

Study of Image Segmentation Schemes and Their Applications to Microscopic Images

S Jagannath Prusty

Rosen Dash



Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

Study of Image Segmentation Schemes and Their Applications to Microscopic Images

*Thesis submitted in partial fulfillment
of the requirements for the degree of*

Bachelor of Technology

in

Computer Science and Engineering

by

S Jagannath Prusty

(Roll: 10506038)

Rosen Dash

(Roll: 10506042)



Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India

May 2009



Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

Certificate

This is to certify that the work in the thesis entitled *Study of Image Segmentation Schemes and Their Application to Microscopic Images* by *S Jagannath Prusty & Rosen Dash* is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the session 2008–2009 in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

NIT Rourkela
11 May 2009

Pankaj Kumar Sa
Lecturer
CSE department of NIT Rourkela

Acknowledgment

It will be simple to name all those people who helped us to get this thesis done, however it will be tough to thank them enough. We will nevertheless try. . .

We would like to gratefully acknowledge the enthusiastic supervision and guidance of Pankaj Kumar Sa, Lecturer, during this work. He is our source of inspiration.

We are very much indebted to Prof. Banshidhar Majhi, Head-CSE, for his continuous encouragement and support.

Our sincere thanks goes to Ratnakar Dash, Research Fellow, for his continuous encouragement. Special thanks goes to Saroj Kumar Panigrahy, Project Officer for helping us at various stages.

We must acknowledge the academic resource that we have got from NIT Rourkela.

S. Jagannath Prusty & Rosen Dash

Abstract

Image Segmentation is one of the most important concerns in digital image processing. It's a long standing problem in computer vision. Often it is viewed as an ill-defined problem in comparison to other vision tasks which have apparently well defined objectives, such as detection, recognition, and tracking. It is fair to say that computer vision or image understanding is all about parsing images.

Different fundamental algorithms of image segmentation are implemented like Hough Transform Algorithm (*Edge Linking & Boundary Detection*), Region Growing (*Similarity Based Segmentation*), Iterative Thresholding Method. A new approach of Image Segmentation based on Maximum A Posteriori is implemented which uses the concept of Baye's Theorem.

A segmentation based problem to count number of white and black nano particles approximately in a microscopic image of a ceramic material is successfully solved.

Contents

Certificate	ii
Acknowledgement	iii
Abstract	iv
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Image Segmentation	2
1.1.1 Discontinuity Based Segmentation	3
1.1.2 Similarity Based Segmentation	9
1.2 Applications of Segmentation	10
1.3 Motivation	11
1.4 Organisation of the Thesis	11
2 Segmentation using Hough Transform	12
2.1 Introduction	12
2.2 Implementation of Hough Transform	15
3 Iterative Thresholding and Region Growing	17
3.1 Iterative Thresholding	17
3.1.1 Implementation of Iterative Thresholding	18
3.2 Region Growing	18
3.2.1 Implementation of Region Growing	20
4 Maximum A Posteriori	21
4.1 Introduction	21

4.2	Mathematical Representation	21
4.3	Segmentation using MAP	22
4.4	Implementation	23
5	Segmentation of a Microscopic Image	25
5.1	Introduction	26
5.2	Steps Involved	26
6	Conclusions	29
6.1	Achievements and Limitations of the work	29
6.2	Further Development	30
	Bibliography	31

List of Figures

2.1	Different types of representation of lines	13
2.2	Transformation from xy -plane to $\rho\theta$ -plane	13
2.3	Hough Transform	15
3.1	Global Thresholding	18
3.2	Region Growing	20
4.1	Region Growing using MAP	24
5.1	Microscopic Image	25
5.2	Input Image	26
5.3	Stages in solving ceramic material problem	28

List of Tables

1.1	A General 3×3 mask	4
1.2	Point Detection Mask	5
1.3	Line masks	6
1.4	Window Under Consideration	8
1.5	(a), (b) - Roberts; (c), (d) - Prewitt, (e), (f) - Sobel	8
4.1	Different MAP Probabilities	24

Chapter 1

Introduction

Digital Image Processing is a rapidly evolving field with growing applications in science and engineering. Image Processing holds the possibility of developing the ultimate machine that could perform the visual functions of all living beings [1]. *Digital Image Processing* methods help us to improve pictorial information for human interpretation and also process image data for storage, transmission and representation for autonomous machine perception [2].

An image may be described as a two-dimensional function I .

$$I = f(x, y) \quad (1.1)$$

where x and y are spatial coordinates. Amplitude of f at any pair of coordinates (x, y) is called intensity I or gray value of the image. When spatial coordinates and amplitude values are all finite, discrete quantities, the image is called digital image [2].

Digital image processing may be classified into various subbranches based on methods whose: [2]

- input and output are images and
- inputs may be images where as outputs are attributes extracted from those images.

Following is the list of different image processing functions based on the above two classes.

- Image Acquisition

- Image Enhancement
- Image Restoration
- Color Image Processing
- Multi-resolution Processing
- Compression
- Morphological Processing
- Segmentation
- Representation and Description
- Object Recognition

For the first seven functions the inputs and outputs are images where as for the rest three the outputs are attributes from the input images. With the exception of image acquisition and display most image processing functions are implemented in software. Image processing is characterized by specific solutions, hence the technique that works well in one area can be inadequate in another. The actual solution of a specific problem still requires a significant research and development [3].

Out of the ten sub-branches of digital image processing, cited above, this thesis deals with image segmentation. Different approaches of segmentation are surveyed and a practical problem based on image segmentation is solved.

This chapter is organized as follows. Image Segmentation is discussed in Section 1.1. The problem definition is described in Section 1.2. Organization of the thesis is outlined in Section 1.4.

1.1 Image Segmentation

Image segmentation refers to the decomposition of a scene into its components [1]. Segmentation subdivides an image into its constituent regions or objects. The level to which the subdivision is carried depends on the problem being solved. That is,

segmentation should stop when the objects of interest in an application have been isolated. For example, the automated inspection of electronic assemblies, interest lies in analysing images of the products with the objective of determining the presence or absence of specific anomalies, such as missing components or broken connection paths. There is no point in carrying segmentation past the level of detail required to identify those elements [2].

Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the eventual success or failure of computerized analysis procedures. For this reason, considerable care should be taken to improve the probability of rugged segmentation [2].

Image segmentation algorithms generally are based on one of two basic properties of intensity values: discontinuity and similarity. In the first category, the approach is to partition an image based on abrupt changes in intensity, such as isolated points, lines and edges in an image. The principal approaches in second category are based on partitioning an image into regions that are similar according to a set of predefined criteria. Thresholding, region growing are examples of methods in this category.

1.1.1 Discontinuity Based Segmentation

The focus of this section is on segmentation methods that are based on detecting sharp, *local* changes in intensity. The three types image features in which we are interested are isolated points, lines and edges. *Edge pixels* are pixels at which the intensity of an image function changes abruptly, and *edges* (or *edge segments*) are sets of connected edge pixels. *Edge detectors* are local image processing methods designed to detect edge pixels. A line may be viewed as an edge segment in which the intensity of the background on either side of the line is either much higher or much lower than the intensity of the line pixels [2].

The most common way to look for discontinuities is to run a mask through the image. For the 3×3 mask shown in Table - 1.1, this procedure involves the sum of products of the coefficients with the gray levels contained in the region encompassed by the mask. The response of the mask at any point in the image

Table 1.1: A General 3×3 mask

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

given by

$$\begin{aligned}
 R &= w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 \\
 &= \sum_{i=1}^9 w_i z_i
 \end{aligned} \tag{1.2}$$

where z_i is the gray level of pixel associated with mask coefficient w_i . As usual, the response of the mask is defined with respect to its center location.

Point Detection

The detection of isolated points in an image is straight forward in principle. Using the mask shown in Table - 1.2, we can say that a point has been detected at the location on which the mask is centered if

$$|R| \geq T \tag{1.3}$$

where T is a nonnegative threshold and R is given by eq. (1.2). Basically, this formulation measures the weighted differences between the center point and its neighbors. The idea is that an *isolated* point will be quite different from its surroundings and thus be easily detectable by this type of mask. A point whose gray level is significantly different from its background and which is located in a homogeneous or nearly homogeneous area is called *isolated* point. The mask shown in Table - 1.2 coefficients sum to zero, indicating that the mask response will be zero in areas of constant gray level.

Line Detection

The next level of discontinuity based segmentation is line detection. If the first mask were moved around a image, it responds more strongly to lines (one pixel

Table 1.2: Point Detection Mask

-1	-1	-1
-1	8	-1
-1	-1	-1

thick) oriented horizontally. With a constant background, the maximum response would result when the line passed through the middle row of the mask. This is easily verified by sketching a simple array of 1's with a line of a different gray level running horizontally through the array. A similar experiment would reveal that second mask in Table - 1.3 responds best to lines oriented at $+45^\circ$; the third mask to vertical lines; and the fourth mask to lines in the -45° direction. These directions can be established also by noting the preferred direction of each mask is weighted with a larger coefficient i.e., 2 than other possible directions. The coefficients sum to zero, indicating that the mask response will be zero in areas of constant gray level.

Let R_1 , R_2 , R_3 and R_4 denote the responses of the masks in Table 1.3, from left to right, where R 's are given by the eq. (1.2). Suppose that the four masks are run individually through an image. If, at a certain point in the image $|R_i| > |R_j|$, for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i . For example, if a point in the image, $|R_1| > |R_j|$ for $j = 2, 3, 4$ that particular point is said to be more likely associated with a horizontal line. Alternatively, we may be interested in detecting lines in a specified direction. In this case, we would use the mask associated with that direction and threshold its output, as in eq. (1.3). In other words, if we are interested in detecting all the lines in an image in the direction defined by a given mask, we simply run the mask through the image and threshold the absolute value of the result. The points that are left are the strongest responses, which, for lines one pixel thick, correspond closest to the direction defined by the mask.

The result of using a threshold equal to the maximum value in the image. The maximum value is a good choice for a threshold in applications such as this

because the input image is binary and we are looking for the strongest responses.

Table 1.3: Line masks

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			$+45^\circ$			Vertical			-45°		

Edge Detection

Although point and line detection certainly are important in any discussion on segmentation, edge detection is by far the most common approach for detecting meaningful discontinuities in gray level. We start by modelling an edge intuitively. This lead us to a formalism in which “meaningful” transitions in gray levels can be measured. An ideal edge is a set of connected pixels (in the vertical direction here), each of which is located at an orthogonal step transition in gray level.

In practice, optics, sampling and other image acquisition imperfections yield edges that are blurred, with the degree of blurring being determined by factors such as the quality of image acquisition system, the sampling rate, and illumination conditions under which the image is acquired. As a result, edges are more closely modeled as having a “ramplike” profile. The slope of the ramp is inversely proportional to the degree of blurring in edge. In this model we no longer have a thin (one pixel thick) path. Instead an edge point now is any point contained in ramp, and an edge then would be set of such points that are connected. The “thickness” of the edge is determined by the length of the ramp, as it transitions from an initial to a final gray level. This length is determined by the slope, which, in turn, is determined by the degree of blurring. This makes sense: Blurred edges tend to be thick and sharp edges tend to be thin.

Gradient operators The gradient of an image $f(x, y)$ at location (x, y) is defined as *vector*

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (1.4)$$

It is well known from vector analysis that the gradient vector points in the direction of maximum rate of change of f at coordinates (x, y) .

An important quantity in the edge detection is magnitude of this vector, denoted $|\nabla f|$, where

$$|\nabla f| = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \quad (1.5)$$

This quantity gives the maximum rate of increase of $f(x, y)$ per unit distance in the direction of ∇f . It is common (although not strictly correct) practice to refer to $|\nabla f|$ also as the *gradient*. We will adhere to convention and also use this term interchangeably, differentiating between the vector and its magnitude only in cases in which confusion is likely.

The *direction* of the gradient vector is also important quantity. Let $\alpha(x, y)$ represent the direction angle of vector ∇f at (x, y) . Then, from vector analysis,

$$\alpha(x, y) = \tan^{-1}(G_y/G_x) \quad (1.6)$$

where the angle is measured with respect to the x -axis. The direction of an edge at (x, y) is *perpendicular* to the direction of the gradient vector at that point.

Computation of the gradient of an image is based on obtaining the partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ at every pixel location. Let the 3×3 area shown in Fig. 1.4 represent the gray levels in a neighborhood of an image. One of the simplest ways to implement a first order partial derivative at point z_5 is to use the following *Roberts cross-gradient operators*:

$$G_x = (z_9 - z_5) \quad (1.7)$$

and

$$G_y = (z_8 - z_6) \quad (1.8)$$

Table 1.4: Window Under Consideration

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Table 1.5: (a), (b) - Roberts; (c), (d) - Prewitt, (e), (f) - Sobel

-1	0
0	-1

(a)

0	-1
-1	0

(b)

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

(c)

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

(d)

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

(e)

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

(f)

These derivatives can be implemented for an entire image by using the masks shown in Table - 1.5.

Masks of size 2×2 are awkward to implement because they don't have a clear center. An approach using masks of size 3×3 is given by

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \quad (1.9)$$

and

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \quad (1.10)$$

In this formulation, the difference between the first and third rows of 3×3 image region approximates the derivative in the x -direction, and the difference between third and first columns approximates the derivative in y -direction. The masks shown in Table - 1.5 (c), (d) called *Prewitt operators*, can be used to implement these two equations.

A slight variation of these two equations uses a weight of 2 in the center coefficient:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (1.11)$$

and

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (1.12)$$

A weight value of two is used to achieve some smoothing by giving more importance to the center point. *Sobel* operators use above two equations. The Prewitt and Sobel operators are among the most used in practice for computing digital gradients. The Prewitt masks are simpler to implement than Sobel masks, but the latter have slightly superior noise suppression characteristics, an important issue when dealing with derivatives.

Hough Transform is another method for detecting straight edges which has been discussed elaborately in next chapter.

1.1.2 Similarity Based Segmentation

In the previous section we discussed the algorithms which are categorized as discontinuity based segmentation. In this section we discuss segmentation based on global or local thresholding value. Thresholding enjoys a central position in application of image segmentation.

To define threshold intensity lets consider an image f composed of light objects on a dark background, in such a way that object and background pixels have intensity values grouped into two dominant modes. One obvious way to extract the objects from the background is to select a threshold T that separates these modes. Then any point (x, y) in the image at which $f(x, y) > T$ is called an object point else it is called a background point. In other words the segmented image $g(x, y)$ is given by

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (1.13)$$

where T is a constant applicable to the entire image, the process given in the above equation is referred to as global thresholding. When the value of T changes over an image we use the term variable thresholding. The term local or regional thresholding is used to denote variable thresholding in which the value of T any point (x, y) in an image depends on properties of a neighborhood of (x, y) (eg. the average intensity of the pixels in the neighborhood). But if T depends on the spatial coordinates (x, y) themselves then the variable thresholding is often referred as dynamic or adaptive thresholding.

There may be an image with three dominant modes having two types of light objects on a dark background, so here we have multiple threshold values. Then any point (x, y) in the image is belonging to the background if $f(x, y) \leq T_1$, to one object class if $T_1 < f(x, y) \leq T_2$, and to the other object class if $f(x, y) > T_2$.

The segmented image is given by

$$g(x, y) = \begin{cases} a & \text{if } f(x, y) > T_2 \\ b & \text{if } T_1 < f(x, y) \leq T_2 \\ c & \text{if } f(x, y) \leq T_1 \end{cases} \quad (1.14)$$

where a, b, c are three distinct intensity values.

One more important method known as region growing which comes under the category of similarity segmentation. As its name suggests, it is a procedure that groups pixels or subregions into larger regions based on predefined criteria for growth. Here the basic approach is to start with a set of “seed” points and from these regions are grown by appending to each seed those neighboring pixels that satisfies the predefined criteria.

1.2 Applications of Segmentation

Some of the practical applications of image segmentation are: Medical Imaging like locate tumors and other pathologies, measure tissue volumes, computer-guided surgery, etc. Various other fields where image segmentation is being used: locate objects in satellite images (roads, forests, etc.), face recognition, fingerprint recognition, traffic control systems, brake light detection, machine vision, etc.

A specific application of image segmentation where we have counted the number of white nano particles in an given SEM image.

1.3 Motivation

In the adjustive study of different segmentation schemes like Hough Transform, Iterative Thresholding Method, Region Growing Method, etc., each scheme works well for a specific type of images.

To segment a microscopic image we cannot directly implement any of them. So we proposed a hybrid segmentation scheme. This hybrid scheme incorporates both the concepts of discontinuity based and similarity based segmentation.

1.4 Organisation of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 covers the methods for discontinuity based segmentation and particularly *Hough Transform* for edge linking and boundary detection is discussed in detail with a suitable example.

Chapter 3 discusses the methods for similarity based image segmentation. The method of *Iterative thresholding* and *Region Growing* are discussed with implementation strategies.

Chapter 4 concentrates on a special method for region based image segmentation i.e., *Maximum A Posteriori* (MAP) which uses the concept of Baye's Theorem and likelihood probability.

In Chapter 5 we discuss a specific approach to count no. of white nano particles in a given ceramic material SEM image. All the steps followed to solve the problem have been discussed along with the output images of each step.

The overall conclusion of this thesis is given in Chapter 6.

Chapter 2

Segmentation using Hough Transform

There are various methods for discontinuity based segmentation as we have discussed in previous chapter. In this chapter we are mainly focused on the concept behind the method of Hough Transform and its implementation strategy. Finally the result obtained from our implementation is added.

2.1 Introduction

The Hough transform is an edge linking and boundary detection technique used in image analysis. The purpose of the technique is to find instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.

Consider a point (x_i, y_i) in the xy -plane and the general equation of a straight line in slope-intercept form, $y_i = ax_i + b$. Infinitely many lines pass through (x_i, y_i) , but they all satisfy the equation for $y_i = ax_i + b$ for varying values of a and b . However writing this equation as $b = -x_i a + y_i$ and considering the ab -plane (also called parameter space) yield the equation of a single line for a fixed pair (x_i, y_i) . Furthermore, a second point (x_j, y_j) also has a line in parameter space associated with it, and unless they are parallel, this line intersects the line associated with (x_i, y_i) at some point (a', b') , where a' is the slope and b' is the intercept of the

line containing both (x_i, y_i) and (x_j, y_j) in the xy -plane. In fact, all the points on this line have lines in parameter space that intersect at (a', b') .

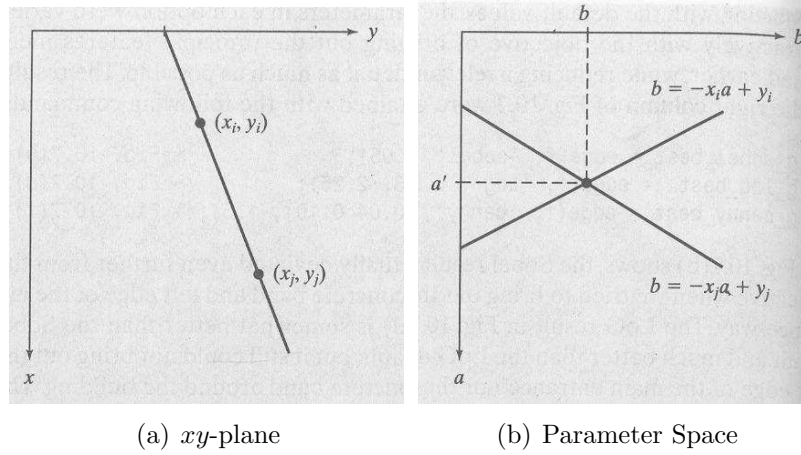


Figure 2.1: Different types of representation of lines

In principle, the parameter-space lines corresponding to all points (x_k, y_k) in the xy -plane could be plotted, and the principal lines in that plane could be found by identifying points in parameter space where large numbers of parameter-space intersect. A practical difficulty with this approach, however, is that a (the slope of a line) approaches infinity as the line approaches the vertical direction. This problem can be solved by using normal representation of a line:

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

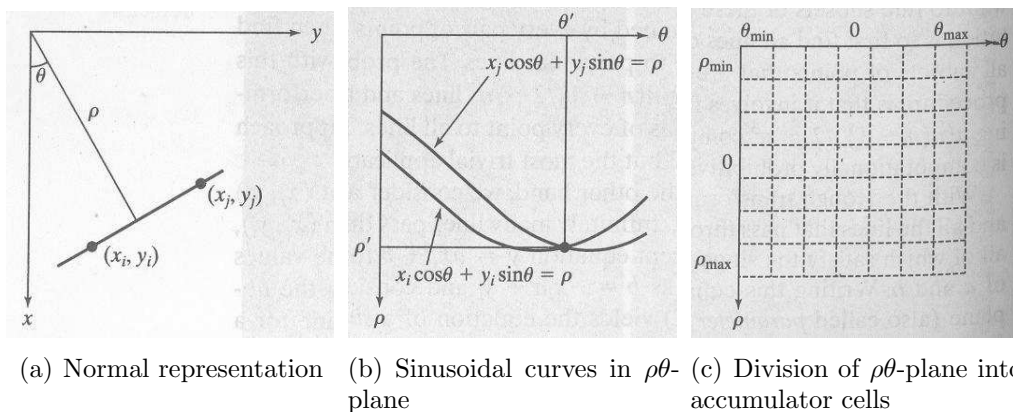


Figure 2.2: Transformation from xy -plane to $\rho\theta$ -plane

Fig. 2.2(a) represents the geometrical interpretation of the parameters θ and ρ . A horizontal line has $\theta = 0^\circ$, with ρ being equal to positive x -intercept. Similarly,

a vertical line has $\theta = 90^\circ$, with ρ being equal to positive y -intercept or $\theta = -90^\circ$, with ρ being equal to negative y -intercept. Each sinusoidal curve as shown in Fig. 2.2(b) represents the family of lines that pass through a particular point (x_k, y_k) in the xy -plane. The intersection point (ρ', θ') as shown in Fig. 2.2(b) corresponds to the line that passes through both (x_i, y_i) and (x_j, y_j) in Fig. 2.2(a)

The computational attractiveness of the Hough Transform arises from subdividing the $\rho\theta$ parameter space into so-called accumulator cells, as shown in Fig. 2.2(c) illustrates, where (ρ_{min}, ρ_{max}) and $(\theta_{min}, \theta_{max})$ are the expected ranges of the parameter values: $-90^\circ \leq \theta \leq 90^\circ$ and $-D \leq \rho \leq D$, where D is the maximum distance between opposite corners in an image. The cell at coordinates (i, j) , with accumulator value $A(i, j)$, corresponds to the square associated with parameter-space coordinates (ρ_i, θ_j) . Initially, these cells are set to zero. Then, for every non-background point (x_k, y_k) in the xy -plane, we let θ equal each of the allowed subdivision values on the θ axis and solve for the corresponding ρ using the equation $\rho = x_k \cos\theta + y_k \sin\theta$. The resulting ρ values are then rounded off to the nearest allowed cell value along ρ axis. If a choice of θ_p results in solution ρ_q , then we let $A(p, q) = A(p, q) + 1$. At the end of this procedure, a value of P in $A(i, j)$ means that P points in the y -plane lie on the line $x \cos\theta_j + y \sin\theta_j = \rho_i$. The number of subdivisions in the $\rho\theta$ -plane determines the accuracy of the collinearity of the points.

The steps for an edge-linking problem using Hough Transform are as follows:

1. Obtain a binary image using any of the techniques discussed in the previous sections.
2. Specify subdivisions in the $\rho\theta$ -plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

Continuity is based on computing the distance between disconnected pixels corresponding to a given accumulator cell. A gap in a line associated with a

given cell is bridged if the length of the gap is less than a predefined specific threshold. The threshold value is defined according to the requirements. This gives a significant advantage over the methods discussed in the previous sections.

2.2 Implementation of Hough Transform

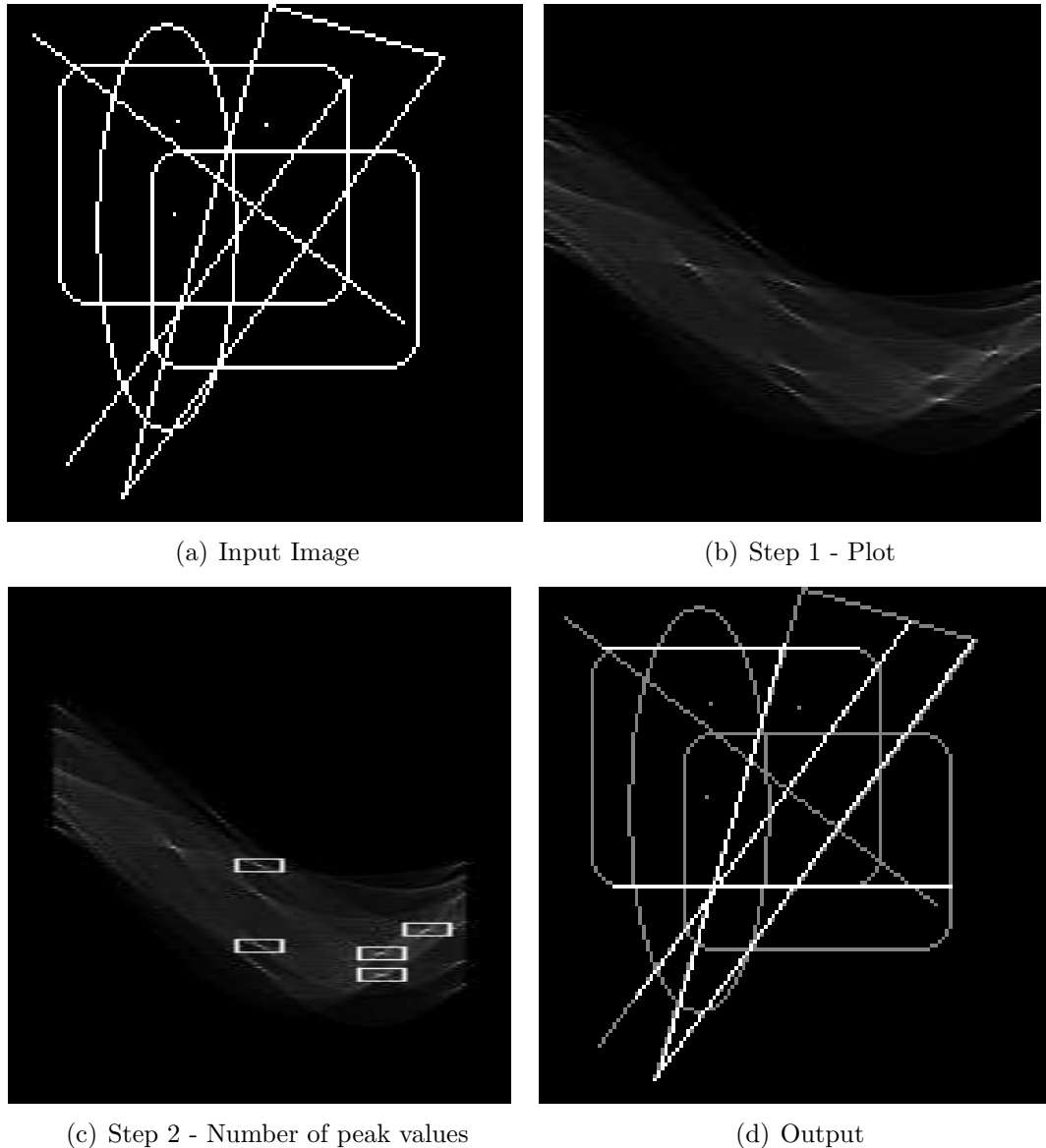


Figure 2.3: Hough Transform

In order to implement Hough Transform we have taken input image as shown in Fig. 2.3(a). Next we have plotted the sinusoidal curves for each white pixel in the input image as shown in Fig. 2.3(b). The point of intersections of the

sinusoidal curves in the $\rho\theta$ - plane are known as peak points. Each peak point represents all the points lying in a straight line in the input image. Then, top five peak points are selected as shown in Fig. 2.3(c) to identify its equivalent straight lines in the input image. Finally the output image as shown in Fig. 2.3(d) shows the identified lines in a brighter shade.

Chapter 3

Iterative Thresholding and Region Growing

In this chapter we are going to discuss the methods of iterative thresholding and region growing for region based segmentation. The various steps involved and the additional criteria which increases the efficiency of these algorithms have been discussed. Finally, the implementation with suitable examples are given.

3.1 Iterative Thresholding

As in most of the applications there is usually enough variability between images that, even if global thresholding is a suitable approach, an algorithm capable of estimating automatically the threshold value for each image is required.

The following are the steps of the iterative thresholding method:

1. Select an initial estimate for the global threshold, T .
2. Segment the image using T , it produces two groups of pixels G_1 consisting of all pixels with intensity values $> T$, and G_2 consisting of pixels with values $\leq T$.
3. Compute the average(mean) intensity values m_1 and m_2 for pixels in G_1 and G_2 , respectively.
4. Compute new threshold value as

$$T = \frac{1}{2}(m_1 + m_2) \quad (3.1)$$

5. Repeat steps 2 through 4 until the difference between values of T in successive iterations is smaller than a predefined parameter ΔT .

Parameter ΔT is used to control the number of iterations in situations where speed is an important issue. In general, the larger ΔT is, the fewer iterations the algorithm will perform. The initial threshold must be chosen greater than the minimum and less than the maximum intensity level in the image. Generally average intensity of an image is chosen as the initial threshold value.

3.1.1 Implementation of Iterative Thresholding

Using Iterative Thresholding the input image given in Fig. 3.1(a) is segmented and the produced output is shown in Fig. 3.1(b)

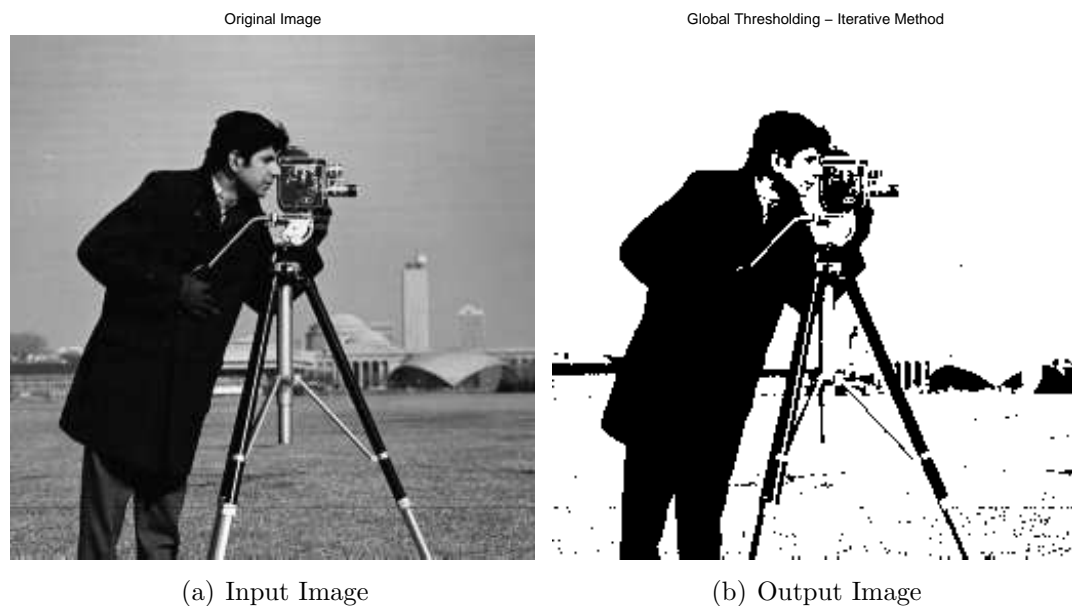


Figure 3.1: Global Thresholding

3.2 Region Growing

It is defined as the procedure that groups pixels or subregions into larger regions based on a predefined criteria for growth. The region growing is done by first selecting a set of seed points. Selecting a set of seed points often can be based on the nature of the problem. When a prior information is not available, the

procedure is to compute at every pixel the same set of properties that ultimately will be used to assign pixels to regions during the growing process. If the result of these computations shows the clusters of values, the pixels whose properties place them near the centroid of these clusters can be used as seeds.

Grouping pixels with the same intensity level to form a “region” without paying attention to connectivity would yield a segmentation result that is meaningless.

The most important thing in region growing is the formulation of a stopping rule. Region growth should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as intensity values, texture, and color are local in nature and do not take into account the history of region growth. Additional criteria that increase the power of a region growing algorithm utilize the concept of size, likeness between a candidate pixel and the pixels grown so far, and the shape of the region being grown.

Let $f(x, y)$ denote an input image array; $S(x, y)$ denote a seed array containing 1's at locations of seed points and 0's elsewhere; and Q denote a predicate to be applied at each location (x, y) . Arrays f and S are assumed to be of the same size. A basic region growing algorithm based on 8-connectivity may be stated as follows:

1. Find all connected components in $S(x, y)$ and erode each connected component to one pixel; label all such pixels found as 1. All other pixels in S are labeled as 0.
2. Form an image f_q such that, at a pair coordinates (x, y) , let $f_q(x, y) = 1$ if the input image satisfies the given predicate, Q , at those coordinates; otherwise, $f_q(x, y) = 0$.
3. Let g be an image formed by appending to each seed point in S all the 1-valued points in f_q that are 8-connected to that seed point.
4. Label each connected component in g with a different region label.

3.2.1 Implementation of Region Growing

This region growing algorithm has been implemented using input image as shown in Fig. 3.2(a). The seed point for the region growing selected at the marked point. The output produced is shown in the Fig. 3.2(b).

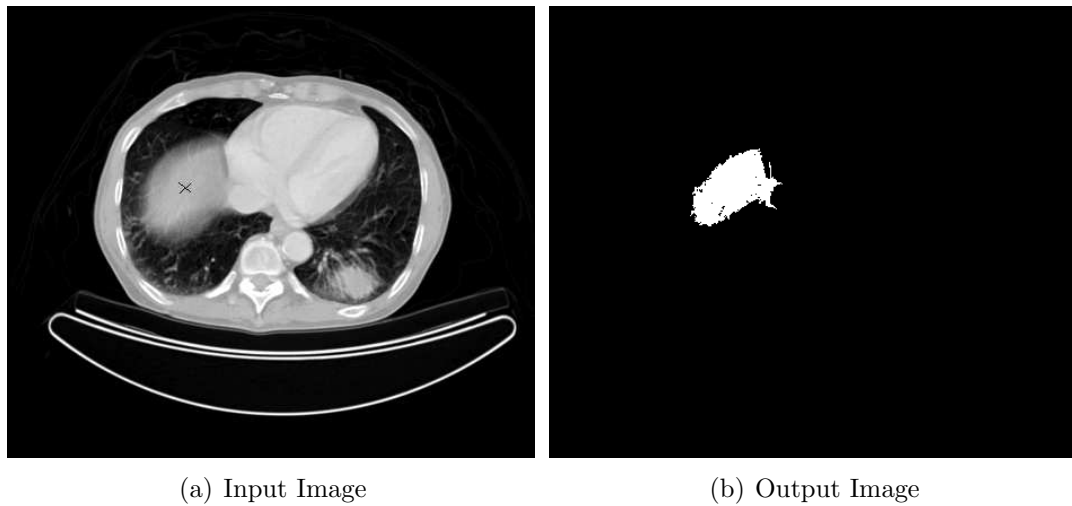


Figure 3.2: Region Growing

Chapter 4

Maximum A Posteriori

In this chapter we are going to discuss *maximum a posteriori* in detail. MAP uses the concepts of Baye's Theorem and likelihood principle. We have implemented this concept for segmentation using an image having two distinct intensity values.

4.1 Introduction

In Bayesian statistics, a *maximum a posteriori* (MAP) estimate is a mode of the posterior distribution. Sometimes we have a priori information about the physical process whose parameters we want to estimate. Such information can come either from the correct scientific knowledge of the physical process or from previous empirical evidence.

4.2 Mathematical Representation

$P(A)$ = prior probability of a model A (without evidence)

$P(B)$ = probability of the evidence B

$P(B|A)$ = likelihood that the evidence B was produced given that the model A

$P(A|B)$ = posterior probability of the model being A , given the evidence is B [4]

MAP finds the model A which maximize the posterior probability $P(A|B)$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \Rightarrow P(A \cap B) = P(A|B)P(B) \quad (4.1)$$

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \Rightarrow P(A \cap B) = P(B|A)P(A) \quad (4.2)$$

$$P(A|B)P(B) = P(B|A)P(A) \Rightarrow P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (4.3)$$

If A_1, A_2, A_3 do not overlap; $(B \cap A_1), (B \cap A_2), \dots$ do not overlap.

$$\begin{aligned} P(B) &= P(A_1 \cap B) + P(A_2 \cap B) + \dots \\ &= P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + \dots \\ &= \sum_{i=1}^n P(B|A_i)P(A_i) \end{aligned} \quad (4.4)$$

$$\therefore P(A|B) = \frac{P(B|A)P(A)}{\sum_{i=1}^n P(B|A_i)P(A_i)} \quad (4.5)$$

4.3 Segmentation using MAP

To implement MAP the intensity values should be divided into different states. For simplicity we have considered an image having two distinct intensity values: dark and bright. So, we have two states: Dark State (S_1), Bright State (S_2). Each pixel will be in either S_1 or S_2 .

$P(S_i)$ = Prior probability

$P(O_j|S_i)$ = likelihood of the outcome O_j given state S_i

Prior Probability for each state is the probability of occurrence of that state in the entire image.

$$P(S_i) = \frac{\text{Number of pixels in } S_i}{\text{Total number of pixels in the image}} \quad (4.6)$$

$P(O_j|S_i)$, the *likelihood probability* is defined as the transition probability from S_i to O_j [5].

$$P(O_j|S_i) = \frac{\text{Number of transitions from } S_i \text{ to } O_j}{\text{Total number of transitions from } S_i \text{ to any state}} \quad (4.7)$$

$$P(S_i|O_j) = \frac{P(S_i \cap O_j)}{P(O_j)} \quad (4.8)$$

$$P(S_i \cap O_j) = P(O_j|S_i)P(S_i) \quad (4.9)$$

$$\begin{aligned} P(O_j) &= P(S_1 \cap O_j) + \dots + P(S_n \cap O_j) \\ &= \sum_{k=1}^{k=n} P(O_j|S_k)P(S_k) \end{aligned} \quad (4.10)$$

$$P(S_i|O_j) = \frac{P(O_j|S_i)P(S_i)}{\sum_{k=1}^n P(O_j|S_k)P(S_k)} \quad (4.11)$$

The next state S_i given the present state O_j will be included in the growing region if $P(S_i|O_j)$ is maximum. So, this method is called *maximum a posteriori*.

4.4 Implementation

We have taken the input image as shown in Fig. 4.1(a) which has two distinct intensity values, so we have two states. The various probability values generated using MAP for that image is shown in Table - 4.1 basing on the posterior probability value generated we grow the region from the seed point.

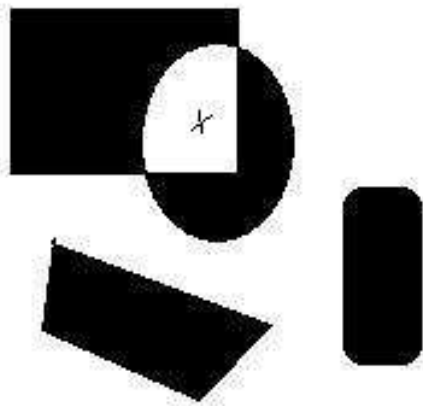
Table 4.1: Different MAP Probabilities

S_1	S_2		S_1	S_2		S_1	S_2	
0.2888	0.7111		S_1	0.9768	0.0231	S_1	0.9766	0.0233
			S_2	0.0094	0.9905	S_2	0.0094	0.9905

Prior Probability

Likelihood Probability

Posterior Probability



(a) Input Image



(b) Output Image

Figure 4.1: Region Growing using MAP

Chapter 5

Segmentation of a Microscopic Image

In this chapter we are going to discuss a specific problem requiring image segmentation and steps involved to solve it. The image is a micrograph taken from Scanning Electron Microscope (SEM) showing black and white nano particles as shown in Fig. 5.1. The nano particles are of dimensions of a few microns.

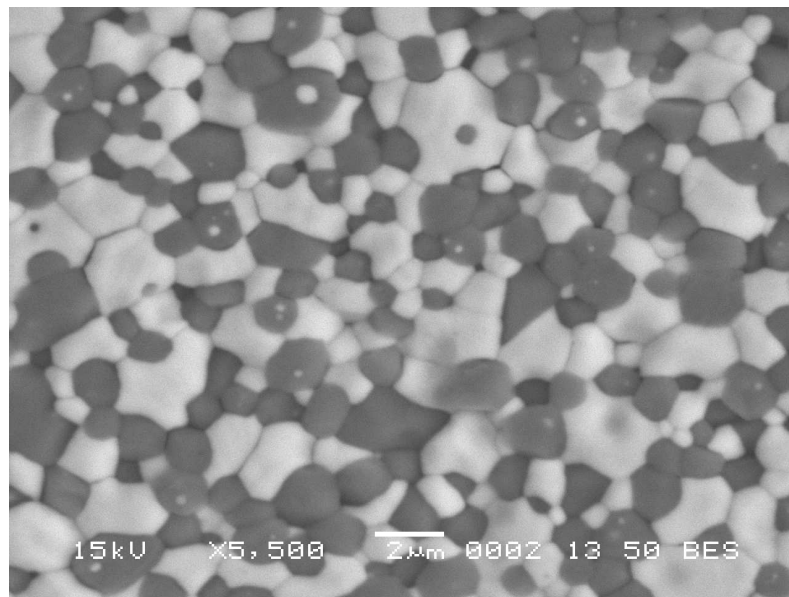


Figure 5.1: Microscopic Image

5.1 Introduction

The problem statement is to count number of white nano particles in the ceramic material gray SEM image as shown in Fig. 5.2. We cannot use any of the segmentation algorithms discussed earlier because the boundary of white structures is not quite distinct.

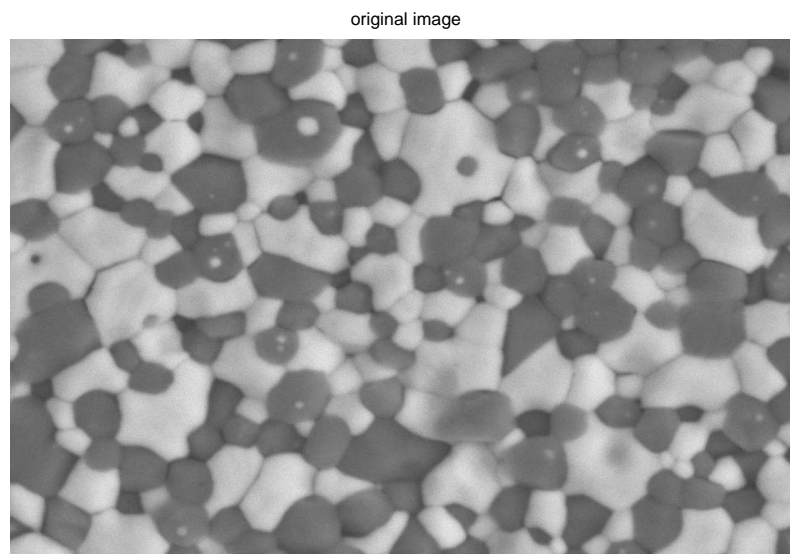


Figure 5.2: Input Image

5.2 Steps Involved

The following steps are used to count number of white nano particles:

1. As we cannot get distinct boundaries for both white and black nano particles at a time, a threshold value is decided analyzing the histogram of the image to focus only on the white nano particles and ignore the rest.

$$\begin{aligned} \text{if } & f(x, y) > T \\ & g(x, y) = f(x, y) \\ \text{else} & \\ & g(x, y) = 0 \end{aligned}$$

where,

$f(x, y)$ is the input image shown in Fig. 5.2.

$g(x, y)$ is the new input image for next step 5.3(a).

T is the threshold value

2. All the black area are changed to gray value 255. This change is required for contrast stretching, so that the difference between the white nano particles and their boundaries will be more distinct. 5.3(b)
3. To get the boundaries of the white nano particles as shown in Fig. 5.3(c), *Canny* edge detection algorithm is used on the new input image.
4. We have taken the inverse as shown in Fig. 5.3(d) of the previous image 5.3(c). So that we can take the logical & of this image with the image shown in Fig. 5.3(a) to get boundaries around the white nano particles.
5. Outcome of the previous step is shown in Fig. 5.3(e).
6. Then the inverse of the previous image is taken as shown in Fig. 5.3(f).
7. The boundaries what we have obtained are double layered so we thickened the boundaries to distinguish the white nano particles which are represented as black patches as shown in Fig. 5.3(g).
8. Finally we took inverse of the previous image to count no. of distinct white nano particles using 8-connectivity method 5.3(h).

By following the above steps the number of white nano particles is 154 for the image 5.2 which is quite acceptable.

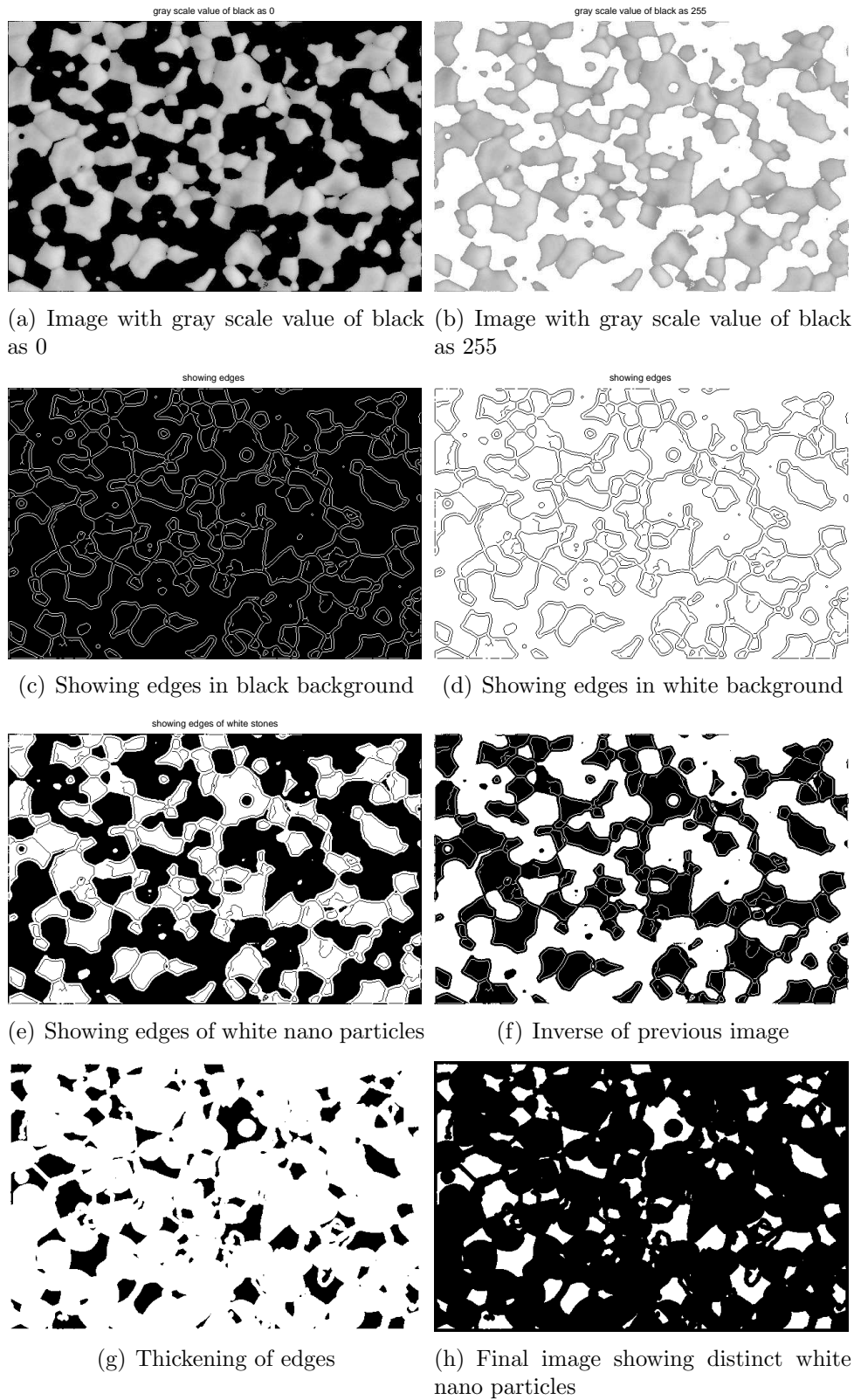


Figure 5.3: Stages in solving ceramic material problem

Chapter 6

Conclusions

The work in this thesis, primarily focuses on different approaches in image segmentation. Using MAP for segmenting an image. And, devised an approach to count number of white nano particles in a given ceramic material SEM image. Section 6.1 lists the pros and cons of the work. Section 6.2 provides some scope for further development.

6.1 Achievements and Limitations of the work

Each image segmentation approaches has some positive and negative aspects. Here we are going to discuss the pros and cons of methods described in this thesis.

Hough transform directly detects the object's edges using image global features; it can link every point to form a closed edge in image field. If the shape or edges of objects is known, using Hough transform edges can be detected and points can be linked together easily. The main advantage of the Hough transform technique is tolerant to the presence of gaps in feature boundary descriptions and is relatively unaffected by image noise. Also the Hough transform provides parameters to reduce the search time for finding lines based on a set of edge points, and that these parameters can be adjusted based on application requirements. The disadvantage of Hough transform is that the calculation quantity is very large. With image size increasing, the quantity of data will become too large and the processing quantity of data will be slow. However, there are two major drawbacks from the Hough transform; these drawbacks are the requirements for a large amount of storage

and high cost in computation.

In iterative thresholding method segmentation is based on the intensity value of the pixels. Pixels belonging to same segment does not have spatial connection with each other. So this method leads to unbounded segments.

Region growing method can correctly separate the regions that have the same properties we define. It can also provide the original images which have clear edges the good segmentation results. Some of its diadvantages: the computation is consuming, no matter the time or power; noise or variation of intensity may result in holes or oversegmentation; this method may not distinguish the shading of the real images.

As we know MAP requires both prior probability and likelihood probability, its not possible to have both of them for all type of images. So it has limited applications.

We are able to count number of white nano particles by following various steps as discussed in Chapter 5. But, this approach is meant for a particular kind of images only. The total area covered by the white nano particles cannot be calculated exactly as we have distorted their boundaries to separate from each other.

6.2 Further Development

A mathematical model like Hidden Markov Model can be used to give a mathematical explanation to the method discussed in chapter 5 for the images having this type of requirement. Various soft computing techniques can be used to have better result.

Bibliography

- [1] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall of India, 2008.
- [2] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison Wesley, 2nd edition, 1992.
- [3] Banshidhar Majhi. *Soft Computing Techniques for Image Restoration*. PhD thesis, Sambalpur University, 2000.
- [4] Claudio Becchetti and Lucio Prina Ricotti. *Speech Recognition*. John Wiley and Sons, 2004.
- [5] Hamid Hassanpour and Payam M. Farahabadi. Using hidden markov model for paper currency recognition. *Elsevier Expert Systems with Applications*, (1):10105 – 10111, 2009.
- [6] William K. Pratt. *Digital Image Processing*. John Wiley-Interscience Publication, 3rd edition, 2001.
- [7] B. Chanda and D. Dutta Majumder. *Digital Image Processing and Analysis*. Prentice-Hall of India, 1st edition, 2002.
- [8] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice Hall of India, New Delhi, 3rd edition, 2002.
- [9] M. Iosifescu. *Finite markov process and their applications*. NY: Wiley, 1980.

- [10] D. Patra P. K. Nanda and A. Pradhan. Unsupervised image segmentation using tabu search and hidden markov random field model. *National Conference on Recent Advances in Power, Signal Processing and Control*, 2004.
- [11] Amir Najmi Jia Li and Robert M. Gray. Image classification by a two dimensional hidden markov model. *Information Systems Laboratory, EE Dept. Stanford University*, pages 3313 – 3316, 1999.
- [12] Pankaj Kumar Sa. On the development of impulsive noise removal schemes. Master's thesis, NIT Rourkela, 2006.