*Learning without thought is a labor lost; thought without learning is perilous*

– Confucius (551 B.C.E - 479 B.C.E).

**University of Alberta**

# ADAPTIVE LOCAL THRESHOLD WITH SHAPE INFORMATION AND ITS APPLICATION TO OIL SAND IMAGE SEGMENTATION

by

**Jichuan Shi**

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

**Master of Science**

Department of Computing Science


Spring 2010
Edmonton, Alberta

## Examining Committee

Hong Zhang, Computing Science

H. Vicky Zhao, Electrical and Computer Engineering

Nilanjan Ray, Computing Science

*To my parents*

*Who have always supported and encouraged me.*

# Abstract

This thesis is concerned with a novel local threshold segmentation algorithm for digital images incorporating shape information. In image segmentation, most local threshold algorithms are based only on intensity analysis. In many applications where an image contains objects with a similar shape, in addition to the intensity information, some prior known shape attributes could be exploited to improve the segmentation. The goal of this work is to design a local threshold algorithm that includes shape information to enhance the segmentation quality. The algorithm adaptively selects a local threshold. Shape attribute distributions are learned from typical objects in ground truth images. Local threshold for each object in an image to be segmented is chosen to maximize probabilities in these shape attributes distributions. Then for the application of the oil sand image segmentation, a supervised classifier is introduced to further enhance the segmentation accuracy. The algorithm applies a supervised classifier trained by shape features to reject unwanted fragments. To meet different image segmentation intents in practical applications, we investigate a variety of combinations of shape attributes and classifiers, and also look for the optimal one. Experiments on oil sand images have shown that the proposed algorithm has superior performance to local threshold approaches based on intensity information in terms of segmentation quality.

# Acknowledgements

I would like to thank my supervisor Hong Zhang, under whose direction, encouragement and inspiration I completed this research. I would like to thank the members of my committee, Nilanjan Ray and Vicky Zhao, in spending their time on reading and commenting on the thesis. Finally, I am very grateful to those in the CIMS lab and Robotics lab who have helped and supported me throughout the work.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Thresholding is widely used in image segmentation. In many digital images, the intensity of targets and that of the background are sufficiently different to make thresholding a simple method of image segmentation. However, many factors, such as illumination, insufficient contrast, similar textures within target and background, prevent a simple threshold from segmenting targets well. Due to those factors, generally speaking, local threshold is superior to global threshold in terms of segmentation quality.

Oil sand is an important natural resource for synthetic crude oil production. In the mining process, the size analysis of oil sand plays an important role in enhancing the efficiency and reliability of operations. Based on computer vision methods, segmenting oil sand targets from digital images provides a feasible solution to obtaining size information without interfering with exploitation of the resource. Thresholding has been adopted as a segmentation algorithm in previous work. Unfortunately, oil sand images captured in an industrial environment are highly degraded by noise. Most local threshold algorithms based merely on intensity analysis can not produce sufficiently accurate segmentation results. However, oil sand ores tend to have a rounded shape with little variation. Therefore, we wish to exploit this shape information to improve local threshold segmentation.

The goal of this research is to incorporate shape information into a local threshold algorithm in order to enhance segmentation quality. For images containing objects with a similar shape, such as the oil sand images, geometric attributes can be learned from target objects and utilized to determine thresholds that make segmented results and prior known targets tend to be alike in shapes. Segmentation results illustrate that the proposed algorithm is superior to a local threshold algorithm based on intensity information alone.

## 1.1 Local Threshold Algorithm

Local thresholding, which is also known as adaptive thresholding, chooses different thresholds for different regions in an image [39]. The key part is the threshold value selection. In most of the research, intensity analysis is the basis of local threshold algorithms. An image threshold is locally

calculated by statistics methods, such as local variance [43], local contrast [2], center-surround [28] as well as surface-fitting thresholding [6]. Some of these methods will be further discussed in Chapter 2.

For local threshold algorithms, threshold value is adaptively decided in accordance with the intensity in a small neighborhood. With this property local threshold algorithms usually focus on searching a suitable threshold for each pixel but do not consider the shape of the object formed by those pixels. It makes local threshold algorithms sensitive to intensity noise. In our research shape information is learned from ground truth images and characterized by shape attributes specific to an application domain. Incorporating the shape information, our algorithm is designed to locally search thresholds producing objects in reasonable shapes. We assume this improvement will enhance the segmentation quality for images containing objects with a similar shape.

## 1.2 Image Segmentation for Oil Sand Mining

Oil sand is an important natural resource of Canada. The production of synthetic crude based on oil sand meets the energy needs not only in Canada, but also in the whole of North America. Before oil is separated from sand, the ores need to be excavated, crushed and then screened. However, due to the main components of oil sand, including bitumen, clay, water and sand, the size of the oil sand varies significantly in different seasons. The oil sand appears as fine pebbly rocks and sand in warm weather, while it may freeze into a giant lump in winter cold. In the oil sand mining process a set of screens are utilized to separate oil sand ores in accordance to their sizes. Therefore, the size information of oil sand ores is a key performance indicator of oil sand exploitation operations [55, 56].

Computer vision approaches are adopted to obtain the size information because they do not interfere with the mining process. Segmenting oil sand objects from images represents a promising approach to determining ore size information. In previous investigations involving oil sand images segmentation, researchers worked with several segmentation algorithms such as watershed [18], graph-cut [46], Markov random field [57], mathematical morphology [10], background subtraction [58] and optical flow. These investigations have successfully tuned existing segmentation algorithms to oil sand images and also indicate that it is too challenging for an algorithm based only on one image property, such as intensity, to generate segmentation results with high enough quality for size analysis [55, 56].

## 1.3 Thesis Objective and Contributions

The work presented in this thesis focuses on incorporating the shape information as prior knowledge to enhance the segmentation quality of local threshold algorithm. In previous research, most local threshold algorithms adopt the intensity information only. However, in addition to the intensity

information, shape information can also be exploited in local thresholding for images containing objects with a similar shape. The goal of our work is to show that a local threshold algorithm utilizing shape information is superior to local threshold algorithms based only on intensity information, and to apply this method in a segmentation algorithm of oil sand images.

In this case the main contributions resulting from this work are divided into two parts: utilizing shape information to improve local thresholding, and encoding the proposed local threshold algorithm into the oil sand image application. In terms of local thresholding, the shape attributes are learned from prior known objects, such as manually produced ground truth fragments. These shape attributes are adopted to adaptively choose thresholds in order to make the segmented fragments tend to have similar shapes to ground truth. Through this process, the segmentation quality of local threshold algorithm is enhanced. With respect to the application of oil sand images, the proposed local threshold method is applied and in combination with a supervised classifier as a post-processing operation to form a segmentation algorithm. Based only on changing threshold value, some fragments in oil sand images cannot produce results of a good shape. In this case the supervised classifier is exploited to eliminate these fragments with unwanted shapes. With this modification the average segmentation accuracy of oil sand images is enhanced by local thresholding and supervised classifier that are both based on prior shape information. Therefore, our investigation improves the local threshold algorithm and applies it to a specific application domain.

## 1.4 Methodology and Results

The proposed local threshold algorithm is incorporated with shape information and applied to oil sand images. Each shape feature is characterized by a probability density distribution. These distributions are learned from ground truth images with kernel estimation. In terms of one shape feature, the higher the probability density value is, the more objects in ground truth images have the same feature value. Therefore, we assume that the probability density value implies the possibility for a segmented fragment to be a good segmentation. In accordance with this assumption, a local threshold is selected to maximize the probability in a shape feature distribution. In thresholding process, an initializing threshold is utilized to produce fragment candidates in an image to be segmented, then a group of thresholds is locally applied to every fragment candidate, and one threshold will be chosen if the corresponding fragment has the highest probability in feature distributions with respect to all local thresholds. Through this process, the shape information chooses local thresholds resulting in that the segmented fragments and ground truth objects tend to be similar in shape. Compared with other local threshold algorithms based only on intensity information, our proposed algorithm is superior to them in terms of segmentation quality.

The proposed local threshold is applied to the oil sand application. Because our interest lies in object size distribution, the accuracy of segmented objects are much more important than the number of objects. Decreasing the number of fragments in one image can be compensated for by increasing

the number of segmented images. In this case, supervised classifiers, such as Support Vector Machine (SVM) and Least Square (LS), are investigated to remove segmented fragments with unwanted shape. Shape attributes of ground truth are used as positive samples, while failure cases in previous segmentation are used as negative ones. A classification boundary is determined based on shape information, and classifies fragments into two clusters: a segmentation with shape similar to ground truth objects or with unwanted shape. Those unwanted fragments are eliminated from segmentation results in order to enhance the average segmentation accuracy. To evaluate the performance of different combinations of classifiers and shape features, several criteria such as object-level consistence score, particle size distribution, and fragment quantity are utilized. Finally we found that the SVM classifier tend to generate results with higher average segmentation accuracy.

## 1.5 Organization

The rest of this thesis is organized as follows. Chapter 2 presents an overview of current local threshold algorithms and shows limitations of them that motivate the investigation of a new local threshold algorithm incorporating shape information. Chapter 3 describes the methodology of local thresholding based on shape information, and illustrates the adopted shape features. Then, Chapter 4 displays that the proposed local threshold algorithm is applied to the oil sand application and adjusted by incorporating a supervised classifiers based on shape features. A variety of criteria are utilized to evaluate the performance of classifiers and shape features in the application. Finally Chapter 5 provides a summary of this work as well as possible future directions for local threshold algorithms including shape information.

# Chapter 2

# Background and Related Work

In the previous chapter, we introduced the research problem of this thesis and presented background information on local thresholding as well as its application to the oil sand mining process. In this chapter we present the background and related work in the field of local threshold algorithms and some segmentation algorithms including shape information. The motivation for developing a local threshold algorithm with shape information is also analyzed.

## 2.1 Introduction

Local thresholding often can be a critical method for image segmentation algorithms. This technology is exploited in many application domains related to computer vision, such as optical character recognition [5, 36, 22, 44], biomedical image segmentation [41, 15], and food quality measuring [25]. The development of local thresholding is mainly focused on two aspects: determining the size of subregions and selecting the value of local thresholds. In some early algorithms images are divided into subregions with fixed size [26]. Then, some algorithms adaptively choose subregion size based on different intensity features, such as edge, contrast, or homogeneity [52, 11]. In determining local threshold values, algorithms vary from analyzing a local histogram [9] to fitting a threshold surface according to energy function [21]. However, most local threshold algorithms are based only on intensity analysis. In some practical applications intensity information is highly degraded by noise. Therefore, extra information from other domains is required to improve segmentation quality.

For images containing objects with similar shape, shape information can be used for segmentation. In current investigations, shape information is often applied to eliminate noise, for example, in [30, 42, 45] shape features of cells, particles, and edges are extracted to remove unwanted segmented fragments. On the other hand, shape information is often highly related to specific application areas. For example, in the biomedical image segmentation domain, roundness, area, and other geometric features are often adopted to characterize cells [8]. For other applications, such as edge detection, thinness, and continuity are applied to describe edges [45].

## 2.2 Local Threshold Algorithms

Local threshold algorithms choose varied threshold values in different regions of an image [39]. The early usage of a local threshold algorithm can be dated back to the 1970s work done by Chow and Kaneko [9]. Thresholds are estimated from an intensity histogram in local ranges. Nowadays, in accordance with different ways of selecting thresholds, local threshold algorithms can be divided into four categories [37]: local variance methods, local contrast methods, center-surround methods, and surface-fitting methods. An overview is presented for each category in the following sections.

### 2.2.1 Local Variance Methods

Local variance methods select a threshold for each pixel based on mean and variance in the neighborhood. Investigations done by previous researchers vary from directly using those statistical variables to integrating the mean or variance into a sophisticated algorithm.

#### Niblack's Method

Niblack [26] designs a method that calculates local threshold in Equation 2.1,

$$T(x,y) = m(x,y) + k \cdot v(x,y) \tag{2.1}$$

$$\text{where} \quad m(x,y) = \frac{1}{N} \sum_{x,y \in S} I(x,y) \tag{2.2}$$

$$\text{and} \quad v(x,y) = \sqrt{\frac{1}{N-1} \sum_{x,y \in S} (I(x,y) - m(x,y))^2} \tag{2.3}$$

in which $T(x,y)$ is the threshold at pixel $(x,y)$ and $k$ is the coefficient parameter that decides the ratio of pixels being segmented as background or objects. When the darker part is considered as background, the greater the $k$ is, the more pixels are labeled as background. In Equations 2.2 and 2.3, $S$ is a neighborhood with size $b$, and $N$ is the quantity of pixels in that neighborhood. The threshold $T$ is directly derived from local mean and variance. In frequency domain this threshold function (Equation 2.1) performs just like a high-pass filter. It removes low frequency signals that are often resulted by illumination changing in background, while keeps high frequency signals that reflect the intensity difference between objects and background. Compared with global thresholding, Niblack's method is more robust to uneven illuminations. However, the coefficient parameter $k$ and neighborhood size $b$ are fixed and need to be manually tuned.

#### Parker's Method

Parker [29] presents a method that labels edge pixels with local thresholding and then segments images by filling inner parts surrounded by those edges. For an original image $I_1$, the minimum intensity difference between a pixel at $(x,y)$ and its eight-neighborhood is calculated as Equation 2.4,

and kept in an image named $I_2$ at the same position indicated by $(x, y)$. After all the difference values have been obtained in pixel level, the image $I_2$ is broken into a group of subregions with size $b \times b$, and then the mean value $m$ and the standard deviation $v$ of each region are calculated. Smoothed by a weighted average, $m$ and $v$ are linearly interpolated as two images, $I_M$ and $I_V$.

$$I_2(x, y) = \min_{i=1,\ldots,8}(I_1(x, y) - I_1(x_i, y_i)) \tag{2.4}$$

For a pixel in $I_2$, if $I_2(x, y) < I_M(x, y) + k * I_V(x, y)$, $I_M(x, y) < -1$ and $I_V(x, y) < -1$, this pixel will be labeled as an edge of object. The rest pixels in $I_2$ will be classified as flat region. After a region growing operation, areas surrounded by labeled edges are filled as objects. In this method the size of subregions still needs to be selected manually. In accordance with experiments adopting different subregion sizes in [29], this algorithm reaches its best performance when the size of subregion is 16.

**Yan *et al.*'s Method**

Yan *et al.* [52] develop an algorithm that choose threshold based on local variance and combine advantages of global as well as local thresholding. The algorithm assumes that gray scale images contain lighter objects and darker background. In this case two global thresholds, $T_0$ and $T_1$, are obtained from an initial step. Pixels with intensity higher than $T_1$ are labeled as objects while pixels with intensity lower than $T_0$ are considered as background. As shown in Figure 2.1, $T_0$ and $T_1$ are derived from an image histogram with percentile measurement, which is presented in Equations 2.5 and 2.6.

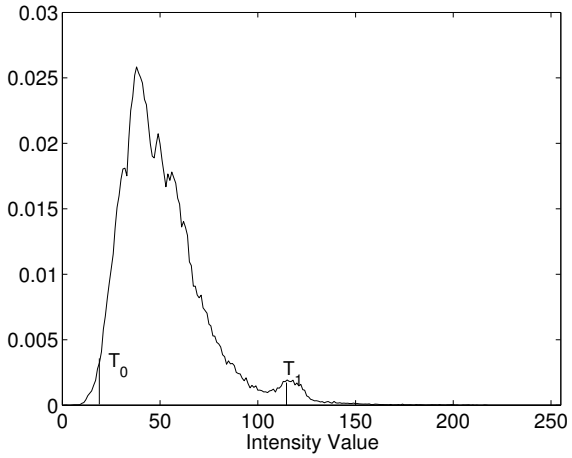


Figure 2.1: Global threshold $T_0$ and $T_1$ obtained from normalized histogram. Reproduced from [52]

$$\int_{-\infty}^{T_0} p_r(t)dt = A\% \tag{2.5}$$

$$\int_{T_1}^{\infty} p_r(t)dt = A\% \tag{2.6}$$

In [52], $A$ is set as 10, experimentally. In addition to pixels that are classified as objects and background by $T_0$ and $T_1$, others are segmented by local thresholds that are calculated by Equation 2.7.

$$T(x, y) = I(x, y) - m(x, y) - k \cdot v(x, y) \tag{2.7}$$

The producing of local threshold values is motivated by [26]. However, other than the method presented in [26], the subregion size $b$ and coefficient parameter $k$ are obtained without manual tuning. The subregion size is determined by the number of edge pixels in it. Different from calculating image gradient, pixels with variance that are twice the minimum variance are selected as edge pixels. To determine the subregion size $b$, an initial value is allocated. The value of $b$ keeps increasing until enough edge pixels are incorporated in this subregion. The coefficient parameter $k$ decides the pixel ratio of objects and background in segmented results. Therefore, for each subregion, this method applies the Otsu algorithm [27] to get a threshold $T$ and then calculates the pixel ratio $\rho$ of objects and background. The intensity distributions of the entire image and its local neighborhoods are assumed to follow the normal distribution $N(\mu, \sigma)$. Finally, the coefficient parameter $k$ can be derived from Equation 2.8.

$$\int_{-\infty}^{k\cdot\sigma+\mu} N_{\mu,\sigma}(x)dx = \rho \tag{2.8}$$

However, those assumptions for coefficient parameter selection are contradictory at some points. The Otsu algorithm that decides the parameter $\rho$ is supposed to search a global threshold for images containing a bimodal histogram [27], while the coefficient parameter $k$ is derived based on the assumption that an image intensity follows normal distribution. Since the normal distribution forms a unimodal histogram, those two assumptions cannot be satisfied at the same time. Therefore, this contradictory assumption between ratio calculation and coefficient parameter may lower the accuracy of local threshold calculation.

**Summary**

The local variance method determines thresholds pixel by pixel in accordance with a linear combination of local mean and variance. The segmentation quality is directly affected by the size of neighborhood and the coefficient parameter. Neighborhood size determination needs to balance the algorithm's robustness to noise and local properties. If the size of the neighborhood is too small, this method will be very sensitive to noise, while if the size is too large, the method will perform like a global threshold algorithm. The coefficient parameter decides the pixel ratio of objects and background in the segmented result. An inappropriate coefficient parameter could lower the segmentation accuracy. Methods in [26] and [29] choose those parameters experimentally, while Yan *et*

*al.* [52] present an approach that automatically selects neighborhood size by counting edge pixels, and also derives the coefficient parameter by the Otsu algorithm and normal distribution. Unfortunately, the contradiction between assumptions penalizes the estimation accuracy of the coefficient parameter.

### 2.2.2 Local Contrast Methods

Local contrast methods choose a threshold based on the intensity contrast between a pixel and its neighborhood. Contrast is often used in threshold algorithms under the assumption that the intensity of objects and that of the background are significantly different.

#### White and Rohrer's method

A local threshold algorithm presented by White and Rohrer [48] adaptively determines thresholds for pixels in accordance with the average intensity in its neighborhood. Adaptively choosing a contrast threshold is motivated by the issue that when the contrast between object and background is high, a higher threshold is required to eliminate noise, but when the contrast is low, the threshold needs to be decreased to prevent objects from being removed.

To enhance the efficiency of the algorithm and save storage space, White and Rohrer adopt a running average. Initially, the running average is calculated in a horizontal way, which is shown in Equation 2.9,

$$I_h(x, y) = f \cdot I(x, y) + (1 - f) \cdot I_h(x - 1, y) \tag{2.9}$$

where $I$ is the original image, $I_h$ is the horizontal running average, and $f$ is a nonlinear weight factor. Using the horizontal running average, the vertical one can be obtained from Equation 2.10.

$$I_v(x, y) = g \cdot I_h(x, y) + (1 - g) \cdot I_v(x, y - 1) \tag{2.10}$$

in which $I_v$ is the vertical running average, and $g$ is the nonlinear weight factor that is similar to $f$. Finally, the threshold $T$ is determined by Equation 2.11,

$$T(x, y) = k \cdot (I_v(x + l, y)) \tag{2.11}$$

in which $l$ is a "delay value" because the running average can only process an average for passed pixels. In [48], the value of $l$ is set as 8, while $k$ in Equation 2.11 is a bias function that is manually selected for different applications.

#### Bernsen's method

To obtain a threshold based on local contrast, Bernsen [2] adopts the midrange value. For each pixel $I(x, y)$, this algorithm calculates the maximum and minimum intensity values in the neighborhood

of the pixel as $I_{max}(x, y)$ and $I_{min}(x, y)$, respectively. The contrast $C$ at a place indicated by $(x, y)$ is calculated by Equation 2.12

$$C(x, y) = I_{max}(x, y) - I_{min}(x, y) \tag{2.12}$$

where $C$ is the contrast at pixel indicated by $(x, y)$. In the binarization step, if the contrast at a pixel is below a constant threshold, which is defined as $15$ in [2], the label of this pixel will be considered as consistent with its previous one; otherwise, this pixel will be classified into another category. This method is quite simple and will be efficient for a noise-free image. However, like the problem we faced in local variance methods, it is also very difficult to decide the size of neighborhood which will reduce effect of noise while keeping local properties.

**Eikvil *et al.*'s method**

Eikvil *et al.* [11] develop a method that uses two sliding windows to obtain the contrast in a local area. As shown in Figure 2.2, the smaller window $S$ and the larger one $L$ share the same centroid. In window $L$, the Otsu algorithm [27] is applied to determine the threshold $T$ in order to classify pixels into two clusters and estimate the mean values $\mu_1$ and $\mu_2$ of those clusters.

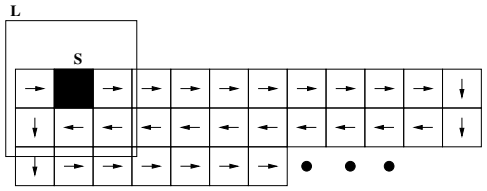


Figure 2.2: Local windows of Eikvil's method. Reproduced from [11]

$$\begin{cases} |\mu_1 - \mu_2| \geq t & \text{segment } S \text{ with } T \\ |\mu_1 - \mu_2| < t & \text{label } S \text{ as background or object} \end{cases} \tag{2.13}$$

If the difference between $\mu_1$ and $\mu_2$ is greater than a preset contrast threshold $t$, pixels in the smaller window $S$ will be segmented by threshold $T$ obtained from the Otsu algorithm. Otherwise, pixels in $S$ will be labeled the same as the cluster that has the latest updated mean value. For the whole image $I$, local window $S$ will move across it in a zig-zag pattern within a step length equal to the side of $S$. The design of applying two local windows is attempting to overcome noise problems: the larger window $L$ is adopted to enhance robustness to noise, while the smaller window $S$ is used to make thresholds work locally. However, the size of windows $L$ and $S$ still need to be chosen manually.

**Summary**

Local contrast methods select thresholds by measuring contrast in a neighborhood. Similar to local variance methods, the size of neighborhood still needs to be tuned manually. Approaches based on

local contrast do not offer any method to derive neighborhood size from image intensity or calculate any other information that can be exploited. Moreover, contrast values obtained from neighborhoods are often classified by another manually picked parameter, such as the constant threshold in [2] or the preset threshold $t$ in [11]. Those manually selected parameters increase the human interactions in local contrast methods and lower the generality.

### 2.2.3 Center-surround Methods

Center-surround methods are mainly adopted in the Optical Character Recognition (OCR) field. In OCR applications, most characters are written or printed in dark ink. Therefore, without losing generality, algorithms listed as follows assume that in a digital image a object has a lower intensity while the background contains a higher one. The original approach is designed by Giuliano *et al.* [12] in a patent for an electronic character-reading system. It is then modified by Palumbo *et al.* [28]. Kamel and Zhao [13] publish a new method introducing logical level technique. Subsequently, Yang and Yan [53] improve Kamel and Zhao's method by automatically choosing parameters for center-surround thresholding.

#### Giuliano *et al.*'s method

In the patent published by Giuliano *et al.* [12], the average intensity value of a central local window and the average intensity values of surrounding local windows are compared to distinguish a pixel into an object or background. As shown in Figure 2.3, a $3 \times 3$ local window is built for pixel located at $(x, y)$. This window will be used as the central one and labeled as $W_c$ in Figure 2.3. The other four windows surrounding $W_c$ in two diagonal directions are considered as neighbor windows, $W_n$.

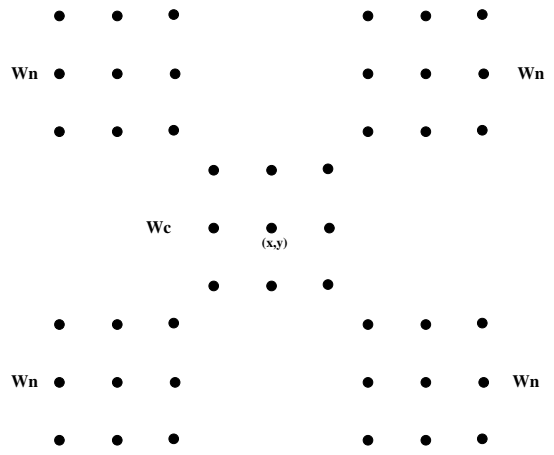


Figure 2.3: Local Window Operation in Giuliano's method. Reproduced from [12]

There are five manually set parameters, named $T_1$ to $T_5$. For a pixel $I(x, y)$, if its intensity is smaller than $T_1$, it will be directly labeled as an object. Otherwise, the average intensity of pixels in $W_c$ is obtained as $A_c$. Moreover, pixels in $W_n$ with intensity higher than $T_2$ are considered as

background. The average intensity of background pixels in $W_n$ is calculated as $A_n$. Finally, $A_c$ and $A_n$ are compared in Equation 2.14, and the result determines a label of this pixel at $(x, y)$.

$$\begin{cases} ((T_3 \times A_n) + T_5) > (T_4 \times A_c) & I(x,y) \in \text{background} \\ ((T_3 \times A_n) + T_5) \leq (T_4 \times A_c) & I(x,y) \in \text{objects} \end{cases} \tag{2.14}$$

where $T_3$, $T_4$ and $T_5$ are parameters affecting the comparison of $A_c$ and $A_n$. The advantage of Giuliano *et al.*'s method is low noise sensitivity, because the comparison is based on local intensity averages. However, preset parameters limit the generality of this approach.

**Kamel and Zhao's method**

To reduce the quantity of preset parameters, Kamel and Zhao [13] introduce a logical level technique. Every pixel in image $I$ is logically compared with the average intensity of four local neighborhoods. These average intensity values are named as $P_i$, $P_{i+1}$, $P'_i$ and $P'_{i+1}$. As shown in Figure 2.4, those neighborhoods surround the compared pixel at $(x, y)$, and the relationship between $P_i$ and $P'_i$ can be described as Equation 2.15.

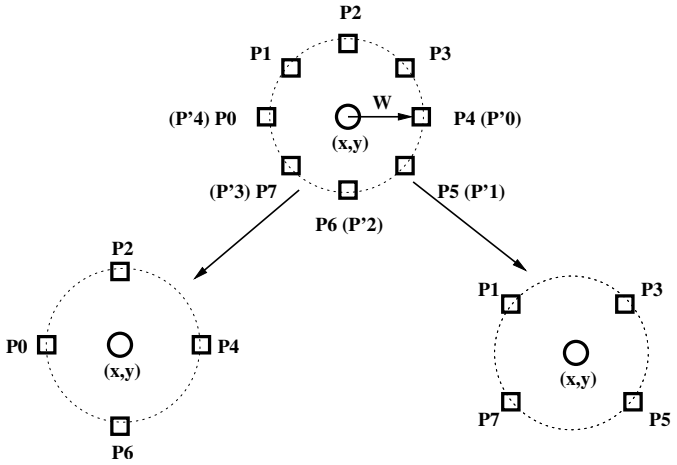


Figure 2.4: Logical Level Technique in Kamel and Zhao's method. Reproduced from [13]

$$P'_i = P_{(i+4) mod 8} \qquad i \in [0, 7] \tag{2.15}$$

The size of neighborhood is $(2w + 1) \times (2w + 1)$, in which $w$ is the manually set object width. A logical expression written as Equation 2.16 classifies pixels into two categories.

$$\bigvee_{i=0}^{3} [L(P_i) \wedge L(P'_i) \wedge L(P_{i+1}) \wedge L(P'_{i+1})] = \begin{cases} \text{true} & I(x,y) \in \text{background} \\ \text{false} & I(x,y) \in \text{objects} \end{cases} \tag{2.16}$$

where $L(P) = ave(P) - I(x, y) > T$. The parameter $T$ is a preset threshold while $ave(P)$ is calculated by Equation 2.17, in which $P_x$ and $P_y$ are the coordinates of $P$.

$$ave(P) = \sum_{-w \le i \le w} \sum_{-w \le j \le w} \frac{f(P_x - i, P_y - j)}{(2 \times w + 1)^2} \tag{2.17}$$

Furthermore, the logical expressions can be decomposed into Equation 2.18 to speed up the operation, and $ave(P)$ can be substituted by running an average defined by Equations 2.9 and 2.10.

$$\bigvee_{i=0}^{3} [L(P_i) \wedge L(P'_i) \wedge L(P_{i+1}) \wedge L(P'_{i+1})] = \bigvee_{i=0,2} [L(P_i) \wedge L(P'_i)] \wedge \bigvee_{i=1,3} [L(P_i) \wedge L(P'_i) \tag{2.18}$$

Similar to Giuliano *et al.*'s method [12], Kamel and Zhao's approach is based on a local intensity average that results in a low noise sensitivity. Moreover, with the logical level technique the quantity of preset parameter is decreased and the calculation speed is enhanced.

**Yang and Yan's method**

Yang and Yan [53] improve Kamel and Zhao's method [13] by automatically choosing all the parameters. It adaptively selects the threshold $T$ in the following steps. Initially, the maximum and minimum intensity values in a neighborhood with center point $P$ are respectively obtained as $I_{w(\max)}(x, y)$ and $I_{w(\min)}(x, y)$. Then an intensity average of the whole neighborhood is also calculated as $ave(P)$. If $|I_{w(\max)}(x, y) - ave(P)| > |I_{w(\min)} - ave(P)|$, this algorithm will consider "object pixels" are more than "background pixels" in this neighborhood. Therefore, the parameter $T$ is chosen as Equation 2.19

$$T = \alpha(\frac{2}{3} I_{w(\min)}(x, y) + \frac{1}{3} ave(P)) \tag{2.19}$$

If $|I_{w(\max)}(x, y) - ave(P)| < |I_{w(\min)} - ave(P)|$, it implies that "background pixels" are more than "object pixels". Then, parameter $T$ is calculated as Equation 2.20.

$$T = \alpha(\frac{1}{3} I_{w(\min)}(x, y) + \frac{2}{3} ave(P)) \tag{2.20}$$

When $|I_{w(\max)}(x, y) - ave(P)| = |I_{w(\min)} - ave(P)|$, the size of neighborhood is enlarged to $(2w + 3) \times (2w + 3)$, then $I_{w(\max)}(x, y)$, $I_{w(\min)}(x, y)$ and $ave(P)$ are calculated from this new neighborhood. If updated differences between those variables are not equal, the parameter $T$ will be determined by equations as stated above; otherwise, the value of $T$ will be allocated as $\alpha \cdot ave(P)$.

In optical character recognition applications, object width is mainly determined by strokes. To automatically estimate object width, Yang and Yan's approach builds a stroke-width histogram for each image. A stroke width with the highest frequency is picked out as the parameter $w$. Therefore, this algorithm can threshold most document images without manually tuning parameters [53].

**Summary**

Center-surround methods choose a threshold by comparing the intensity contrast between a central neighborhood and the surrounding ones. Both center-surround and local contrast methods exploit intensity contrast to threshold selection. However, to reduce the interruption of noise, center-surround methods do not measure the intensity contrast based on individual pixels, but rely on the average intensity of neighborhoods. Kamel and Zhao's method [13] develops a logical comparison technique in this area. This approach incorporates more neighborhood intensity information into threshold selection and enhances the algorithm robustness to noise. On the other hand, most center-surround methods are applied to the optical character recognition domain. In this application area objects (characters) in one image have almost the same size. This simplifies the problem of determining neighborhood size by estimating the stroke width of characters.

### 2.2.4 Surface-fitting Methods

In the range of a two-dimensional (2D) digital image, intensity can be considered as a surface. To segment images with an unevenly illuminated background, which are quite common in many applications, a threshold surface fitting the changing intensity is needed [54]. Chow and Kaneko [9] present an idea that adaptively varies threshold surface in accordance with a local histogram. This idea is further developed by Yanowitz and Bruckstein [54] through designing a threshold surface that interpolates at a local maximal gradient points of intensity surface. Based on a variational method, Chan *et al.* [6] improve Chow and Kaneko's algorithm by combining the step of locating maximal gradient points and interpolating the threshold surface with Poisson functions. Moreover, to eliminate the "ghost object" in Chow and Kaneko's algorithm, Liu *et al.* [21] design a new external force to locate the active threshold surface. In recent research, Saha and Ray [35] publish a method that automatically chooses parameters for an active surface in accordance with variational minimax optimization.

**Yanowitz and Bruckstein's method**

Yanowitz and Bruckstein's approach [54] adopts both edge and intensity information to produce a threshold surface. An image is segmented by this threshold surface and then validating gradient value is used to remove mistakenly segmented fragments. The process can be divided into the following steps. First of all, an image is smoothed by an average filter, and then the gradient magnitude is obtained from the smoothed image. Then, thresholding and thinning algorithms are applied to the gradient magnitude image and obtain local maximal gradient points that indicate boundaries of objects. Intensities of the smoothed image at local maximal gradient points are sampled and used as local threshold values. A threshold surface is produced by interpolating those threshold values. Finally, the image is segmented with this threshold surface. Fragments whose average edge values are lower than the validating value will be removed from the final result.

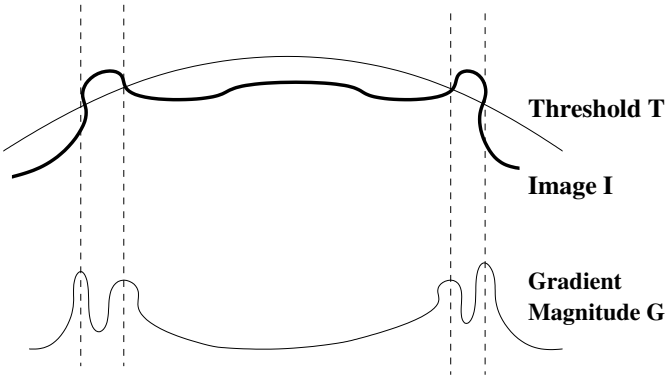


Figure 2.5: Adaptive threshold surface for image segmentation. Reproduced from [54]

As shown in Figure 2.5, local threshold values, which are also known as "supporting points" in [21], are located at the local maximum gradient points of intensity surface. Threshold surface is interpolated in accordance with the following minimal surface algorithm that chooses a surface with minimal area between those indicated threshold points. Therefore, the threshold surface is derived from Equation 2.21,

$$\frac{\partial^2 I}{\partial x^2}[1+(\frac{\partial I}{\partial y})^2]+\frac{\partial^2 I}{\partial y^2}[1+(\frac{\partial I}{\partial x})^2]-2\frac{\partial I}{\partial x}\frac{\partial I}{\partial y}\frac{\partial^2 I}{\partial x \partial y}=0 \tag{2.21}$$

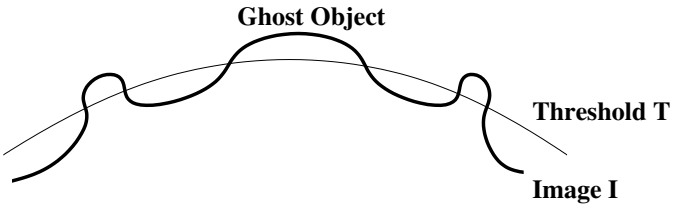


Figure 2.6: Ghost Object. Reproduced from [21]

The main problem of Yanowitz and Bruckstein's method is the "ghost objects" produced in the segmentation result. One assumption behind this method is that a threshold surface only intersects with the intensity surface at those chosen local threshold values. However, if the threshold surface and the intensity surface are quite close to each other, there might be some intersections between them besides those chosen ones, as shown in Figure 2.6. This will produce false objects named ghost objects in an area that should be background, and vice versa. In [54], the algorithm tries increasing local threshold values to get rid of ghost objects, but that will result in removing some true objects. The solution for eliminating ghost objects is further developed in [21].

**Liu *et al.*'s method**

To obviate ghost objects, Liu *et al.*'s approach [21] applies an active surface instead of the "supporting points" used in [54], and designs an external force to keep the threshold and intensity surface

away. The active surface is moved by an energy function defined as Equation 2.22,

$$E = \int_{\Omega} w|\nabla t(x,y)|^2 + E_{ext}(t(x,y))dxdy \tag{2.22}$$

in which $t(x,y)$ is the active threshold surface. The first term in Equation 2.22 is an internal energy that keeps the active surface smooth, while the second term is an external energy driving the active surface to designed features such as high gradient points, in this case. The energy function is designed to be minimized by moving the active surface to a stable status. Therefore, the threshold indicated by this active surface fits the background changing.

Different from Yanowitz and Bruckstein's method that does not consider the distance between the threshold and intensity surface, Liu *et al.*'s method includes this factor by designing an external energy as Equation 2.23,

$$E_{ext}(t(x,y)) = \exp(-(t(x,y) - i(x,y))^2/\sigma^2) \tag{2.23}$$

where $i(x,y)$ is the intensity surface. This external energy reaches its maximum when $t(x,y) = i(x,y)$ and decreases with the growth of difference between $t(x,y)$ and $i(x,y)$. From Equation 2.23, a repulsive force is derived as Equation 2.24,

$$F_{ext} = (t(x,y) - i(x,y))\exp(-(t(x,y) - i(x,y))^2/\sigma^2) \tag{2.24}$$

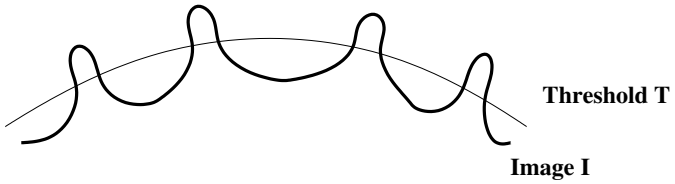


Figure 2.7: Threshold Surface in Liu *et al.*'s method. Reproduced from [21]

During the process of active surface searching, the threshold surface obtains its stable status at high gradient points. As shown in Figure 2.7, the energy function reaches the local minimum with the mean value of two neighbor points lying in the area of objects and background, respectively. Therefore, this active threshold surface is looking for high gradient points that indicate boundaries of objects. Moreover, with the effect of repulsive force, the whole threshold surface keeps a distance away from intensity threshold, and successfully eliminates the ghost objects.

**Saha and Ray's method**

In practical applications, a weight parameter is often required to balance the internal and external force of active surface energy function. For most algorithms, this weight parameter needs to be fine-tuned manually. Saha and Ray [35] present a method that automatically chooses this weight based on minimax optimization.

The energy function described in Equation 2.22 can also be changed into the form in Equation 2.25. In this case the equation is transformed to a function with respect to threshold surface $t$ as well as weight parameter $w$.

$$E(t, w) = \sqrt{1 - w^2} E_{int}(t) + w E_{ext}(t) \tag{2.25}$$

in which $E_{int}$ is the internal energy and $E_{ext}$ is the external one.

The minimax optimization supplies a relatively conservative solution to $w$, which is based on the principle that minimizes the energy function in its worst possible case. [35] In the parameter space with respect to $w$ and $t$, energy function $E$ has a concave-convex nature. Since the duality gap is zero [33], the order of minimizing and maximizing can be interchanged. Therefore, we get a function as Equation 2.26,

$$t^* = \arg \max_w \min_t E(t, w) = \arg \min_t \max_w E(t, w) \tag{2.26}$$

Then the maximal value of $w$ is derived from energy function as Equation 2.27.

$$w^* = \frac{E_{ext}(t)}{\sqrt{E_{int}(t)^2 + E_{ext}(t)^2}} \tag{2.27}$$

The energy function is minimized with respect to $t$ by a gradient descent approach. Finally, the VM algorithm [23] is applied to search the solution to this minimax problem iteratively.

**Summary**

Thresholds in surface-fitting methods are obtained from a surface that fits image intensity. Different from previous local threshold algorithms, surface-fitting methods do not adopt neighborhoods to localize a threshold searching range. This avoids the problem of choosing local neighborhood size. On the other hand, energy function is introduced to move the threshold surface. The threshold surface can be pushed by an external force to a place with selected intensity features, while an internal force keeps the smoothness of the threshold surface. This means the threshold surface can be driven by different intensity features through modifying the external force. However, a main drawback of this type of methods is that they often tend to be sensitive to noise. As mentioned in [54], a preprocessing step removing part of the noise, such as an average smooth, is required before gradient map generation. In some application domains, images are highly degraded by intensity noise, which makes the gradient map too fuzzy to produce a suitable threshold surface.

## 2.3 Shape Information

The algorithm presented in this thesis is based on shape information. In related investigations, shape information is also exploited by different algorithms, such as global thresholding [30, 42], local thresholding [19], boundary segmentation [8], and edge detection [45]. Since shape information is

often highly related to application domains, the motivation and methodologies of these algorithms are briefly analyzed in this section.

**Global Thresholding**

In the global thresholding area, Pikaz and Averbuch [30], as well as Torabi *et al.* [42], exploit different shape information in their methods. Pikaz and Averbuch [30] adopt the shape information to eliminate noise points during segmentation process. The method is based on an assumption that the area of noise point changes significantly with varied thresholds while the size of object tends to be stable. In this case Pikaz and Averbuch introduce a size-threshold function $N_s(t)$, where $s$ is a size; $t$ is a threshold, and $N$ is the number of fragments whose areas are greater than $s$. With the area changing assumption, for noise points their shape information, measured by $N_s(t)$, is fiercely varied while the threshold is changing. When an image contains only white noise following Gaussian distribution, the $N_s(t)$ curve is a bell shape with a sharp peak, as shown in Figure 2.8.

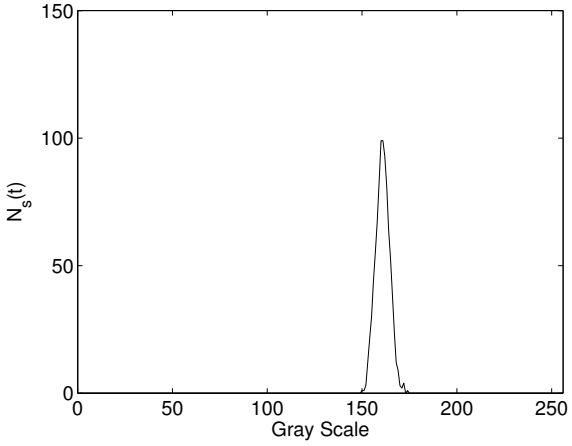


Figure 2.8: The $N_s(t)$ curve for an image containing white noise only. Reproduced from [30]

Therefore, when both objects and noise points exist in one image, the $N_s(t)$ curve is shown as Figure 2.9, in which the widespread plateau is generated by objects while the sharp peak is produced by noise. In [30], the global threshold is chosen as the middle value of the widest plateau of the $N_s(t)$ curve.

To remove noise points, Torabi *et al.* [42] introduce the area of segmented fragments as shape information. Without losing generality, the area of objects is assumed to be larger than that of noise points. Based on this assumption, if all the fragments in a segmented image have larger size, we may consider it to be a noise-free segmented result. In this case the global threshold is designed to maximize the minimum fragment, which is described in Equation 2.28.

$$T_{\max\min} = \max_{j=0\ldots k}[\min_{i=1\ldots n}(A_{i,j})] \tag{2.28}$$

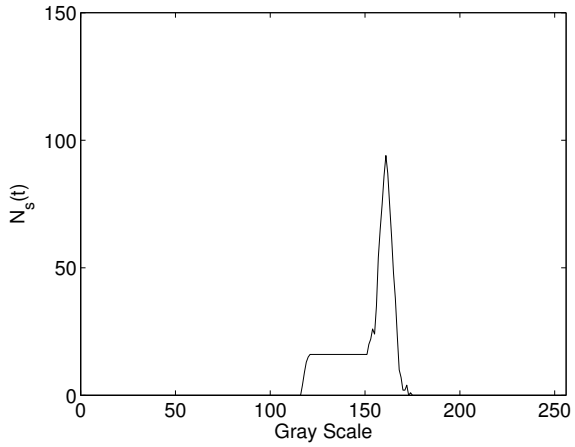


Figure 2.9: The $N_s(t)$ curve for an image containing objects and noise. Reproduced from [30]

where $A_{i,j}$ is the area of the $i^{th}$ fragment produced by the $j^{th}$ threshold, and $T_{\max\min}$ is the chosen global threshold.

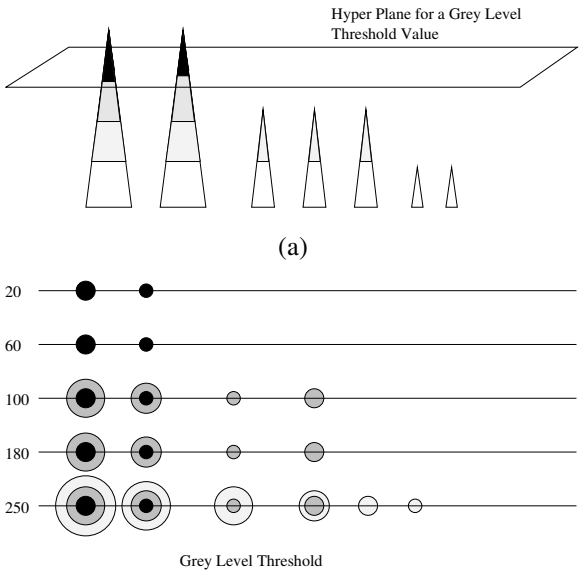


Figure 2.10: The process of choosing an optimal global threshold: (a) Threshold hyperplane in the grey scale depth of an image with two objects, (b) Particle count and size results at different threshold values. Reproduced from [42]

In an example given in [42], the algorithm is applied to an image containing two dark objects, a few smaller noise points, and a bright background. Considering intensity as "depth", we can get a side view, as shown in Figure 2.10a. The darker the fragment is, the higher its side view will be. A global threshold is applied as a hyperplane to cut the intensity of fragments. For each threshold, as shown in Figure 2.10b, the fragment size is decided by the projection of that fragment

on the threshold hyperplane. Finally, one global threshold will be picked out when it maximizes the smallest fragment. However, this method also has a limitation related to intensity. As shown in the example image, part of the object needs to be darker than noise points. Otherwise, if the intensity of those noise points is the same as the lowest intensity of objects, this method will not be able to exclude all the noise points. Therefore, for images containing noise with random intensity, the robustness of this method may be penalized.

**Local Thresholding**

Li and Najarian [19] introduce shape information for local thresholding. In this case local Regions of Interest (ROI) are generated in preprocessing by a global threshold algorithm. Similar to [30], the shape of objects is assumed to be almost the same while the threshold changes in a small range. In this case features describing shape information tend to be stable. In [19] the algorithm adopts shape features such as normalized compactness (NC) describing the roundness of objects and region expanding rate (RER) characterizing the stability. These features are defined as Equations 2.29 and 2.30, respectively.

$$NC = C^2/A \tag{2.29}$$

$$RER = \frac{\sqrt{\frac{A_n}{A_{n-1}} - 1}}{T_n - T_{n-1}} \tag{2.30}$$

In Equation 2.29, $C$ is an object contour length and $A$ is an area of object in a local ROI. For the application in [19], biomedical cells with rounded shape are objects of segmentation. Therefore, a smaller normalized compactness indicates a better segmentation. In Equation 2.30, $T_n$ and $T_{n-1}$ are thresholds applied to a local ROI, while $A_n$ and $A_{n-1}$ are object areas with respect to threshold $T$. A smaller region expanding rate implies a more stable shape that is preferred for obtaining a better segmentation. To simplify the criterion process, both normalized compactness and region expanding rate are integrated into one index $\phi$ that is defined as follows:

$$\phi = \sqrt{NC^2 + \alpha RER^2} \tag{2.31}$$

in which $\alpha$ is a weight to balance two features. The relationship between integrated index $\phi$ and threshold $T$ is presented in Figure 2.11.

To calculate an optimal threshold for a ROI, this algorithm selects a threshold resulting in the smallest $STB$ rather than simply choosing a threshold at the minimum point of $\phi$-Threshold curve. $STB$ is the stability of a window defined in Equation 2.32 and presented as the small rectangles in Figure 2.11.

$$STB = \frac{\max(\phi) - \min(\phi)}{\max(T) - \min(T)} \tag{2.32}$$

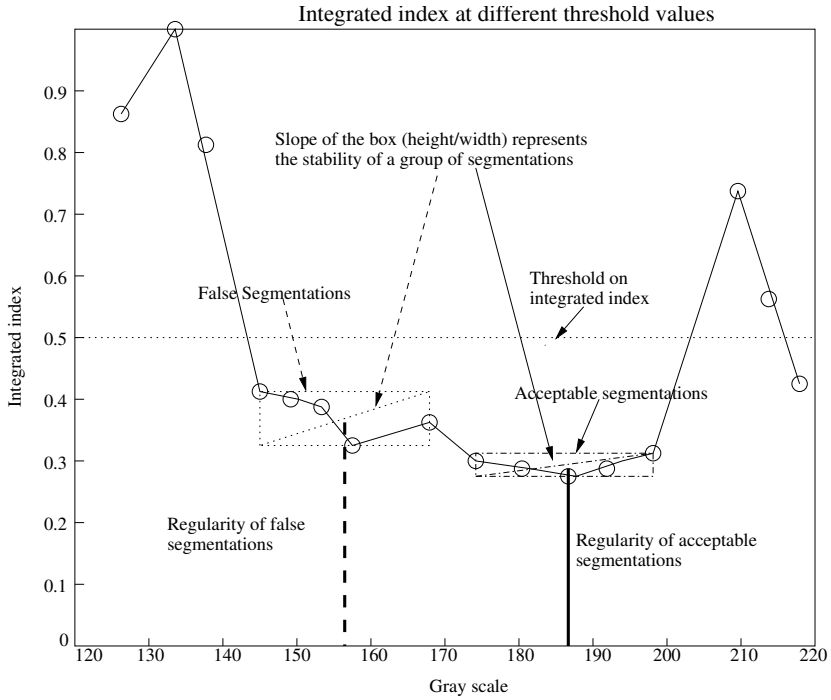


Figure 2.11: The curve of integrated index $\phi$ and threshold $T$. Reproduced from [19]

In this case, the smaller the $STB$ is, the more stable a fragment shape is and the higher the segmentation quality will be. Therefore, the shape information is incorporated in local thresholding by picking out threshold values that minimize the $STB$ value.

**Edge Detection**

In edge detection, the result is often affected by noise, which appears as short line fragments. To remove those noise line fragments in the final result, Venkatesh and Rosin [45] evaluate edges with local shape information. Generally, edges are assumed to be long and continuous lines; therefore, thinness and continuity are adopted as shape features. All edges are evaluated pixel by pixel. An edge pixel is measured in its $3 \times 3$ neighborhood. The continuity requires that edge pixels connect with each other. Therefore, as one edge pixel locates at the center of the neighborhood, there should be two more pixels around it having the same direction as the center one. To meet the requirement of thinness, the rest of the 6 pixels in this neighborhood should not belong to any edge. These shape features of edge are derived as follows:

$$A(\alpha, \beta) = \frac{\pi - |\alpha - \beta|}{\pi} \tag{2.33}$$

As shown in Equation 2.33, $A(\alpha, \beta)$ describes the agreement of two angles. With this measurement the continuity of an edge to its left and right are calculated by Equations 2.34 and 2.35 respectively,

$$L(k) = \begin{cases} A(d_c, d_k) \times A(\frac{\pi}{4}, d_c + \frac{\pi}{2}), & \text{if neighbor k is an edge pixel} \\ 0, & \text{otherwise} \end{cases} \tag{2.34}$$

$$R(k) = \begin{cases} A(d_c, d_k) \times A(\frac{\pi}{4}, d_c - \frac{\pi}{2}), & \text{if neighbor k is an edge pixel} \\ 0, & \text{otherwise} \end{cases} \tag{2.35}$$

in which $d_c$ is an edge gradient direction of the center point, while $d_k$ is an edge gradient direction at the $k^{th}$ neighborhood point. Neighborhood points are labeled in an anti-clockwise direction starting at the left point next to the center one. Therefore, the continuity is calculated from Equation 2.36, and the thinness is obtained from Equation 2.37, in which $nE$ is the number of observed non edge pixels in the $3 \times 3$ neighborhood, while RP equals 6, representing the ideal number of non edge pixels. [45].

$$C = \frac{1}{2}(\max[L(i)] + \max[R(i)]) \tag{2.36}$$

$$T = \frac{nE}{RP} \tag{2.37}$$

$$E = \gamma C + (1 - \gamma)T \tag{2.38}$$

Finally, as shown in Equation 2.38, the continuity $C$ and thinness $T$ are linearly combined with a normalized weight $\gamma$. The combined feature index $E$ indicates a perfect edge when it reaches $1$ and a worst case when it is $0$. Only edges with an $E$ value higher than a fixed threshold are selected for the final result.

**Summary**

Shape information is adopted to enhance segmentation quality. From previous investigations, shape features are exploited for images containing objects with similar shape, such as biomedical images with cells [42, 19] or images for edge detection [45]. Moreover, shape information is usually highly related to specific application domains. In most cases shape features can only characterize one kind of shape in an image. If the category of target objects is changed, new shape information will be required. Finally, in most presented cases, shape information is used to remove noise as a post-processing step such as removing segmented noise in global thresholding [30] or eliminating unwanted line fragments in edge detection [45]. In general incorporating shape information could improve segmentation quality, while approaches including the shape information still need to be developed.

## 2.4 Summary

In this chapter, some previous investigations related to local thresholding and shape information are briefly introduced. A lot of local threshold algorithms are based on statistical features of intensity, such as local variance methods, local contrast methods, and center-surround methods. Local neighborhood size in these methods often needs to be manually tuned to reduce noise while keeping local properties. Another type of local threshold method determines threshold values by fitting a threshold surface with an intensity one. In this case an energy function is introduced to incorporate intensity features as well as to keep the smoothness of the threshold surface. Almost all of these local threshold methods are based merely on intensity information. However, in many application domains intensity is often degraded by types of noise and cannot supply enough information to produce segmentation results satisfying practical requirements. On the other hand, some investigations include shape information in algorithms working on images containing objects with similar shape. In most of them shape information is used in a post-processing step to remove noise or unwanted fragments. With the aid of shape information those algorithms often tend to be more robust to noise. Therefore in this thesis, shape information is incorporated into a local threshold algorithm to enhance segmentation quality. Different from just adopting shape information in a post-processing step, local thresholds in our algorithm are selected by shape features. Further details of our algorithm will be presented in Chapter 3.

# Chapter 3

# Local Threshold Algorithm Based on Shape Information

The previous chapter introduced investigations into local thresholding and shape information. Some drawbacks of local thresholding based merely on intensity information were analyzed. To enhance segmentation quality, we plan to include shape information learned from ground truth images into a local threshold algorithm. This algorithm is presented in greater detail in Chapter 3.

## 3.1 Introduction

Local thresholding is a simple and efficient method for image segmentation. However, most local threshold algorithms are based merely on the intensity information, which makes them quite sensitive to noise. In general shape information characterizing geometric features of objects tends to be more robust to noise than intensity information. In this chapter, we present an algorithm that selects local threshold values based on shape features learned from ground truth images as prior knowledge. Experimental results for the proposed algorithm are compared with competing local threshold algorithms. Our method is demonstrated to be superior to others with respect to segmentation accuracy.

## 3.2 Outline of Local Threshold Process

In our algorithm shape information is incorporated into a local threshold algorithm in order to improve the segmentation quality. The process of this method is briefly presented in Figure 3.1.

In the first stage shape information is encoded and then derived from objects in ground truth images. Solidity and angle-curve difference are adopted in the proposed algorithm as shape features. Then, distributions of shape features are generated by kernel density estimation to approximate the nature of particles. In the second stage a global threshold algorithm is applied to a gray-scale image to produce fragment candidates. Bonding boxes of those fragment candidates are used as Regions of Interest (ROI), and a group of thresholds are locally applied to every ROI. One threshold will be picked if the corresponding fragment has the highest probability in feature distributions with respect

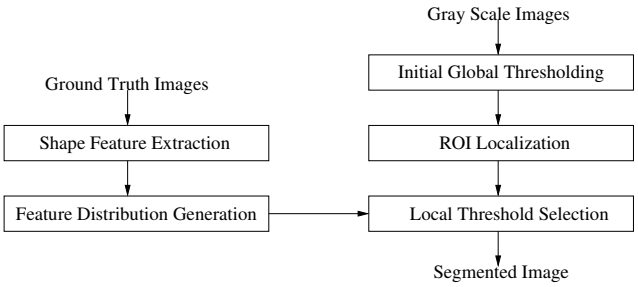


Figure 3.1: Outline of Local Threshold based on Shape Information

to all the other local thresholds. Finally, the result is segmented by a local threshold selected for each ROI, respectively.

## 3.3 Shape Features Selection

Shape information is highly related to specific application domains. Geometric features that characterize object shape usually vary from one kind of application to another. In oil sand images objects often tend to have a rounded convex shape and a relatively smooth boundary. Therefore, we adopt solidity to describe the convexity of segmented fragments and angle-curve difference to measure the smoothness of fragment boundaries. These shape features can distinguish objects from background with the specific process stated above.

### 3.3.1 Solidity

The definition of solidity follows Equation 3.1 [24], in which "Convex Area" is the area of the smallest convex polygon bounding the fragment.

$$Solidity = \frac{Area}{Convex\ \ Area} \tag{3.1}$$

This feature is used to distinguish convex objects from the background as well as to identify concave spurious shapes. In the oil sand image segmentation domain, well segmented fragments are mostly convex, while fragments that are under-segmented due to object fusion tend to have a concave shape. Through simple experiments we investigate the relationship between solidity value and segmentation quality that is measured by a segmentation score [32] with respect to the ground truth. Solidity and segmentation score are obtained from previous segmentation results. For segmented fragments having the same solidity value, an average segmentation score is calculated. The relationship of the average segmentation score and the solidity is plotted in Figure 3.2.

From this figure we can see that most fragments with a higher segmentation quality have solidity values concentrated around $0.9$. This matches the observation that objects in oil sand images tend to have a convex rounded shape. Moreover, the segmentation score decreases drastically when solidity is close to $1$. This phenomenon is primarily resulting from two factors: over segmentation and noise

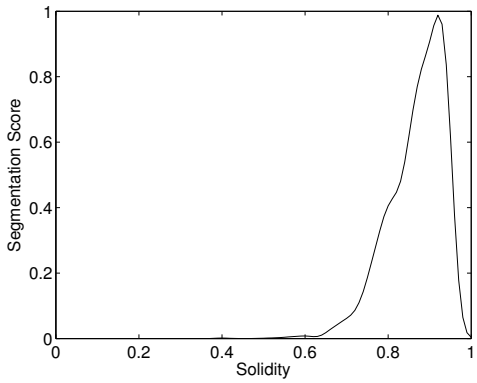


Figure 3.2: Solidity vs. Average Segmentation Score

points among segmented fragments. In some cases fragments are over segmented, and part of those over segmented fragments could have an almost rounded shape. This means that the solidity of those fragments could be very high, but their segmentation score is relatively lower. Another factor is the noise point. As shown in Figure 3.1, fragment candidates are generated by a global threshold algorithm. Some noise points will also be segmented and included as candidates. These points can produce a fragment with extremely high solidity, but their segmentation scores tend to be $0$. In this case, the average segmentation score is significantly decreased when the solidity is too high.

### 3.3.2 Angle-curve Difference

Angle-curve Difference [16] characterizes the smoothness of object boundaries. It measures the difference between the central angle curve of a segmented fragment and the standard one learned from ground truth objects. A central angle is illustrated in Figure 3.3. One side of the central angle $\phi$ is the horizontal axis, and the other moves along the fragment boundary in an anticlockwise direction.

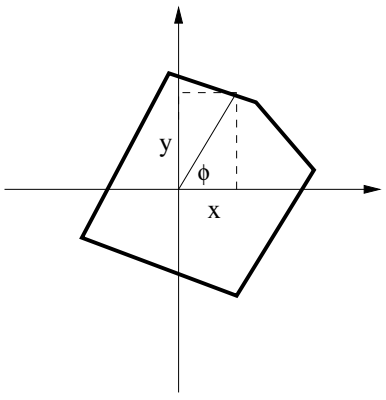


Figure 3.3: Central Angle

Every point $(x(l), y(l))$ on the fragment boundary is a parameter with respect to the normalized length $l$, where $0 \leq l \leq 1$. The normalization step causes fragments in different sizes to have the same scale. The central angle $\phi$ can be derived from Equation 3.2.

$$\phi(l) = \arctan \frac{y(l)}{x(l)} \tag{3.2}$$

If we plot the central angle curve in a plane, as shown in Figure 3.4, the horizontal axis is the normalized length, and the vertical axis is the central angle in the range of $[-\pi, \pi]$. The standard angle curve is defined as the mean of central angle curves corresponding to all the objects in ground truth images. In Figure 3.4 dash lines $\phi_g$ are central angle curves produced by ground truth objects. A mean point $m(i)$ is calculated as an average value of all the points in $\phi_g$ sharing the same index $i$ at the horizontal axis. In Equation 3.3 $K$ is the number of ground truth objects equal to the total number of $\phi_g$. Those mean value points $m(i)$ along the horizontal axis form the standard angle curve $\phi_m$ indicated by the solid line in Figure 3.4. Therefore, by generating this standard angle curve $\phi_m$, the shape information about the central angle curve is retrieved from objects in ground truth images. The difference between an arbitrary central angle curve and this standard one is derived to define the similarity between a segmented fragment and ground truth objects.

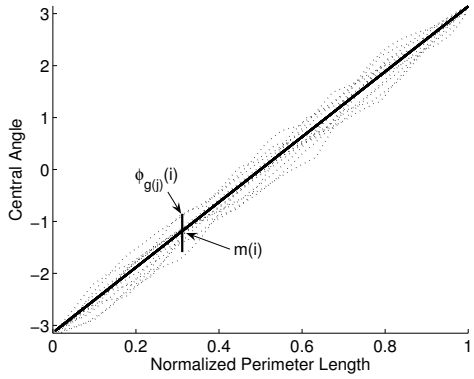


Figure 3.4: Standard Angle-curve Calculation

$$m(i) = \sum_{j=1}^{K} \phi_{g(j)}(i) \tag{3.3}$$

As shown in Figure 3.5, the difference between an arbitrary central angle-curve $\phi$ and a standard angle-curve $\phi_m$ is measured point by point, and can be written as $\phi(i) - m(i)$, in which $\phi(i)$ is a point in the central angle-curve and $m(i)$ is the equivalent one in the standard angle curve. For a segmented fragment the angle-curve difference is defined as the average of absolute values of point differences along the length of the normalized perimeter, as shown in Equation 3.4, where $N$ is the number of points along the horizontal axis. Since the central angle is calculated when its one side moves along the fragment boundary, the angle-curve difference is quite sensitive to the boundary

variation. While most objects in ground truth images have a smooth boundary, this shape feature value indicates the smoothness of a fragment.

$$Diff_{Angle\ Curve} = \frac{1}{N}\sum_{i=1}^{N}|\phi(i) - m(i)| \tag{3.4}$$

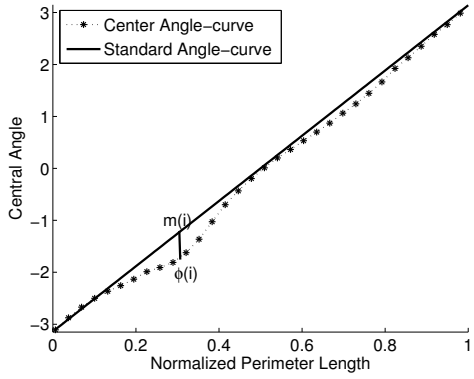


Figure 3.5: Difference between Arbitrary Central Angle-curve and Standard Angle-curve

We also investigate the relationship between the angle-curve difference and the segmentation quality indicated by a segmentation score [32]. Similar to the solidity, the angle-curve difference and the segmentation score are obtained from each segmented fragment and plotted in Figure 3.6. As with the observation that oil sand ores often have a smooth boundary, fragments with a high segmentation score tend to have a low angle-curve difference. Resulting from over segmentation and noise points, when angle-curve difference is close to $0$, the segmentation score also decreases significantly.

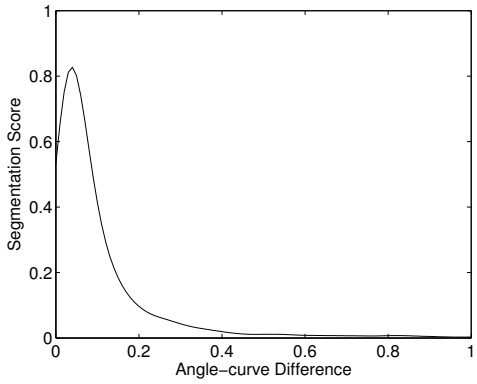


Figure 3.6: Angle-curve Difference vs. Average Segmentation Score

## 3.4 Feature Distribution Generation

To simulate the nature of particles in a real application, we build probability distributions of selected shape features. The solidity value and angle-curve difference of objects in ground truth images are adopted as observations, then kernel density estimation [40] is exploited to generate probability density functions. In this case we assume the probability value derived from observation frequency indicates the possibility of a fragment with specific features being a good segmentation.

The kernel estimator is one of most commonly used and studied estimators [40]. Following kernel properties, a continuous and differentiable probability distribution can be derived from discrete observation data. Moreover, without manual selection, a smooth parameter in the kernel can be automatically calculated from input samples. These features make the kernel estimator be able to handle a lot of probability estimation problems accurately and conveniently. Specifically, the probability density distribution is estimated by Equation 3.5 [40],

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{3.5}$$

in which $K$ is a kernel, $h$ is the smoothing window size, $n$ is the number of observations, $x$ is the mean value of these observations, and $X_i$ is an arbitrary observation value. The Gaussian kernel is chosen for the proposed algorithm. On the other hand, the smoothing window size $h$ is calculated by Equation 3.6, in which $\sigma$ is the standard deviation of observations and $IQR$ is the interquartile range.

$$\begin{aligned} h_{opt} &= 0.9A * n^{-1/5} \\ A &= min(\sigma, IQR/1.34) \end{aligned} \tag{3.6}$$

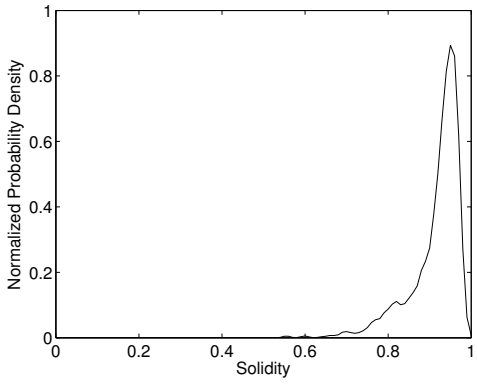


Figure 3.7: Probability Density Distribution of Solidity

From ground truth images in our application, we are able to obtain the probability density distributions of solidity and angle-curve difference shown in Figures 3.7 and 3.8. Since the probability density values retrieved from these two distributions will be utilized to determine the local threshold

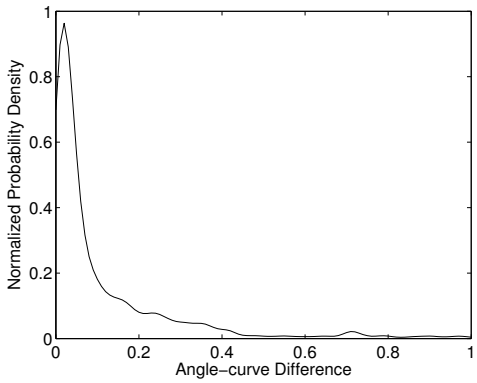


Figure 3.8: Probability Density Distribution of Angle-curve difference

value, to balance the effect of these two features, the probability density distributions are normalized into the same scale. In these two probability distributions, most oil sand objects in ground truth images have a high solidity value and a low angle-curve difference. These distributions also match our previous observation that oil sand ores tend to have a rounded shape and smooth boundary. Since the probability density distributions are obtained from ground truth images, we assume that for a segmented fragment, the probability density values retrieved from these distributions reflect the possibility of this fragment being well segmented. On the other hand, from those two figures, it can be also noted that when the solidity value is close to $1$ or the angle-curve difference is close to $0$, both of their probabilities are significantly decreased. This implies that even in ground truth images, objects are seldom have a "perfect" shape. Considering that the probability value indicates the possibility of a fragment being well segmented, we can say a fragment with an extremely high solidity or an extremely low angle-curve difference is too good to be true. This conclusion is quite similar to the result retrieved from curves with respect to their segmentation score and shape features. In previous segmented results, we also found when solidity is too high or angle-curve difference is too low, fragments often acquire a bad segmentation, which may resulted from over-segmentation or noise points. The two disciplines learned from ground truth images and segmentation results are matched with each other. Therefore, when we choose a threshold to maximize the probability density values of those two features - which will be described in detail in next section - we can avoid both under segmented spurious fragments as well as over segmented fragments.

## 3.5 Local Threshold Determination

A local threshold value is selected for each fragment in a local region to maximize a joint probability of shape features. In our algorithm the initial threshold is chosen by a global algorithm such as Otsu [27], and this threshold is used to produce the initial segmentation of the image into connected components or fragments. Bounding boxes of these fragments are used as local regions of

interest (ROI). A group of thresholds are then applied to a ROI, each producing a slightly different segmented fragment. Shape features, including solidity and angle-curve difference of the fragment, are obtained. Probability densities of these shape features are retrieved from probability density distributions learned from objects in ground truth images, respectively. Finally, we choose the optimal local threshold as the one that maximizes a joint probability of shape features.

The process of local threshold selection is described by Equation 3.8, in which $t$ stands for a group of thresholds varying from 0 to 255. Shape feature values such as solidity and angle-curve difference are changed with the variation of the threshold. We can therefore map the relationship between them as a function whose domain is the local threshold value $t$ and whose range is the value of the shape feature, which can be expressed as $S(t)$ and $A(t)$, representing solidity and angle-curve difference, respectively. The probability density values, which are written as $P(S)$ and $P(A)$, are retrieved from the distributions estimated with kernel method. The correlation of these two probability density values are investigated as follows.

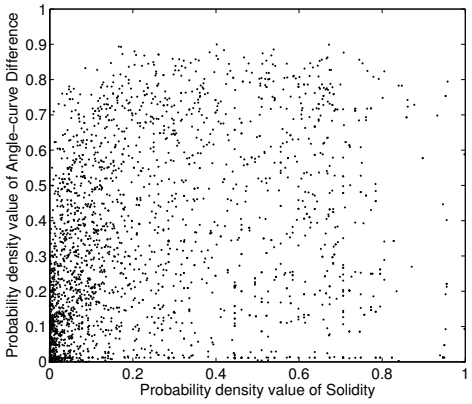


Figure 3.9: Probability Density Value Points of Shape Features

In the Figure 3.9 all the probability density value points are derived from previous segmentation results that include both good and bad segmentations. With respect to one probability density value of solidity, the probability density value of angle-curve difference appears randomly in the plane, and vice versa. There is no pattern formed by these two feature values. Moreover, in accordance to the definition of variable independence, two random variables $S$ and $A$ are statistically independent from each other if and only if the conditional probability $P(S|A)$ of $S$ given $A$ satisfies $P(S|A) = P(S)$. In this case we choose an arbitrary probability density value of angle-curve variance $P(a)$ and estimate the probability density distribution of solidity with respect to the chosen $P(a)$. This probability density distribution is the conditional probability $P(S|A)$

The Figure 3.10 illustrates the two probability density distributions: one is with respect to the probability of solidity and another is with respect to the conditional probability. As shown in this figure, these two distributions are almost overlapped with each other. We also calculated the Bhattacharyya

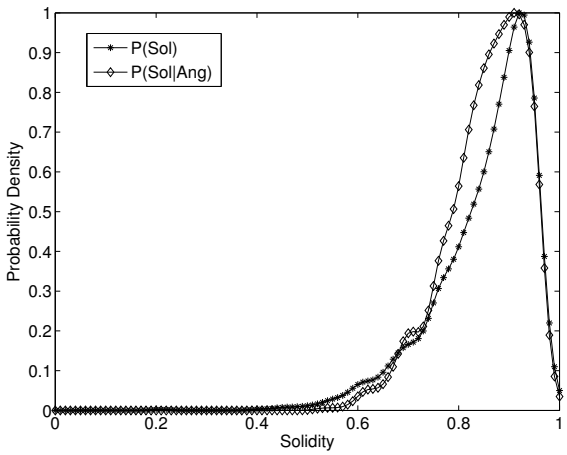

Figure 3.10: Probability Density Distributions of Solidity

correlation [3] between these distributions, which measures the similarity between probability density distributions. The result of Bhattacharyya correlation is $0.9799$ and indicates that the $P(S|A)$ is almost equals with the $P(S)$. Therefore, we consider these two probability density $P(S)$ and $P(A)$ are independent from each other, and the joint probability $P(S, A)$ is written as Equation 3.7,

$$P(S, A) = P(S) \cdot P(A) \tag{3.7}$$

$$t^* = \arg\max_t [P(S(t)) \cdot P(A(t))] \quad t \in [0, 255] \tag{3.8}$$

The optimal local threshold value $t^*$ is selected to maximize the joint probability, which is written as Equation 3.8. With the assumption that the probability of shape features indicates a possibility of a fragment being well segmented, a threshold maximizing the joint probability results in an optimal segmented fragment in the local region.

## 3.6 Experimental Results

The local threshold algorithm presented in this thesis is applied to a synthetic image containing regular geometric shapes. The intensity of each fragment is randomly varied, which simulates the illumination changes in practice usages. Some random noise points are also included to test the algorithm's robustness to noise. Several local threshold algorithms are chosen for comparison. They include Liu *et al.*'s locally adaptive threshold based on active-surface [21], Saha and Ray's variational minimax optimized threshold [35], and Yan *et al.*'s multistage adaptive thresholding (MAT) [52]. These local threshold algorithms represent different aspects. Liu *et al.*'s approach thresholds images by minimizing energy function; Saha and Ray's approach focuses on choosing parameters automatically; and Yan *et al.*'s algorithm belongs to intensity analysis approaches and combines global and local thresholds together.

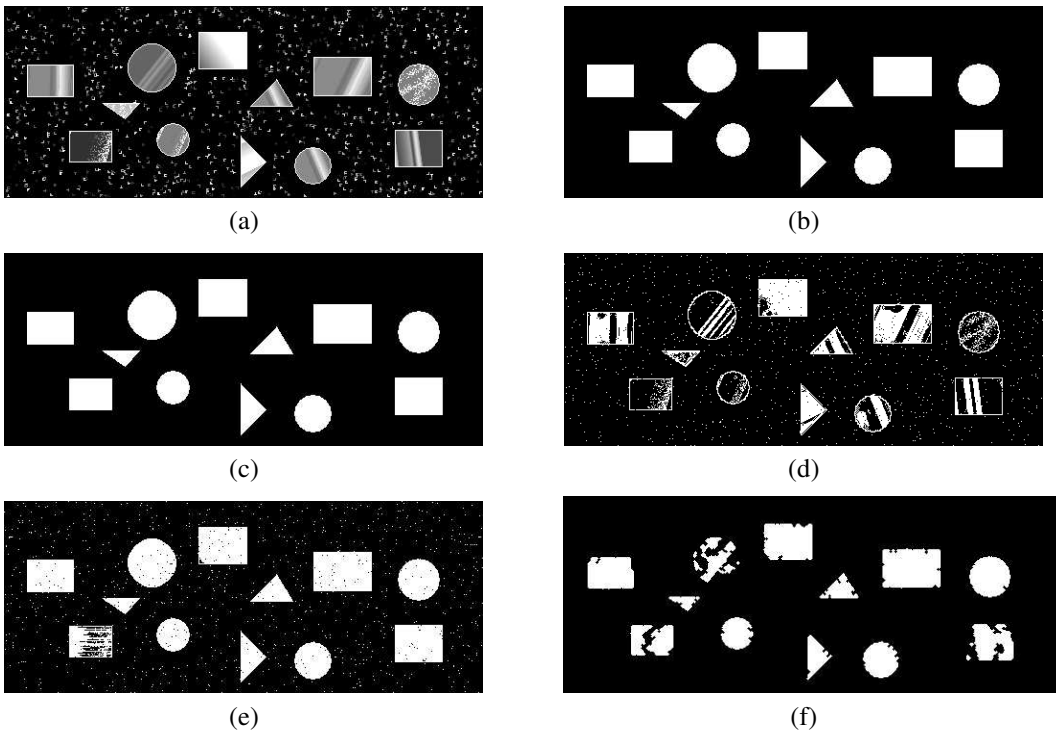

Figure 3.11: Artificial Image: (a) Original image, (b) Ground truth image, (c) Local Threshold based on Shape Information, (d) Variational Minimax, (e) Active surface, (f) Multiple-stage Adaptive Threshold (MAT)

As shown in Figure 3.11, the segmentation result produced by our algorithm is compared with these local threshold algorithms and demonstrates that the proposed algorithm is more robust to noise and can handle illumination changes better. It also shows that the local threshold algorithm including shape information is superior to those algorithms based only on intensity analysis.

The proposed local threshold algorithm is also applied to oil sand images and compared with the same competing algorithms stated above. To evaluate the performance of each algorithm, ground truth images serve as references. The Object level Consistency Error (OCE) [32], which is defined in Equation 3.9, is used as the criterion.

$$\begin{aligned} OCE(I_g, I_s) &= min(E_{g,s}, E_{s,g}) \\ E_{g,s}(I_g, I_s) &= \sum_{j=1}^{M} \left[1 - \sum_{i}^{n} \left( \frac{|A_j \bigcap B_i|}{|A_j \bigcup B_i|} \times \frac{\delta(|A_j \bigcap B_i|)|B_i|}{\sum_{k=1}^{N} \delta(|A_j \bigcap B_k|)|B_k} \right) \right] \frac{A_j}{\sum_{l=1}^{M} |A_l|} \end{aligned} \tag{3.9}$$

where $A_j$ is the area of the $j^{th}$ fragment in ground truth image $I_g$, and $B_i$ is the area of $i^{th}$ fragment in segmented image $I_s$. In accordance with the definition of the OCE score, this criterion considers a fragment with greater size to be more critical than a smaller one. This property matches well with requirements in oil sand mining because a larger oil sand ore is more valuable while more easily results in mechanical problems such as jamming a crusher. The scores of $E_{g,s}$ and $E_{s,g}$ are one-directional measurements of segmentation quality with either the ground truth image $I_g$ or

| | $E_{g,s}$ | $E_{s,g}$ | OCE |
|---|---|---|---|
| Solidity based local threshold | **0.5464** | **0.5617** | **0.5762** |
| Active surface | 0.2555 | 0.2935 | 0.2553 |
| Variational minimax | 0.4106 | 0.4589 | 0.4908 |
| MAT | 0.5076 | 0.5677 | 0.5057 |

Table 3.1: Object level criterion result of local threshold segmentation

the segmented image $I_s$ serving as the reference, while the OCE score compares the result with reference images in two directions and penalizes both over-segmentation and under-segmentation.

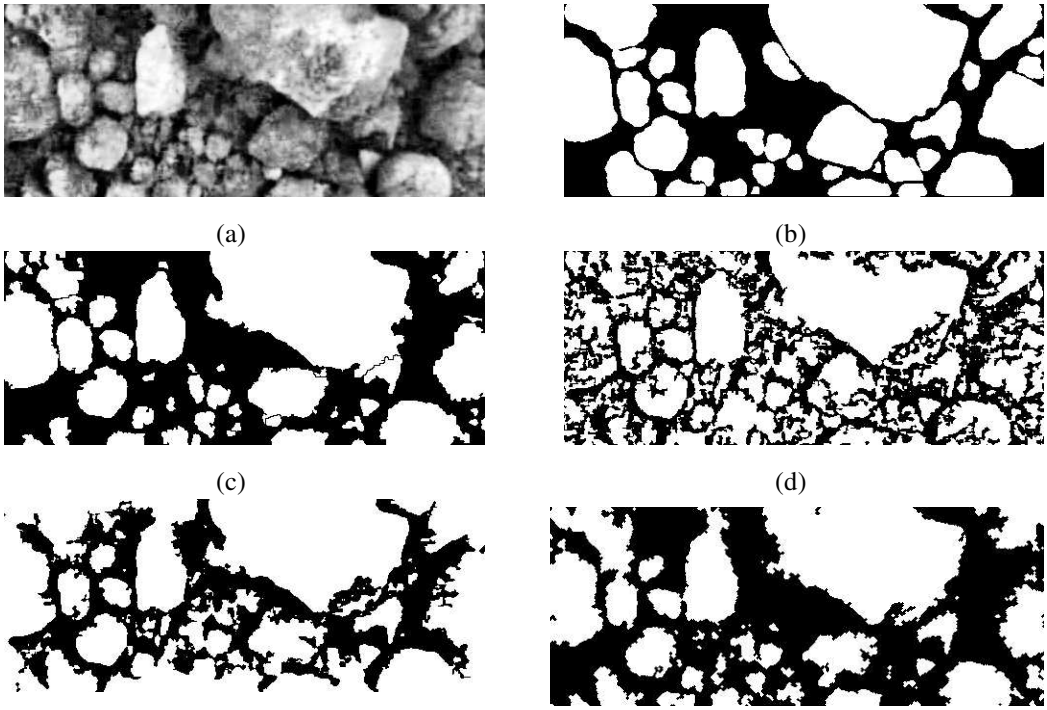


Figure 3.12: Segmentation Result 1: (a) Original image, (b) Ground truth image, (c) Local Threshold based on Shape Information, (d) Variational Minimax, (e) Active surface, (f) Multiple-stage Adaptive Threshold (MAT)

In our experiments, the test dataset includes 60 oil sand images, each of them containing $200 \times 500$ pixels. Average $E_{g,s}$, $E_{s,g}$ and OCE score are compared among competing algorithms. From Table 3.1 it can be found that among compared local threshold methods, our local threshold with shape information gets the highest score in these three criteria. This also demonstrates that the proposed algorithm is superior to other local thresholding in the oil sand application domain. On the other hand, although the segmentation score indicates that our algorithm has the best segmentation quality, a higher segmentation accuracy is still required by the oil sand mining industry. Further investigations on oil sand application are developed in the next chapter.

Segmented images are presented in Figures 3.12 and 3.13. The results demonstrate that the proposed local threshold method contains fragments of higher quality than others, while there are

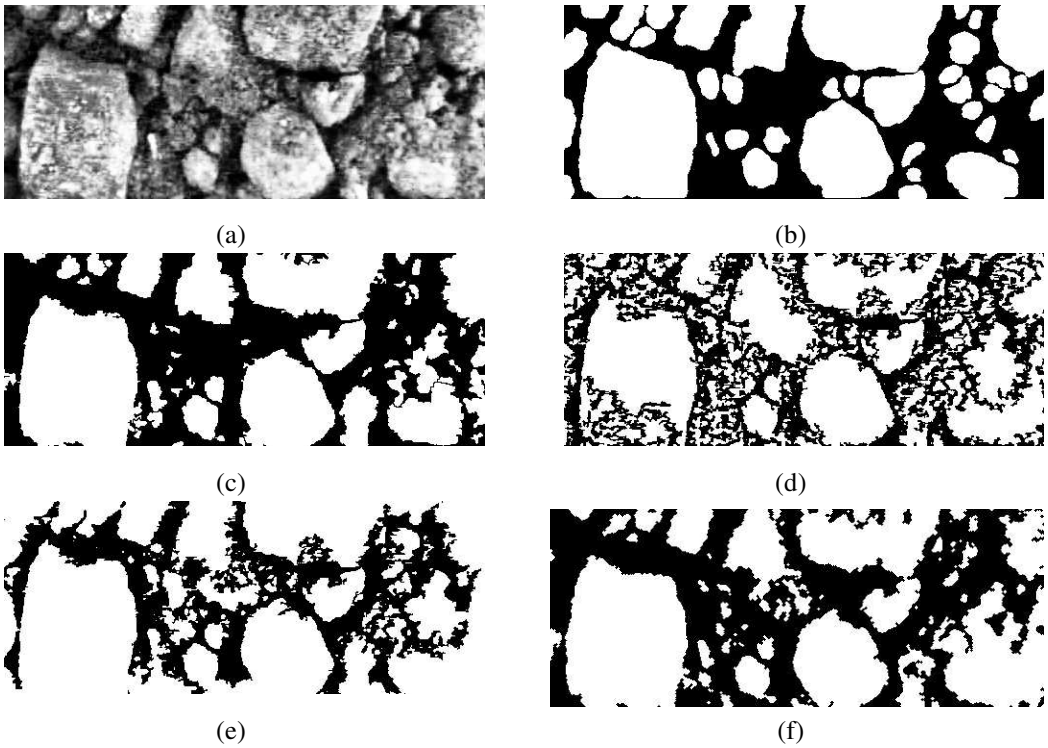

Figure 3.13: Segmentation Result 2: (a) Original image, (b) Ground truth image, (c) Local Threshold based on Shape Information, (d) Variational Minimax, (e) Active surface, (f) Multiple-stage Adaptive Threshold (MAT)

also some unwanted fragments in the final results which penalize the segmentation accuracy. The results of segmented images also support the conclusion derived from the OCE metrics.

## 3.7 Summary

In this chapter, a local threshold algorithm based on shape information is presented. This method incorporates shape features, such as solidity and angle-curve difference, to characterize objects and adaptively selects local thresholds by maximizing a joint probability of shape features. Experiments on a synthetic image as well as oil sand images illustrate that the proposed algorithm is superior to other local threshold algorithms in terms of segmentation quality. However, the segmentation accuracy needs to be further enhanced to meet requirements in the oil sand industry. Additional investigations with respect to oil sand applications will be presented in the next chapter.

# Chapter 4

# Application to Oil Sand Images

In the previous chapter we introduced a local threshold algorithm based on shape information and applied it to oil sand images. The experimental results demonstrate that our proposed algorithm is superior to the other local threshold algorithms in terms of segmentation quality. However, for an industrial application some fragments in oil sand images cannot produce results with good shapes only through changing threshold values. These fragments with unwanted shapes need to be removed, and make the average segmentation accuracy high enough for the practical application. In this chapter our investigation is focused on designing a classifier based on shape information to eliminate unwanted fragments from local thresholding results. A variety of classification algorithms and shape features are explored as candidates. Derived from different aspects, several criteria are introduced to evaluate the quality of classification. Experiments are designed in order to measure the performance of classifiers and shape features in terms of different metrics.

## 4.1 Introduction

Oil sand is an important natural resource and economic support for the province of Alberta. The crude oil produced from oil sand not only meets the energy requirements of Canada but also North America. Because of its main components the size of oil sand can change significantly in warm weather and extremely low temperatures. A set of screens are utilized in the mining process to separate oil sand ore in accordance to their size. The efficiency of these screens are evaluated by the ratio of the amount of ores passed the screen over the the amount of ores should have passed. [31] To capture the size information that indicates the efficiency of screens, a camera is set over the convey belt and obtains oil sand images. The size of oil sand ores can be derived from the size of segmented fragments in the oil sand images. Therefore, we would like to produce more accurate segmentation results to obtain the size information with higher accuracy.

The oil sand's size is analyzed statistically. For a group of images, the sizes of objects are calculated from fragments in segmented images, then a size distribution is produced for further analysis. If the number of fragments decreases in one image, we can incorporate more images into

the group to compensate for the loss of the fragment quantity. In this case the accuracy of the size distribution primarily depends on the segmentation quality of included fragments, while the fragment quantity is a much less critical factor.

With the specific situation in the oil sand application we introduce a classifier based on shape information in order to reject unwanted segmented fragments and enhance the average segmentation accuracy. Since the particle size information needs to be statistically analyzed, we assume that the introduced classifier dose not have any bias on fragment size. This assumption can be validated by the size distribution of particles in experimental results. As candidates of the classifier, two classification algorithms including Support Vector Machine (SVM) and Least Square (LS) are investigated. The SVM is a widely used classification algorithm. It has the property to change a non-linear separable problem to a linear separable one through mapping the problem to a higher dimensional space. On the other hand, the LS algorithm is simple and efficient in the classification domain. In addition to the selected classification algorithms, shape features utilized by these classifiers are also investigated to look for a shape feature that matches a classifier best. Several classification criteria such as the Object-level Consistency Error (OCE) score, quantity of segmented fragments, Particle Size Distribution (PSD), and the convergence curve of PSD are adopted to evaluate the performance of different combinations of classifiers and shape features. Finally, conclusions are drawn from the experimental results.

## 4.2 Classifier with Shape Information

A classifier based on shape information is trained by both positive and negative samples and designed to reject unwanted fragments. The objects in ground truth images are utilized as positive samples, while fragments in failure cases are adopted as negative samples. The quantity of positive and negative samples is balanced to keep the decision boundary from being biased.

The shape information adopted by classifiers is characterized by the solidity as well as the angle-curve difference. Different from using the value of these shape features directly, probability density values learned from shape feature distributions are exploited to build a shape feature plane. The probability density values of features are utilized because they match the forms of features that select the local threshold in the previous step. The feature plane built by probability density values is shown as Figure 4.1, where the horizontal and vertical axes are the probability density value of solidity and angle-curve difference, respectively; the “stars” in the plane stand for positive samples which are derived from the ground truth images; while the “diamonds” represent negative samples derived from failure cases in previous research. These positive and negative samples shown in Figure 4.1 are utilized as the training data for the classifiers introduced as follows. A classification boundary needs to be determined by the following classification algorithms.

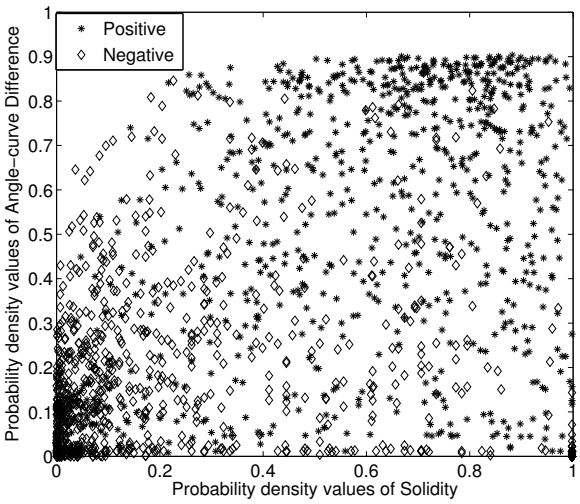


Figure 4.1: Training Samples

### 4.2.1 Support Vector Machine

The Support Vector Machine (SVM) is widely used in classification problems. For example, Shanahan and Roma [38] use the SVM for text classification, Waring and Liu [47] adopt the SVM to detect human faces, and Barutcuoglu *et al.* [1] exploit this classification approach to gene function prediction.

The main advantage of the SVM classifier is mapping a non-linear separable problem into a linear separable problem using kernel methods. For mapping problems the kernel method plays a critical role. However, the kernel method needs to assess all the possible data to obtain a solution. This is not executable in a practical application domain and limits the generality of kernel methods. To solve this problem, the SVM algorithm supplies a sparse solution by choosing parameters in accordance with a convex optimization process that indicates a local optimum solution can also be a global optimum solution. Therefore, the SVM classifier is able to make a prediction based only on a subset of training data [4]. Since the SVM classifier can handle both linear and non-linear separable problems as well as predict with a subset of training data, this method is able to work with many complex applications, even including some overlapping class distributions. In this case the SVM classifier is applied to the space generated by shape features of oil sand fragments. The training set for the SVM classifier includes positive samples derived from ground truth images while negative samples came from the failure cases in previous research. The feature space and sample points are illustrated in the Figure 4.1. The segmented fragments produced by the proposed local threshold algorithm are adopted as the testing set. The SVM classifier labels the data in the testing set into two categories: the “good segmentation” and the “unwanted segmentation”. Those fragments that are labeled as unwanted are removed from the final segmentation result. Therefore, we incorporate the same shape information utilized in the proposed local threshold algorithm to select fragments

similar to the expected segmentations that are indicated by the ground truth images.

The Libsvm [7], which is a software library for the SVM application, is utilized to incorporate the SVM classifier in our algorithm. This software library includes the $C$-SVM, $\mu$-SVM, and $\varepsilon$-SVM. The $C$-SVM is chosen for our classification problem because of the convenience of its selecting parameters. Since in the oil sand application testing data need to be classified into two classes, the linear classification boundary is written as Equation 4.1 for the two-class classification problem,

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{4.1}$$

in which $\mathbf{x}$ is a training vector containing multiple features, $y$ is a label belonging to $[-1, 1]$, and $\phi$ is a factor of the kernel function $\mathbf{K}(\phi(x), \phi(x'))$. This boundary function is derived by $C$-SVM method, shown as Equation 4.2:

$$\min \ \ \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l} \varepsilon_i \tag{4.2}$$

where $C$ is the parameter penalizing errors indicated by $\varepsilon_i$. The classification boundary is primarily decided by the parameter matrix $\mathbf{w}$ that minimizes the kernel function.

The kernel selection plays an important role in SVM classification. Several kernels are often used in classification problems, such as linear kernel, polynomial kernel, radius basis function (RBF), and sigmoid kernel. In our algorithm the RBF kernel defined by Equation 4.3 is adopted.

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2), \quad \gamma > 0 \tag{4.3}$$

The RBF kernel non-linearly maps the data set into a higher dimensional space [7]. In this case the SVM classifier can handle the nonlinear classification problems. In accordance with the investigation by Keerthi and Lin [14], the linear kernel can be considered as a special case of RBF, because with the same penalty parameter $C$ linear and RBF kernels perform the same in linear classification problems. The research done by Lin and Lin [20] also shows that using certain parameters the performance of RBF and sigmoid kernels are quite similar. With respect to hyper parameters, RBF is superior to a polynomial kernel in terms of calculation complexity [7]. Parameters such as $C$ and $\gamma$ are tuned by cross validation and grid-search that are listed in the guidance paper provided with libsvm [7]. At the beginning a loose search is applied to parameters. Using the recommended value in [7], the value of $C$ ranges from $-5$ to 7 and $\gamma$ varies from $-7$ to 3. The average accuracy of a 5-fold cross validation is then obtained for each point in the grid shown as Table 4.1. The average accuracy reaches its highest value when the value of $C$ and $\gamma$ fall into the range of $[-5, -3]$ and $[-7, -5]$, respectively. After that another grid shown as Table 4.2 is produced for a fine search. Finally, we get the highest average accuracy $79.38\%$ of 5-fold cross validation, when $C$ is $0.0442$ $(2^{-4.5})$ and $\gamma$ equals $0.0078$ $(2^{-7})$.

| $C\backslash\gamma$ | $2^{-7}$ | $2^{-5}$ | $2^{-3}$ | $2^{-1}$ | $2^{1}$ | $2^{3}$ |
|---|---|---|---|---|---|---|
| $2^{-5}$ | **79.1922** | **79.1310** | 78.4578 | 78.0294 | 52.3868 | 52.3868 |
| $2^{-3}$ | **79.0698** | **79.1310** | 79.1310 | 77.9070 | 54.5900 | 52.3868 |
| $2^{-1}$ | 79.1922 | 79.1922 | 78.8862 | 77.9682 | 77.7846 | 52.3868 |
| $2^{1}$ | 72.3378 | 72.2766 | 72.0318 | 71.9706 | 77.9070 | 52.3868 |
| $2^{3}$ | 69.5838 | 69.5838 | 69.1554 | 77.2338 | 76.1322 | 52.3868 |
| $2^{5}$ | 69.9718 | 68.5435 | 67.9315 | 76.1322 | 68.9106 | 52.3868 |
| $2^{7}$ | 68.5435 | 68.2987 | 69.6450 | 71.6646 | 76.8054 | 52.3868 |

Table 4.1: Loose Search for $C$ and $\gamma$

| $C\backslash\gamma$ | $2^{-7.5}$ | $2^{-7}$ | $2^{-6.5}$ | $2^{-6}$ | $2^{-5.5}$ | $2^{-5}$ |
|---|---|---|---|---|---|---|
| $2^{-5.5}$ | 78.9474 | 78.8862 | 78.8862 | 78.8862 | 78.9474 | 78.9474 |
| $2^{-5}$ | 79.1922 | 79.1922 | 79.1922 | 79.1310 | 79.0698 | 79.1310 |
| $2^{-4.5}$ | 79.3146 | **79.3758** | 79.3146 | 79.2534 | 79.1922 | 79.1310 |
| $2^{-4}$ | 79.1310 | 79.1310 | 79.1310 | 79.2534 | 79.2534 | 79.1922 |
| $2^{-3.5}$ | 79.1310 | 79.1310 | 79.1310 | 79.1310 | 79.1310 | 79.1310 |
| $2^{-3}$ | 79.0698 | 79.0698 | 79.0698 | 79.0698 | 79.0698 | 79.1310 |
| $2^{-2.5}$ | 79.1310 | 79.1310 | 79.1310 | 79.1310 | 79.1922 | 79.1922 |

Table 4.2: Fine Search for $C$ and $\gamma$

### 4.2.2 Least Square

Least Square (LS) is a method of fitting data and searching the classification boundary. The most suitable parameter for classification is searched by this approach through minimizing the sum of squared residuals. As shown in Equations 4.4 and 4.5, $r$ is a residual between a labeled value $y$ and a predicted value produced by the model $f(x, \mathbf{w})$.

$$S = \sum_{i=1}^{n} r_i^2 \tag{4.4}$$

$$r_i = y_i - f(x_i, \mathbf{w}) \tag{4.5}$$

The LS method has been applied to many applications. It is able to fit parameters in both linear and non-linear models. For a linear regression model all the parameters are linear combined. Therefore, the parameter matrix **w** is estimated by Equation 4.6:

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y \tag{4.6}$$

in which **X** is a matrix derivative from $\partial f/\partial w_i$. For a non-linear model the parameters are estimated iteratively until they are convergent, which is a non-closed form approximation. [51].

To simplify the classification process in our application, we intuitively consider that the data set can be separated by a linear classification boundary. The method for a linear regression model is exploited to estimate parameters. Compared with the non-linear model, the linear one does not

require an initial value and does not have the non-convergence problem because the linear least square method is concave globally. [51]

Considering the simplicity and high efficiency of the LS classifier, we incorporate it as a comparison to the SVM classifier. The training set and the testing set for the LS classifier is the same as the data utilized by the SVM classifier. A classification boundary is searched by the LS classifier among the positive and negative training samples, while fragments labeled as "unwanted" are removed by this classifier.

### 4.2.3 Summary

In this section we present two classifiers, including the SVM and the LS approaches. Those classifiers are designed to classify segmented fragments into two clusters - the good segmentation as well as unwanted fragments. Through selecting fragments with shapes similar to the expected segmentation and rejecting fragments with unwanted shape, the average segmentation accuracy is expected to be enhanced. Both classifiers are based on shape information characterized by the probability density values of solidity and angle-curve difference. After being trained by those shape features, a classification boundary is determined by a classifier. For a new segmented image, fragments surviving the classification operation tend to have a higher segmentation quality, indicating that the average segmentation accuracy is improved by classifiers. The improvements resulting from different classifiers are evaluated by several classification criteria that are introduced in the following sections.

## 4.3 Classification Criteria

To appraise the performance of classifiers and shape features, several classification criteria are introduced in this section. These criteria focus on different aspects of classification. Since the main purpose of designing a classifier is to enhance the average segmentation accuracy, we measure the segmentation accuracy to evaluate the effectiveness of a classifier. The Object-level Consistency Error (OCE) score calculates the segmentation accuracy for each fragment and the Particle Size Distribution (PSD) curve evaluates the segmentation accuracy of the whole image set statistically. Before the PSD curve is applied as a criterion, the stability of PSD with respect to the fragment number is investigated. In addition to the segmentation accuracy the quantity of segmentation fragments also provides the total number of fragments that survive the classification operation. Definitions and the main principles of these criteria are presented below.

### 4.3.1 Object-Level Consistency Error Score

The Object-level Consistency Error score (OCE) is a metric evaluating the segmentation quality for images containing multiple objects [32]. This method measures the difference between segmented images and manually produced ground truth images. Different from other empirical discrepancy

approaches, the OCE score incorporates the existence, size, shape, and position of fragments as factors of the criterion. The definition of the OCE score is presented in Section 3.6. This criterion is chosen for the images in the oil sand application because its property of evaluating segmentation of multiple objects matches the attributes of oil sand images.

As stated in the introduction to this chapter, segmentation accuracy plays a more critical role than the quantity of fragments in the oil sand application. Although one segmented image may miss some fragments, this image is still preferable as long as it has a higher segmentation accuracy with respect to each fragment. This preference is implemented by incorporating a classifier based on shape information. Therefore, in order to evaluate the performance of proposed classifiers and selected shape features, we use only one-directional OCE score $E(s, g)$, shown as Equation 4.7, which does not penalize missing fragments

$$E(s,g) = \sum_{j=1}^{M} \left[ 1 - \sum_{i}^{n} \left( \frac{|A_j \bigcap B_i|}{|A_j \bigcup B_i|} \times W_{ij} \right) \right] \times W_j \tag{4.7}$$

$$W_{ij} = \frac{\delta\left(|A_j \bigcap B_i|\right)|B_i|}{\sum_{k=1}^{N} \delta\left(|A_j \bigcap B_k|\right)|B_k|} \tag{4.8}$$

$$W_j = \frac{A_j}{\sum_{l=1}^{M} |A_l|} \tag{4.9}$$

In Equation 4.7 $A_j$ and $B_i$ are fragments in segmented image and reference one, and $W_{ij}$ and $W_j$ are area weights defined in Equations 4.8 and 4.9, respectively. The OCE score $E(s, g)$ measures the segmentation accuracy of every fragment in the segmented images. In this case we consider the OCE criterion as a fragment level accuracy evaluator. The enhancement of the OCE score indicates the efficiency of a classifier: a higher OCE score implies a more efficient classifier.

### 4.3.2 Particle Size Distribution

To examine the segmentation accuracy statistically, we apply the Particle Size Distribution (PSD) curve that is a cumulative distribution of the segmented fragment size. In practical usage the diameter is exploited to measure the size of fragments other than the area. To simplify the calculation, all the fragments are assumed to be regular circles. In this case the diameter is derived as $2 \times \sqrt{A/\pi}$, where $A$ is the area of a fragment.

As shown in Figure 4.2, the horizontal axis is the length of diameter while the vertical axis is the percentage of fragments with diameter smaller than the equivalent value indicated on the horizontal axis. This cumulative distribution can be expressed as Equation 4.10, where $P$ is the percentage, $N$ is the total number of fragments, and $n(x < d)$ is the number of fragments with diameter smaller than $d$.

$$P(d) = \frac{n(x < d)}{N} \times 100\% \tag{4.10}$$

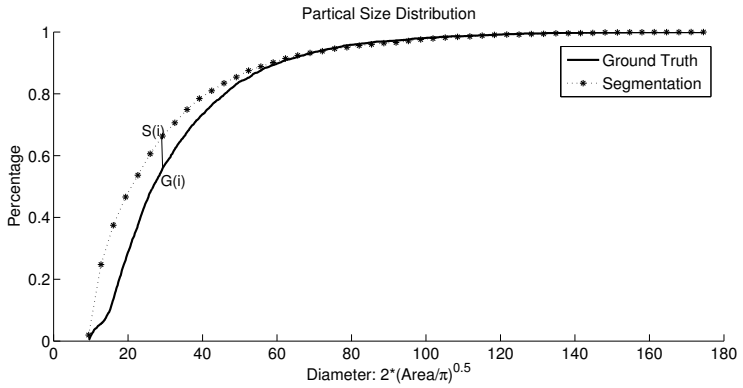


Figure 4.2: Particle Size Distribution

The PSD curve evaluates the segmentation results globally. Different from the OCE score, which measures segmentation quality fragment by fragment, this criterion combines all the segmentation results into one curve and supplies an overview on the size information of oil sand ores.

To measure the segmentation quality mathematically, a PSD curve produced by objects in ground truth images is adopted as a reference. The hypothesis is that the segmentation quality can be indicated by the similarity between PSD curve $S$ and $G$ which are derived from a group of segmented images as well as the ground truth images, respectively. The more similar between these two curves, the higher the quality of these segmented images will be. Therefore, we investigate several similarity metrics including Bhattacharyya distance [3], Hellinger distance [34], and Kullback-Leibler divergence (KL divergence) [17]. All of them are widely used to measure the similarity between two probability distributions.

The Bhattacharyya distance is based on the Bhattacharyya coefficient, and measures the overlap amount between two statistical samples. [49] The definition of the Bhattacharyya distance is given as follows,

$$D_B(s,g) = -\ln(\sum_{x\in X}(s(x)g(x))^{\frac{1}{2}}) \tag{4.11}$$

In Equation 4.11 the $s(x)$ and $g(x)$ are probability density values derived from the PSD curve $S$ and $G$, respectively. In this case a smaller $D_B(s,g)$ indicates a higher similarity between PSD curves generated by segmented images as well as ground truth images.

The Hellinger distance is also based on Bhattacharyya coefficient. However, different from the Bhattacharyya distance, Hellinger distance follows the triangle inequality. The definition of this metric can be written as Equation 4.12,

$$D_H(s,g) = \sqrt{1-\sum_{x\in X}(s(x)g(x))^{\frac{1}{2}}} \tag{4.12}$$

| | Solidity | Angle-curve Difference | Joint Probability |
|---|---|---|---|
| Non-classifier | 0.1522 | 0.1566 | 0.1484 |
| Least Square | 0.2455 | 0.2234 | 0.1709 |
| SVM | 0.2703 | 0.2991 | 0.2481 |

Table 4.3: Bhattacharyya Distance to Ground Truth PSD curve

| | Solidity | Angle-curve Difference | Joint Probability |
|---|---|---|---|
| Non-classifier | 0.3758 | 0.3808 | 0.3713 |
| Least Square | 0.4666 | 0.4475 | 0.3963 |
| SVM | 0.4889 | 0.5085 | 0.4688 |

Table 4.4: Hellinger Distance to Ground Truth PSD curve

in which the $\sum_{x \in X}(s(x)g(x))^{\frac{1}{2}}$ is the Bhattacharyya coefficient. Similar to the Bhattacharyya distance, a smaller Hellinger distance also indicates a higher similarity between two distributions.

In addition to the Bhattacharyya and Hellinger distances both based on the Bhattacharyya coefficient, the KL divergence describes the similarity of two distributions from the aspect of information theory. With respect to two probability distributions $G$ and $S$, if data are selected from distribution $S$ to represent $G$, the KL divergence will evaluates the required quantity of extra bits. Generally the distribution $G$ is the theoretical distribution, and the $S$ is a distribution of observation or an approximation of $G$. [50]

$$D_{KL}(g \parallel s) = \sum_{x \in X} g(x) \log \frac{g(x)}{s(x)} \tag{4.13}$$

In accordance to the definition of KL divergence, the more similar between two probability distributions, the smaller KL divergence value will be.

These metrics are applied to measure the similarity between ground truth PSD curve and PSD curves derived from segmentation results with respect to different shape features as well as classifiers. The results are listed in the following tables.

In these tables the distances derived from segmentation results without using any shape classifier are smaller than the distances derived from segmentation results passed the shape classifiers. However these similarity values are contradictory to the reality that a shape filter removes fragments with unwanted shapes and leads the segmentation PSD curve be closer to the ground truth one. The reason behind this phenomenon is that the $s(x)$ and $g(x)$ in Equations 4.11, 4.12, and 4.13 are the

| | Solidity | Angle-curve Difference | Joint Probability |
|---|---|---|---|
| Non-classifier | 0.2019 | 0.2159 | 0.2209 |
| Least Square | 0.2508 | 0.2578 | 0.2179 |
| SVM | 0.2678 | 0.2848 | 0.2618 |

Table 4.5: KL Divergence to Ground Truth PSD curve

probability density values. In this case, all of these listed similarity metrics only evaluate the similarity of two probability density distributions, but not the probability cumulative distributions that are utilized as the Particle Size Distribution. If we take a look at the probability density distribution of segmented fragment size, which is shown as the Figure. 4.3, we can find that the density values, even including the density values of ground truth, are too noisy for comparison. Therefore we abandon the metrics that are widely utilized to measure the similarity between probability distributions, and introduce another criterion that is highly related to the specific application of oil sand mining.

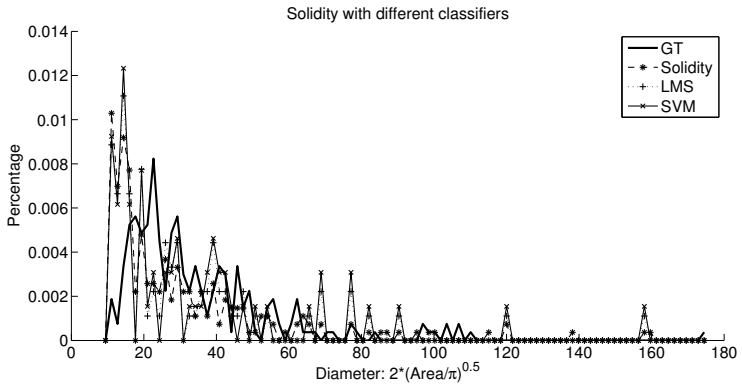


Figure 4.3: Particle Size Density Distribution

$$Diff_{s,g}(x) = |s(x) - g(x)| \qquad x \in [15, 30, 60] \tag{4.14}$$

As stated in the introduction part, the size information is a critical indicator to evaluate the performance of the screening process. The efficiency of a screen is measured by the ratio of amount of fragments passed the screen over the amount of fragments should have passed the screen [31]. Therefore, the accuracy of a PSD curve can be represented by the differences between a segmentation PSD curve and the ground truth one at the points that indicate the screen sizes. These differences can be derived by Equation. 4.14, in which $x$ is selected in accordance with the screen size. A smaller difference value infers a higher segmentation quality. On the other hand, the curve difference is also highly related to the specific application of the oil sand mining.

### 4.3.3 PSD Convergence Curve

With the number of fragments increasing, the PSD curve built by those fragments tends to remain the same. The PSD convergence curve measures the stability of a PSD and indicates the number of fragments required to make a PSD curve stable. In this thesis the number is known as the "convergence number." The convergence curve is formed with average difference that is the average value of difference at every point along the segmentation PSD curve as well as the ground truth PSD curve. This average difference can be derived by the Equation. 4.15. The process of vertical convergence

curve derivation is explained in the following.

$$AveDiff_{s,g} = \sum_{x \in X} |s(x) - g(x)| \tag{4.15}$$

As shown in Figure 4.4, each number in the horizontal axis indicates the number of fragments. The value on the vertical axis is the average difference between two PSD curves. One PSD curve is derived from an image data set with the fragment number represented by the current horizontal index, while the other is derived from the image data set with the fragment number represented by the previous index. For example, the value of a point of $250$ is an average difference between PSD curves generated with $250$ fragments and $225$ fragments, respectively. Therefore, if a PSD curve tends to be stable while the number of fragments is increasing, the PSD convergence curve will tend to be $0$.

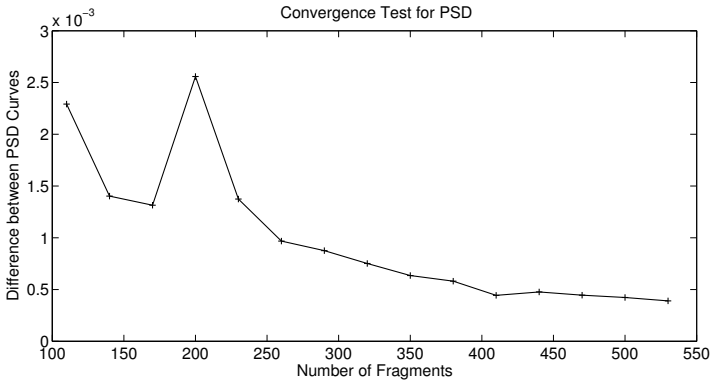


Figure 4.4: Convergence curve of PSD

Using the PSD convergence curve, a convergence number can be discovered to build a stable PSD. The convergence number affects several aspects, as follows. For the industrial application, this convergence number implies a required system running time to produce enough fragments for the size distribution analysis of oil sand. During the mining process the monitoring system continues to gather oil sand images and produce the segmented fragments. The smaller the convergence number is, the faster a PSD curve will become stable, and the less running time is required. Moreover, since in our thesis the PSD curve is used as a segmentation criterion, the performance of different classifiers and shape features can be compared with each other only when their PSD curves become stable. Therefore, the convergence number is searched for each classifier and shape feature. This step is implemented in further experimental investigations.

### 4.3.4 Segmented Fragments Quantity

Although in the oil sand image application the segmentation accuracy is quite important, the quantity of fragments is also a factor in measuring the performance of different classifiers and shape features.

An ideal classifier is assumed to produce fragments with the highest segmentation accuracy while maintaining the quantity of fragments as much as possible. In practical design if two classifiers have a similar performance in terms of accuracy enhancement, the one which keeps more fragments will be preferred. In order to calculate the quantity, the total number of fragments is accumulated in a group of segmented images.

### 4.3.5 Summary

In this section we present several criteria for evaluating classifiers in different aspects such as fragment and group levels segmentation accuracy as well as the quantity of segmented fragments. In addition, the number of fragments producing a stable PSD curve is also investigated. Accuracy improvement indicates the effectiveness of a classifier, while the segmented fragment quantity indicates the number of fragments surviving the classification operation. The definition and main usages of these criteria are introduced. Further discussion and experimental results will be presented in the next section.

## 4.4 Experimental Results

This section illustrates the results produced by different classifiers and shape features. These experimental results are evaluated by a variety of criteria: the segmentation accuracy reflecting the effectiveness of classifiers is measured by the OCE score and PSD curve; the number of fragments surviving the classification operation is represented by the segmented fragment quantity; and the number of fragments resulting in a stable PSD curve is investigated.

Experimental results are compared in two directions. To evaluate different shape features, classification results generated by one classifier but a variety of shape features are compared with each other to find out which feature is more suitable for the specific classifier than the others. To check the performance of classifiers, the classification results produced by different classifiers using the same shape feature are compared, and the classifier that best matches the selected shape feature is looked for. In the experiments the test data is the same as that used in Section 3.6, which includes $60$ images. A conclusion is drawn with respect to every criterion.

#### OCE Score

The OCE score measures the accuracy of segmentation results passing by the classification operation. Since the purpose of including a classifier is to remove unwanted fragments and enhance the average segmentation accuracy, a higher OCE score often implies a superior classifier in terms of effectiveness. The OCE score derived from our proposed local threshold algorithm without using any classifier is incorporated as a reference in order to show the improvement made by classifiers.

As stated above, the segmentation accuracy plays a critical role in oil sand image segmentation. In this specific application domain we adopt only one-directional OCE score $E(s, g)$ that measures

| | $E(s,g)$ | | $E(s,g)$ | | $E(s,g)$ |
|---|---|---|---|---|---|
| Solidity | 0.5991 | Angle-curve Difference | 0.5498 | Joint Probability | 0.5762 |
| Solidity LS | 0.7444 | Angle-curve Difference LS | 0.6966 | Joint Probability LS | 0.6939 |
| Solidity SVM | **0.7701** | Angle -curve Difference SVM | 0.7482 | Joint Probability SVM | 0.7588 |

Table 4.6: Segmentation Accuracy of Different Classifiers measured by OCE score

the segmentation accuracy and does not penalize missing fragments in a segmented image. This criterion evaluates the segmentation accuracy in fragment level and weights the score in accordance with the fragment area. With $60$ test images an average $E(s,g)$ score is calculated and compared to every combination of classifier and shape feature.

Table 4.6 shows the OCE scores of different classifiers and shape features. The data in each row of this table is produced by the same classifier; for example, the SVM or LS. The first row is the OCE score of local thresholding results without using a classifier. This row is incorporated as a reference. Each column in this table shares the same shape feature such as solidity, angle-curve difference, or joint probability. In Table 4.6 the $E(s,g)$ score is enhanced with the aid of classifiers based on the shape information.

In addition to the OCE score, the segmentation results filtered by classifiers with different features are presented in Figures 4.5 and 4.6. In a group of segmented images the first row is the original gray-scale image and ground truth image. Then, from the second to the fourth row the images in each column are produced by the same classifier while the images in each row are generated by the same shape feature. In every line, the order of classifiers is non-classifier, LS classifier, and SVM classifier; and in every column the order of shape features is solidity, angle-curve difference, and joint probability. From these segmentation results we can see that the fragments in images produced with classifiers tend to have a higher average segmentation accuracy than the results produced without classifier. In accordance with previous discussion, images containing fragments which are segmented more accurately are preferred in the oil sand mining application. Therefore, both the OCE score and segmentation images prove our hypothesis that incorporating classifiers allows us to remove unwanted fragments and improve the average segmentation accuracy.

In further investigation, from both the OCE score listed in Table 4.6 and segmented fragments presented in Figures 4.5 and 4.6, solidity is superior to the other shape features in terms of fragment level segmentation accuracy, while for presented shape features, the SVM approach performs better than the other classifiers. The OCE score reaches its maximum value when we choose solidity as a shape feature and the SVM as a classifier. Therefore, if we want to ensure each fragment has its most accurate segmentation, especially for a fragment with a bigger size, the SVM classifier based on solidity will be selected.

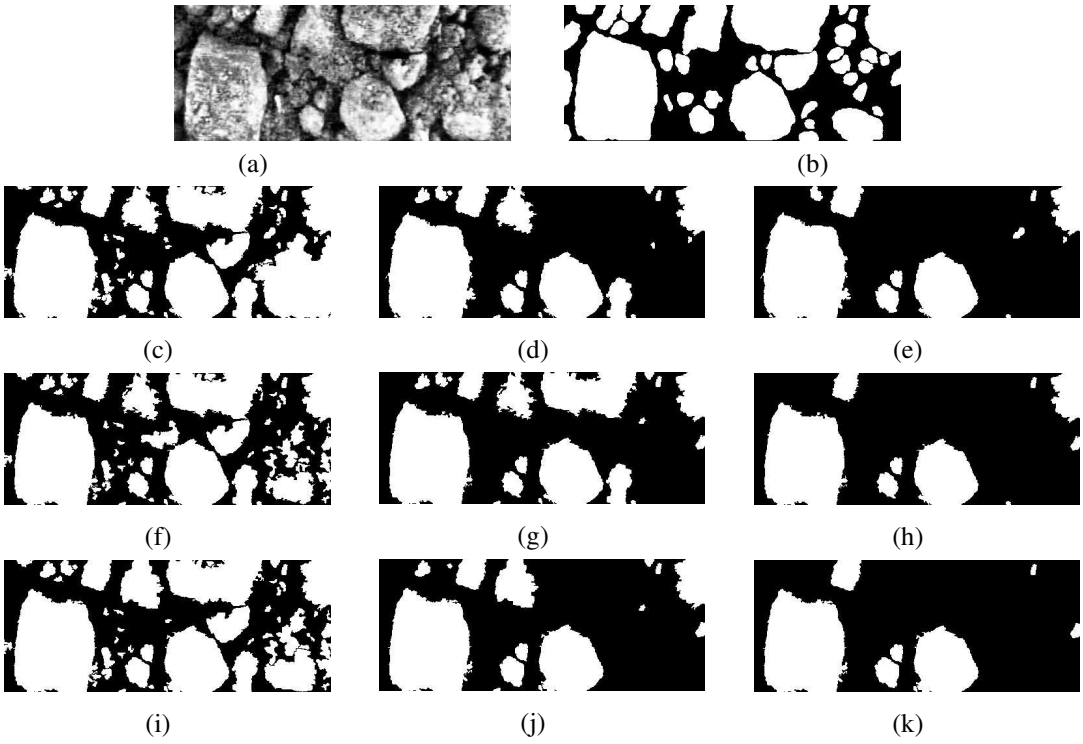


Figure 4.5: Segmentation Result 1 with respect to different classifiers and shape features: (a) Original image, (b) Ground truth image, (c) Solidity, (d) Solidity with LS classifier, (e) Solidity with SVM classifier, (f) Angle-curve Difference, (g) Angle-curve Difference with LS classifier, (h) Angle-curve Difference with SVM classifier, (i) Joint Probability, (j) Joint Probability with LS classifier, (k) Joint Probability with SVM classifier

**PSD Convergence Curve**

The PSD convergence curve is utilized to search for a fragment number that makes a PSD curve stable. To determine whether a PSD curve is stable or not, a reference threshold is introduced. If a PSD curve tends to be almost the same while the fragment number is increasing, the value in a PSD convergence curve will be small. Therefore, a PSD convergence curve will be considered as convergent when its value is continually smaller than the introduced threshold. The minimum horizontal index of a point that belongs to the convergent part of a PSD convergence curve is known as the convergence number and indicates the least number of fragments required to make a PSD convergence curve stable. For example, in Figure 4.7 the PSD convergence curve labeled "Solidity" is convergent from horizontal index 260, which means that the PSD curve tends to be stable when there are at least 260 fragments incorporated.

As stated before, all convergence numbers for different classifiers and shape features need to be determined. Similar to the OCE score the PSD convergence curve looks for a convergence number of classifiers and shape features separately. The experimental results are presented in detail in the following.

The PSD convergence curves in Figure 4.7 are derived from image results produced by different

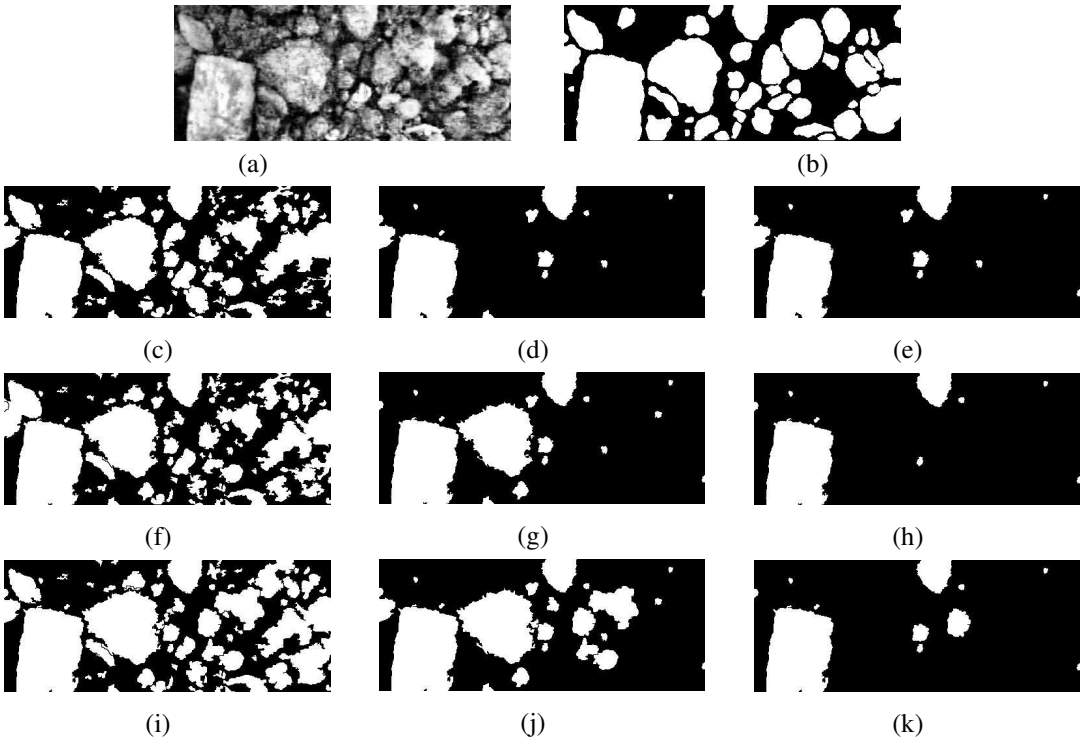


Figure 4.6: Segmentation Result 2 with respect to different classifiers and shape features: (a) Original image, (b) Ground truth image, (c) Solidity, (d) Solidity with LS classifier, (e) Solidity with SVM classifier, (f) Angle-curve Difference, (g) Angle-curve Difference with LS classifier, (h) Angle-curve Difference with SVM classifier, (i) Joint Probability, (j) Joint Probability with LS classifier, (k) Joint Probability with SVM classifier

| | Solidity | Least Square | SVM |
|---|---|---|---|
| Convergent No. | 260 | 160 | 300 |

Table 4.7: The least number of fragments required for convergence (Solidity)

classifiers adopting solidity as a shape feature. The curve labeled Solidity indicates segmentation results generated by local thresholding without using a classifier. This curve is adopted as a reference to show the effects of classifiers. In Table 4.7 the convergence number with respect to SVM and LS classifiers are $160$ and $300$ respectively. These convergence numbers also match the convergence curves, as shown in Figure 4.7. The result indicates that based on solidity the LS classifier needs fewer fragments than the SVM classifier to obtain a stable PSD curve. However, when PSD curves of LS and SVM classifiers tend to be flat, such as when the number of fragments is more than $350$, these two curves are intertwined with each other which implies that those two classifiers have quite similar performance with respect to PSD stability.

| | Angle-curve Difference | Least Square | SVM |
|---|---|---|---|
| Convergent No. | 240 | 200 | 240 |

Table 4.8: The least number of fragments required for convergence (Angle-curve Difference)

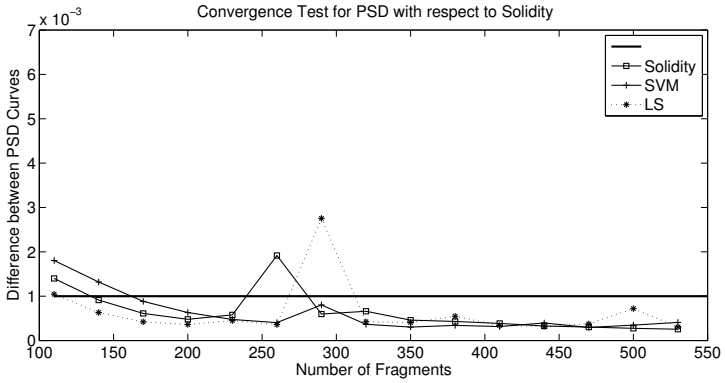


Figure 4.7: PSD convergence curve with respect to Solidity

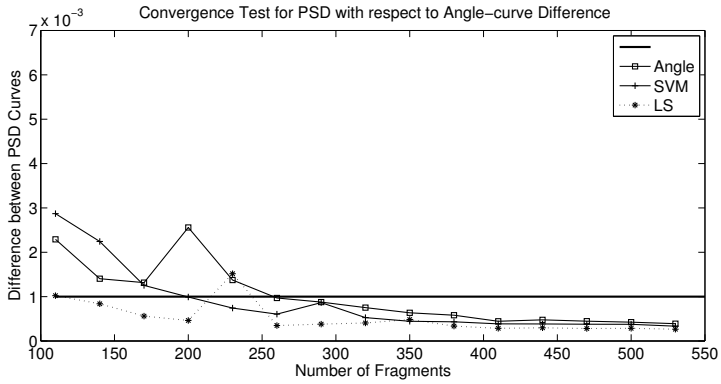


Figure 4.8: PSD convergence curve with respect to Angle-curve Difference

In Figure 4.8 PSD convergence curves are derived from image results produced by classifiers using the angle-curve difference alone. The curve labeled "Angle" represents segmentation results generated by local thresholding without using a classifier and is also adopted as a reference. In Figure 4.8 convergence curves of SVM and LS classifiers converge after $200$ and $240$, respectively. This result matches the convergence numbers listed in Table 4.8. In this case the SVM and LS classifiers need almost the same number of fragments to generate a stable PSD curve. The PSD convergence curves derived from the classification results of LS and SVM classifiers are also almost merged with each other in the flat region where the number of fragments is more than $350$.

| | Joint Prob | Least Square | SVM |
|---|---|---|---|
| Convergent No. | 260 | 540 | 550 |

Table 4.9: The least number of fragments required for convergence (Joint Probability)

In Figure 4.9 PSD convergence curves are derived from fragments surviving the classification operation based on the joint probability of shape features such as solidity and angle-curve difference.

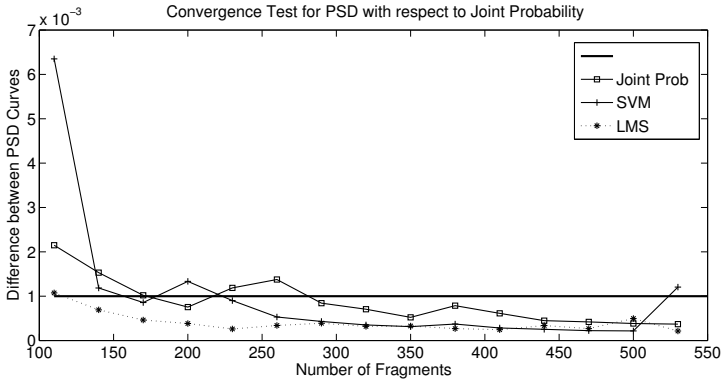


Figure 4.9: PSD convergence curve with respect to Joint Probability

Similar to previous cases, the curve labeled "Joint Prb" represents segmentation results generated by local thresholding without a classifier and is used as a reference. As shown in Table 4.9, the convergence numbers of LS and SVM classifiers are quite close. In Figure 4.9 the convergence curves of LS and SVM classifiers are also intertwined with each other, especially when those curves are almost converged. This also illustrates that when these two classifiers both adopt joint probability as a shape feature, they have almost the same performance in terms of the stability of the PSD curve.

As demonstrated by the experiments shown above, when a shape feature is fixed, the number of fragments required by the LS classifier to produce a stable PSD curve is almost as large as the fragment number needed by the SVM classifier. In this case, between the LS and SVM classifiers, whichever is chosen, the same fragment number can be exploited to generate a stable PSD curve for further study. On the other hand, the PSD curves produced by these two classifiers have almost the same stability because the PSD convergence curves of LS and SVM classifiers are intertwined with each other when they tend to be convergent.

In addition to comparing the PSD convergence curves of different classifiers, the convergence curves derived from a variety of shape features are also compared with each other when a classifier is fixed. Further discussions and experimental results are presented below.

| | Solidity | Angle-curve Difference | Joint Probability |
|---|---|---|---|
| Convergent No. | 300 | 240 | 550 |

Table 4.10: The least number of fragments required for convergence (SVM Classifier)

Figure 4.10 presents the convergence curves derived from image results with respect to different shape features when the SVM is the only classifier used in the classification step. From Table 4.10 we can see that joint probability has a larger convergent number than other shape features such as solidity and angle-curve difference. However, this is resulting only from a slight uprising in the last part of the PSD convergence curve of joint probability. In the whole range of PSD convergence

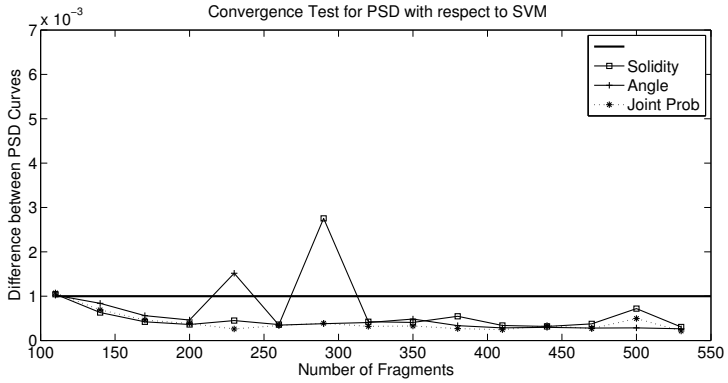


Figure 4.10: PSD convergence curve with respect to SVM Classifier

curves, shape features such as solidity, angle-curve difference, and joint probability tend to be almost convergent which implies that those shape features have similar performance in terms of the stability of PSD curves.

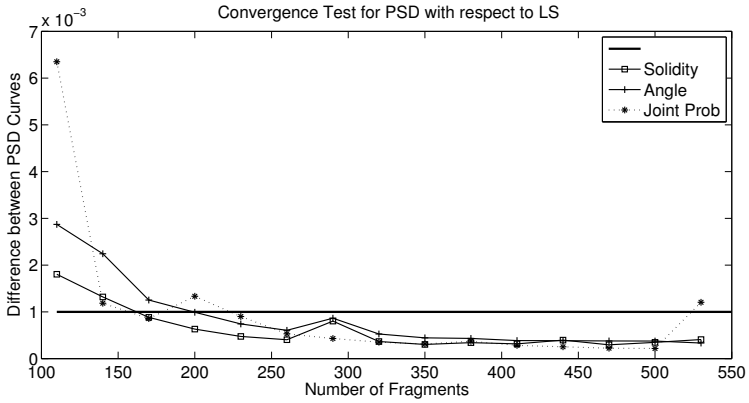


Figure 4.11: PSD convergence curve of Least Square Classifier

| | Solidity | Angle-curve Difference | Joint Probability |
|---|---|---|---|
| Convergent No. | 160 | 200 | 540 |

Table 4.11: The least number of fragments required for convergence (LS Classifier)

The Figure 4.11 shows the convergence curves with respect to different shape features when the LS classifier is chosen for the classification process. As in results generated by the SVM classifier, the solidity and angle-curve difference have similar convergence numbers, and both are significantly smaller than the convergence number of joint probability. However, this is also resulting from a slight uprising in the last part of the joint probability PSD convergence curve. The PSD convergence curves of solidity, angle-curve difference, and joint probability are intertwined with each other for

the most part. Therefore, we still consider that those shape features lead to almost the same PSD curve stability.

Overall, the convergence numbers are obtained for different classifiers and shape features. Based on these convergence numbers, the LS and SVM classifiers require almost the same number of fragments to make a PSD curve stable. This implies that the convergent number tends to be unchanged to a variety of classifiers. In this case a constant number can be chosen as the convergence number for both classifiers. With respect to different shape features, although joint probability needs a larger convergence number, the PSD convergence curves are merged with each other in most of the range. This demonstrates that those shape features have almost the same property in terms of their PSD curve stability. Therefore, a constant number is also selected for a variety of shape features, which simplifies the calculation process.

**Particle Size Distribution**

The Particle Size Distribution (PSD) curve represents the cumulative distribution of the sizes of segmented fragments. Different from the OCE score which operates at fragment level, this criterion evaluates the segmentation accuracy globally and statistically. In this criterion a PSD curve derived from ground truth images is introduced as a reference. The hypothesis is that if a PSD curve is closer to the reference curve, the group of images producing this PSD curve will tend to have a higher segmentation quality. In terms of the application related to the screen efficiency measurement, the differences between segmentation PSD and ground truth PSD are calculated at the points indicating the screen sizes. In the oil sand mining process, the sizes of screens are $5.5$ cm, $11$ cm, and $22$ cm, and the transform ratio is $0.37987$ cm/pixel, which changes the size of fragments in images into the size of real oil sand ores. Therefore, the difference between two PSD curves are measured at the points $15$, $30$, and $60$.

In accordance with the experimental results of the PSD convergence curve, all the PSD curves are generated with a number of fragments greater than the convergence number to ensure the PSD curve is stable when it is used as a criterion. Similar to previous experiments, results are compared with respect to classifiers and shape features. The investigations related to different classifiers combined with a variety of shape features are presented in details as follows:

| | $\text{Diff}_{s,g}(15)$ | $\text{Diff}_{s,g}(30)$ | $\text{Diff}_{s,g}(60)$ |
|---|---|---|---|
| Solidity | 0.2504 | 0.1134 | 0.0214 |
| Least Square | 0.2505 | 0.1108 | 0.0173 |
| SVM | 0.2215 | 0.0835 | 0.0101 |

Table 4.12: Difference between PSD Curves and Ground Truth Reference (Solidity)

In Figure 4.12 the PSD curves with respect to a variety of classifiers are presented. All these classifiers are based on solidity. The PSD curve labeled "Sol" is produced by local thresholding without using any classifier. From Figure 4.12 we can see that the PSD curve generated by the SVM

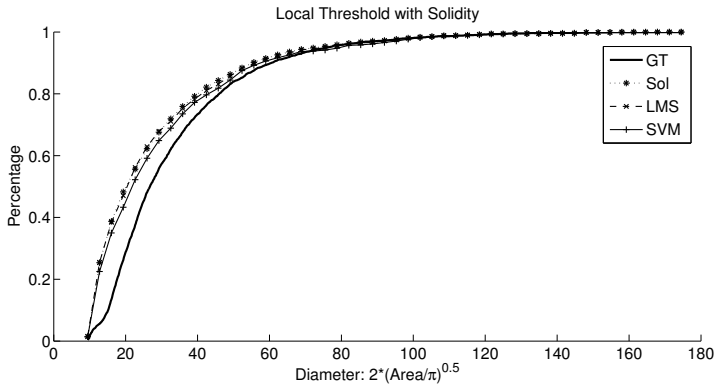


Figure 4.12: Particle Size Distribution with respect to Solidity

classifier is closer to the reference curve than the other two curves. This result also matches the differences listed in Table 4.12, where the SVM classifier has the smallest $\text{Diff}_{s,g}$ values. Therefore, when all the classifiers adopt solidity as a shape feature, the SVM classifier is superior to other approaches in terms of enhancing segmentation quality.

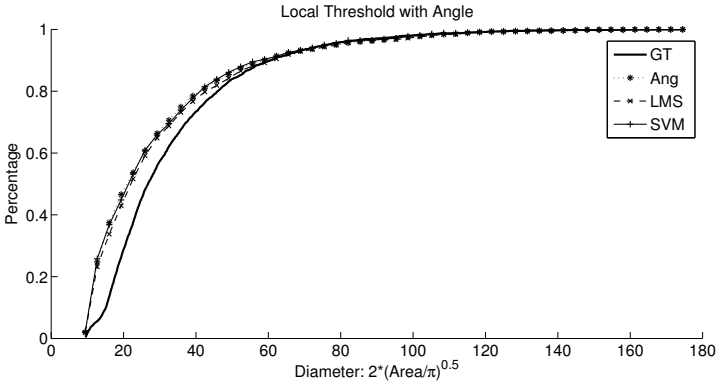


Figure 4.13: Particle Size Distribution with respect to Angle-curve Difference

| | $\text{Diff}_{s,g}(15)$ | $\text{Diff}_{s,g}(30)$ | $\text{Diff}_{s,g}(60)$ |
|---|---|---|---|
| Angle-curve Difference | 0.2404 | 0.1028 | 0.0074 |
| Least Square | 0.2095 | 0.0853 | 0.0035 |
| SVM | 0.2475 | 0.0948 | 0.0074 |

Table 4.13: Difference between PSD Curves and Ground Truth Reference (Angle-curve Difference)

Experimental results with respect to angle-curve difference are presented in Figure 4.13 and Table 4.13. Similar to the previous case the PSD curve labeled "Angle" is produced by local thresholding without using any classifier. The PSD curves in Figure 4.13 are close to each other. However, in Table 4.13, the differences of the LS classifier is smaller than the results produced by the other ap-

proaches, which indicates that the LS classifier is slightly better between two classifiers with respect to the angle-curve difference.

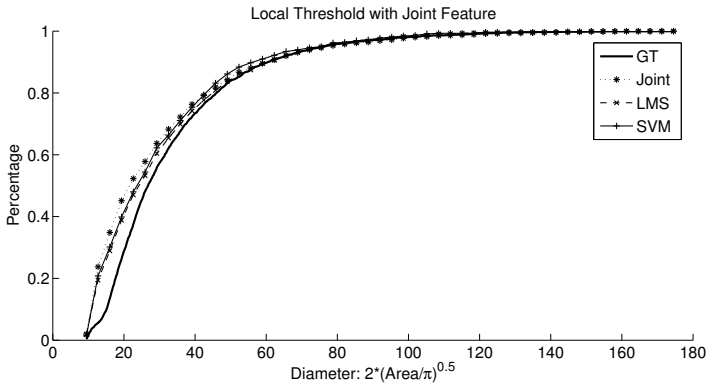


Figure 4.14: Particle Size Distribution with respect to Joint Features

| | $\text{Diff}_{s,g}(15)$ | $\text{Diff}_{s,g}(30)$ | $\text{Diff}_{s,g}(60)$ |
|---|---|---|---|
| Joint Probability | 0.2168 | 0.0751 | 0.0036 |
| Least Square | 0.1638 | 0.0445 | 0.0042 |
| SVM | 0.1810 | 0.0561 | 0.0177 |

Table 4.14: Difference between PSD Curves and Ground Truth Reference (Joint Probability)

In Figure 4.14 and Table 4.14 experimental results produced by the joint probability of both solidity and angle-curve difference are displayed. The PSD curve labeled "Joint" also indicates the segmentation results generated by local thresholding without any classifier. In Figure 4.14 the PSD curve produced by the LS classifier is the one closest to the reference curve, and the LS classifier has the smallest differences in Table 4.14. These experimental results illustrate that the LS classifier adopting the joint probability of two shape features is superior to the SVM classifier in terms of segmentation quality enhancement.

As stated above we compare the performance of different classifiers based on the same shape feature, and determine the most suitable classifier for each specific shape feature. When solidity is selected, fragments surviving the SVM classifier tend to have a higher segmentation quality. However, when the angle-curve difference or the joint probability of both two shape features is adopted, the result of the LS classifier is better than the others. Generally speaking, with the aid of a classifier, most PSD curves are closer to the ground truth reference curve than those PSD curves derived from images produced without a classifier. This also indicates that the introduced classifier can reject unwanted fragments and enhance the segmentation accuracy. On the other hand, since the PSD curves are compared with the ground truth reference curve in the whole range, it could validate our assumption that the investigated classifiers do not have a bias on size information.

In the following sections we evaluate the segmentation quality of different shape features with

respect to the same classifier and try to determine the best feature for each type of classifier.

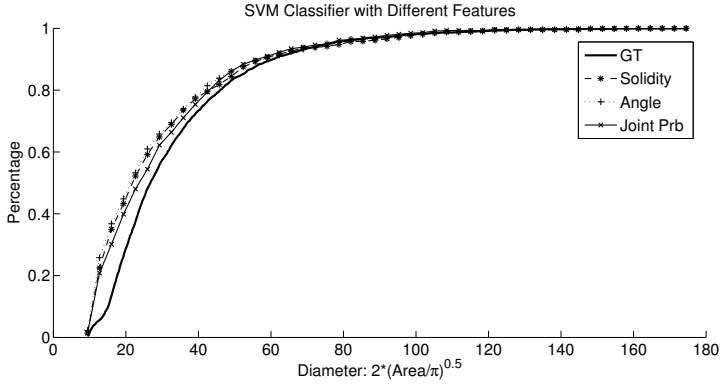


Figure 4.15: Particle Size Distribution based on SVM Classifier

| | $\text{Diff}_{s,g}(15)$ | $\text{Diff}_{s,g}(30)$ | $\text{Diff}_{s,g}(60)$ |
|---|---|---|---|
| Solidity | 0.2215 | 0.0835 | 0.0101 |
| Angle-curve Difference | 0.2475 | 0.0948 | 0.0074 |
| Joint Probability | 0.1810 | 0.0561 | 0.0177 |

Table 4.15: Difference between PSD Curves and Ground Truth Reference (SVM Classifier)

When adopted by the proposed SVM classifier, joint probability of solidity and angle-curve difference results in the highest segmentation quality. As shown in Figure 4.15, although PSD curves produced by different shape features are close to each other, the PSD curve derived from results of joint probability is the one nearest to the reference curve, and the differences listed in Table 4.15 also indicate that joint probability results in has the smallest $\text{Diff}_{s,g}$ value. Both these experimental results illustrate that joint probability is the most suitable feature characterizing shape information for the SVM classifier.

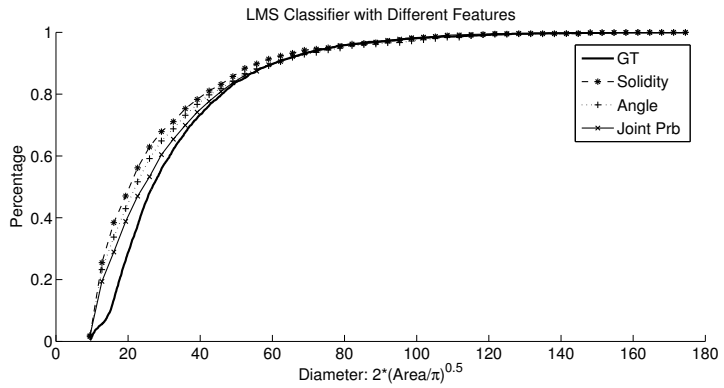


Figure 4.16: Particle Size Distribution based on Least Square Classifier

| | $\text{Diff}_{s,g}(15)$ | $\text{Diff}_{s,g}(30)$ | $\text{Diff}_{s,g}(60)$ |
|---|---|---|---|
| Solidity | 0.2505 | 0.1108 | 0.0173 |
| Angle-curve Difference | 0.2095 | 0.0853 | 0.0035 |
| Joint Probability | 0.2505 | 0.1108 | 0.0173 |

Table 4.16: Difference between PSD Curves and Ground Truth Reference (LS Classifier)

For the LS classifier the difference between different shape features is not very significant. The Figure 4.16 illustrates that the PSD curve produced by joint probability is closer to the reference curve than are the other two. On the other hand, the Table 4.16 shows that the gap distances of angle-curve difference measured by the $\text{Diff}_{s,g}$ at selected size points are smaller than results of LS classifiers based on solidity or joint probability. Therefore, it is difficult to tell which shape feature is superior to the others.

From experiments evaluating classifiers as well as shape features, a PSD curve with the smallest differences at points indicating the screen size is produced by the LS classifier with the angle-curve difference. However, in general, the SVM classifier is superior to the LS classifier in terms of segmentation quality enhancement, while the joint probability of two shape features has better performance than each single shape feature of solidity or angle-curve difference. Therefore, in the oil sand application the SVM classifier adopting joint probability would be used in the classification operation to generate the best particle size distribution.

**Segmented Fragment Quantity**

In addition to the OCE score and PSD curve, the quantity of fragments in the segmentation result is also utilized as a criterion. As stated in Section 4.3.4, when two classifiers have similar segmentation accuracy with respect to the segmentation accuracy, a classifier that keeps more fragments will be selected. The reason behind this choice is that we assume an ideal classifier would have the highest precision as well the highest recall. However, in the oil sand application the segmentation accuracy plays a much more important role than the fragment quantity. In this case the fragment quantity is selected under constraints with respect to the segmentation accuracy. Therefore, a one-directional OCE score $E(s, g)$ is included as an accuracy constraint for the fragment quantity comparison.

| | Solidity | Solidity LS | Solidity SVM |
|---|---|---|---|
| E(s,g) | 0.5991 | 0.7444 | 0.7701 |
| Quantity | 2721 | 903 | 649 |
| | AC Difference | AC Difference LS | AC Difference SVM |
| E(s,g) | 0.5498 | 0.6966 | 0.7482 |
| Quantity | 2584 | 999 | 620 |
| | Joint Prob | Joint Prob LS | Joint Prob SVM |
| E(s,g) | 0.5762 | 0.6939 | 0.7588 |
| Quantity | 2616 | 1200 | 577 |

Table 4.17: Segmentation Accuracy of Different Classifiers measured by OCE score

In Table 4.17 the fragment quantity produced by different classifiers and shape features is presented. The data in each column are generated by the classifier in that column. Utilized as a reference, the data in the first column is derived from segmentation results without applying a classifier. In general, the LS classifier keeps more fragments than the SVM classifier, while the latter has a higher OCE scores. Since the difference between two classifiers with respect to their OCE score cannot be ignored, the SVM classifier is still preferred for the classification operation.

**Summary**

In the Experimental Results section we compared the performance of classifiers and shape features with different kinds of criteria. To measure the accuracy improvement produced by classifiers, two criteria are introduced: the Object-level Consistency Error (OCE) score focuses on fragment level accuracy while the Particle Size Distribution (PSD) curve characterizes the segmentation quality globally and statistically. The Support Vector Machine (SVM) classifier adopting solidity obtains the highest OCE score while the Least Square (LS) classifier utilizing the joint probability of solidity and angle-curve difference generates a PSD curve that is closer to the ground truth reference curve than any other combination of classifiers and shape feature. Moreover, in general, joint probability is superior to the other shape features in terms of segmentation quality measured by the PSD curve, and the SVM classifier results in a higher average segmentation accuracy than the LS classifier.

The number of fragments surviving classification operation is also investigated by the criterion of segmented fragment quantity. The LS classifier tends to have more fragments in classified results while the SVM classifier often has a higher segmentation quality. In addition to these criteria, the number of fragments that makes a PSD curve stable, which is called the convergence number, is searched for different classifiers as well as for various shape features. Then we adopt more fragments than the convergence number to produce a stable PSD curve. Therefore, the PSD curves used as a criterion will not be changed by the variation of fragment numbers. In accordance with our experiments we find that the convergence number is almost the same to the LS and SVM classifiers. A constant number could be selected as the convergence number for these classifiers. Finally, from the results above we can see that both LS and SVM classifiers are able to enhance the accuracy of segmentation. However, the SVM classifier tends to have an even higher average segmentation accuracy while the LS classifier often keeps more fragments as the result of the classification operation. Therefore, in the oil sand application domain if the segmentation accuracy is the only factor considered, the SVM classifier will be the best choice.

## 4.5 Summary

In this chapter experimental problems in the oil sand application are investigated. To enhance the segmentation accuracy of proposed local threshold algorithm, two classifiers - the Support Vector Machine and Least Square - are introduced. These classifiers are based on the same shape features

adopted by our local threshold algorithm, which includes the probability density values of solidity, angle-curve difference, and the joint probability of these two shape features. To measure the performance of the chosen classifiers and shape features, several criteria are applied to evaluate different aspects such as fragment level and group level segmentation accuracy which indicate the effectiveness of a classifier, and the quantity of fragments surviving the classification operation. In addition, the number of fragments resulting in a stable PSD curve is also looked for. Based on the results of these criteria, the SVM classifier tends to be superior to the LS classifier in terms of segmentation accuracy enhancement, while the LS classifier often has a better performance on fragment quantity.

# Chapter 5

# Conclusion and Future Work

## 5.1 Local Threshold based on Shape Information

A local threshold algorithm incorporating shape information is presented in this thesis. The shape information characterizes geometric features of objects, then a threshold is selected adaptively and locally to maximize shape feature probabilities of distributions learned from manually segmented ground truth images. The proposed algorithm is utilized in an application of oil sand image segmentation. To further enhance the segmentation accuracy for this application, a supervised classifier, which is also based on shape information, is introduced to remove segmented fragments with unwanted shapes. Different combinations of classifiers and shape features are investigated. Criteria focusing on a variety of aspects are adopted to evaluate the performance and properties of these classifiers.

The local threshold algorithm based on shape information obtains a higher segmentation quality than algorithms adopting only intensity information. In the experiments several local thresholding algorithms based on intensity information are included as comparison. Both the proposed method and the competing ones are applied to oil sand and synthetic images. In terms of segmentation quality, the experimental results show that our proposed local threshold algorithm based on shape information is superior to local threshold algorithms based only on intensity information. Therefore, in accordance with the experiments we conclude that the shape information is able to enhance the segmentation quality for digital images containing similar objects.

Two classifiers - the Support Vector Machine (SVM) and the Least Square (LS) - are investigated. Several criteria are exploited to evaluate classifier candidates in a variety of aspects. The Object-level Consistency Error score measures the average segmentation accuracy changes at fragment level while the Particle Size Distribution curve shows the average segmentation accuracy improvement at group level. In addition to segmentation accuracy measurement, the quantity of fragments surviving the classification operation is also calculated. Using those criteria, the classifier based on the SVM algorithm is superior to the LS classifier in terms of the average segmentation accuracy enhancement.

A number of contributions are made in this work. The shape information is extracted from target

objects. For local thresholding, this information is introduced to threshold selection. The proposed local threshold algorithm with shape information obtains superior segmentation quality to other local threshold approaches based only on intensity information. For the oil sand application, this information is incorporated by classification operation that removes segmented fragments with unwanted shapes in order to enhance the average segmentation accuracy. Therefore, our investigation on the process of utilizing shape information to improve the segmentation quality benefits the general local threshold methods as well as a specific application domain.

## 5.2 Future Work

In the future, the work presented in this thesis can be developed by including different shape features and extending the investigation to more segmentation algorithms as well as applying them to additional application domains.

In the current algorithm, the solidity and angle-curve variance are used to characterize the geometric features of objects in oil sand images. Solidity measures the convexity of a segmented shape while the angle-curve variance focuses on the smoothness of fragment boundaries. In fact, in addition to these two shape features, there are other features that could be exploited to represent the shape of oil sand ores. Further work would investigate additional shape features and incorporate them into the proposed algorithm.

To further enhance the segmentation quality of oil sand images, another segmentation algorithm could be investigated and exploited. Currently, images are segmented by the local threshold algorithm. However, the local thresholding method also has some limitations. In some cases it is difficult to calculate a threshold which will produce good segmentation because the intensity in an object is distributed irregularly. Therefore, in future investigations, shape information could be incorporated into a stronger segmentation algorithm to generate a segmentation result with higher quality.

The future work proposed above focuses on the application domain of oil sand images. This algorithm could also be utilized in other application areas such as segmentation for biomedical cell images or other images containing rounded objects. However, for a new application area, in accordance with the process of the proposed algorithm, new shape feature distributions need to be generated based on a group of ground truth images related to the new application domain.

Therefore, the future work outlined would further improve the segmentation quality of oil sand images and also extend this algorithm to some other application domains.

# Bibliography


[1] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22:830–836, 2006.

[2] J. Bernsen. Dynamic thresholding of gray level images. *ICPR'86: Proc. Intl. Conf. Patt. Recog.*, pages 1251–1255, 1986.

[3] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.

[4] C. M. Bishops. *Pattern Recognition and Machine Learning*. Springer, New York, USA, 2006.

[5] I. Blayvas, A. Bruckstein, and R. Kimmel. Efficient computation of adaptive threshold surfaces for image binarization. *Pattern Recognition*, 39:89–101, 2006.

[6] F. H. Y. Chan, F. K. Lam, and H. Zhu. Adaptive thresholding by variational method. *IEEE Trans. Image Process.*, IP-7:469–473, 1991.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[8] Q. Chen, Z. M. Zhou, M. Tang, P. A. Heng, and D. S. Xia. Shape statistics variational approach for outer contour segmentation of left ventricle mr images. *IEEE Trans. on Information Technology in Biomedicine*, 10(3):588–597, 2006.

[9] C. K. Chow and T. Kaneko. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research*, 5:388–410, 1972.

[10] F. Dornaika and H. Zhang. Granulometry with mathematical morphology and motion. *Proc. IAPR Workshop on Machine Vision Application*, pages 51–54, Nov. 2000.

[11] L. Eikvil, T. Taxt, and K. Moen. A fast adaptive method for binarization of document images. *Proc. First Intel. Conf. Document Analysis and Recognition*, pages 435–443, 1991.

[12] E. Giuliano, O. Paitra, and L. Stringer. Electronic character reading system. *U.S. Patent*, (4,047,15), Sep 1977.

[13] M. Kamel and A. Zhao. Extraction of binary characters/graphics images from grayscale document images. *Graph. Models Image Process*, 55(3):203–217, 1993.

[14] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation*, 15:1667–1689, 2003.

[15] D. Y. Kim and J. W. Park. Connectivity-based local adaptive thresholding for carotid artery segmentation using mra images. *Image and Vision Computing*, 23:1277–1287, 2005.

[16] V. V. Kindratenko. On using functions to describe the shape. *Journal of Mathematical Imaging and Vision*, 18:225–245, 2003.

[17] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[18] I. Levner and H. Zhang. Classification driven watershed segmentation. *IEEE Trans. Image Process.*, 16(5):1437–1445, 2007.

[19] Z. Li and K. Najarian. Biomedical image segmentation based on shape stability. *IEEE Intl. Conf. Image Process.*, VI:281–284, Sep. 2007.

[20] H.-T. Lin and C.-J. Lin. A study on sigmoid kernels for svm and the training of non psd kernels by smo-type methods. Technical report, Department of Computer Science, National Taiwan University, 2003.

[21] F. Liu, Y. Luo, X. Song, and D. Hu. Active surface model-based adaptive thresholding algorithm by repulsive external force. *Journal of Electronic Imaging*, 12:299–306, April 2003.

[22] Y. Liu and S. N. Srihari. Document image binarization based on texture features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:540–544, 1997.

[23] V. M. Lo. Heuristic algorithms for task assignment in distributed systems. *IEEE Trans. Computers*, pages 1384–1397, 1988.

[24] The MathWorks. Matlab. http://www.mathworks.com/products/matlab/, 2007.

[25] D. Mery and F. Pedreschi. Segmentation of color food images using a robust algorithm. *Journal of Food Engineering*, 66:353–360, 2005.

[26] W. Niblack. *An introduction to digital image processing*. Prentice Hall, Englewood Cliffd, NJ, 1986.

[27] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Tans. Systems Man. Cyber*, 9:62–66, 1979.

[28] P. W. Palumbo, P. Swaminathan, and S. N. Srihari. Document image binarization: Evaluation of algorithms. *Proc. SPIE*, 697:278–286, 1986.

[29] J. R. Parker. Grey levelthresholding in badly illuminated images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(8):813–819, 1991.

[30] A. Pikaz and A. Averbuch. Digital image thresholding, based on topological stable-state. *Pattern Recognition*, 29:829–843, 1996.

[31] M. Polak and R. Matti. Porting dirt tv-screen efficiency to syncrude research and osa findfrags. Technical report, Center for Intelligent Mining Systems, University of Alberta, Jul 2008.

[32] M. Polak, H. Zhang, and M. Pi. An evaluation metric for image segmentation of multiple objects. *Image and Vis. Comput*, 27:1223–1227, 2009.

[33] A. Ruszczynski. *Nonlinear optimization*. Princeton University Press, Princeton, NJ, 2006.

[34] M. S. Ruszczynski. *Encyclopaedia of mathematics: Hellinger distance*. SpringerLink, 2002, New York.

[35] B. N. Saha and N. Ray. Image thresholding by variational minimax optimization. *Pattern Recognition*, 42:843–856, 2008.

[36] A. E. Savakis. Adaptive document image thresholding using foreground and background clustering. *IEEE Intl. Conf. Image Process.*, 3:785–789, 1998.

[37] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–145, 2004.

[38] J. G. Shanahan and N. Roma. *Boosting support vector machines for text classification through parameter-free threshold relaxation*. ACM, New York, NY, USA, 2003.

[39] L. G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice Hall, New Jersey, 2001.

[40] B. W. Silverman. *Density estimation for statistics and data analysis*. CRC Press, London, UK, 1986.

[41] S. Singh and K. Bovis. *Medical image segmentation in digital mammmography*. Springer-Verlag New York, Inc., New York, NY, 2001.

[42] K. Torabi, S. Sayad, and S. T. Balke. Adaptive image thresholding for real-time particle monitoring. *Inte. Journal Imaging Syst. Tech.*, 16:9–14, 2006.

[43] O. D. Trier and A. K. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-17:1191–1201, 1995.

[44] O. D. Trier and T. Taxt. Evaluation of binarization methods for document images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:312–315, 1995.

[45] S. Venkatesh and P. L. Rosin. Dynamic threshold determination by local and global edge evaluation. *Graphical Model and Image Processing*, 57:146–160, 1995.

[46] H. Wang, N. Ray, and H. Zhang. Graph-cut optimization of the ratio of functions and its application image segmentation. *IEEE Intl. Conf. Image Process.*, pages 737–740, Oct. 2008.

[47] C. A. Waring and X. Liu. Face detection using spectral histograms and SVMs. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 35:467–476, 2005.

[48] J. M. White and G. D. Roher. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM J. Res. Dev.*, 27:400–411, 1983.

[49] Wikipedia. Bhattacharyya distance. http://en.wikipedia.org/wiki/Bhattacharyya_distance, 2009.

[50] Wikipedia. Kullback-leibler divergence. http://en.wikipedia.org/wiki/Kullback-Leibler_divergence, 2009.

[51] Wikipedia. Least square. http://en.wikipedia.org/wiki/Least_squares, 2009.

[52] F. Yan, H. Zhang, and C. R. Kube. A multistage adaptive thresholding method. *Pattern Recognition Letters*, 26:1183–1191, 2005.

[53] Y. Yang and Y. Yan. An adaptive logical method for binarization of degraded document images. *Pattern Recogn.*, 33:787–807, 2000.

[54] S. D. Yanowitz and A. M. Bruckstein. A new method for image segmentation. *Comput. Graph. Image Process*, 46:82–95, 1989.

[55] H. Zhang. iCore ICE annual report. Technical report, Center for Intelligent Mining Systems, University of Alberta, May 2005.

[56] H. Zhang. iCore ICE annual report. Technical report, Center for Intelligent Mining Systems, University of Alberta, May 2006.

[57] D. Zhou and H. Zhang. Modified GMM background modeling and optical flow for detection of multiple moving objects. *IEEE Intel. Conf. System Man and Cyber.*, pages 10–13, Oct. 2005.

[58] D. Zhou, H. Zhang, and N. Ray. Texture based background substraction. *IEEE Intl. Conf. Information and Automation*, pages 601–605, Jun. 2008.