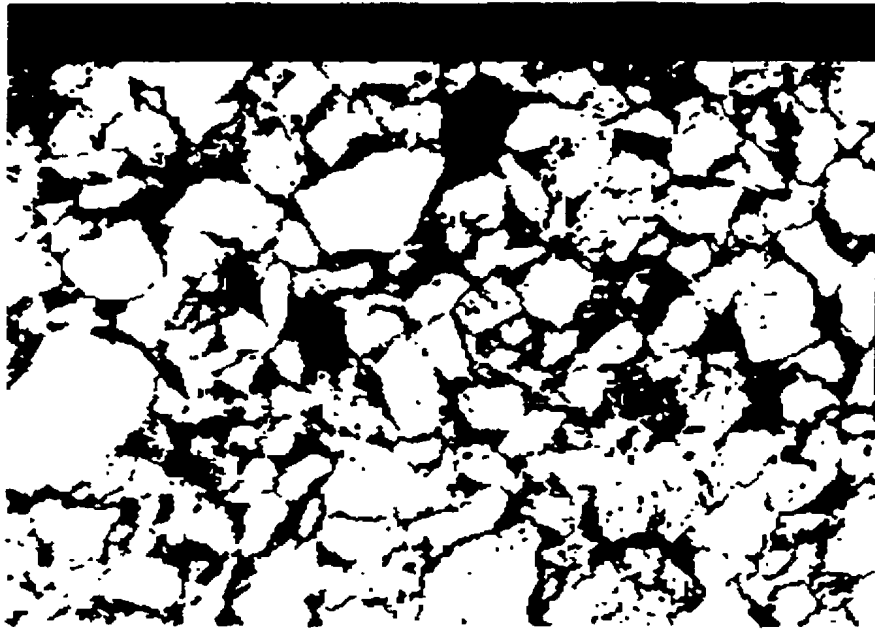


Figure 5.5 Binary image after adaptive thresholding

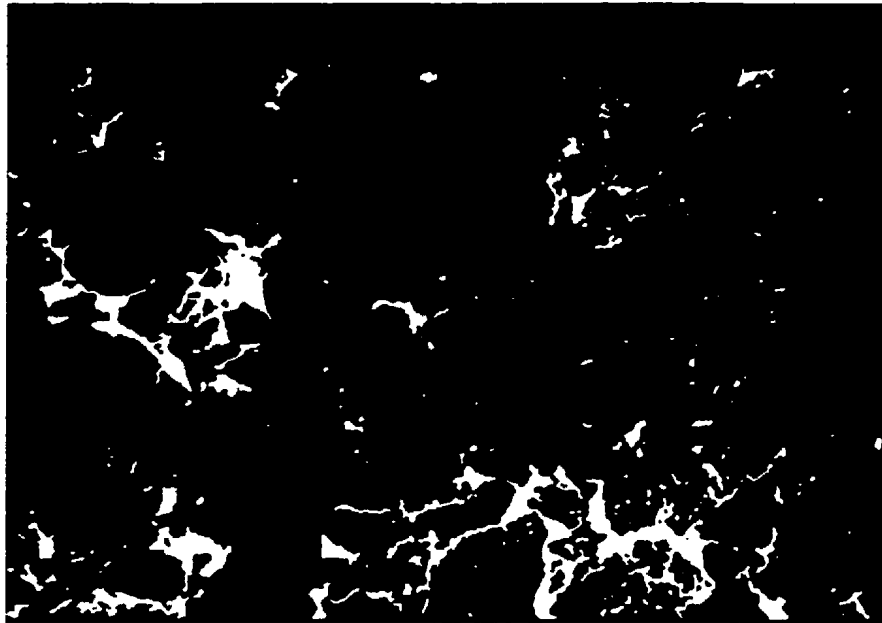
5.2.4 Median Filter for Speckles Removal

In Figure 5.5, most quartz grains are found, and show very precise boundaries. In this stage, the image is “noisy” due to the existence of speckles and black spots that should be included inside the grains during the grain size measurement.

Very tiny speckles can be removed using a median filter. This approach is superior to neighborhood averaging methods since it preserves the sharpness of the edges. One important parameter for median filter is the number of repetitions of the operation. With few repeating, many of the speckles remain, while more repeating degrades the shape of the blob. It was found that repeating the median filter five times gave the best result. Figure 5.6 (a) shows the image after applying the median filter five times, and most speckles inside the blobs are omitted.

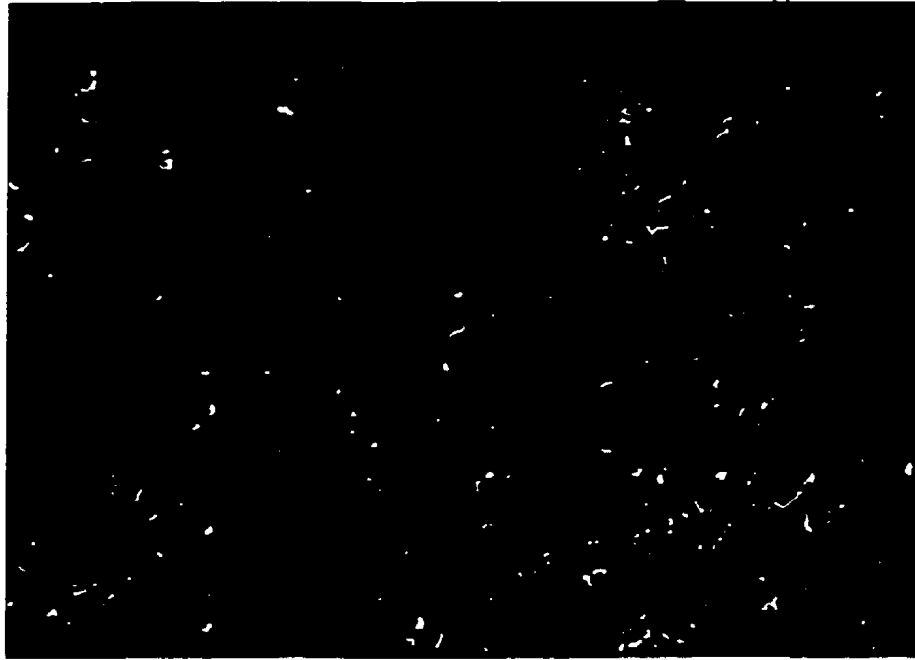


(a)

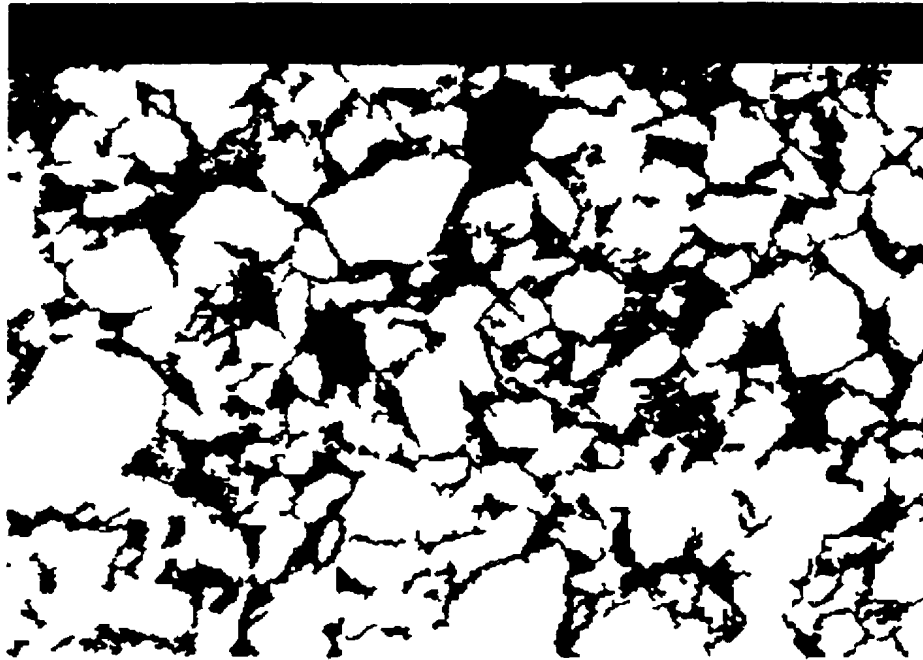


(b)

Figure 5.6 (a) Binary image after applying median filter. (b) all holes extracted from blobs in (a).



(c)



(d)

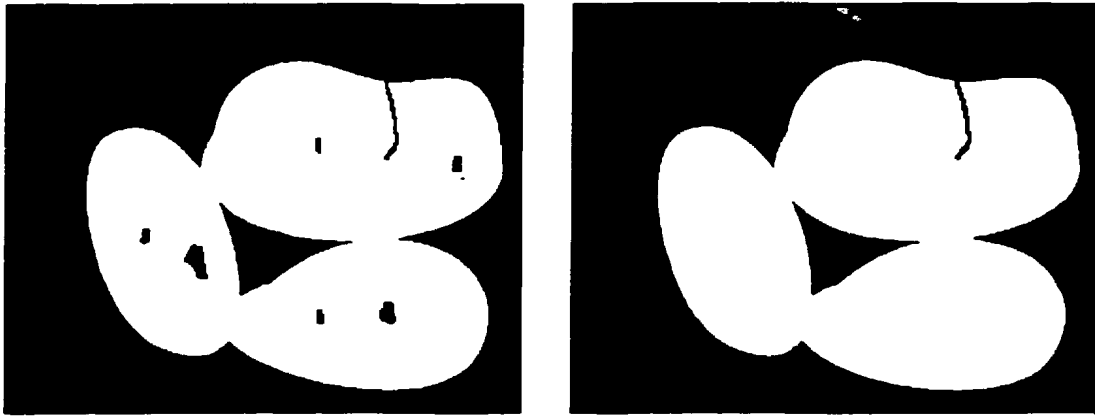
Figure 5.6 (c) Small holes. (d) Binary image after small hole filling. (Continued)

5.2.5 Blob Analysis for the Small Hole Filling

After the speckles are omitted, blobs in the binary image shown in Figure 5.6 (a) are much cleaner, although there are still some holes corresponding to the small black spots inside the quartz grains. These holes should also be included in the corresponding quartz grain during the quartz grain size measuring process. For holes in such a range of sizes, usually 20 – 100 pixels, the median filter does not function well since it works over a small neighborhood, normally a 3×3 window. The blob analysis can be employed, however, to extract holes inside the blobs and fill them. Figure 5.7 illustrates this blob analysis function. The procedure of this operation is following:

1. Extract holes inside all blobs in the given image.
2. Select the area as the blob analysis feature.
3. Calculate the areas of all extracted holes.
4. Select holes whose areas are less than a preset hole area threshold T_{hole} . This is because the bigger holes usually correspond to the pore space in the original image. T_{hole} is set to 100 pixels in this work. Figure 5.6 (b) gives an image with all holes extracted from the image in Figure 5.6 (a). Small holes are black spots, while big ones are the pore space or edges between two grains.
5. Fill the small holes. Figure 5.6 (c) shows the image after hole filling.

Based on the operation applied to this part, the quartz grains are very clean. The side effect of the filling in small hole is that some short edges are also filled and vanish. This will result in the merging of some individual grains.



(a)

(b)

Figure 5.7 Hole filling in blob analysis. (a) An original adjoining blob with holes, a partial edge and a pore. (b) The blob after hole filling

5.3 Watershed Transformation and Adjoining Quartz Grains Separation

In order to measure the grain size accurately, all individual quartz grains should be presented both in terms of shape and size. Some of the objects in Figure 5.6 (d) are touching quartz grains resulting from the quartz overgrowth and the missing edges. Mathematical morphology provides solutions to such problems and the watershed transformation can be applied to disconnect touching objects.

The watershed transformation can be used in conjunction with other operations to disconnect touching particles. According to Vincent [1993] and Matrox Imaging Library [1999], the procedure is as follows:

1. Apply the $\frac{3}{4}$ chamfer distance transformation to the binary object image. The distance for each pixel inside objects is assigned as the distance to the nearest pixel of the background.

2. Perform a watershed transformation. The distance transformation provides a maximum for each blob, also known as a marker, and these maxima are used for forming the catchment basins. Because many maxima are caused by noise, the minimum variation between maxima should be given to obtain real watersheds. In this case, it is set to 2.
3. Perform an AND operation to the original binary image and the image in which contains watersheds for adding watersheds to the objects.

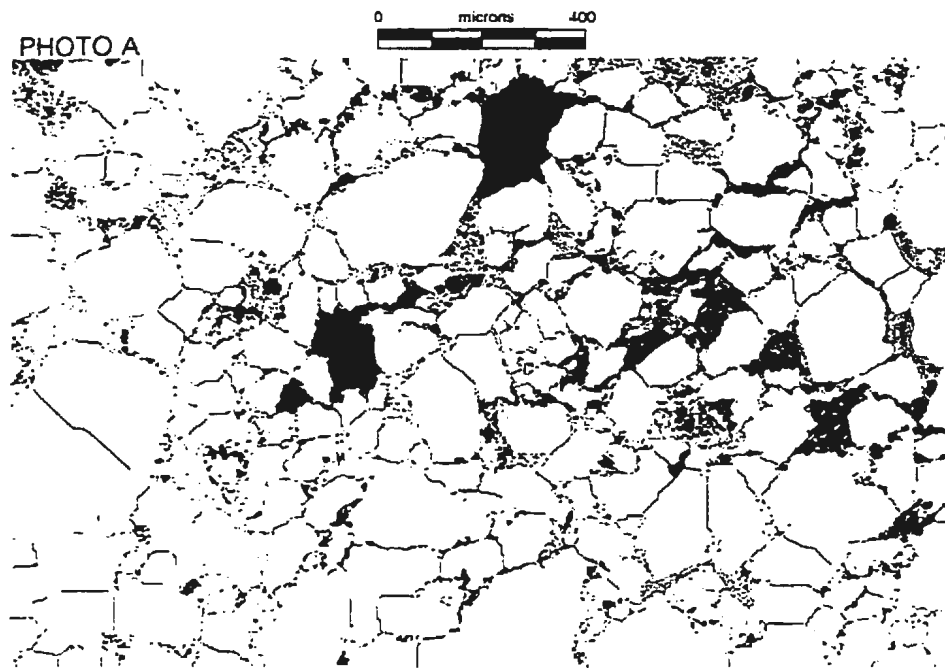


Figure 5.8 Watershed lines added quartz grains

Applying this operation to the binary image shown in Figure 5.6 (d), and Figure 5.8 is the resulting image. It is found that most watersheds correctly split the adjoining grains, while some of them are inside the individual grains in this image. These false lines are marked in Figure 5.9.

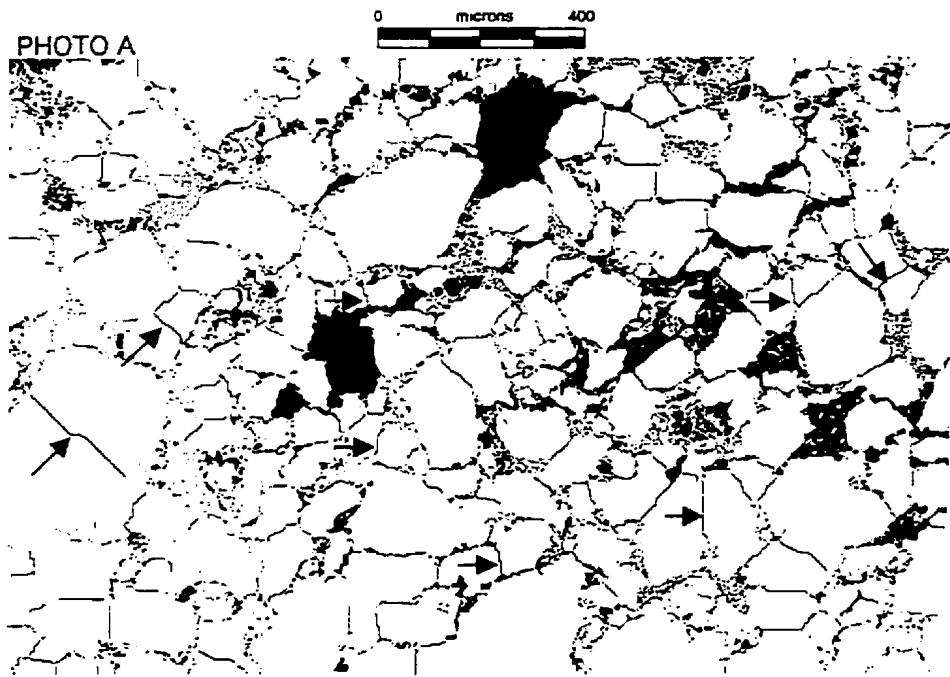


Figure 5.9 Arrow marked false watersheds

Chapter 6

Modified Watershed Transformation and Quartz Grain Measurements

In the first part of this chapter, the watersheds that are obtained by applying a common watershed transformation to the binary image of the segmenting quartz grains are examined. A modified watershed transformation based on the physical nature of quartz grains is developed for correctly disconnecting adjoining grains. The second part of this chapter describes the algorithms for determining the individual quartz grain size, distribution and sorting.

6.1 Modified Watershed Transformation

As shown in Figure 5.9, their natural boundaries plus the watersheds provide the outline for the quartz grains in this image. The watersheds split quartz grains that have been fused by the recrystallization. Also as explained in Section 5.2.5, some edge pixels are removed by the process of filling in small holes in the image objects correspond to quartz grains.

As elaborated in Section 5.3, the maxima of the distance transformation image are used for forming the catchment basins, and some of maxima are caused by noise. Therefore, a minimum variation between maxima should be preset. It has been set to 2 in this work, and proven to be efficient. From the image shown in Figure 5.8, it is evident

that many watersheds split the connected grains correctly, while some of watersheds are incorrect due to oversegmentation as shown in Figure 5.9.

In order to eliminate these false watershed lines, it is necessary to analyze the nature of quartz grains, and find the source of these errors.

6.1.1 Physical Nature of Quartz Grains

Quartz grains are natural materials. In thin section images from the test database, quartz grains appear white with black spots inside. As mentioned in early section, the quartz grain is measured in terms of size, shape and orientation. Since the orientation is chosen randomly during the image digitization, it is not a major factor. The size of the quartz grain is described as very fine (from 0.062mm to 0.125mm), fine (from 0.125mm to 0.25mm), medium (from 0.25mm to 0.5mm), coarse (from 0.5mm to 1mm) and very coarse (from 1mm to 2mm). In the image that is shown in Figure 1.1, the size is fine – medium. Roundness as a shape factor includes very angular, angular, subangular, subrounded, rounded and well rounded. [Scholle, 1979] The comparison chart of the measure of roundness is shown in Figure 6.1.

To summarize, quartz grains varies in size and shape in a great deal. As a result, the generation of watershed lines should respect these factors.

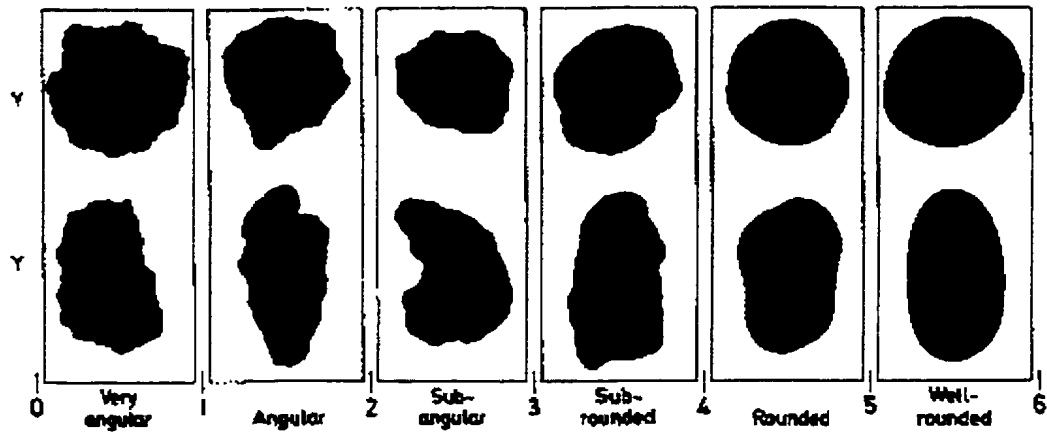


Fig. A-2. Roundness images and classes. Columns show grains of similar roundness but different sphericity (Redrawn from Powers, 1953, Fig. 1)

Figure 6.1 Comparison chart for roundness [Scholle, 1979]

6.1.2 Error Analysis

The watershed lines marked with arrows in Figure 5.9 are false. Some of them are inside individual grains, while the rest corresponds to some dark areas that belong to the quartz grains, but which were missed during the thresholding operation and the small hole filling process. For the false watersheds inside quartz grains, when quartz grains are more angular, the problem arises since the watersheds are generated by setting a minimum variation between extrema, and this variation does not adapt to the angularity factor. In the operation of a common watershed transformation, this value is set to 2. Obviously, the variation factors in some grains are out of range, and result in false watersheds. If this variation is set larger, the watersheds in smaller grains will be omitted since the variation of extrema will be less in these grains. This introduces the need for another factor, the grain size, to guide the watershed generation.

The other error is related to dark spots in the image that is shown in Figure 5.9. In some cases, the holes that do not touch the blob border are extracted during the blob analysis. In practice, some dark parts touch the border, and remain black after the hole filling operation. These spots are actually inside the blob, but appear to have the background color, causing incorrect variation, and resulting in false watersheds. In order to solve this problem, more complex image analysis methods should be developed. For example, texture-based techniques exploit the properties in a pixel's neighborhood. Classification based on different texture patterns may help.

The second type of black spots are inside blobs. As mentioned in Section 5.2.5, a threshold of the maximum grain area T_{hole} is set for the extraction of certain single range of holes. The reason for setting this range is to avoid extracting the pore space whose area is usually larger. Although adaptive thresholding is applied when obtaining the quartz grain, some weak edges are missed, and only the stronger parts obtained in the object image that is shown in Figure 5.4 (b). These partial edges are also extracted as holes. Since the areas of these edges are usually larger than those of black spots, a threshold of the area is selected in order to maintain these edges. The black spots with the same size range as edges, however, are also maintained.

In this research work, only the false watersheds inside quartz grains are considered, while those caused by black spots will be the future work.

6.1.3 Introduction to the Modified Watershed Transformation

It was found that using only the common watershed transformation would produce an image that require significant manual editing. In view of this, a modified

watershed transformation was developed which would operate with regard for the physical nature of quartz grains.

In this thesis, the approach is to examine all watersheds that are obtained by applying the common watershed transformation, add true watersheds and deleting false. The main idea behind this approach is that it was found that there is a significant difference between the lengths of the watershed and lines in parallel to the watershed. By establishing an acceptable difference which is adaptive to the grain size, false watersheds can be found and deleted using an approach illustrated in Figure 6.2.

Figure 6.2 (a) illustrates two different types of grains. The left image represents two cemented grains, while the right image is a single long grain. Figure 6.2 (b) shows the resulting grains after applying the common watershed transformation, and two watersheds in which the left one is true, and the right is false are added to the grains. In Figure 6.2 (c), two lines are drawn in parallel with the watershed in each grain. It is apparent that the length differences between the watershed and its two parallel lines are larger for the left hand grain, while smaller in the right. Figure 6.2 (d) shows the resulting grains after the watershed line examination. The left hand-watershed is retained, while the right hand grain has the watershed line removed.

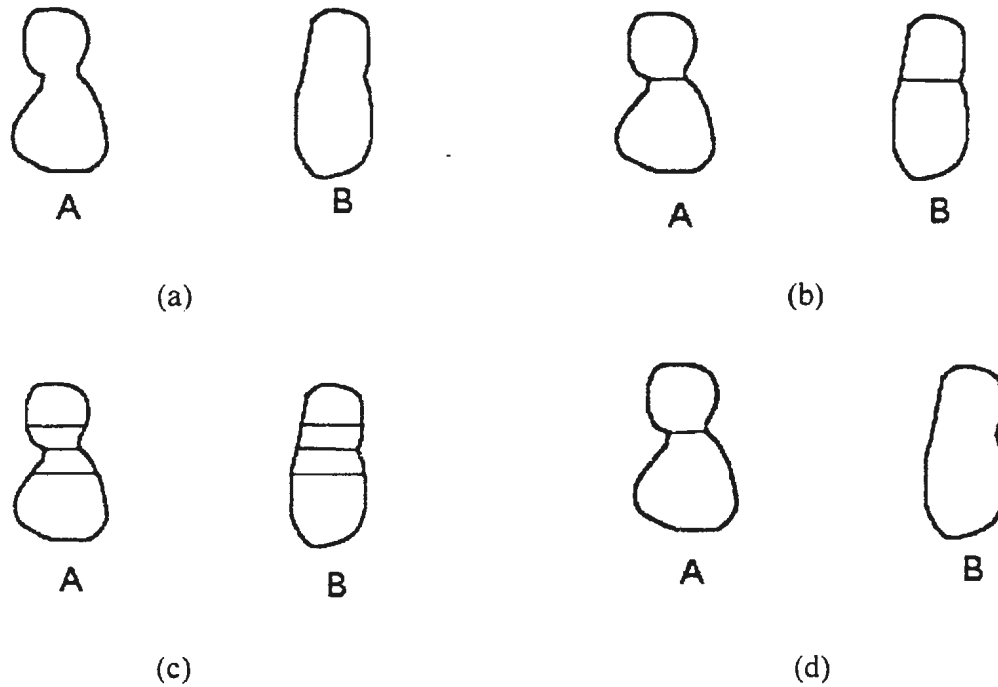


Figure 6.2 Illustration of the watershed line examination

6.1.4 Algorithms for Modified Watershed Transformation

The algorithm starts from the examination of watersheds after the common watershed transformation has been applied. The procedure is as follows:

1. Measure the lengths of all watershed lines, and extract those longer lines and add shorter ones to the object image directly. This step is applied because blob analysis is time consuming, and it is desirable to reduce the number of blobs. In addition, shorter lines are usually correct since the error commonly occurs in the large and long grains. A line length threshold T_{line} is set to 18 pixels experimentally.
2. Examine individual watersheds, obtain and store the central points of all watersheds.

3. Reconstruct every individual watershed from its central point.
4. Draw two lines in parallel with each watershed in a certain distance d which is chosen according to the average maximal blob diameter. $d = 12$ in this case.
5. Measure the lengths of every two lines corresponding to each watershed and calculate the differences.
6. Determine the difference threshold $T_{linedist}$ based on the length of the current watershed. Preset the $Threshold_Base = 16$ experimentally, assume the length of the watershed is $L_{watershed}$ and calculate the ratio.

$$\left. \begin{array}{l} ratio = 0, \quad L_{watershed} < 45 \\ ratio = (L_{watershed} - 45) / 60, \quad L_{watershed} > 45 \end{array} \right\} \quad (6.1)$$

Then

$$T_{linedist} = (1 + ratio) \times Threshold_Base \quad (6.2)$$

7. Compare the two distances to the threshold.
8. Add the watershed line to the object image if both distances are less than the threshold, or skip this watershed if one or both of two distances are greater than the threshold.

6.1.5 The Line Drawing Algorithm

Since the watershed line examination involves a line length comparison inside the grains, linear geometry and blob analysis are employed in the development of the algorithm. After the lengths of all watersheds are measured, each watershed is examined individually. Two lines are drawn in parallel at a set distance d from the watershed. One typical blob with the related lines and points is drawn in Figure 6.3. In this figure, the

line A is the watershed, while B and C are lines in parallel with it and D is a perpendicular line of A . B and C and passes the point a . a is the central point of the line A , which is the watershed, while b and c are the intersections of the lines B and D , and C and D , respectively.

Assume the coordinate of the point a is (x_0, y_0) , and the slope of the line A is k_0 . From these known data, the points b and c , as well as the equations of the lines B and C are derived.

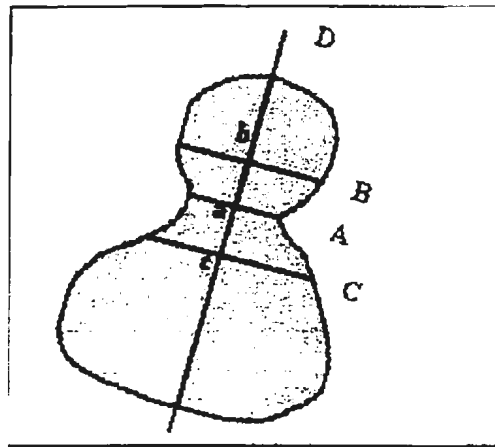


Figure 6.3 A typical blob

Assume the coordinates of points b and c are (x_b, y_b) and (x_c, y_c) , respectively, and that the distance d that between A and B (and A and C) is known. Then

$$\begin{aligned}
x_b &= x_0 - \frac{d}{\sqrt{1 + 1/k_0^2}} \\
y_b &= y_0 + \frac{d}{k_0 \sqrt{1 + 1/k_0^2}} \\
x_c &= x_0 + \frac{d}{\sqrt{1 + 1/k_0^2}} \\
y_c &= y_0 - \frac{d}{k_0 \sqrt{1 + 1/k_0^2}}
\end{aligned} \tag{6.3}$$

Let

$$\begin{aligned}
a_1 &= \frac{d}{\sqrt{1 + 1/k_0^2}} \\
a_2 &= \frac{d}{k_0 \sqrt{1 + 1/k_0^2}}
\end{aligned} \tag{6.4}$$

Then

$$\begin{aligned}
B: \quad y &= k_0 x + y_0 - a_2 - k_0(x_0 + a_1) \\
C: \quad y &= k_0 x + y_0 - a_2 - k_0(x_0 + a_1)
\end{aligned} \tag{6.5}$$

Based on Equation 6.3, the two lines are drawn inside the current blob according to the following procedure:

1. Find the start point and the end point for each line. The minimum and maximum coordinates of the current blob restrict the line in a certain range, while the coordinates of the start and end points are determined by these values. First select the x-coordinate, and the y-coordinate is calculated based on the Equation 6.3. There is a very important aspect in drawing lines in a digital image since the coordinates calculated based on a line equation are

possibly decimal, but they should be integer. According to the Bresenham algorithm [Appendix II], the y-coordinate is determined by finding the nearest integer of the calculated value.

2. Drawing two lines from their start and end points. In the Bresenham algorithm, the coordinate of each pixel on a line is determined by the "nearest integer" rule to form a step function, and the line will look most natural to human eyes.

These two lines are drawn in a box that is defined by the minimum and maximum coordinate of the current blob. Then an AND operation is performed between these lines and the blob for obtaining the lengths of lines inside the blob.

6.1.6 Special Cases in Implementation of the Line Drawing

Algorithm

Due to factors, such as the irregular shapes of grains, boundary problem, some special cases occur during the line drawing implementation procedure.

Case 1: The watershed is reconstructed from its centroid. Unfortunately, this point is sometimes not on the watershed due to the integer and decimal conversion problem. The distance between the centroid and its real position on the watershed is usually one pixel. A local neighborhood search solves this problem.

Case 2: When a watershed near the image boundary, the y-coordinate is checked to confirm if it is out of range.

Case 3: Figure 6.4 shows an irregular blob. In this case, three lines will be obtained after the AND operation is performed. The line on the right part of the blob can

not be used for the line comparison since it does not reflect the local variation of the blob width, and needs to be removed. The solution is to find the line near the watersheds which is in the left part of the blob, and delete the other.

Case 4: When the angle of a watershed is 0° or 90° , the start and end point should be determined separately since invalid values will be produced using Equation 6.3.

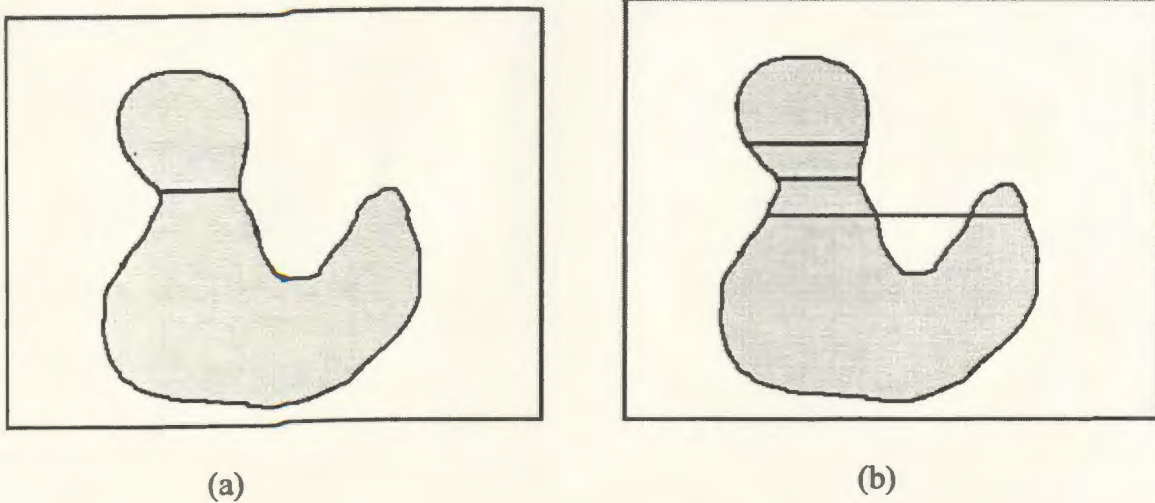


Figure 6.4 An irregular grain

Case 5: When a watershed touches the image border, the situation is extremely complicated. It is difficult to determine the threshold for the length variation that is determined by the watershed length since it cannot be predicted that the partition of a grain inside the image, such as a half, more than a half or less than a half. In this case, all watersheds that touch the image borders are added directly to the object image.

There are some other cases in this procedure, such as two connected watersheds. These connected lines are added directly into the image. Figure 6.5 gives the resulting image after the watershed line examination. It was found all watersheds inside the long and large grains are deleted.

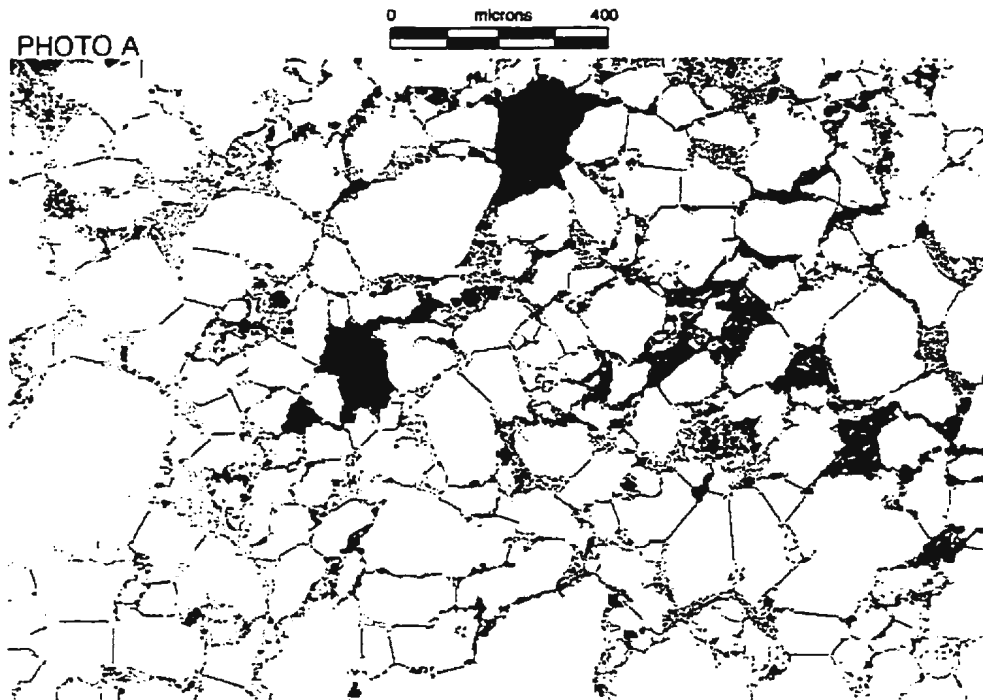


Figure 6.5 Resulting image after the watershed line examination

6.2 Quartz Grain Measurement

As mentioned in Section 5.2, the grain size is a very important parameter in thin section analysis since it is a visible property. Size is also considered as the basis for grain classification. Reservoir geologists and engineers are interested in information about individual grains as well as statistical derived from a large number of grains, such as the size mean, distribution histogram and sorting.

6.2.1 NSD of the Grain

According to Kennedy and Mazzullo [1991], the nominal sectional diameter (NSD) is an appropriate size descriptor. The NDS is defined as “the diameter of a circle with the same area as the maximum projection profile of a single particle”. In thin section

image, the objects are on a single plane, so that the maximum projection profile of a particle will be the area of this particle, and this area is readily converted into its NSD.

6.2.2 NSD Distribution Histogram

The statistical information of the grain size is generated from an entire image or even a whole slide. After the area of each grain is available, its NSD can be determined by converting the area to this diameter. In this research work, the NSD distribution histogram is plotted by percentage vs. microns. This provides for a direct comparison between samples. The software is designed to determine the size distribution in a whole image or in a user-selected range. The histogram of quartz grains for the image shown in Figure 5.1 is given in Figure 6.6.

6.2.3 Sorting

Sorting of the object sizes in a thin section image is useful for core analysis. The sorting class provides an idea of whether the grains are fairly uniform in size. Usually, the grains are not perfectly sorted. According to Folk [1980], the sorting is most likely to be decided by the size range in the image. If the size range is large, it will not be well sorted. The conventional sorting classes include very well sorted, well sorted, moderately sorted and poorly sorted. Figure 6.8 shows the sorting images.

In Folk's book, four values in the Phi (ϕ) standard deviation shown in Figure 6.7, correspond to the sorting classes, very well sorted ($\phi=0.35$), well sorted ($\phi=0.5$), moderately sorted ($\phi=1.0$) and poorly sorted ($\phi=2.0$).

In order to determine the sorting class of grains within an image, a estimation is made to obtain the values related to the sorting classes. From statistics, it is known that the standard deviation is the most frequently used measure of dispersion, and "takes into account every observation in the data set". [Levin, 1984] The following formula is used to calculate the standard deviation:

$$\sigma = \sqrt{\frac{\sum(x - \mu)^2}{N}} \quad (6.4)$$

where σ is the standard deviation, x is the observation, μ is the population mean, and N is the total number of elements in the population.

From the equation 6.4, it is found that the standard deviation is still related to the grain size. In other words, if the values of all elements in a data set are relatively large, the standard deviation will be large since $x - \mu$ is large. A value of the relative difference should be found. A relative deviation is defined as follows:

$$S = \sigma / \mu \quad (6.5)$$

where S is the relative deviation, σ is the standard deviation and μ is the population mean. Table 1 shows a comparison of values in Figure 6.8 and using equation 6.5.

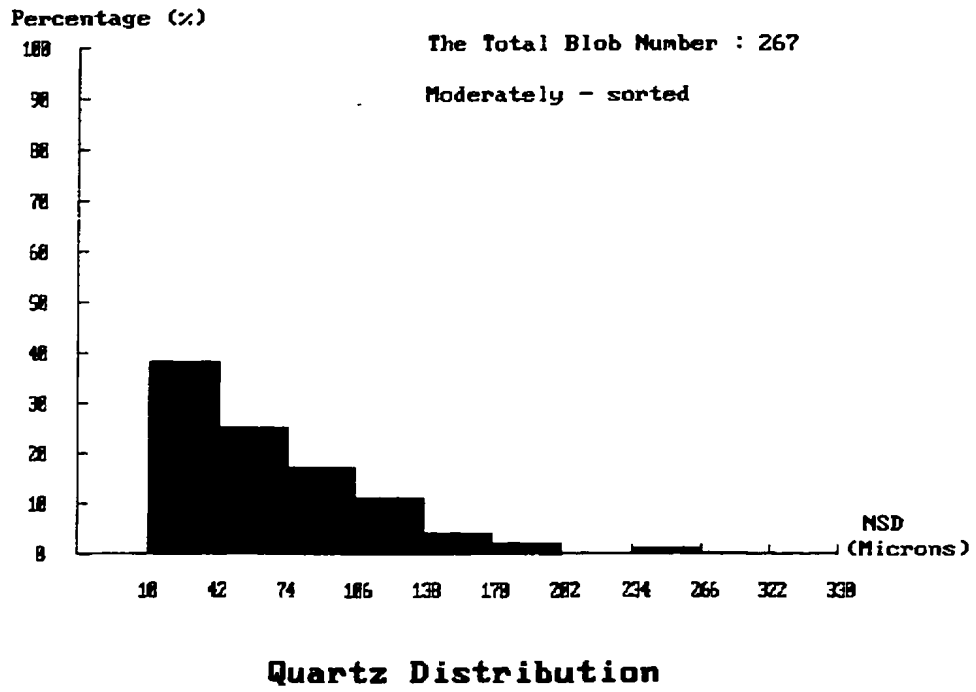


Figure 6.6 Quartz grain distribution histogram

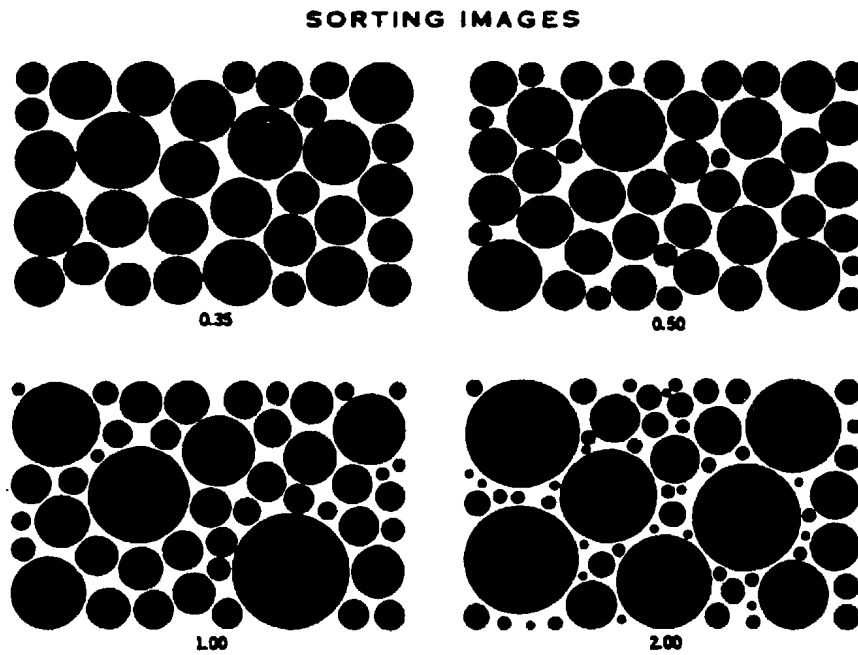


Figure 6.7 Sorting images Folk [1980]

Table 1 Comparison chart for sorting and sorting classes

| PHI standard deviation | Relative standard deviation | Verbal scale |
|------------------------|-----------------------------|-------------------|
| 0.35 | 0.2377 | very well sorted |
| 0.5 | 0.4017 | well sorted |
| 1 | 0.5542 | moderately sorted |
| 2 | 0.9989 | poorly sorted |

From Table 1, it is found that the range for the relative standard deviation is narrower than that for PHI standard deviation. Nevertheless, it separates four sorting classes and is easy to compute.

Chapter 7

Results, Integration into Windows Tools and Discussions

In this chapter, the algorithms for both porosity determination and quartz grain measurements are tested and the results are presented. These algorithms have also been integrated into a graphic user interface (GUI) of the PetroGraFX software tool which is under development at C-CORE. Discussions about the results are also given in this chapter.

7.1 Results and Discussions

The test image set is from #B-16 2 well of the Hibernia oilfield. It contains 70 thin section images in two resolutions. X63 and X125. In this test image set, X63 images contain about 100-400 grains, and are good for quartz grain measurements, especially the distribution and sorting, while X125 images display the inter or intra-crystal porosity in a certain part of the thin section clearly.

The PetrograFX software is implemented using Microsoft Visual C++ 6.0 and Matrox Imaging Library 6.0 in which many built-in functions for image processing and analysis, such as the distance transformation, the watershed transformation and blob analysis, are provided.

7.1.1 Porosity Determination

All 70 images were tested for the pore space segmentation and porosity determination. The pore space is converted into pink in all resulting images. Currently, the manual (visual) estimated porosity database is being built, and is not available in the present, so that all resulting images were visually reviewed by a specialist. It was found that 67 image processed using the algorithm in this thesis presented very good pore space segmentation based on a visual assessment by an expert. Only 3 of segmented images showed any evidences of missing very small pores. When the resulting images are obtained from X125 original images, the region growing in the pore space show details along the boundaries. Figure 7.1 gives a X125 image and its resulting image. In this figure, the pixels of light blue near the boundaries are picked up. Figure 7.2 gives four original images shown in Figure 1.5 and their corresponding resulting images.

From Figure 7.2, the resulting images (b), (d) (h) exhibit good pore space segments. all visible pores were found accurately. Figure 7.2(f) illustrates a case where there are some small pores missing, in which the color is very light annotate the image. In fact, Figure 7.2 (f) represents a class of images, in which pores are dimmer, and quartz grains are darker than usual. In this experiment, the parameters for thresholding and region growing are adjusted automatically according to the red channel histogram for this class of images, so that most pores are found.

There are 3 images which display missing pores in this experiment. Figure 7.3 (a) and (c) are the other two originals, while (b) and (d) are the resulting images. In Figure 7.3 (b), there are several very tiny pores missing due to the light color. Figure 7.3 (c) is a special case, in which the pores appear an unusual dark blue. Some regions are too dark

to be recognized as the pore space; therefore, it results in some central parts of pores being omitted.

Overall, the defect images in the pore space segmentation are less than 5% in total. These images have been reviewed by reservoir specialists and it has been noted that the pore space segmentation is very good.

7.1.2 Quartz Grain Measurements

Since the manual segmentation images are very expensive, the limited testing for quartz grain measurements has been carried on three thin section images for which a reservoir specialist provided manually segmented images as a basis for comparison. Figure 7.4 illustrates segmented quartz grains that are converted into white in the auto segmentation mode ((a), (c) and (e)) and the yellow manual segmentation by the reservoir specialist ((b), (d) and (f)). Table 2 provides a comparison chart for these two modes.

From Figure 7.4 and Table 2, some observations are made:

1. The percentage of quartz grains in manual mode is higher than that in auto mode in Figure 7.4 (a) and (b). The reason for this is that some dark quartz grains that contain rich texture properties are excluded in the auto mode segmentation process. Noticed that this factor in two modes is nearly identical in Figure 7.4 (c) and (d), and (e) and (f) since most quartz grains are texture free, while there are 7.2 % differences between (a) and (b).
2. The numbers of grains in Figure 34 (c) and (d) are very close. This number in (c) is larger than that in (d) due to the merging of some small grains caused by missing edges. The number difference is more significant between (a) and (b),

and (e) and (f) since some small grains were omitted in the manual mode and then included in the auto mode.

3. The relative standard deviation in manual mode is smaller than in the auto mode. The main reason for this is that some individual quartz grains which contain rich textural properties are decomposed into several small grains during the thresholding operation. The minor reason is some relatively big black spots inside the quartz grains are omitted from the small hole filling process and lead to false watersheds to form smaller segments. From the NSD mean, the difference of this factor between Figure 7.4 (a) and (b) is significant because the texture reason, while this difference is also large between Figure 7.4 (e) and (f) due to the small grain exclusion in the manual mode.

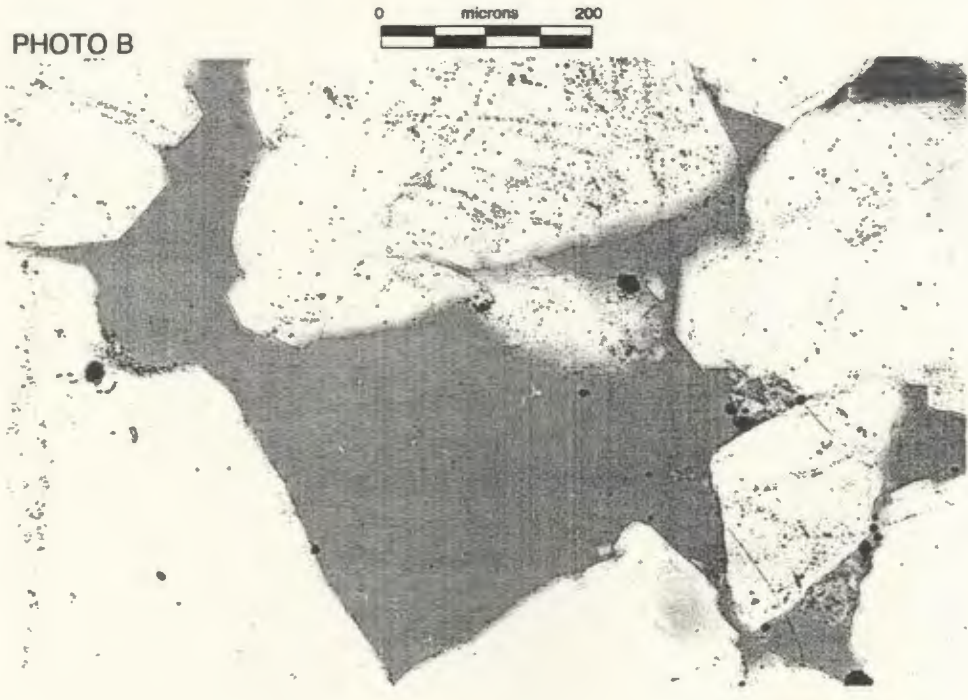
Despite the differences in the extracted parameters, it was concluded by the reservoir specialist that the initial grain segmentation results were very promising.

Table 2 Comparison Chart for Auto and Manual Quartz Grain Segmentation

| Image Number | Segmentation Mode | Total Grain Number | Relative Standard Deviation | NSD Mean (Microns) | Percentage of Quartz Grain |
|--------------------------------------|-------------------|--------------------|-----------------------------|--------------------|----------------------------|
| 417870a (Figure 7.4 (a) & (b)) | Manual | 233 | 0.3682 | 41.2146 | 67.968 |
| | Auto | 278 | 0.5645 | 33.0576 | 60.742 |
| 418527a (Figure 7.4 (c) & (d)) | Manual | 281 | 0.6074 | 34.6940 | 70.212 |
| | Auto | 267 | 0.7521 | 33.1423 | 69.461 |
| 434646a (Figure 7.4 (e) & (f)) | Manual | 215 | 0.5226 | 36.8419 | 56.276 |
| | Auto | 303 | 0.8020 | 27.6469 | 57.609 |

7.2 Integration into the Windows Tool

The software for porosity determination and the quartz grain measurement has been integrated into a graphic user interface (GUI) using MFC in Microsoft Visual C++ 6.0 in Windows environment. Besides the two processing tools, the GUI also provides a user interface for examining thin section images. Small windows within an image can be selected for processing if desired. Figure 7.6 illustrates the PetrograFX GUI window.

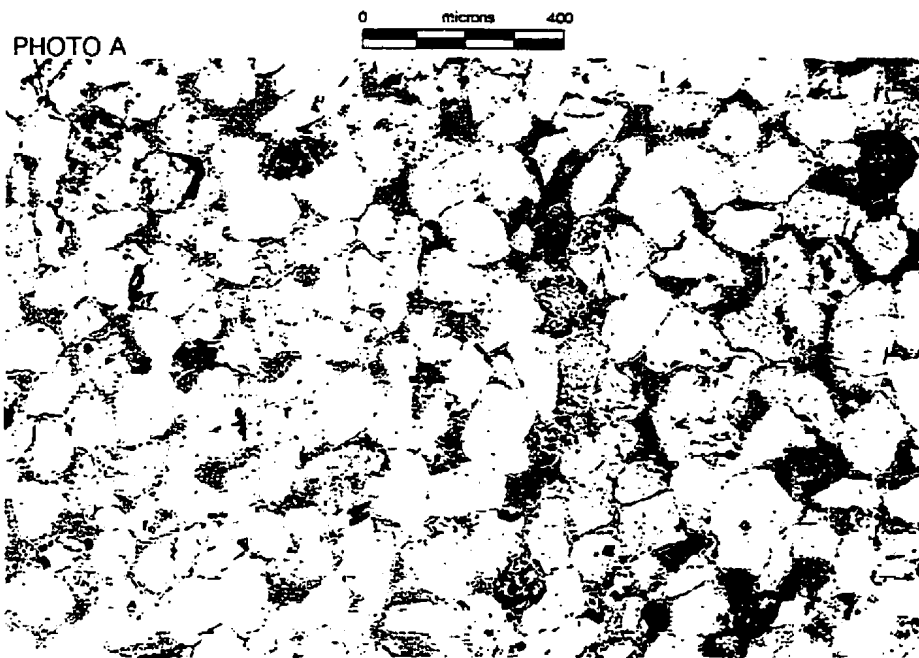


(a)

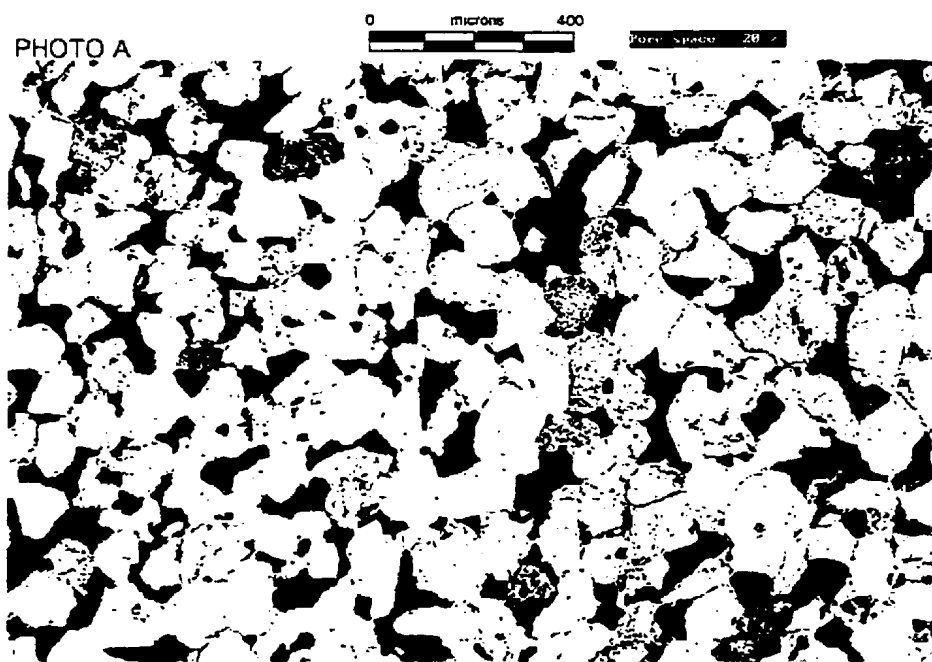


(b)

Figure 7.1 (a) X125 original, (b) pore space resulting images,

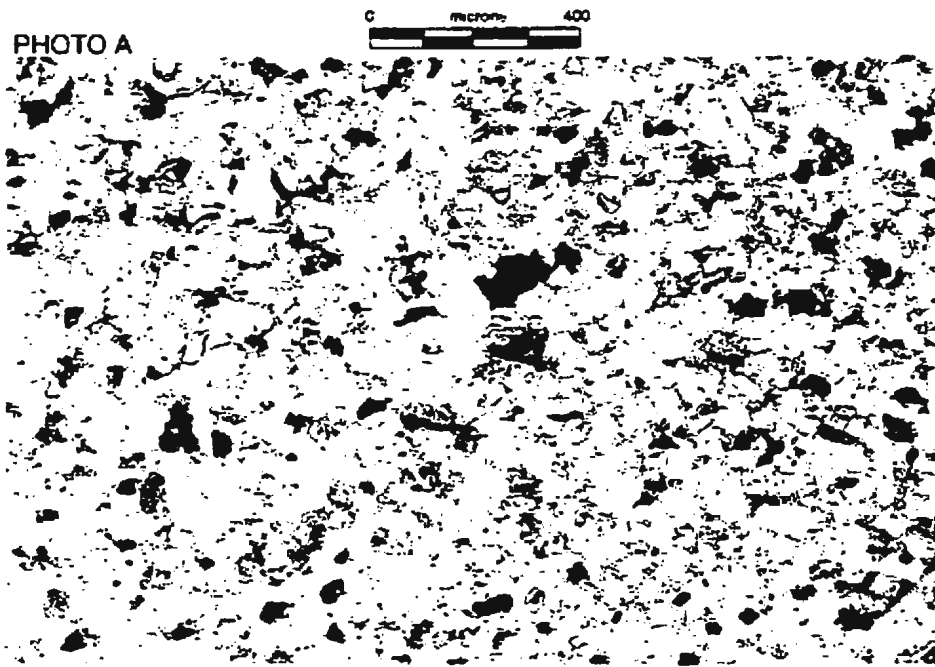


(a)

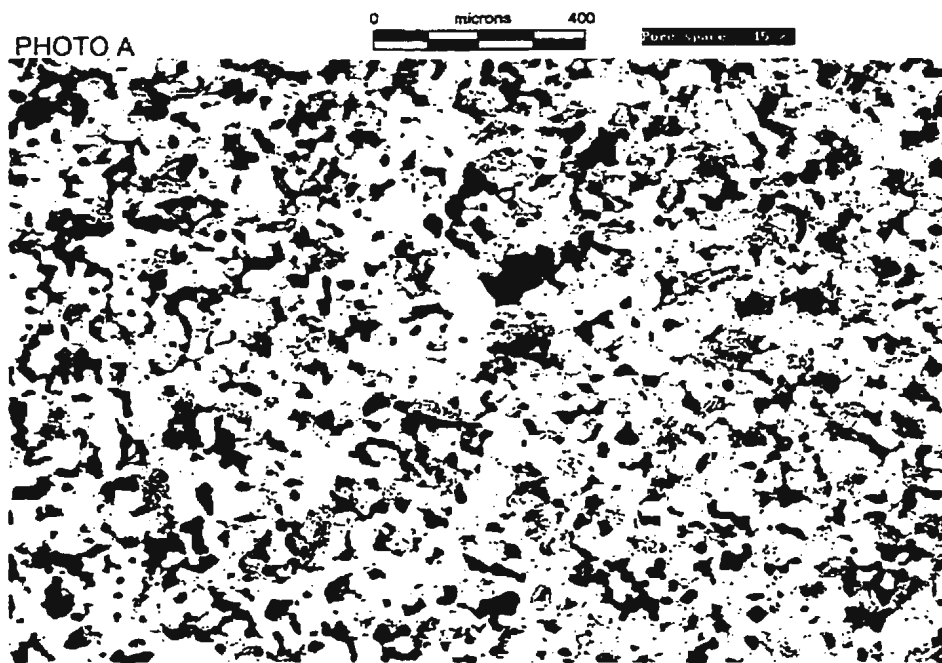


(b)

Figure 7.2 Resulting images for the pore space segmentation

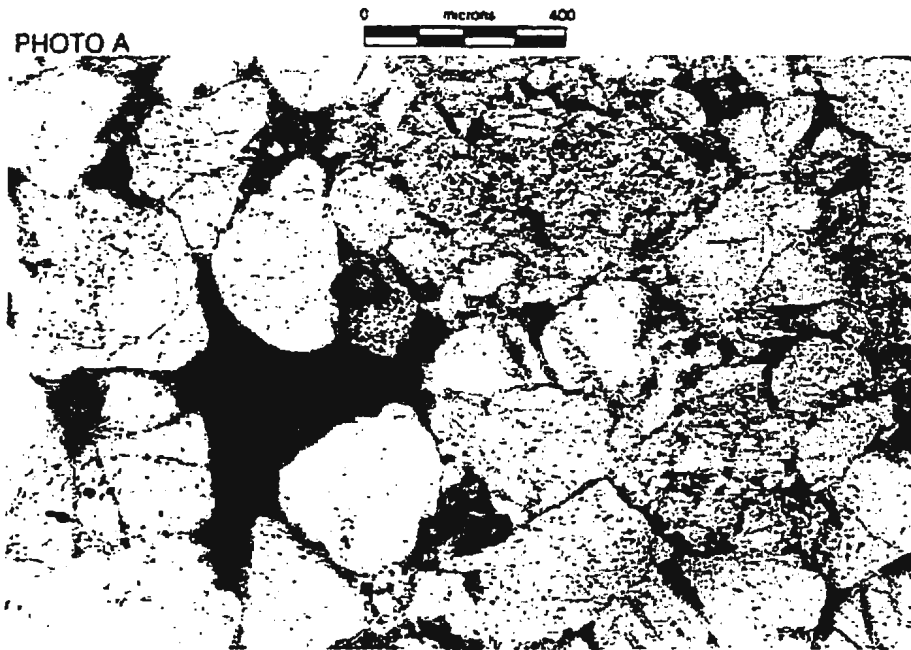


(c)

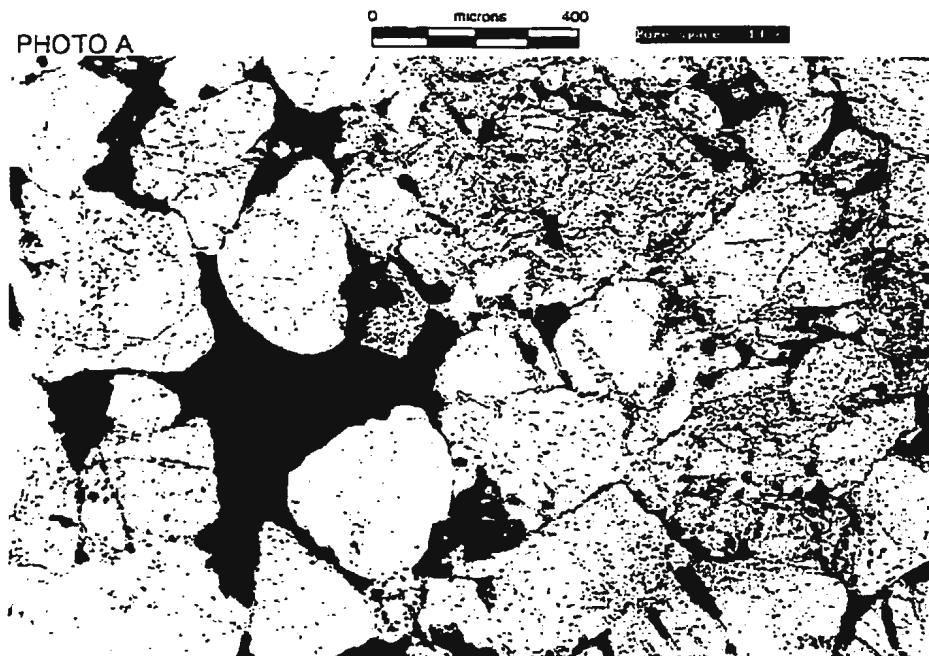


(d)

Figure 7.2 Resulting images for the pore space segmentation (Continued)

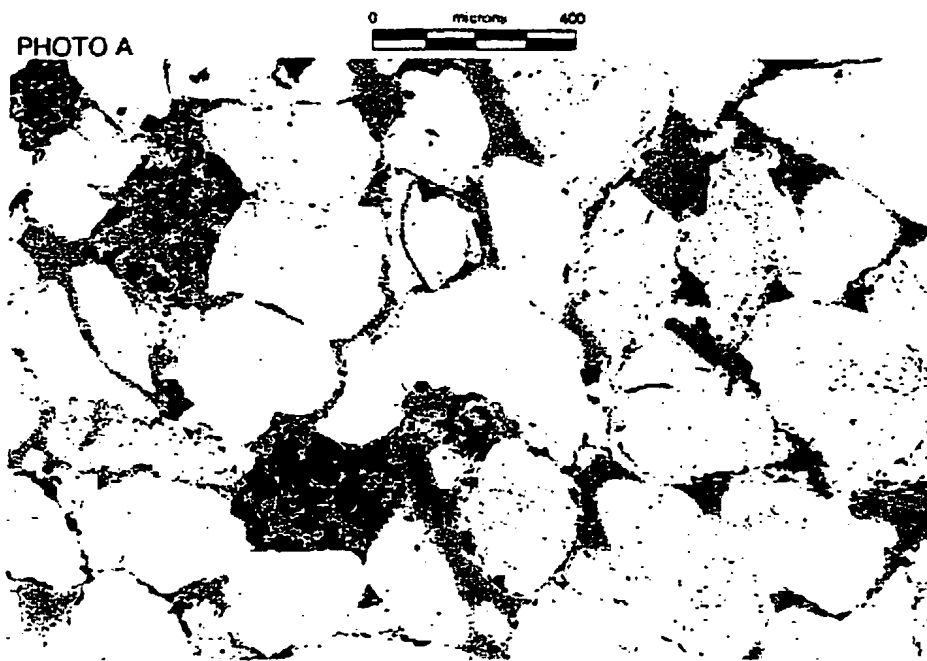


(e)

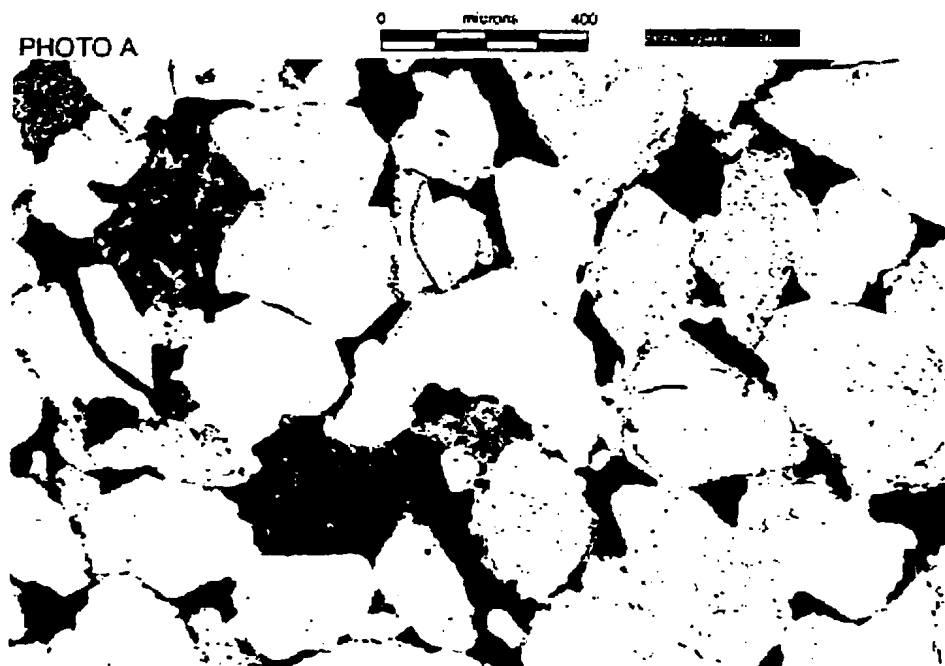


(f)

Figure 7.2 Resulting images for the pore space segmentation (Continued)

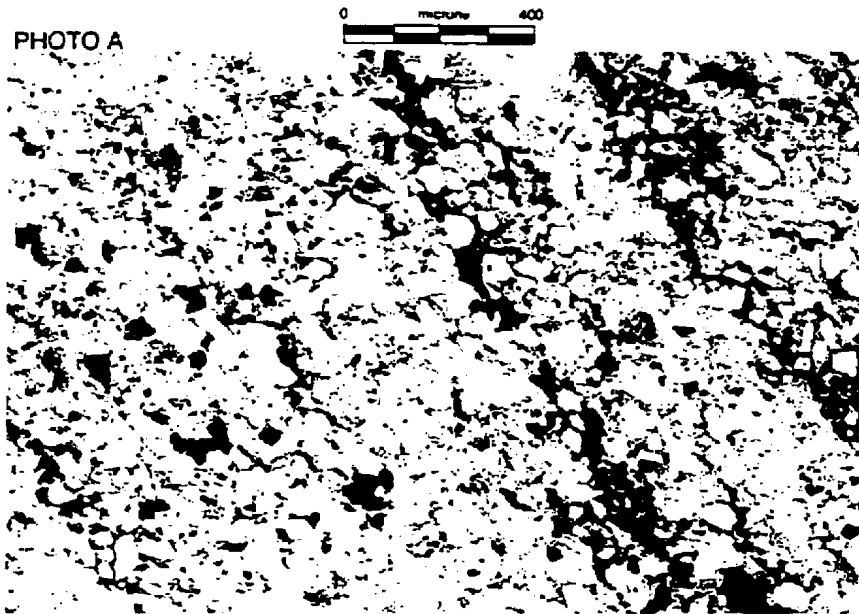


(g)

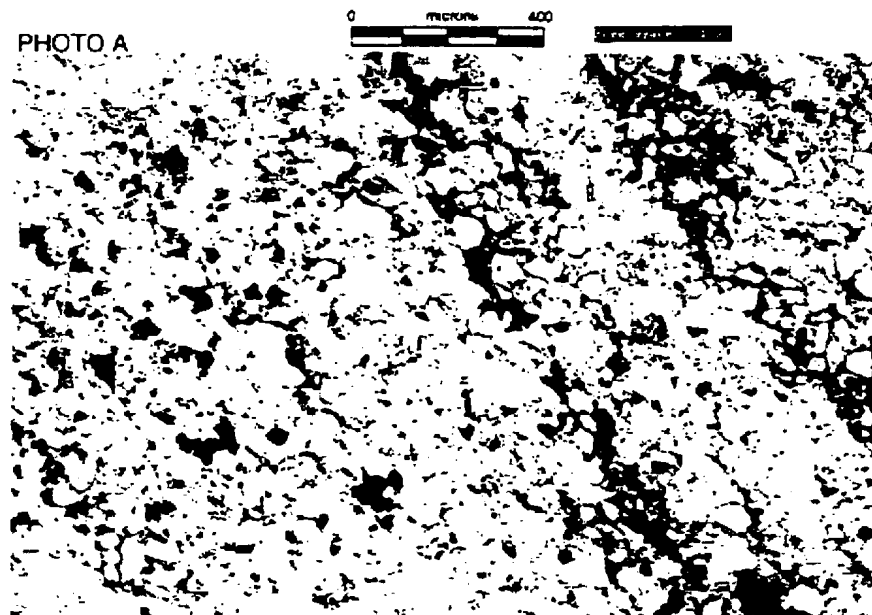


(h)

Figure 7.2 Resulting images for the pore space segmentation (Continued)

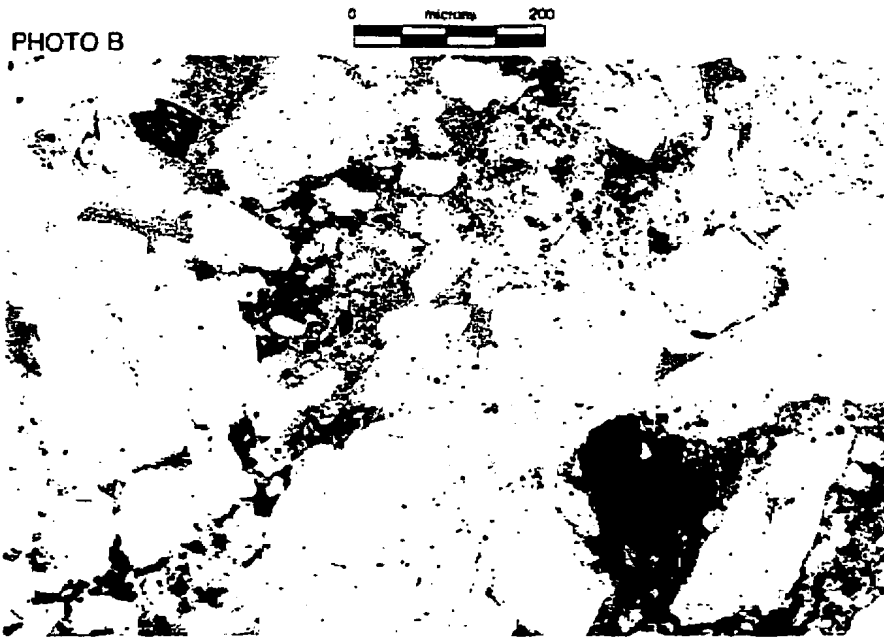


(a)

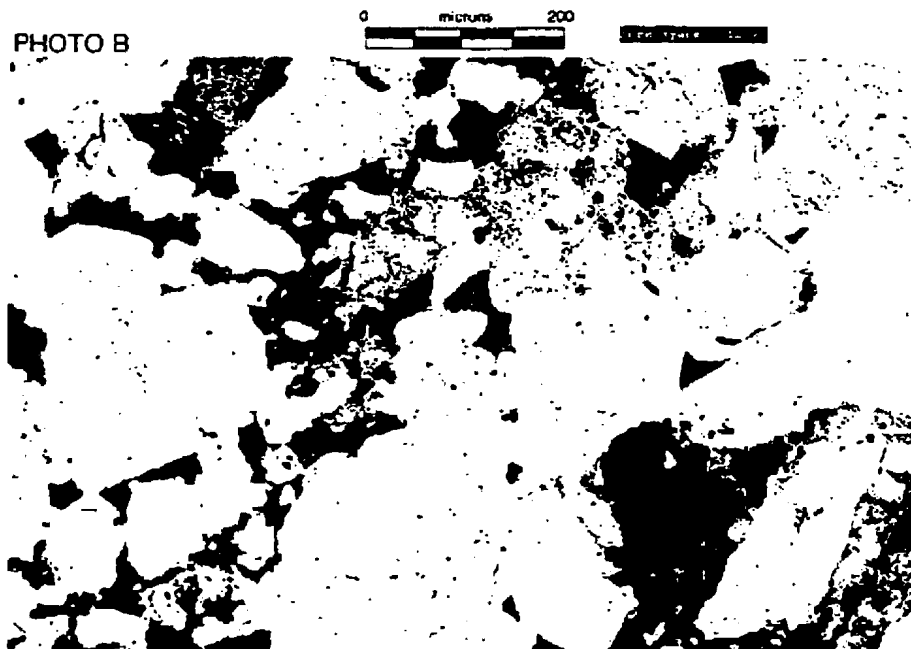


(b)

Figure 7.3 Defect resulting images for the pore space segmentation

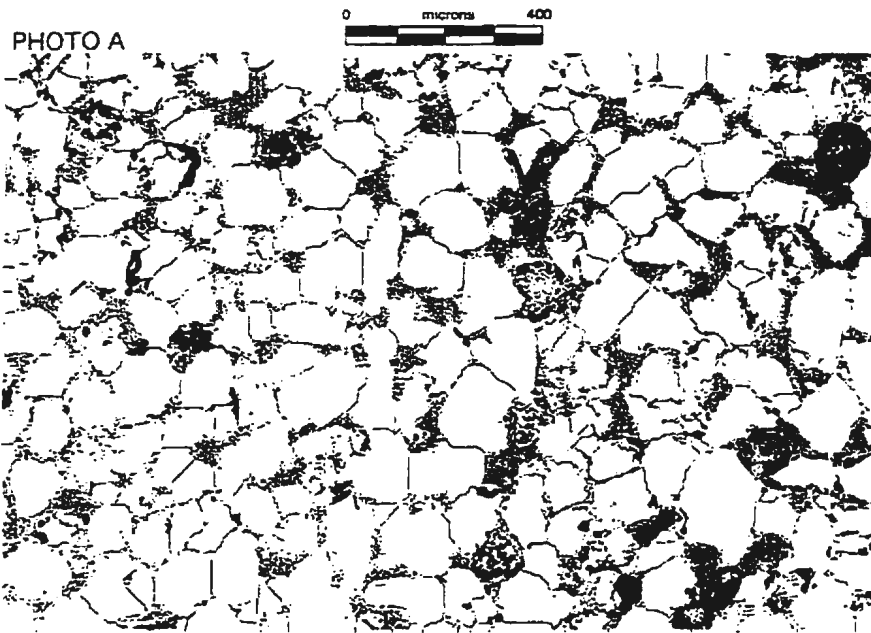


(c)

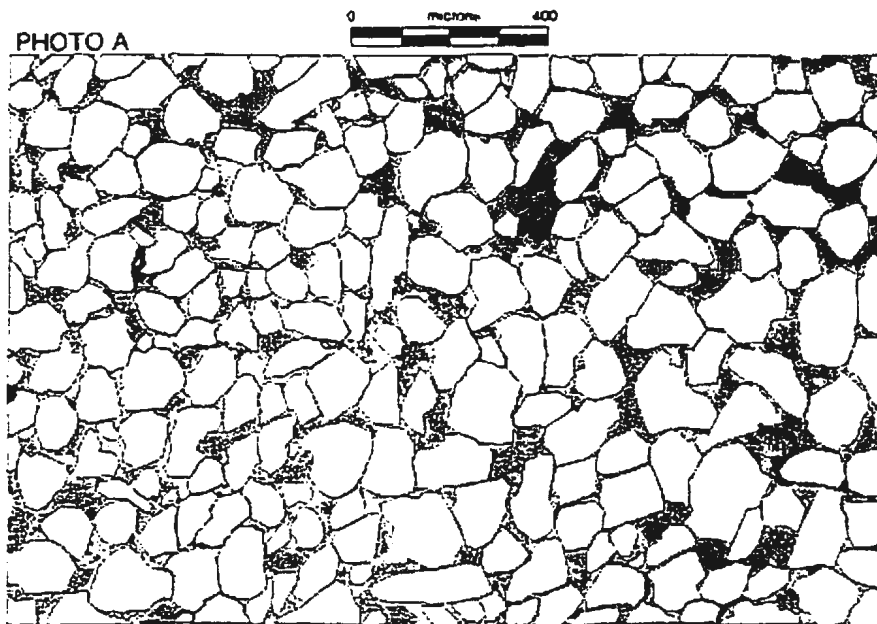


(d)

Figure 7.3 Defect resulting images for the pore space segmentation (continued)

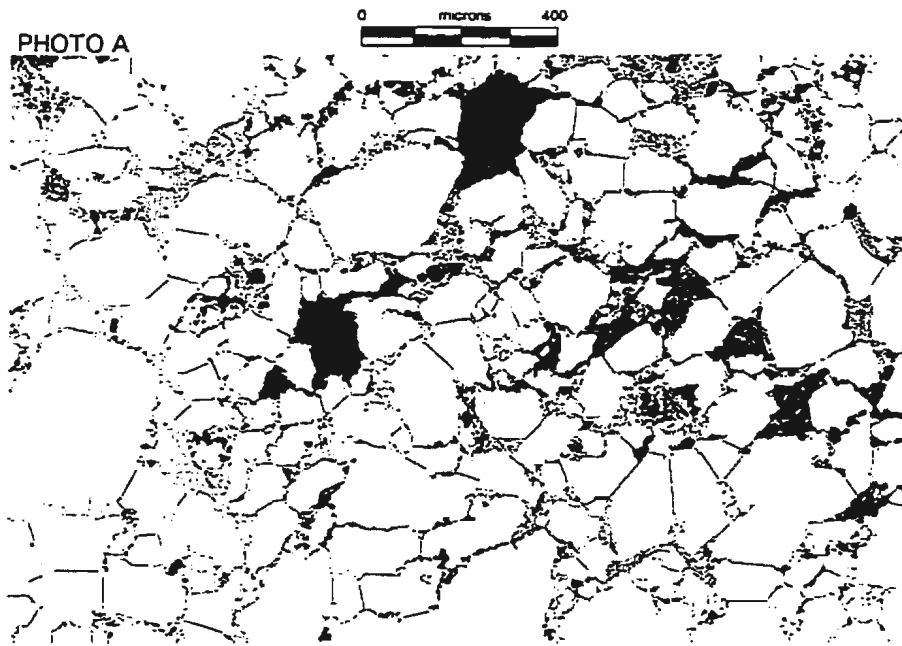


(a)

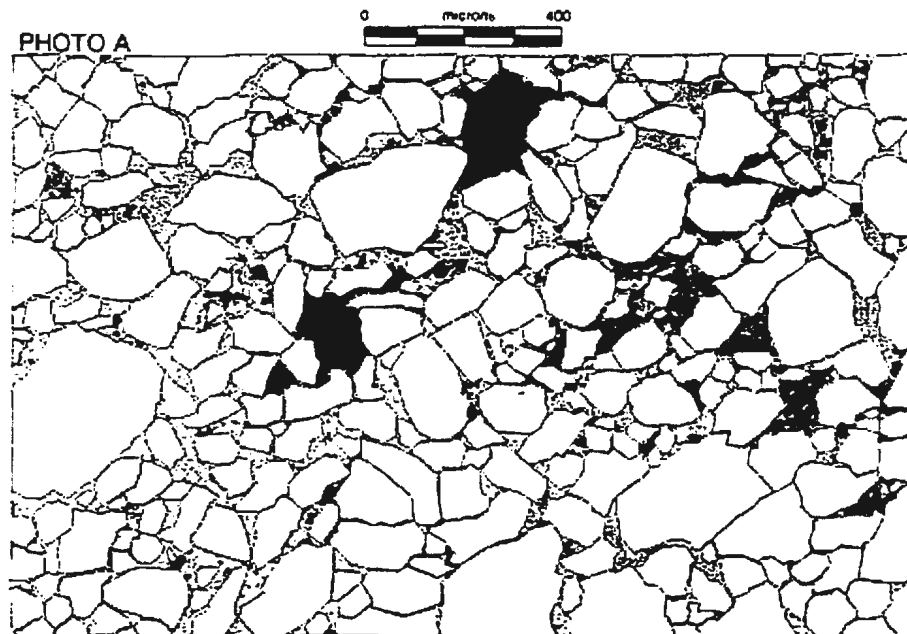


(b)

Figure 7.4 Auto and manually segmented quartz grains

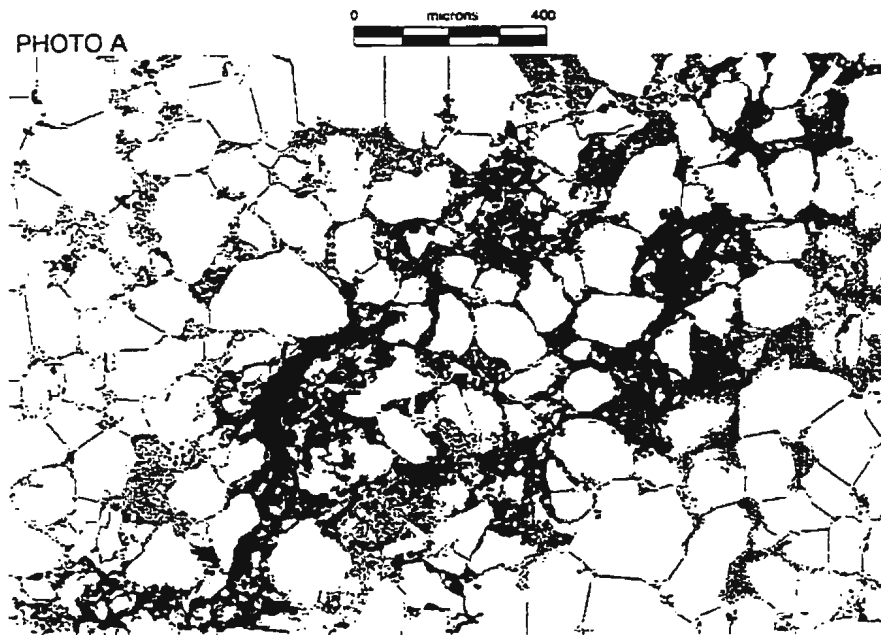


(c)

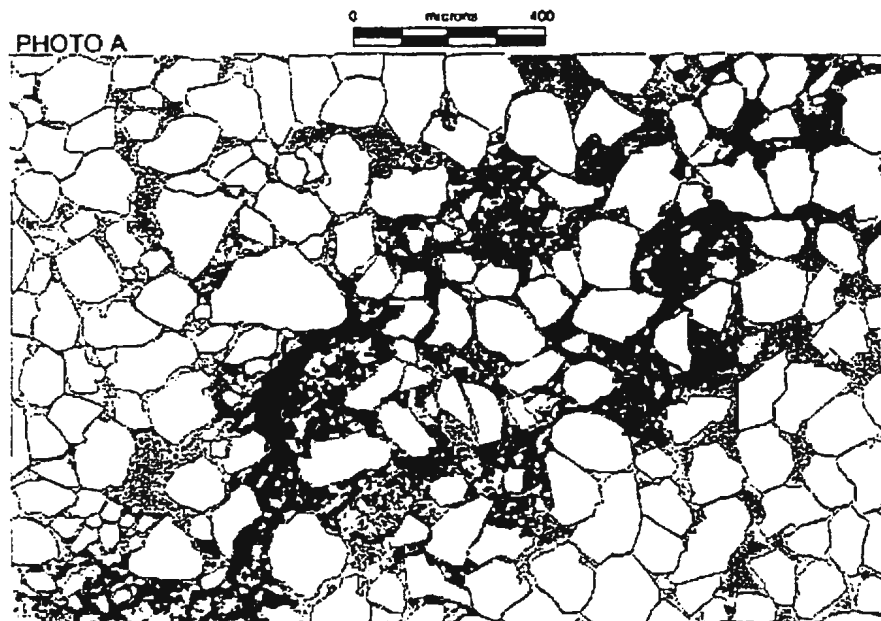


(d)

Figure 7.4 Auto and manually segmented quartz grains (Continued)



(e)



(f)

Figure 7.4 Auto and manually segmented quartz grains (continued)

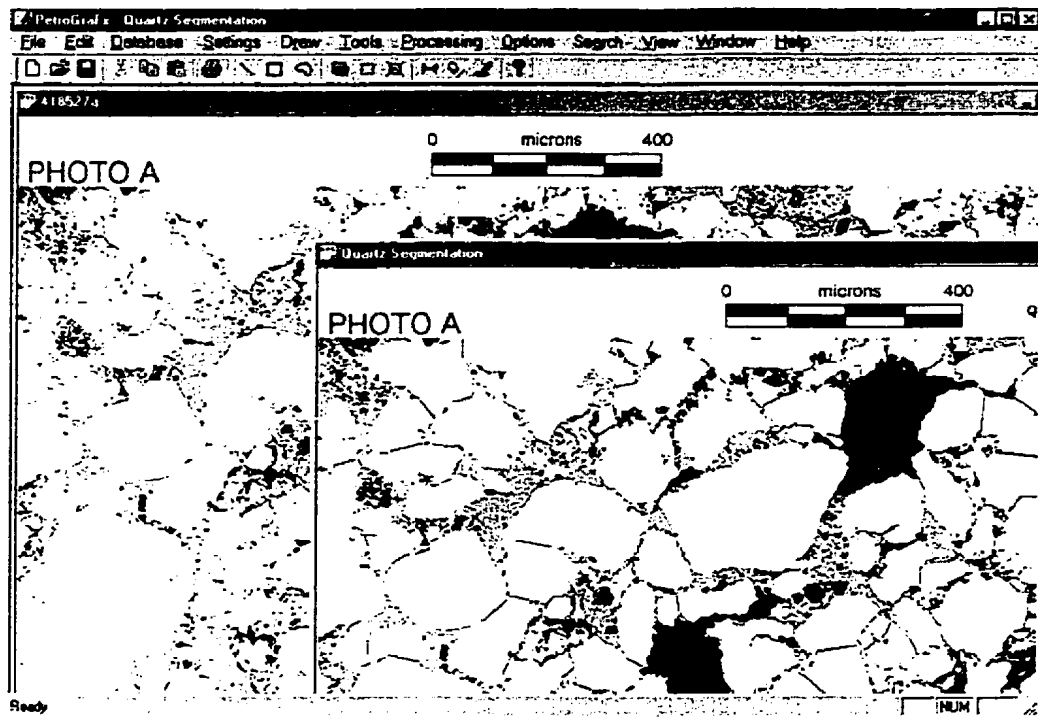


Figure 7.5 PetrograFX GUI window

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

This thesis has presented the algorithms for automated image analysis for petrographic thin section images including pore space segmentation, porosity determination, pore space distribution, individual quartz grain segmentation and the NSD measurement, quartz grain distribution and sorting.

From the comparison of auto mode and manual mode segmentation results, it is concluded as follows:

- The region growing scheme in color space successfully segments the pore space with an error rate less than 5%. The software is designed to be adaptive to variations in color, lighting condition and other parameters in different images.
- Image segmentation, image preprocessing and blob analysis are employed to perform the automated quartz grain segmentation and measurements, such as determining the NSD of an individual grain, the quartz grain distribution in a certain size range or the full range, the sorting class in a whole image. The comparison of results between the auto mode and manual mode has shown that the most quartz grains are measured to an acceptable accuracy.

The major contribution of this thesis is the development of image processing and image analysis algorithms for a particular image class, the petrographic thin section images. Since most segmentation techniques are application dependent, this thesis focuses on the selection and development of the most appropriate methods based on the clients' requirements, as well as to assess the performance of the software. In the porosity determination phase, the seeded region growing scheme in color space is designed, while the seeds are generated automatically from all pore space regions that are obtained from the feature space. The feature is extracted from the absolute R – B difference image. It utilizes color signatures for forming the most appropriate feature space for highlighting the pore space. Meantime, the effects of color and lighting variations are also suppressed since the feature image is generated from two channels of the same image. In the quartz grain measurement phase, the adaptive thresholding, median filter and blob analysis are used for obtaining the quartz grains, and the modified watershed transformation is designed to insert acceptable approximation of grain boundaries between fused grains.

8.2 Recommendations

This thesis provides the basis for future recommendation and development on both pore space segmentation and quartz grain boundary identification. This will include:

- Estimating the porosity in a whole slide. This will likely require optimization of the speed since many more blobs are involved. Also there will likely be issues related to image registration to be addressed since the entire image will need to be processed in blobs.

- Adapting to various color stained pore space. In this thesis, only the blue-stained pore space is processed although stains of various colors are used by industry.
- Segmenting rich texture quartz grains - Although most quartz grains are texture free and included in the resulting image, there are still a few grains in which rich texture properties are found. These grains are omitted from the quartz grain segments in the current software, but this issue needs to be considered.
- Recovering missing edges. Some weak edges that belong to the quartz grain boundaries are removed during the blob analysis and small hole filling process. By analyzing the shape or other related factors, these edges could be recovered.
- Extracting other features (eg. angularity, roundness, connectedness etc.) for the image.

References

- Adams, Rolf and Bischof, Leanne (1994). Seeded Region Growing. IEEE Tans. On Pattern Analysis and Machine Intelligence, Vol 16, No. 6, June.
- Beghdadi, Azeddine (1995). Entropic Thresholding Using a Block Source Model. Graphical Models and Image Processing, Vol. 57, No. 3, May.
- Biederman, Edwin W. Jr. (1986). Atlas of Selected Oil and Gas Reservoir Rocks from North America. A Wiley-Interscience Publication, John Wiley & Sons.
- Borgefors, Gunilla (1986). Distance Transformations in Digital Images. Computer Vision, Graphics, and Image Processing 34, 344-371.
- Brink, A. D. (1995). Minimum Spatial Entropy Threshold Selection. IEE Proc.-Vis. Image Signal Process, Vol. 142, No. 3, June.
- Burdick, Howard E. (1997) Digital Imaging, Theory and Applications, McGraw-Hill.
- Canny, John (1986). A Computational Approach to Edge Detection. IEEE Tans. On Pattern Analysis and Machine Intelligence. Vol. PAMI-8. No. 6. November.
- Chang, Y L and Li, X. (1994) Adaptive Image Region-growing. IEEE Trans on Image Processing, 2:868-872.
- Chen, Chi-hau (1978). Statistical Pattern Recognition. Spartan Books.
- Clausi, David A. and Jernigan (Waterloo). M. Ed. Towards a Novel Approach for Texture Segmentation of SAR Sea Ice Imagery. Department of Systems Design Engineering and Department of Geography, University of Waterloo.
- Ehrlich, Robert, Crabtree, Sterling James, Horkowitz, Kathleen O., and Horkowitz, John P. (1991). Petrography and Reservoir Physics I: Objective Classification of Reservoir Porosity, The American Association of Petroleum Geologists Bulletin, V75, No. 10 October.
- Ford, Ray (1998). Petrographic Image Analysis (PIA),
<http://www.cs.umd.edu/DOE/PIA/home.html>

- Folk, Robert L. (1980). *Petrology of Semimentary Rocks*, hemphill Publishing Company, Austin, Texas 78703
- Goins, Elizabeth and Reedy, Chandra (2000), *New Image Analysis Technologies for Thin-Section Examination In Objects and Architectural Conservation*.
<http://aic.stanford.edu/aic2000/goins.html>.
- Gonzalez, R. C. and Wintz, P. (1987). *Digital Image Processing*, Second Edition.
- Goshtasby, Ardeshir (1985). *Description and Discrimination of Planar Shapes Using Shape Matrices*. IEEE Trans. On Pattern Analysis and Machine Intelligence. Vol. PAMI-7, No. 6.
- Gosine, Ray and Burden, Elliott (1999). *PetrograFX: A petrographic Image Analysis and Database Management Toolkit. A Proposal to Seed Collaborative Reserch in Information Technology and Informatics*, C-CORE & Earth Sciences, Memorial University of Newfoundland.
- Haralick, Robert M., Sternberg, Stanley R. and Zhuang, Xinhua (1987). *Image Analysis Using Mathematical Morphology*, IEEE Trans. On Pattern Anal. Mech. Intell., 9(4) 532-549,
- Haralick, R M and Shapiro, L. G. (1985). *Image Segmentation Techniques*. Computer Vision, Graphics, and Image Processing, 29.
- Hibernia. HMDC_web_page\Core_data\Petrographic_data\Petrographic_web_pages\Hibernia_B-16_2\Hib_B16-2_pet_inventory.html.
- Hojjatoleslami, S. A. and Kittler, J. (1998). *Region Growing: A New Approach*. IEEE Trans. On Image Processing, Vol. 7, No. 7, July.
- Jahne, Bernd (1991). *Digital Image Processing, Concepts, Algorithms, and Scientific Applications*, Springer-Verlag.
- Jain, Anil K., and Farrokhnia, Farshid (1991). *Unsupervised Texture Segmentation Using Gabor Filters*, Pattern Recognition, Vol. 24, No.12.
- Kennedy, Stephen K. and Mazzullo, Jim (1991). *Image Analysis Method of Grain Size Measurement*. In *Principles, Methods and Application of Particle Size Analysis*. Syvitski, J.P.M (ed). pp 76-87.
- Kittler, J. and Illingworth, J.(1986). *Minimum Error Thresholding*, Pattern Recognition, Vol. 19, No. 1.
- Levin, Richard I. (1984). *Statistics for Management*, third edition, Pretice-Hall, Inc.,

New Jersey 07632.

- Loustau, John and Dillon, Meighan (1993). *Linear Geometry with Computer Graphics*. Marcel Dekker, Inc.
- Matrox Imaging Library User's Guide (1999). Version 6.0. Matrox Electronic Systems Ltd..
- Ng, Isaac, Tan, Tele and Kittler, Josef (1992). On Local Linear Transform and Gabor Filter Representation of Texture, Proc. Int'l Conf. Pattern Recognition, pp. 627-631. Int'l Assoc. for Pattern Recognition.
- Ohta, Yu-Ichi, Kanade, Takeo, and Sakai, Toshiyuki (1980). Color Information for Region Segmentation. *Computer Graphics and Image Processing*. 13.
- Parker, J. R. (1997). *Algorithms for Image Processing and Computer Vision*. New York, Wiley Inc.
- Rabbani, Majid and Jones, Paul W. (1991). *Digital Image Compression Techniques*, Spie Optical Engineering Press, Bellingham, Washington USA.
- Ramesh, N., Yoo, J.-H. and Sethi, I.K.(1995). Thresholding Based on Histogram Approximation, IEE Proc.-Vis. Image Signal Process., Vol. 142, No. 5, October.
- Randen, Trygve, Husoy, John Hakon (1999). Filtering for Texture Classification: A Comparative Study. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 21, No. 4, April.
- Schettini, Raimondo (1993). A Segmentation Algorithm for Color Images. *Pattern Recognition Letters* 14.
- Scholle, Peter A. (1979). *A Color Illustrated Guide To Constituents, Textures, Cements, and Porosities of Sandstones and Associated Rocks*, Memoir 28. The American Association of Petroleum Geologists.
- Sharma, Gaurav and H. Joel Trussell (1997). *Digital Color Imaging*, *IEEE Trans. On Image Processing*, Vol. No. 7, July.
- Sonka, Milan, Hlavac, Vaclav, and Boyle, Roger (1998). *Image Processing, Analysis, and Machine Vision*, 2nd edition, PWS Publishing.
- Antoine Taza and Ching Y. Suen (1989). Discrimination of Planar Shapes Using Shape Matrices, *IEEE Trans. On SMC*, Vol. 19 No. 5
- Taza, Antoine and Suen, Ching Y. (1989). Discrimination of Planar Shapes Using Shape Matrices, *IEEE Trans. On SMC*, Vol. 19 No. 5.

- Therrien, Charles (1991). *Decision Estimation and Classification*. Wiley.
- Vincent, Luc and Soille, Pierre (1991). *Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol 13, No 6. June.
- Vincent, Luc (1993). *Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms*. *IEEE Trans. on Image Processing*. Vol 2. No. 2. April.
- Weszka, Joan S. (1978). *A Survey of Threshold Selection Techniques*. *Computer Graphics and Image Processing*, 7.
- Williams, Howel. Turner, Francis J. and Gilbert Charles M. (1983) *Petrography. An Introduction to the Study of Rocks in Thin Section*, Second Edition. W. H. Freeman and Company, San Francisco.
- Wilson, Andrew (1988), *What Color is Color*, *ESD: magazine*, January.
- Wu, Jian-Shing and Leou, Jin-Jang (1993). *New Polygonal approximation Schemes for Object Shape Representation*, *Pattern Recognition*. Vol. 26. No. 4, pp 471-484.
- Zahn, C. T. and Roskies, R. Z. (1972). *Fourier Descriptors for Plane Closed Curves*. *IEEE Trans. On Computers*. Vol. C-212 pp. 269-281.

Appendix I

Lossless Predictive Coding

Generally, the gray scales of adjacent pixels in natural images are highly correlated since there are many uniform regions. The predictive coding idea is to exploit the relationship between a pixel and its surrounding neighbors, and obtain the value of this pixel by inspecting its neighboring pixel values. This can be modeled as a Markov source. "An m th-order Markov source is a source in which the probability of occurrence of a source symbol s_i depends upon a finite number m of the preceding symbols". [Rabbani, 1991] It is defined by the set of conditional probabilities

$$p(s_{i_1}, s_{i_2}, s_{i_3}, \dots, s_{i_m}) \quad \text{where} \quad i, j_p (p = 1, \dots, m) = 1, 2, \dots, n.$$

If an image is modeled by a m th-order Markov source, any pixels that are represented by k bits in this image take one of values of K from 2^k possible values. A given pixel x_m depends on the values of the m previous pixels, x_0, x_1, \dots, x_{m-1} .

The differential pulse code modulation (DPCM) uses the most probable estimate for x_m instead of all of the possible estimates. The value \hat{x}_m is the maximum of $p(x_m | x_{m-1}, \dots, x_0)$. This is the most likely prediction. The difference between x_m and \hat{x}_m is called the differential or error signal e_m ,

$$e_m = x_m - \hat{x}_m$$

Appendix II

Bresenham Algorithm

The line drawing procedure is known as the Bresenham algorithm. In a computer graphics system, the line drawing procedure belongs to the display process unit (DPU).

According to Loustau and Dillon [1993], if the coordinates of two pixels are given, the line equation $y = mx + b$ is determined. The problems arise in rendering the graph on a computer screen. Only the pixels with integer coordinates can be displayed. Also, since the slope of the line is not integer in most case, the integer x-coordinate associates with a decimal y-coordinate. So that the line can not be represented properly.

A good line drawing algorithm should satisfy the following criteria:

- The displayed pixels should be closest to the actual pixels
- The displayed pixels should be distributed evenly
- Begin and end accurately
- Be efficient

The Bresenham algorithm provides a good approach to solve this problem. From the two end pixels, (x_1, y_1) and (x_2, y_2) , each integer between x_1 and x_2 is utilized to calculate the corresponding y-coordinate using the line equation. Then the integer y-coordinate is determined by finding its nearest integer. This is called a Round function. The x-coordinate is increased 1 greater than the previous until the last pixel is generated.



