12-2016

New marked point process models for microscopy images

Dae Woo Kim
Purdue University

Graduate School Form
30 Updated 12/26/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Dae Woo Kim

Entitled
New Marked Point Process Models for Microscopy Images

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Mary L. Comer
Chair

Edward J. Delp

Charles A. Bouman

Mireille Boutin

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Mary L. Comer

Approved by: Venkataramanan Balakrishnan 10/13/2016
Head of the Departmental Graduate Program Date

NEW MARKED POINT PROCESS MODELS FOR MICROSCOPY IMAGES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Dae Woo Kim

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2016

Purdue University

West Lafayette, Indiana

To my wife Bo Young Pak and my children Hee Chan Kim and Bethany Sion Kim

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Mary Comer for her continuous support of my Ph.D study. Prof. Comer enlightened me with a Bayesian approach to solve the image segmentation problems and also introduced me a new stochastic framework to detect objects in microscope images. Without her patience, motivation, and immense knowledge, my Ph.D could never be possible. Her guidance helped me in all the time of research and writing of this thesis.

I would also like to thank my thesis committee members, Prof. Edward Delp, Prof. Charles Bouman, and Prof. Mireille Boutin. Their insightful comments and encouragement widened my research from various perspectives. It is my great honor to have these professors, who are well-known in the field of image processing, as my thesis committee members.

During the course of this research, through the MURI project, I had many opportunities to cooperate with a group of wonderful material scientists from other prestigious universities and the Air Force Office of Scientific Research(AFOSR). Especially, I would like to thank Dr. Jeff Simmons and Prof. Marc De Graef, for providing many discussion and useful information which helped me learn about the materials science. Being involved in this MURI project became a priceless asset for my career in the area of microscope image processing.

I would like to thank all of my colleagues who supported to strive towards my goal: Hsiao-Chiang Chuang, Huixi Zhao, Shruthi Kubatur, Camilo G Aguilar Herrera. Also, I thank all ECE department staff who helped me with my academic affairs, business trip, and more. Their kind deeds made Purdue University a very special place for me.

I appreciate my family for their prayers and support: my wife, both of my son and daughter born here, my mother-in-law, father-in-law, my mother. Their prayers

are what sustained me this far. I have another family in the Lord who are as close as my blood family: Pastor Del, Nelson and Janet, Stan and Barb, Lyailya, John and Nancy, Mike and Angie, Steve, Fred, and more.

Most of all, I thank God for giving a chance to study at Purdue University and for having been faithful all the way through the end.

This work was partially funded by the DARPA YFA Program and the Air Force Office of Scientific Research MURI Managing the Mosaic of Microstructure

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

SYMBOLS

Chapter 3

$\mathbf{X}$	marked point process (random variable)
$\mathbf{x}$	configuration of marked point (realization of $\mathbf{X}$)
$\mathcal{C}$	space of a configuration of marked point
$\mathbf{Y}$	image (random variable)
$\mathbf{y}$	realization of $\mathbf{Y}$
$\mathbf{Z}$	segmentation label field (random variable)
$\mathbf{z}$	realization of $\mathbf{Z}$
M	mark space
K	rectangular lattice
$S = M \times K$	space of a marked point
n	number of marked point in a configuration
$i \in \{1, \cdots, n\}$	index of marked point in a configuration
$X_i = (S_i, N_i)$	ith marked point (random variable)
$s = (a, b) \in K$	spatial location
$s_i = (a_i, b_i) \in K$	center position of the ith marked point (realization of S_i)
y_s	element of image $\mathbf{y}$ at s
n_i	mark of ith marked point(realization of N_i)
c_i	channel type of the ith marked point
l_i	length of the ith marked point
w_i	width of the ith marked point
θ_i	orientation of the ith marked point
M_0	mark space of necking channel mark

M_1	mark space of denting channel mark
A_c^i	channel area of the ith marked point
A_{nc}^i	non-channel area of the ith marked point
R_c^i	image intensity data in channel area of the ith marked point
R_{nc}^i	image intensity data in non-channel area of the ith marked point
P_B, P_D	probability of choosing birth kernel and death kernel (RJMCMC)
n_f, n_s, n_d	number of free, single and double segment (candy model)
$\lambda_o, \lambda_c, \lambda_{int}$	control parameters of Gibbs energy
$\omega_f, \omega_s, \omega_d, \omega_c, \omega_{io}$	control parameters of candy model
R_c	set of connected segment pairs (candy model)
R_{io}	set of connected segment pairs which have internal bad orientation relation (candy model)
β_{seg}	interaction parameter (EM/MPM)

Chapter 4

$\mathbf{W}$	marked point process (random variable)
$\mathbf{w}$	configuration of marked point (realization of W)
$\mathcal{C}$	space of a configuration of marked point
$\mathbf{Y}$	image (random variable)
$\mathbf{y}$	realization of Y
M	mark space
K	rectangular lattice
$S = M \times K$	space of a marked point
n	number of marked point in a configuration
$i \in \{1, \cdots, n\}$	index of marked point in a configuration

$p = (x, y, z) \in K$	spacial location
y_p	element of image $\mathbf{y}$ at p
r, t	parameters to describe shape of superellipse and superellipsoid
D_i, m_i, v_i, T_d	notations used to calculate data potential
w_i	ith marked point (2D object)
$p_i = (p_i^x, p_i^y, p_i^z) \in K$	location of the ith marked point (2D object)
$p_i^{xy} = (p_i^x, p_i^y)$	x and y coordinates of the ith marked point (2D object)
a_i, b_i	semi-axis lengths of the ith marked point (2D object)
θ_i	orientation of the ith marked point (2D object)
T_c, T_p, n_z	control parameters of clustering
N_c	number of clusters
C_k	kth cluster
N_k	number of 2D objects in the kth cluster
$\hat{C}_k$	set of points on the outline of 2D object in the kth cluster
$s_m \in \hat{C}_k$	point on the outline of 2D object
$w_{k,i}$	ith 2D object in the kth cluster (after clustering)
$p_{k,i}$	location of the ith 2D object in k-th cluster (after clustering)
w_i^{3D}	ith marked point (3D object)
p_i^{3D}	location of the ith marked point (3D object)
$a_i^{3D}, b_i^{3D}, c_i^{3D}$	semi-axis lengths of the ith marked point (3D object)
$\alpha_i, \beta_i, \gamma_i$	orientation of the ith marked point (3D object)
σ	control parameter to generate center position map
l_p	line fitted with center positions of 2D objects

$\mathbf{v} = (v_x, v_y, v_z)$	direction vector of the line l_p
ϕ	tilt angle of a 3D object
T_o, σ_o	control parameter to orientation map
$r_p^m, r_p^M, z_p^m, z_p^M, c_p^z, c_p^M, c_p^m$	notations to generate semi-axis length map
$\varepsilon_r, \sigma_r, \varepsilon_c, \sigma_c$	control parameters to generate semi-axis length map
σ_n	control parameter to generate noise in synthetic dataset

ABSTRACT

Kim, Dae Woo Ph.D. Student, Purdue University, December 2016. New Marked Point Process Models for Microscopy Images. Major Professor: Mary L. Comer.

In developing new materials, the characterization of microstructures is one of the key steps. To characterize the microstructure, many microscope modalities have been devised and improved over decades. With the increase in image resolution in the spatial and time domains, the amount of image data keeps increasing in the fields such as materials science and biomedical engineering. As a result, image processing plays a critical role in this era of science and technology. In materials image analysis, image segmentation and feature detection are considered very important.

The first part of this research aims to resolve the segmentation problem caused by blurring artifacts in scanning electron microscopy(SEM) images. This blurring issue can lead to a bridged channel problem, which becomes an obstacle in analyzing the microstructures. To tackle the problem, we propose a joint deconvolution and segmentation (JDS) method. As a segmentation method, we use the expectation-maximization/maximization of the posterior marginals (EM/MPM) method, using the Markov random field(MRF) prior model. Experiments show the proposed method improves the segmentation result at object boundaries.

The next phase of the image segmentation is detecting image features. In the second part of this research, we detect channel configurations in materials images. We propose a new approach of channel identification, based on the marked point process(MPP) framework, to effectively detect channels in materials images. To describe a higher level of structures in an image, the MPP framework is more effective than the MRF prior model. The reversible-jump Markov chain Monte Carlo (RJMCMC) algorithm embedded with simulated annealing is used as an optimization method, and a new switching kernel in an RJMCMC is used to reduce computational time. The channel configuration is useful in characterizing materials images. In addition, this information can be used to reduce the bridged channel problem more effectively.

In materials image processing, one of the most important goals of feature detection is identifying the 3D structure of objects from 3D microscope datasets. The final part of this research is to perform fast and accurate estimation of 3D object configurations from a 3D dataset. We propose a fast 3D fitting method to improve the computational complexity over a full-search 3D MPP method. Experiments show that the fast 3D fitting method significantly decreases execution time compared to the full 3D MPP method.

1. INTRODUCTION

Historically, the development of new materials has had a huge impact on human society. As a result of discoveries or developments of new materials, people have consistently experienced significant changes in their daily life. It seems no wonder that history is categorized and named after materials - Stone Age, Bronze Age, and Iron Age. New materials, for example those used in military applications, have decided the destiny of some civilizations. Recently, biomaterials used in medical applications have not merely extended human lifespans but also improved the quality of life: titanium alloys for prostheses, stainless steel for stent construction in tubular body parts, ceramics for dental implants and polymeric materials for blood vessels or soft tissue. Furthermore, great advances have been brought to modern society by materials which are used in energy production, construction, transportation, electronic products, etc.

Therefore, many scientists and engineers are involved in developing new materials through various kinds of research. Since the first microscopy studies of material structures were performed about 100 years ago, succeeding materials research has built a foundation of materials science and engineering discipline [1]. In developing new materials, it is important to understand the microstructures of materials and to relate them to their properties. From the microstructure, relationships between process and structure can also be determined. This process-structure-property relationship is very useful in computational material design.

The quantification of microstructures of materials is called characterization. For many years, the use of microscopy has been essential for the characterization of materials. Now, improvement of microscopy techniques enhances the precision of instrumentation to the extent that we can observe materials systems at an atomic scale. Nowadays, microscopes can capture 3D data with techniques such as automated serial sectioning [2, 3]. Moreover, some imaging techniques such as x-ray tomography can take 3D data evolving over time without destroying the material specimen [4]. Accordingly, the amount of available microscopy data has been growing exponentially, so that image processing techniques have become essential in analyzing this enormous quantity of microscopy data.

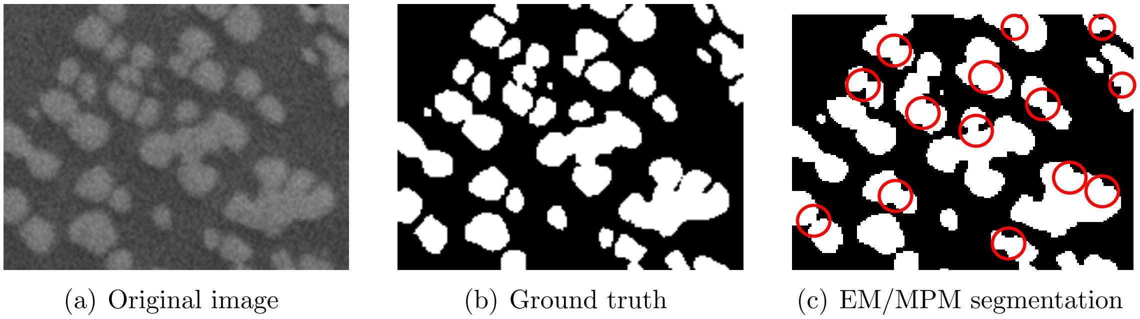
It is a difficult task to develop image processing methods for materials analysis and design because of the random nature of materials microstructures and noise and distortion created by image acquisition devices. Users of image acquisition equipment often need to develop their own unique image processing procedures [5]. However, Duval et al. [6] summarized a relatively standard workflow for image processing in material science: image acquisition, reconstruction, enhancement and filtering, segmentation, analysis, multi-modality, and microstructure stochastic modeling.

In the early step of materials image analysis, the image segmentation is essential. Segmentation is the process of partitioning an image into multiple regions that are homogeneous in some sense. The next image analysis step is detecting microstructure features. These features are useful to contruct stocastic microstructure models of materials. They can be also used to get more precise boundaries in the segmentation step [7,8].

1.1 Segmentation and Detection of Microstructural Features in Microscope Images of Materials

In a material characterization, various kinds of image acquisition modalities are used, such as light microscopy, x-ray diffraction, transmission electron microscopy (TEM), scanning electron microscopy (SEM), scanning probe microscopy [9]. Before performing image analysis, methods such as x-ray tomography require reconstruction from raw measured data to image data. Sometimes, the acquired images might have degradation problems such as blurring, shading(irregular illumination) and noise(random or structural). Most of these artifacts come from the nature of image acquisition methods.

In the segmentation of materials images, it is important to localize exact region boundaries. However, if the images have degradation problems such as blurring or low contrast, a precise boundary localization becomes a complex task. For the input materials images with blurring degradation, conventional segmentation methods give results which show the swelling of objects. This causes some neighboring objects to merge together and forms bridged channels. Figure 1.1 shows an example of this segmentation problem. When we compare a segmentation result (Figure 1.1(c)) with a ground truth (Figure 1.1(b)), we can see many bridged channels marked with red circles. In this example, a Markov random field(MRF) is used as a prior model for segmentation. These bridged channel defects are partly originated from the nature of the MRF prior model and from the blurring artifact in the original image acquired from SEM. This blurring is quite usual in image acquisition devices. In the image acquisition process of the SEM, a high-energy electron beam is emitted from an electron gun and interacts with the material specimen, and then the reflected secondary electrons or back-scattered electrons are detected. This process of emission and detection scans the whole surface of a specimen and constructs a microscope image. During the electron interaction, electrons travel around the surface or the inside of the material specimen. When electrons come out from the surface, they bring information from around the initial entry position of the emitted electrons. This causes blurring in an SEM image. To tackle the blurring problem, we model this blurring in microscopy images and propose a joint deconvolution and segmentation (JDS) method which in-



(a) Original image (b) Ground truth (c) EM/MPM segmentation

Fig. 1.1. Merging problem of segmentation of material image

corporates the blurring model into an existing segmentation method. In addition, we also propose channel detection methods to reduce the smoothness effect of an MRF prior model in the channel area which partly causes the bridged channel defect.

1.2 Detection of 3D Microstructure Features

Estimating the relationship between the 3D structure and properties of materials is one of the most important tasks in materials development. To predict microstructure-property relationships in computational modeling of materials, the ability to generate digital 3D microstructures is required. Related to this requirement, Groeber describes two microstructure representation approaches [10]: an explicit representation by experiments and a statistical representation.

The former approach is a deterministic 3D representation of a microstructure. First, features are segmented from microscope images of an observed material. Then, the surface meshes of these features are constructed. Usually, such geometrical features are contours of a grain boundary, simple geometric shapes or tessellations.

In the latter approach, once a microstructure is identified from a microscope image of a material specimen, with this limited amount of information, we can estimate a 3D stochastic and geometric model which represents the microstructure of the material. From this stochastic model, a synthetic 3D microstructure can be generated. During this step, structure variations might be obtained by changing model parameters. After that, material properties of interest are estimated by simulation with this 3D microstructure and the information about compositions of the material. To construct a stochastic model, the representation of a microstructure should be able to specify such statistical quantities in the material volume as [5]: the number of structural objects, the size and shape of the object, and the form of their distribution.

Traditionally, statistical modeling has been done using first and second order statistics and univariate distributions. However, it is more effective to use advanced models from probability theory, such as a Markov rnadom field (MRF) or marked point process (MPP). In an MPP, a mark carries the parameters describing the shape and the size of each object of interest, and a point process conveys the information about the random number and spatial locations of the objects. Since the MPP model was introduced in the field of signal and image processing by Baddeley and Van Lieshout [11], it has been widely used as a methodology for the extraction of multiple objects from images, such as simple circular objects [12], building outlines [13], tree crowns [14, 15], road networks [16, 17], people in surveillance videos [18], superellipse objects [8] and channel features in materials image [7]. As we can see, most of these applications are aimed at detection of 2D objects in images. However, the ultimate goal of the material characterization is identifying 3D structure. We need to expand this application area of MPP method to the detection of 3D objects, although this is not an easy task due to its high computational cost.

1.3 Dissertation Organization

In Chapter 2, we propose a joint deconvolution and segmentation(JDS) method and a minimum area increment(MAI) constraint in the EM/MPM algorithm to improve the segmentation of microscope images of materials. We incorporate the new scheme of adaping the value of the spatial interaction parameter, so that segmentation results at object boundaries are improved. Experimental results are presented to demonstrate the proposed segmentation method and prior model.

In Chapter 3, we present a new approach for channel identification based on a MPP framework to effectively detect channels in materials images. In a materials image, a channel is a narrow structure between two objects. We propose two methods of the channel modeling, where a channel object and a related Gibbs energy are defined. The RJMCMC algorithm embedded with a simulated annealing is used as an optimization method, and the switching kernel in an RJMCMC is newly designed to decrease execution time. The new channel model is useful for analyzing materials images. Moreover, we also propose a method for exploiting detected channel configurations to reduce bridged channel defects in the segmentation result by adaptively controlling the interaction parameter. Experimental results demonstrate that the proposed channel modeling methods are successful to detect channels. The results also show that the proposed interaction parameter control method can be used to improve boundary precision in the segmentation of microscopic images of materials.

In Chapter 4, we propose a fast 3D fitting method which improves computational complexity over a full 3D MPP method in detecting object configurations in 3D datasets. This method detects 2D object configurations from 2D image slices first. Then, a clustering step clusters all the 2D objects that are expected to belong to the same 3D object. Conventionally, 3D objects can be estimated by fitting the predefined 3D object model to these clustered 2D objects. However, the fast 3D fitting method estimates a 3D object configuration from the original 3D dataset through a 3D MPP method. The fast 3D fitting method employs the information of the clustered 2D object configuration detected in the previous step to reduce computation time over full-search 3D detection. On the other hand, one advantage of this method over the 3D clustering method is that the proposed fast 3D fitting method is more robust for the errors in a 2D object detection and clustering step. Another advantage of the fast 3D fitting method is that its estimated object parameters are more precise than those of the 3D clustering method.

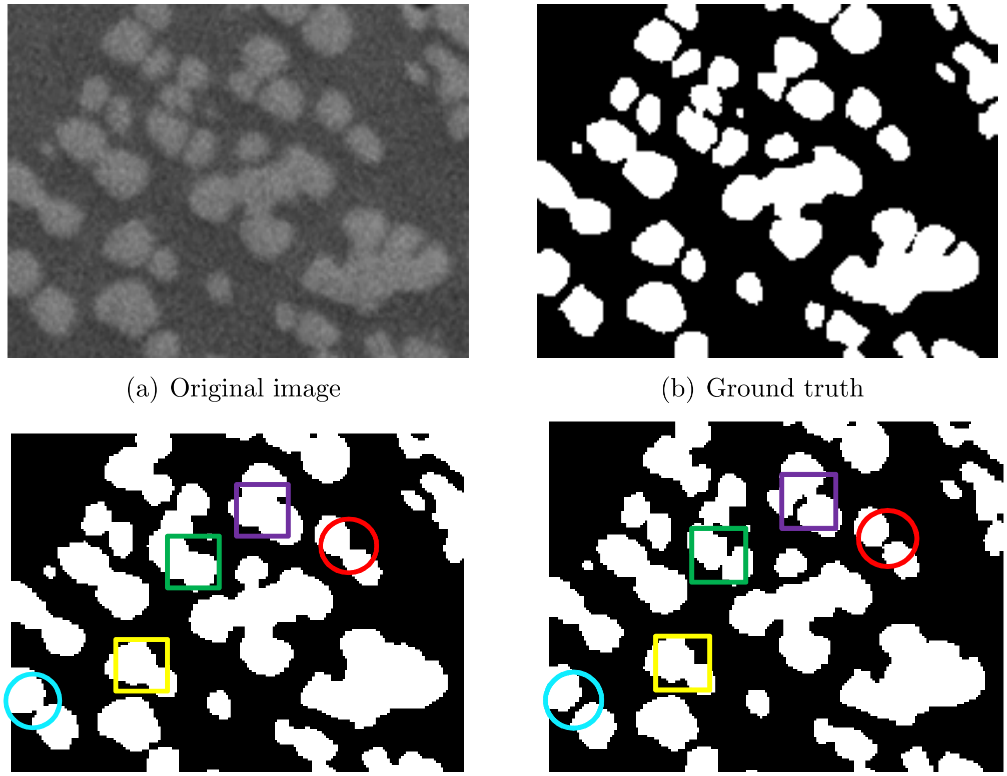
2. JOINT DECONVOLUTION/SEGMENTATION OF MICROSCOPE IMAGES OF MATERIALS

2.1 Introduction

Due to the complex electron interactions with a specimen, it is common to see the blurring artifacts in the SEM images during the image acquisition processes. For the input material images with this blurring degradation, the conventional EM/MPM segmentation method [19] shows the swelling of objects in its results. This swelling makes some objects to merge together, which forms small bridged channels in the segmented result. To improve these problems, we propose the joint deconvolution and segmentation(JDS) method which incorporates the obtained information about blurring degradation into the existing EM/MPM method. Figure 2.1 helps us to compare the results between the conventional EM/MPM segmentation and the JDS method. Focusing on the inside regions of the circles in Figure 2.1(c) and of Figure 2.1(d), we can observe that the JDS method makes the merged objects to be shrunk and disconnected from each other. As a result, the JDS method shows a clearer segmentation result than the conventional EM/MPM segmentation. However, as we see the inside of rectangles, some bridged channels still remain unresolved.

To improve these remained bridged channel defects, we propose a minimum area increment(MAI) constraint and apply it to the existing EM/MPM algorithm. To make this constraint more effective, the simulated annealing(SA) scheme is used to control the interaction parameters of the MRF model. We will call this as the MAI/SA method. Experimental results demonstrate the proposed JDS method and the MAI/SA method are effective for removing bridged channel defects.

In Section 2.2, we will review the imaging mode of the SEM device and the innate issue of blurring degradation occurring in the image acquisition process. In Section



(a) Original image

(b) Ground truth

(c) Original EM/MPM segmentation

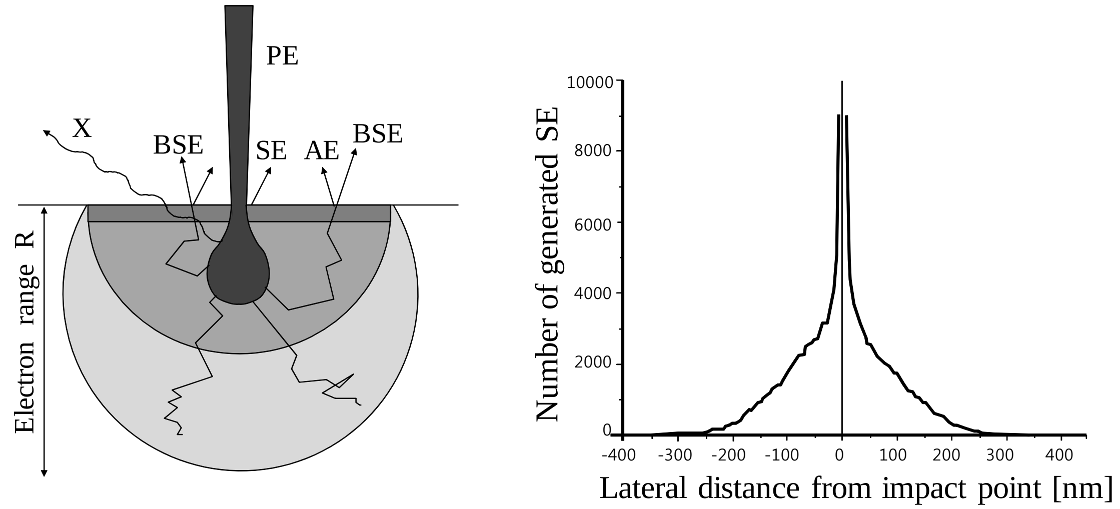
(d) Joint deconvolution and segmentation method

Fig. 2.1. Merging problem of segmentation of material image

2.3, we will go through the EM/MPM framework including the image model, the prior model and the EM/MPM algorithm. In Section 2.4, the JDS EM/MPM algorithm will be explained in detail and a new blurring image model will be proposed. And, in Section 2.5, the MAI/SA method will be proposed to improve the JDS EM/MPM algorithm. In Section 2.6, our proposed modified EM/MPM algorithm will be examined through several experiments. The results will be shown in detail.

2.2 SEM Image Acquisition

The sequence of SEM images are acquired by the ion milling and the imaging process alternately. The ion milling process mills the surface of the material specimen and the imaging process captures the newly produced surface. A high-energy electron beam emitted from an electron gun interacts with the surface of the materials specimen during the imaging process. At the same time, some electrons come out from the surface. These reflected electrons are detected with the built-in detectors of the SEM. Figure 2.2(a) is the cross-sectional diagram which shows the trajectory of each kind of electron when they are injected into the specimen. The SEM imaging process has several imaging modes according to the energy bend of the electrons. Among those imaging modes, the secondary electron(SE) imaging mode and the backscattered electron(BSE) imaging mode are two major modes. The energy of the SE is low so that they travel only the surface of specimen and come out while the energy of the BSE is big enough to go deep inside of the specimen. Therefore, in the SE imaging mode, blurring degradation can be modeled with a 2D blurring filter. The filter coefficients can be obtained from the distribution of the lateral number of the generated SE on the surface which can be modeled as an exponential [21].



(a) Diffusion cloud of electron range R for normal incidence of the primary electron (PE). [20]

(b) The lateral distribution of generated SE(Monte Carlo simulation, silicon, 5kV) [21]

Fig. 2.2. Interaction between injected electrons with a specimen

$$N_{SE}(x) = A\exp[-x/w] \tag{2.1}$$

where N_{SE} is the number of the generated SE, x is the distance from the impact point, and ω is the width of the exponential. For the BSE imaging mode, blurring can be modeled with a 3D blurring filter [22].

$$h(x, y, z) = A \exp\left(-\frac{\sqrt{x^2 + y^2 + (z/\delta)^2}}{\omega}\right) \tag{2.2}$$

where x and y are the horizontal and vertical distance from the impact point on the surface, z is the depth from the surface, ω is the width of the distribution, and δ is the parameter to control the information depth.

2.3 The EM/MPM Framework

The EM/MPM algorithm was proposed to address the problem of segmenting a textured image [19]. A multidimensional field, which has the same spatial resolution as the observed image, is generated during the segmentation. Its individual pixel value at a given location reflects the texture around the corresponding location in the observed image. This individual pixel classifications are called labels and a multidimensional field comprised of these labels is called a label field.

The EM/MPM follows the Bayesian approach, in which the label field and the observed image are modeled as random fields. And the EM/MPM segmentation becomes a statistical estimation problem. For the label field, a Markov random field(MRF) model is used. The MRF model imposes the spatial constraint that neighboring pixels are likely to be of the same class. For the observed image model, a Gaussian random field is used. As an estimation criterion, the minimization of the expected value of the number of misclassified pixels is used. The estimate which optimizes this criterion is known as the "maximizer of the posterior marginals" (MPM) estimate. However, in order to use the MPM algorithm, we need to know the values of all parameters of the image models. To address this problem, the EM/MPM algorithm which combines the MPM algorithm for segmentation with the EM algorithm for the parameter estimation was proposed [19].

2.3.1 Image Model

Throughout this chapter, $\mathbf{X}$ represents the label field and $\mathbf{Y}$ represents the observed image. $\mathbf{X}$ and $\mathbf{Y}$ are defined on rectangular lattice S. The elements in $\mathbf{X}$ are denoted as the random variables X_s and the elements in $\mathbf{Y}$ are denoted as the random variables Y_s at the spatial location $s \in S$. In this chapter, $\mathbf{x} = [x_1, x_2, \cdots, x_N]^{\mathrm{T}}$ will represent the sample realization of $\mathbf{X} = [X_1, X_2, \cdots, X_N]^{\mathrm{T}}$, where N is the total number of pixels in S. And $\mathbf{y} = [y_1, y_2, \cdots, y_N]^{\mathrm{T}}$ will represent the sample realization of $\mathbf{Y} = [Y_1, Y_2, \cdots, Y_N]^{\mathrm{T}}$, where the pixel of the vector $\mathbf{x}$ and $\mathbf{y}$ are a raster scan order as below.

$$\begin{aligned}
\mathbf{x} &= [x_1, x_2, \cdots, x_r, \cdots, x_N]^{\mathrm{T}} \\
&= [x_{(1,1)}, \cdots, x_{(1,M)}, x_{(2,1)}, \cdots, x_{(2,M)}, \\
&\quad \cdots, x_{(r_1,r_2)}, \cdots, x_{(M,1)}, \cdots, x_{(M,M)}]^{\mathrm{T}}
\end{aligned} \tag{2.3}$$

where the set of all rectangular lattice point S be $[1, \cdots, M]^2$ and $r = (r_1, r_2)$ is an ordered pair with each coordinate taking on values in the range of 1 to M and $N = M^2$.

To impose a smoothness constraint on the segmentation result, the MRF model is used as a prior model. In this chapter, a 8-neighborhood system $\mathcal{N}$ is used to improve the smoothness of object boundary shape. A random variable $\mathbf{X}$ is said to be a Gibbs random fields(GRF), which is the same as the MRF according to the Hammersley-Clifford theorem [23], on S w.r.t. $\mathcal{N}$ if and only if its distribution takes the form

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{z} \exp\left(-\frac{1}{T} U(\mathbf{x})\right) \tag{2.4}$$

where a normalizing constant $Z = \sum_{\mathbf{x}\in\mathbf{X}} \exp -\frac{1}{T}U(\mathbf{x})$ is called the partition function. A constant T is called the temperature, which is assumed to be 1 usually. And a function $U(\mathbf{x})$ is called the energy function, which is the sum of clique potentials $V_c(x)$. This energy function can be written as

$$\begin{aligned} U(\mathbf{x}) = &\sum_{\{r\}\in\mathcal{C}_1} V_1(r, x_r) + \sum_{\{r,s\}\in\mathcal{C}_2} V_2(r, s, x_r, x_s) \\ &+ \sum_{\{r,s,t\}\in\mathcal{C}_3} V_3(r, s, t, x_r, x_s, x_t) + \cdots . \end{aligned} \tag{2.5}$$

In this chapter, we assume that this energy function has only the second-order energy which is the special case when only the cliques of size up to two are considered. And we want to divide the pairwise clique set $\mathcal{C}_2$ into $\mathcal{C}_{HV}$ for horizontal and vertical cliques and $\mathcal{C}_D$ for diagonal cliques. Let $V_1(r, x_r) = \gamma_{x_r}$ and $V_2(r, s, x_r, x_s) = \beta_{x_r,x_s} t(x_r, x_s)$ when $r, s \in \mathcal{C}_{HV}$. And let $V_2(r, s, x_r, x_s) = \beta'_{x_r,x_s} t(x_r, x_s)$ when $r, s \in \mathcal{C}_D$. Then

$$U(\mathbf{x}) = \sum_{\{r,s\}\in\mathcal{C}_{HV}} \beta_{x_r,x_s} t(x_r, x_s) + \sum_{\{r,s\}\in\mathcal{C}_D} \beta'_{x_r,x_s} t(x_r, x_s) + \sum_{\{r\}\in\mathcal{C}_1} \gamma_{x_r}. \tag{2.6}$$

where

$$t(x_r, x_s) = \begin{cases} 0, & \text{if} \quad x_r = x_s \\ 1, & \text{if} \quad x_r \neq x_s \end{cases} \tag{2.7}$$

. Let $\beta_{x_r,x_s} = \beta'_{x_r,x_s}/\sqrt{2}$, considering the difference between $\text{dist}(r, s)|_{(r,s)\in\mathcal{C}_{HV}}$ and $\text{dist}(r, s)|_{(r,s)\in\mathcal{C}_D}$. And let β_{x_r,x_s} depend on only x_r. Then

$$U(\mathbf{x}) = \sum_{r\in S} \beta_{x_r} \Big(\sum_{s\in\mathcal{N}_{HV}} t(x_r, x_s) + \frac{1}{\sqrt{2}} \sum_{s\in\mathcal{N}_D} t(x_r, x_s) \Big) + \sum_{\{r\}\in\mathcal{C}_1} \gamma_{x_r}, \tag{2.8}$$

,where $\mathcal{N}_{HV}$ is a horizontal and vertical neighbor and $\mathcal{N}_D$ is a diagonal neighbor. The parameter $\{\beta_k\}$ is a set of spatial interaction parameters for pairwise cliques, and $\{\gamma_k\}$ is a set of model parameters for single-pixel cliques. If γ_k is high, a class k is less likely to occur than classes with lower values. This parameters $\{\gamma_k\}$ can be used for the applications if the relative sizes of the various classes are known. If there is no such information, γ_k is assumed to be zero. More detail of a MRF model is discussed in [24, 25].

For the observation image model, we assume that the grayscale value of each pixel in the observed image is the realization of a Gaussian random variable as below

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = \prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{x_r}^2}} \exp\left(-\frac{(y_r-\mu_{x_r})^2}{2\sigma_{x_r}^2}\right). \tag{2.9}$$

2.3.2 EM/MPM Algorithm

The EM/MPM algorithm alternately performs the MPM segmentation algorithm and the EM parameter estimation algorithm [19]. For the MPM algorithm, the segmentation problem is an optimization problem in which the minimization of the expected number of the misclassified pixels is used as an optimization criterion. This is equivalent to maximizing $P(X_s = k|\mathbf{Y} = y)$ over all $k \in \{1, 2, \cdots, L\}$, for every $s \in S$. But the exact computation of this probability is computationally infeasible. Marroquin et al. [26] proposed an algorithm for approximating these marginal probabilities. Suppose we have an initial segmentation $\hat{X}(0)$ and an initial parameter estimate $\hat{\theta}(0)$. The generated Markov Chain from the Gibbs sampler visits all pixels in a raster scan order to produce a set of estimates of the marginals of the posterior pmf given as (2.10). After that, these estimates of the marginals are used to find the parameter estimates. This alternating process continues until a certain stopping criterion has been met.

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) = \frac{1}{zf_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})}\left[\prod_{r=1}^{N}\frac{1}{\sqrt{2\pi\sigma_{x_r}^2}}\right]\cdot\exp\left(-\sum_{r=1}^{N}\frac{(y_r-\mu_{x_r})^2}{2\sigma_{x_r}^2} - U(\mathbf{x})\right) \tag{2.10}$$

2.4 Joint Deconvolution/Segmentation Method

2.4.1 Observed Image Model with 2D Blurring Information

We can make the blurring matrix $\mathbf{H}$ with a window size $(2W+1) \times (2W+1)$ be a vector $\mathbf{h}$ with a raster scan order, so that

$$\begin{aligned}\mathbf{h} &= [h_{-D}, h_{-D+1}, \cdots, h_s, \cdots, h_{D-1}, h_D]^{\mathrm{T}} \\ &= [h_{(-W,-W)}, h_{(-W,-W+1)} \cdots, h_{(-W,W)}, \\ &\quad h_{(-W+1,-W)}, h_{(-W+1,-W+1)}, \cdots, h_{(-W+1,W)}, \\ &\quad \cdots, h_{(s_1,s_2)}, \cdots, \\ &\quad h_{(W,-W)}, h_{(W,-W+1)}, \cdots, h_{(W,W)}]^{\mathrm{T}}\end{aligned} \tag{2.11}$$

where $D = ((2W+1)^2 - 1)/2$.

If we write the conditional probability mass function for the image model as

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2(r)}} \exp\left(-\frac{(y_r - \mu(r))^2}{2\sigma^2(r)}\right), \tag{2.12}$$

in the original EM/MPM segmentation method, we can express that

$$\mu(r) = \mu_{x_r} \quad \text{and} \quad \sigma^2(r) = \sigma^2_{x_r}. \tag{2.13}$$

However, to improve the segmentation result of the image which has blurring degradation, we would like to use the information of the blurring which the input image $\mathbf{y}$ has.

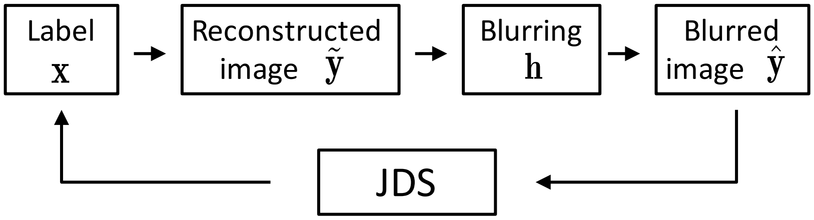

Label $\mathbf{x}$
Reconstructed image $\tilde{\mathbf{y}}$
Blurring $\mathbf{h}$
Blurred image $\hat{\mathbf{y}}$
JDS

Fig. 2.3. Joint deconvolution/segmentation algorithm

Figure 2.3 shows the concept of the joint deconvolution/segmentation(JDS) method. Where $\mathbf{x}$ is the result of the JDS method, $\tilde{\mathbf{y}}$ is the reconstructed image from $\mathbf{x}$. And $\hat{\mathbf{y}}$ is the observed image which has blurring degradation caused by blurring filter $\mathbf{h}$. Let $\hat{y}_r$ be the pixel of $\hat{\mathbf{y}}$ and $\tilde{y}_r$ be the pixel of $\tilde{\mathbf{y}}$ at location $r \in S$. Then, $\hat{y}_r$ can be expressed as

$$\hat{y}_r = \sum_{s=-D}^{D} h_s \tilde{y}_{r-s} \tag{2.14}$$

where $r - s$ means $(r_1 - s_1, r_2 - s_2)$. In addition, we can express $\mu(r)$ and $\sigma^2(r)$ as

$$\mu(r) = E[\hat{y}_r] \quad \text{and} \quad \sigma^2(r) = Var[\hat{y}_r]. \tag{2.15}$$

If we reconstruct $\tilde{\mathbf{y}}$ as $\tilde{y}_r = \mu_{x_r}$ for all $r \in S$, then

$$\begin{aligned} \mu(r) &= E\left[\sum_{s=-D}^{D} h_s \tilde{y}_{r-s}\right] \\ &= \sum_{s=-D}^{D} h_s E[\tilde{y}_{r-s}] \\ &= \sum_{s=-D}^{D} h_s \mu_{x_{r-s}} \end{aligned} \tag{2.16}$$

and

$$\begin{aligned} \sigma^2(r) &= Var\left[\sum_{s=-D}^{D} h_s \tilde{y}_{r-s}\right] \\ &= \sum_{i=-D}^{D} \sum_{j=-D}^{D} h_i h_j \sigma(\tilde{y}_{r-i}, \tilde{y}_{r-j}). \end{aligned} \tag{2.17}$$

To make the calculation simple, we assume the covariance terms are zero. Then, (2.17) can be rewritten as

$$\sigma^2(r) = \sum_{s=-D}^{D} h_s^2 \sigma_{x_{r-s}}^2. \tag{2.18}$$

However, this multi variance form also needs much computation to get the estimates of variances $\{\sigma_k\}$. So, if the computational cost is important for some applications, we can use the single variance form (2.13) instead of using (2.18). Using (2.16) and (2.18), we propose the modified observed image model as

$$\begin{aligned} f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = \prod_{r=1}^{N} & \frac{1}{\sqrt{2\pi \sum_{s=-D}^{D} h_s^2 \sigma_{x_{r-s}}^2}} \\ & \cdot \exp\left(- \frac{\left(y_r - \sum_{s=-D}^{D} h_s \mu_{x_{r-s}}\right)^2}{2\sum_{s=-D}^{D} h_s^2 \sigma_{x_{r-s}}^2} \right) \end{aligned} \tag{2.19}$$

Finally, we propose a new conditional probability mass function of $\mathbf{X}$ given $\mathbf{Y}$ by using this 2D blurring observed image model. By applying (2.19) to (2.10), we can get a new pmf as below:

$$\begin{aligned} p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) = & \frac{1}{z f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \left[\prod_{r=1}^{N} \frac{1}{\sqrt{2\pi \sum_{s=-D}^{D} h_s^2 \sigma_{x_{r-s}}^2}} \right] \\ & \cdot \exp\left(- \sum_{r=1}^{N} \left(\frac{\left(y_r - \sum_{s=-D}^{D} h_s \mu_{x_{r-s}}\right)^2}{2\sum_{s=-D}^{D} h_s^2 \sigma_{x_{r-s}}^2} \right) - U(\mathbf{x}) \right) \end{aligned} \tag{2.20}$$

2.4.2 EM Algorithm for the Proposed Conditional pmf

To find the MPM estimate of X, it is necessary to find the value of k which maximizes $P(X_s = k|\mathbf{Y} = y)$ for each $s \in S$. The exact computation of this marginal probability mass function is computationally infeasible. A Gibbs sampler is used to approximate these probabilities. At each EM iteration, the expectation step and the maximization step are performed. Let $\boldsymbol{\theta}(p)$ be the estimate of $\boldsymbol{\theta}$ at the pth iteration. Then, in the expectation step at iteration p, the function

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) =& E[\log f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)] \\ & + E[\log p_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)] \end{aligned} \tag{2.21}$$

is calculated. After that, the estimate $\boldsymbol{\theta}(p)$ is obtained in the maximization step which maximizes the value of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1))$.

For the means $\mu_k(p)$, by substituting our 2D blurring observed image model and MRF model into (2.21), differentiating and setting to zero(refer to Appendix A), we can get the number of the L linear equations (2.24) from which we can obtain all parameters $\mu_k(p)$, where

$$\begin{aligned} v(l_1, l_2, \cdots, l_{D+1}, \cdots, l_{2D+1}) = \\ h_{-D}^2 \sigma_{l_1}^2 + h_{-D+1}^2 \sigma_{l_2}^2 + \cdots + h_D^2 \sigma_{l_{2D+1}}^2 \end{aligned} \tag{2.22}$$

For the variances $\sigma_k(p)$, if we use (2.18) as a variance term, we cannot get the closed form solution. We have to use the optimization method to find the optimal estimation. Details will be discussed in Section 2.4.4. However, if we just use the single variance form (2.13), we can get the closed form solution (2.23) of $\sigma_k^2(p)$.

$$
\begin{aligned}
N_k &= \sum_{r=1}^{N} p_{X_r|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p-1)) \\
\sigma_k^2 &= \frac{1}{N_k}\Bigg(\sum_{r=1}^{N}\sum_{l_1=1}^{L}\cdots\sum_{l_D=1}^{L}\sum_{l_{D+2}=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L}(y_r-(h_{-D}\mu_{l_1}+\cdots+h_{-1}\mu_{l_D}+h_0\mu_k+h_1\mu_{l_{D+2}}+\cdots+h_D\mu_{l_{2D+1}}))^2 \\
&\qquad \cdot p_{X_{r+D},\cdots,X_r,\cdots,X_{r-D}|\mathbf{Y}}(x_{r+D}=l_1,\cdots,x_r=k,\cdots,x_{r-D}=l_{2D+1}|\mathbf{y},\boldsymbol{\theta}(p-1))\Bigg) \qquad (2.23)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}(p-1))}{\partial \mu_k} &= \frac{\partial}{\partial \mu_k}\sum_{r=1}^{N} E\Bigg[-\sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L}\delta(x_{r+D}=l_1)\delta(x_{r+D-1}=l_2)\cdots\delta(x_{r-D}=l_{2D+1}) \\
&\qquad \cdot\left(\frac{(y_r-(h_{-D}\mu_{l_1}+h_{-D+1}\mu_{l_2}+\cdots+h_D\mu_{l_{2D+1}}))^2}{2v(l_1,l_2,\cdots,l_{D+1},\cdots,l_{2D+1})}\right)\Bigg|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)\Bigg] \\
&= \frac{\partial}{\partial \mu_k}\sum_{r=1}^{N}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L}\Bigg(\frac{(y_r-(h_{-D}\mu_{l_1}+h_{-D+1}\mu_{l_2}+\cdots+h_D\mu_{l_{2D+1}}))^2}{2v(l_1,l_2,\cdots,l_{D+1},\cdots,l_{2D+1})} \\
&\qquad \cdot p_{X_{r+D},X_{r+D-1},\cdots,X_{r-D}|\mathbf{Y}}(x_{r+D}=l_1,x_{r+D-1}=l_2,\cdots,x_{r-D}=l_{2D+1}|\mathbf{y},\boldsymbol{\theta}(p-1))\Bigg) \\
&= a_{k,1}\mu_1+a_{k,2}\mu_2+\cdots+a_{k,L}\mu_L-b_k=0 \qquad (2.24)
\end{aligned}
$$

2.4.3 EM/MPM Algorithm for the Proposed Conditional pmf

In the EM/MPM algorithm, the MPM segmentation algorithm and the EM parameter estimation are performed at each iteration step p. During the MPM segmentation, the Markov chain $\mathbf{X}(p,t)$ is generated by using the Gibbs sampler. Then, we can get the approximations of $p_{X_r|\mathbf{Y}}(k|\mathbf{y},\boldsymbol{\theta}(p))$ and $p_{X_{r+D},\cdots,X_{r-D}|\mathbf{Y}}(l_1,\cdots,l_{2D+1}|\mathbf{y},\boldsymbol{\theta}(p))$ and the MPM estimate of $\mathbf{x}$. After that, these approximations of the marginal conditional probability mass functions are used to calculate the estimate $\hat{\boldsymbol{\theta}}(p)$ of $\boldsymbol{\theta}$ in the EM parameter estimation. Details of our EM/MPM algorithm are specified below.

1. Initialize $\boldsymbol{\Theta}(0)=\hat{\boldsymbol{\theta}}(0)$ for some $\hat{\boldsymbol{\theta}}(0)\in\Omega_\theta$, where the random variable $\boldsymbol{\Theta}(p)$ is the estimate of $\boldsymbol{\theta}$ at stage p. And set $p=1$.

2. Markov chain $\mathbf{X}(p,t)$ is generated, where $t=1,2,\cdots,T_p$. The state $X_{q_t}(p,t)$ of pixel q_t is determined by sampling from the conditional probability mass function

$$p_{X_{q_t}|\mathbf{Y},X_r,r\in\mathcal{G}_{q_t}\cup\tilde{\mathcal{G}}_{q_t}}(k|\mathbf{y},x_r(t-1),r\in\mathcal{G}_{q_t}\cup\tilde{\mathcal{G}}_{q_t},\boldsymbol{\theta}) \qquad (2.25)$$

   where $q_t=(q_{t,1},q_{t,2})\in S$ is the pixel visited at time t. And $\mathcal{G}_{q_t}$ is the neighbors of spatial location q_t and $\tilde{\mathcal{G}}_{q_t}=\{(q_1,q_2)|q_{t,1}-D\leq q_1\leq q_{t,1}+D, q_{t,2}-D\leq q_2\leq q_{t,2}+D,(q_1,q_2)\neq(q_{t,1},q_{t,2})\}$. And $p_{\mathbf{X}|\mathbf{Y},\boldsymbol{\Theta}(p)}(\mathbf{x}|\mathbf{y},\hat{\boldsymbol{\theta}}(p))$ has the same form of (2.23) except that $\boldsymbol{\Theta}(p)$ is used instead of $\boldsymbol{\theta}$.

3. $\boldsymbol{\Theta}(p)=[M_1(p),S_1(p),\cdots,M_L(p),S_L(p)]$ is computed by using the EM parameter estimation. Detailed explanation is in Section 2.4.2. But the conditional

probability mass functions in Section 2.4.2 are very difficult to obtain. Therefore, as an approximation of $p_{X_r|\mathbf{Y}}(k|\mathbf{y},\boldsymbol{\theta}(p-1))$, the $v_{k,r}(p,t)$ is used where

$$
\begin{aligned}
v_{k,r}(p,t) &= \frac{1}{t}\sum_{i=1}^{t} u_{k,r}(p,i), \\
u_{k,r}(p,t) &= \begin{cases} 1, & \text{if } x_r(p,t) = k \\ 0, & \text{if } x_r(p,t) \neq k, \end{cases} .
\end{aligned} \tag{2.26}
$$

$v_{l_1 l_2 \cdots l_{2D+1},r}(p,t)$ is used as an approximation of $p_{X_{r+D},\cdots,X_{r-D}|\mathbf{Y}}(l_1,\cdots,l_{2D+1}|\mathbf{y},\boldsymbol{\theta}(p-1))$.

$$
\begin{aligned}
v_{l_1\cdots l_{2D+1},r}(p,t) &= \frac{1}{t}\sum_{i=1}^{t} u_{l_1\cdots l_{2D+1},r}(p,i), \\
u_{l_1\cdots l_{2D+1},r}(p,t) &= \begin{cases} 1, & \text{if } x_{r+D}(p,t) = l_1,\cdots \\ & ,x_{r-D}(p,t) = l_{2D+1} \\ 0, & \text{else} \end{cases}
\end{aligned} \tag{2.27}
$$

4. Set $p = p + 1$ and repeat from step 2 through step 4 until a stopping criterion is met.

2.4.4 Optimization Method of Finding Variances

From (A.3), using (2.27), we can get

$$
\begin{aligned}
\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}(p-1))}{\partial \sigma_k} &= \frac{\partial}{\partial \sigma_k}\frac{1}{t}\sum_{i=1}^{t}\sum_{r=1}^{N}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L} \\
&\Bigg(-\frac{(y_r-(h_{-D}\mu_{l_1}+\cdots+h_D\mu_{l_{2D+1}}))^2}{2v(l_1,\cdots,l_{2D+1})} \\
&-\frac{1}{2}\log v(l_1,\cdots,l_{2D+1})\Bigg)\cdot u_{l_1\cdots l_{2D+1},r}(p-1,t)
\end{aligned} \tag{2.28}
$$

Let

$$m(i,r) = h_{-D}\mu_{l_1} + \cdots + h_D \mu_{l_{2D+1}} \tag{2.29}$$

and

$$\mathbf{v}(p) = [\sigma_1^2(p), \sigma_2^2(p), \cdots \sigma_L^2(p)]^{\mathrm{T}}. \tag{2.30}$$

Then from (2.22),

$$v(l_1, \cdots, l_{2D+1}) = \mathbf{k}(i,r)^{\mathrm{T}} \cdot \mathbf{v} \tag{2.31}$$

where the kth component of $\mathbf{k}(i,r)$ is $\sum_{m=-D}^{D} h_m^2 u_{k,r+m}(p-1,i)$. Finding estimation $\hat{\mathbf{v}}$ is the optimization problem as below:

$$\begin{aligned}
\hat{\mathbf{v}} &= \arg\max_{\mathbf{v}} Q(\boldsymbol{\theta}(\mathbf{v}, \cdots), \boldsymbol{\theta}(p-1)) \\
&= \arg\min_{\mathbf{v}} \frac{1}{t} \sum_{i=1}^{t} \sum_{r=1}^{N} \left(\frac{(y_r - m(i,r))^2}{2\mathbf{k}(i,r)^{\mathrm{T}} \cdot \mathbf{v}} + \frac{1}{2} \log \left(\mathbf{k}(i,r)^{\mathrm{T}} \cdot \mathbf{v} \right) \right) \\
&= \arg\min_{\mathbf{v}} R(\mathbf{v})
\end{aligned} \tag{2.32}$$

To find this estimation, we use Conjugate Gradient Algorithm(CGA) [27] as Algorithm 1.

2.4.5 3D Blurring Observed Image Model

First, we define $\mathbf{y}_n$ as the observed n-th image vector in a stack of images and $\mathbf{x}_n$ as the label vector of the n-th slice. Let the set of all lattice point S be $[1, \cdots, M]^2$ and the order of the pixel of x_n and y_n be raster scan order as below.

$$\begin{aligned}
\mathrm{x}_n &= [x_{n,1}, x_{n,2}, \cdots, x_{n,r}, \cdots, x_{n,N}] \\
&= [x_{n,(1,1)}, \cdots, x_{n,(1,M)}, x_{n,(2,1)}, \cdots, x_{n,(2,M)}, \\
&\quad \cdots, x_{n,(r_1,r_2)}, \cdots, x_{n,(M,1)}, \cdots, x_{n,(M,M)}]^{\mathrm{T}}
\end{aligned} \tag{2.34}$$

Algorithm 1 Algorithm for finding $\hat{\mathbf{v}}$

```
Initialization: Set $n \Leftarrow 0$ and select the initial point $\mathbf{v}_0$
g_0: $\mathbf{g}_0 \Leftarrow \nabla R(\mathbf{v}_0)$
if $\mathbf{g}_0 = 0$ then
    stop
else
    set $\mathbf{d}_0 \Leftarrow -\mathbf{g}_0$
end if
repeat
    Find $\alpha_n = \arg\min_{\alpha \geq 0} R(\mathbf{v}_n + \alpha \mathbf{d}_n)$ using Newton's algorithm
    $\mathbf{v}_{n+1} \Leftarrow \mathbf{v}_n + \alpha_n \mathbf{d}_n$
    $\mathbf{g}_{n+1} \Leftarrow \nabla R(\mathbf{v}_{n+1})$
    if $\mathbf{g}_{n+1} = 0$ then
        stop
    else
        Use Fletcher-Reeves formula.
            $\delta_n \Leftarrow \frac{\mathbf{g}_{n+1}^{\mathrm{T}} \mathbf{g}_{n+1}}{\mathbf{g}_n^{\mathrm{T}} \mathbf{g}_n}$    (2.33)
    end if
    $\mathbf{d}_{n+1} \Leftarrow -\mathbf{g}_{n+1} + \delta_n \mathbf{d}_n$
    $n \Leftarrow n + 1$
until $n \geq n_{max}$
```

$$\begin{aligned}\mathrm{y}_n &= [y_{n,1}, y_{n,2}, \cdots, y_{n,r}, \cdots, y_{n,N}]\\ &= [y_{n,(1,1)}, \cdots, y_{n,(1,M)}, y_{n,(2,1)}, \cdots, y_{n,(2,M)},\\ &\quad \cdots, y_{n,(r_1,r_2)}, \cdots, y_{n,(M,1)}, \cdots, y_{n,(M,M)}]^{\mathrm{T}}\end{aligned} \tag{2.35}$$

And let $\mathbf{h}_{3D}$ be the blurring 3D filter vector defined as (2.36), where $h_{3D}(s_1, s_2, m)$ is the filter coefficients of the 3D blurring filter which can be obtained from (2.2). And the size of the 3D blurring filter is $(2W+1) \times (2W+1) \times (2W+1)$

$$\begin{aligned}\mathbf{h}_m &= [h_{m,-D}, h_{m,-D+1}, \ldots, h_{m,s}, \cdots, h_{m,D-1}, h_{m,D}]\\ &= [h_{3D}(-W,-W,m), h_{3D}(-W,-W+1,m), \ldots,\\ &\quad h_{3D}(s_1,s_2,m), \cdots, h_{3D}(W,W-1,m), h_{3D}(W,W,m)]\\ \mathbf{h}_{3D} &= [\mathbf{h}_{-D}, \mathbf{h}_{-D+1}, \ldots, \mathbf{h}_m, \cdots, \mathbf{h}_{D-1}, \mathbf{h}_D]\end{aligned} \tag{2.36}$$

where $D = ((2W+1)2-1)/2$. And $s = (s_1, s_2)$ is an ordering index of the points $(x, y, z = m)$ with raster scan ordering. By using (2.34),(2.35) and (2.36), we can expand the 2D blurring observed image model (2.19) and its pmf (2.20) to the 3D blurring observed image model and its relevant pmf with (2.37) and (2.38) respectively if we use the single variance form where $\tilde{\mathbf{x}}_{n-1}, \cdots, \tilde{\mathbf{x}}_{n-W}$ are the already segmented label vectors of the previous slices.

$$\begin{aligned}f_{\mathbf{Y}_n|\mathbf{X}_n}(\mathbf{y}_n|\mathbf{x}_n, \boldsymbol{\theta}_n, \tilde{\mathbf{x}}_{n-1}, \cdots, \tilde{\mathbf{x}}_{n-W}) &= \prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2_{n,x_{n,r}}}}\\ \exp\left(-\frac{(y_{n,r} - \sum_{s=-D}^{D} h_{0,s}\mu_{n,x_{n,r-s}} - \sum_{m=1}^{W}\sum_{s=-D}^{D} h_{m,s}\mu_{n,\tilde{x}_{n-m,r-s}})^2}{2\sigma^2_{n,x_{n,r}}}\right)&\end{aligned} \tag{2.37}$$

$$p_{\mathbf{X}_n|\mathbf{Y}_n}(\mathbf{x}_n|\mathbf{y}_n, \boldsymbol{\theta}_n, \tilde{\mathbf{x}}_{n-1}, \cdots, \tilde{\mathbf{x}}_{n-W}) = \frac{1}{z f_{\mathbf{Y}_n}(\mathbf{y}_n|\boldsymbol{\theta}_n)} \left[\prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2_{n,x_{n,r}}}} \right]$$
$$\cdot \exp\left(-\sum_{r=1}^{N} \frac{(y_{n,r} - \sum_{s=-D}^{D} h_{0,s}\mu_{n,x_{n,r-s}} - \sum_{m=1}^{W} \sum_{s=-D}^{D} h_{m,s}\mu_{n,\tilde{x}_{n-m,r-s}})^2}{2\sigma^2_{n,x_{n,r}}} \right.$$
$$\left. - \sum_{\{r,s\}\in\mathcal{C}} \beta t(x_{n,r}, x_{n,s}) \right) \tag{2.38}$$

2.5 Minimum Area Increment(MAI) Constraint

2.5.1 Area Increment Measuring Function

To improve the bridged channel defect, we propose the MAI(minimum area increment) constraint. We want the small particles to merge together according to the MRF property but the large objects not to merge together easily.

First, we define the bridging class set as the set of classes whose objects have a bridged channel problem. And we define the class-k connecting point as a point where more than two disconnected class k regions exist over neighboring pixels. Figure 2.4 shows a 4 neighbor and a 12 neighbor system. The center point can be a connecting point and the surrounding shaded points are its neighboring pixels.

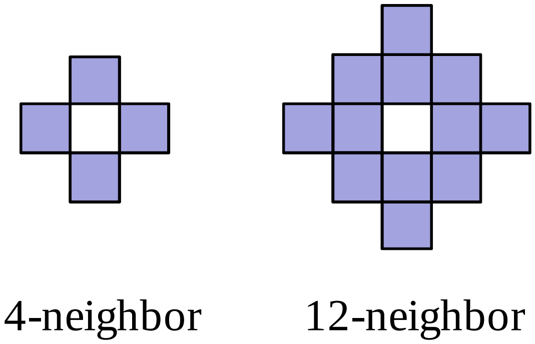


Fig. 2.4. A connecting point and its neighbors

Now, we propose the area increment measuring function $g_{w_s,r}(x_r)$ as the amount of area increase on largest-area region which exists among the neighbors of pixel location r, when the class label of that pixel is x_r. (where w_s is window size and $\forall r, g_{w_s,r}(0) = 0$)

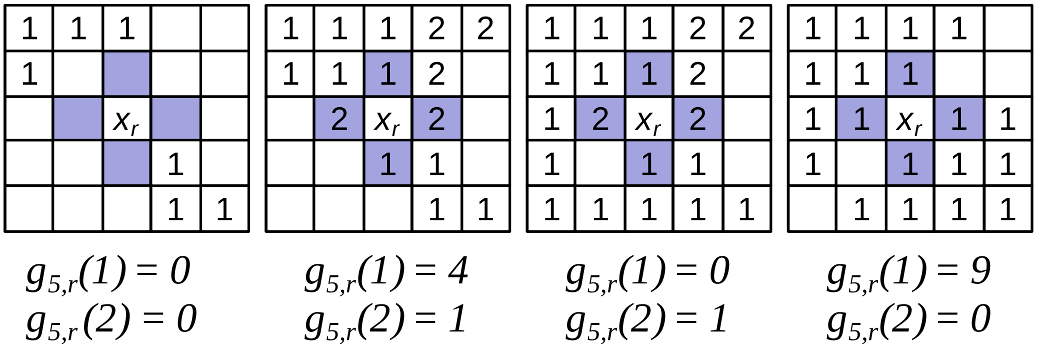
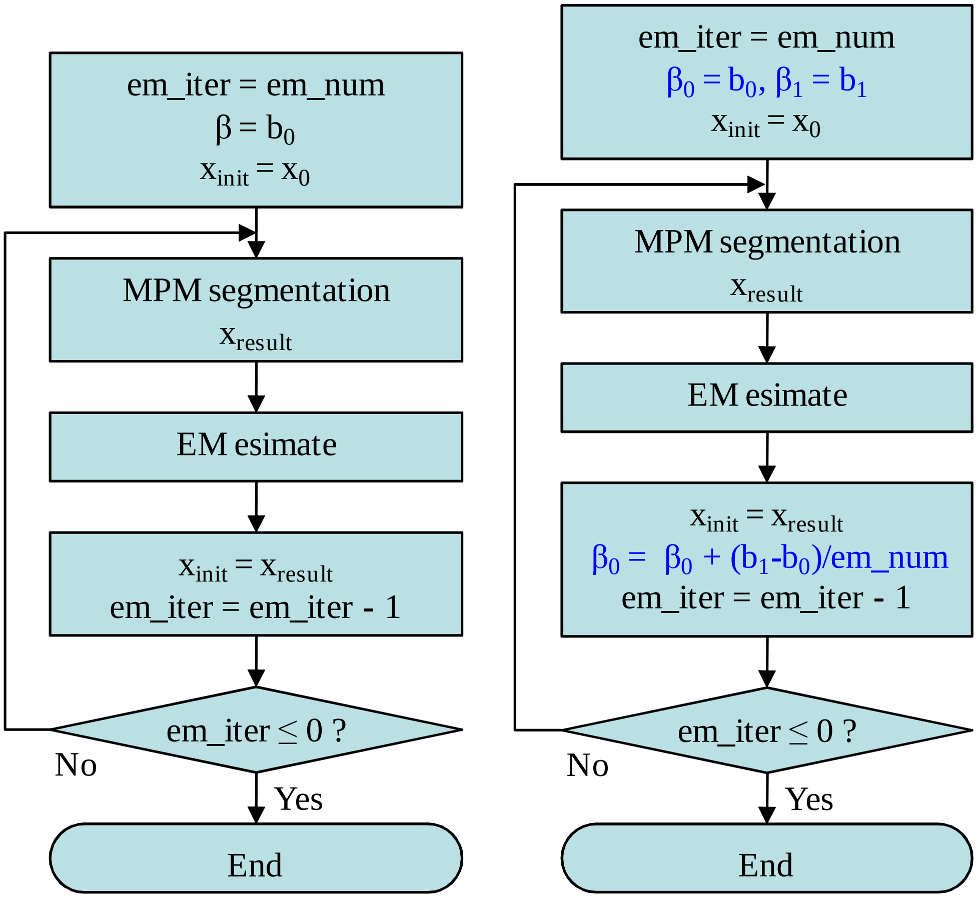


Fig. 2.5. Examples of the area increment measuring function (searching 4 neighbors)

Figure 2.5 shows some examples of the area increment measuring function when the class number is 3 and the window size $w_s = 5$. In this figure, the pixels labeled with 1 and 2 are the class 1 and the class 2 pixels respectively. And the pixels with no label is the class 0 pixels. In this example, the class 1 and the class 2 are bridging classes while the class 0 is a non-bridging class. In the first image, the center point is not a connecting point because there is no more than two disconnected same class areas over neighboring pixels. In the second image, the center point is both the class-1 connecting point and the class-2 connecting point. If the x_r is the class 1, the top left of the class 1 area and the bottom right of the class 1 area are connected. And the top left region has larger area than the bottom. Therefore, the area increment $g_{5,r}(1)$ becomes 4 which is the area of the bottom right region. Likewise, $g_{5,r}(2)$ becomes 1. In the third image, the center point is only the class-2 connecting point because all the pixels of the class 1 have already been connected. This constraint measured as $\tau g_{w_s,q_t}(x_{q_t})$ at time t can be incorporated into 2.25 when generating Markov chain, where τ is a control parameter to adjust the amount of this constraint.

2.5.2 Modified EM/MPM Iteration



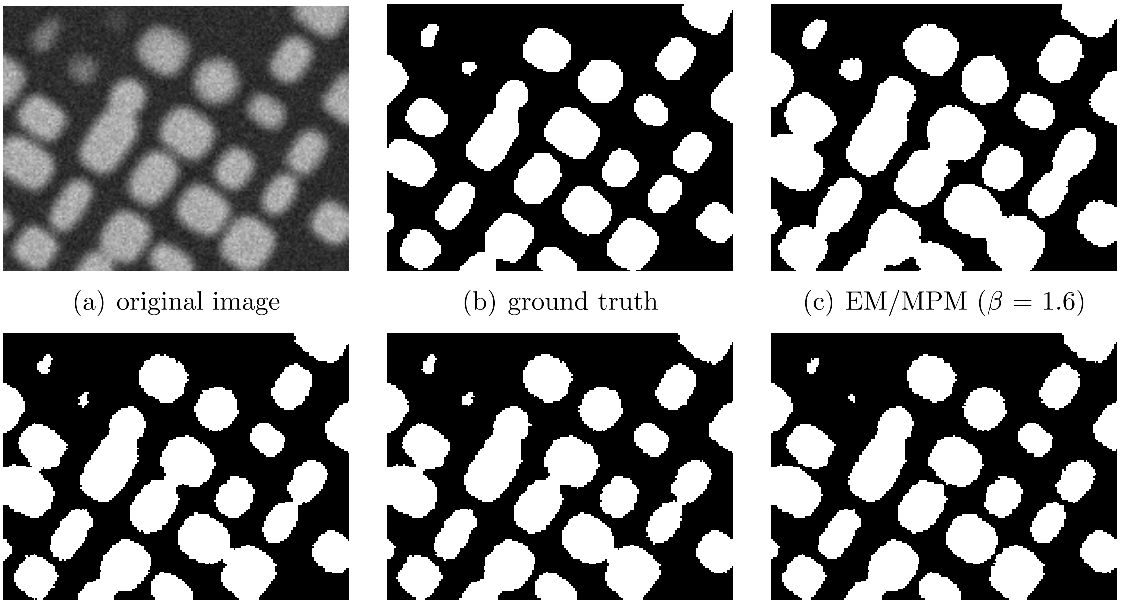
(a) Original EM/MPM algorithm (b) Proposed SA scheme

Fig. 2.6. EM/MPM algorithm flowchart

To make the proposed MAI constraint more effective, we applied the SA(simulated annealing) scheme during the iteration. But, different from the conventional SA, we gradually change the β value only for the non-bridging classes. Figure 2.6 illustrates flowcharts of the original EM/MPM algorithm and the proposed SA scheme for the segmentation of two classes. The class 0 is not in the bridging class set. And the class 1 is in the bridging class set in which we are interested for its merging problem. In these flowcharts, 'em_num' is the EM/MPM iteration step number. $\mathbf{x}_0$ is the initial class label for the whole EM/MPM algorithm. $\mathbf{x}_{init}$ is used as an initial class label for the MPM segmentation. And, $\mathbf{x}_{result}$ is the result of that segmentation. In the proposed SA scheme, the spatial interaction parameter β_0 for the non-bridging class is initially set smaller than β_1 of the bridging class. And then it is gradually increased toward the same value as β_1 according to the iteration step. And b_0 and b_1 is the initial spatial interaction parameters of class 0 and class 1 respectively.

2.6 Experimental Results

In this section, we present the visual and numerical results obtained by using the joint deconvolution segmentation(JDS) EM/MPM algorithm and the MAI constraint. In our experiments, the percentage of misclassified pixels(PMP) is used to evaluate the performance of our algorithm numerically.



(a) original image

(b) ground truth

(c) EM/MPM ($\beta = 1.6$)

(d) Single variance JDS EM/MPM ($\beta = 1.6$, $\omega = 0.52$, $\delta = 0$)

(e) Multi variance JDS EM/MPM ($\beta = 1.6$, $\omega = 0.52$, $\delta = 0$)

(f) Single variance JDS EM/MPM with MAI/SA ($b_0 = 1.2$, $b_1 = 1.6$, $\omega = 0.52$, $\delta = 0$, τ=1.5)

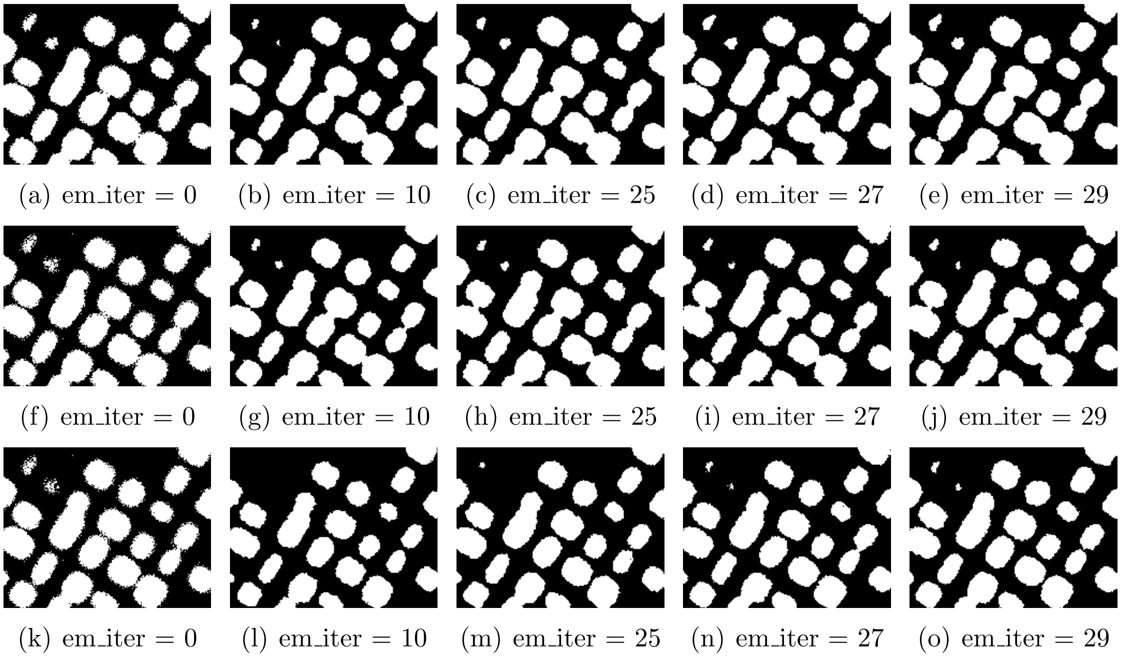
Fig. 2.7. Result images of Ni-Al-Cr superalloy images

The first test image and its results are shown in Figure 2.7. Figure 2.7(a) is the original image and Figure 2.7(b) is the ground truth segmentation image of the original image. We apply the original EM/MPM algorithm to the original image, JDS EM/MPM algorithm using the single variance form (2.13), the JDS EM/MPM algorithm using the multi variance form (2.18) and the JDS EM/MPM algorithm using single variance with MAI constraint and SA. Figure 2.7(c), Figure 2.7(d), Figure 2.7(e) and Figure 2.7(f) are the results of those methods respectively. We can see that the class 1 (white area) objects in the result image of the original EM/MPM segmentation are so expanded and have some bridged channel defects. The results of JDS EM/MPM algorithm show that this method makes the class 1 area to be more shrunk by reducing blurring artifacts, but it still have some bridged channels left. Figure 2.7(f) shows that the JDS EM/MPM algorithm with MAI/SA can eliminate these bridged channels effectively. In this experiment, we used a 12 neighbor system and a window size $w_s = 7$ for MAI. And the em_num, the EM iteration number is 30.

Figure 2.8 shows that the MAI constraint is effective only when it is used with the proposed SA scheme and the JDS EM/MPM algorithm. The MAI/SA method aims for the objects to be well-disconnected and eventually fit to the object area without bridged channels. However, in the intermediate results of the MAI/SA method without the JDS EM/MPM algorithm (first row) and of the MAI constraint without the SA (second row), many of the class 1 objects have already been merged even in the early stage of the EM/MPM iteration. But the JDS EM/MPM algorithm with the MAI/SA (third row) makes the MAI constraint more effective.

The second test image group and their results are shown in Figure 2.9 and Figure 2.10. In Figure 2.9(c) and Figure 2.10(c), we can see the conventional EM/MPM algorithm produce particles with some bridged channels. However, in the Figure 2.9(f) and Figure 2.10(f), the JDS EM/MPM segmentation method with the MAI/SA can mitigate those bridged channels.

Table 2.1 shows the PMP value of each test image result. Our JDS EM/MPM method shows better PMP value compared to original EM/MPM method. And, JDS EM/MPM with MAI/SA method gives the best result. In addition, single variance



(a) em_iter = 0 (b) em_iter = 10 (c) em_iter = 25 (d) em_iter = 27 (e) em_iter = 29

(f) em_iter = 0 (g) em_iter = 10 (h) em_iter = 25 (i) em_iter = 27 (j) em_iter = 29

(k) em_iter = 0 (l) em_iter = 10 (m) em_iter = 25 (n) em_iter = 27 (o) em_iter = 29

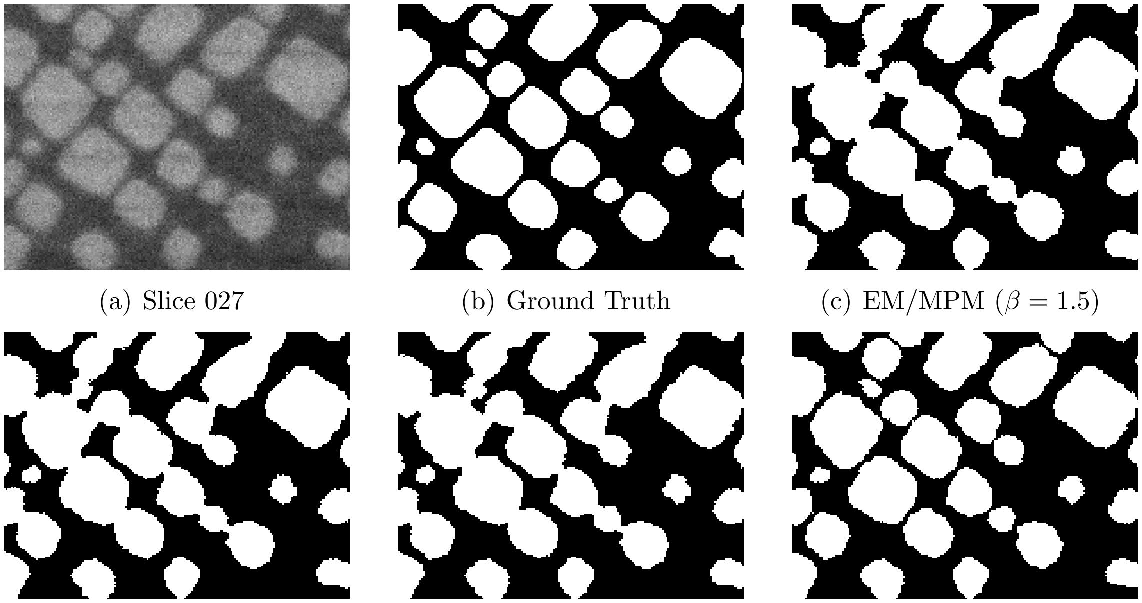
Fig. 2.8. The MAI constraint applied segmentation results during the EM iteration: (first row) The MAI constraint only with SA; (second row) The MAI constraint only with JDS; (third row) The MAI constraint with both JDS and SA

Table 2.1
PMP of Test and Ni-Al-Cr superalloy images

	Fig. 2.7(a)	Fig. 2.9(a)	Fig. 2.10(a)
Original EM/MPM	11.60	6.18	5.27
Single variance JDS EM/MPM	5.37	6.04	5.38
Multi variance JDS EM/MPM	5.35	6.08	5.29
Single variance JDS EM/MPM with MAI/SA	4.63	4.97	4.63

form shows as much favorable result as muti variance form even reducing the computational complexity.

For the third test image dataset, we used the images of a polycrystalline nickel-base superalloy (René 88 DT) material [28]. Images with the slice number from 013



(a) Slice 027

(b) Ground Truth

(c) EM/MPM ($\beta = 1.5$)

(d) Single variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.47$, $\delta = 0.5$)

(e) Multi variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.47$, $\delta = 0.5$)

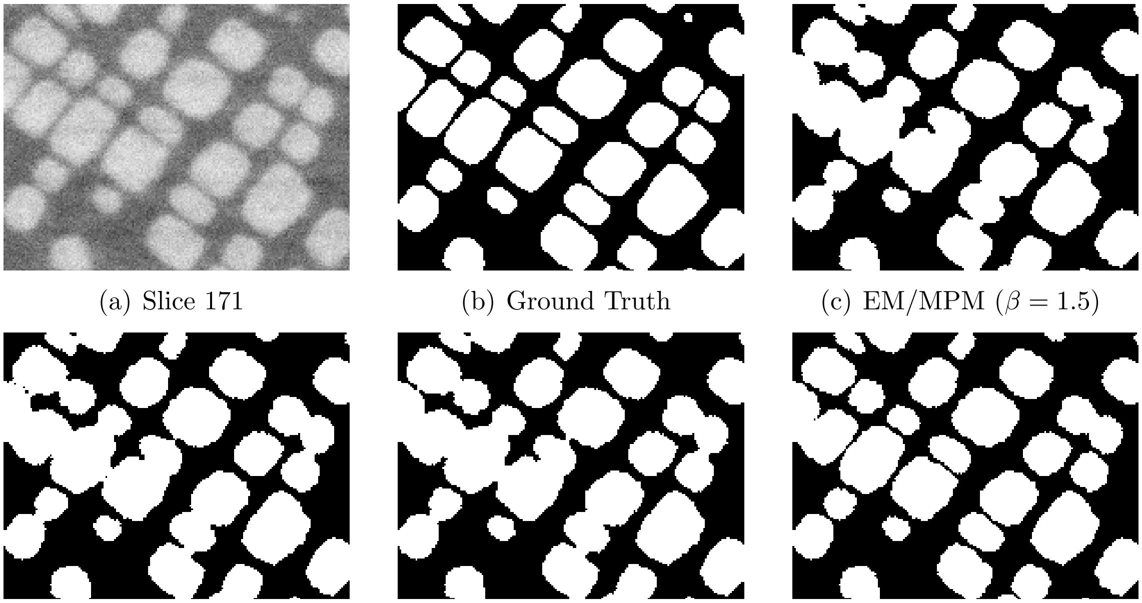
(f) Single variance JDS EM/MPM with MAI/SA ($b_0 = 1.1$, $b_1 = 1.5$, $\omega = 0.47$, $\delta = 0.5$, $\tau = 1.5$)

Fig. 2.9. Result images of slice number 27 of Ni-Al-Cr superalloy image sequence

Table 2.2
PMP of René 88 DT images

	Slice 017	Slice 152	Slice 170
Original EM/MPM	7.27	7.01	5.43
Single variance JDS EM/MPM	5.62	5.23	4.42
Multi variance JDS EM/MPM	5.62	5.17	4.36
Single variance JDS EM/MPM with MAI/SA	5.03	4.74	4.21

to 017, from 148 to 152 and from 165 to 170 were used (Figure 2.11). The light-colored phase(class 1) is γ' and the gray matrix(class 0) is γ. Figure 3.14, 3.15 and 3.16 show the results of the last images of those image sequences. The result images demonstrate that the JDS EM/MPM with MAI/SA method makes the area of the



(a) Slice 171

(b) Ground Truth

(c) EM/MPM ($\beta = 1.5$)

(d) Single variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.38$, $\delta = 0.5$)

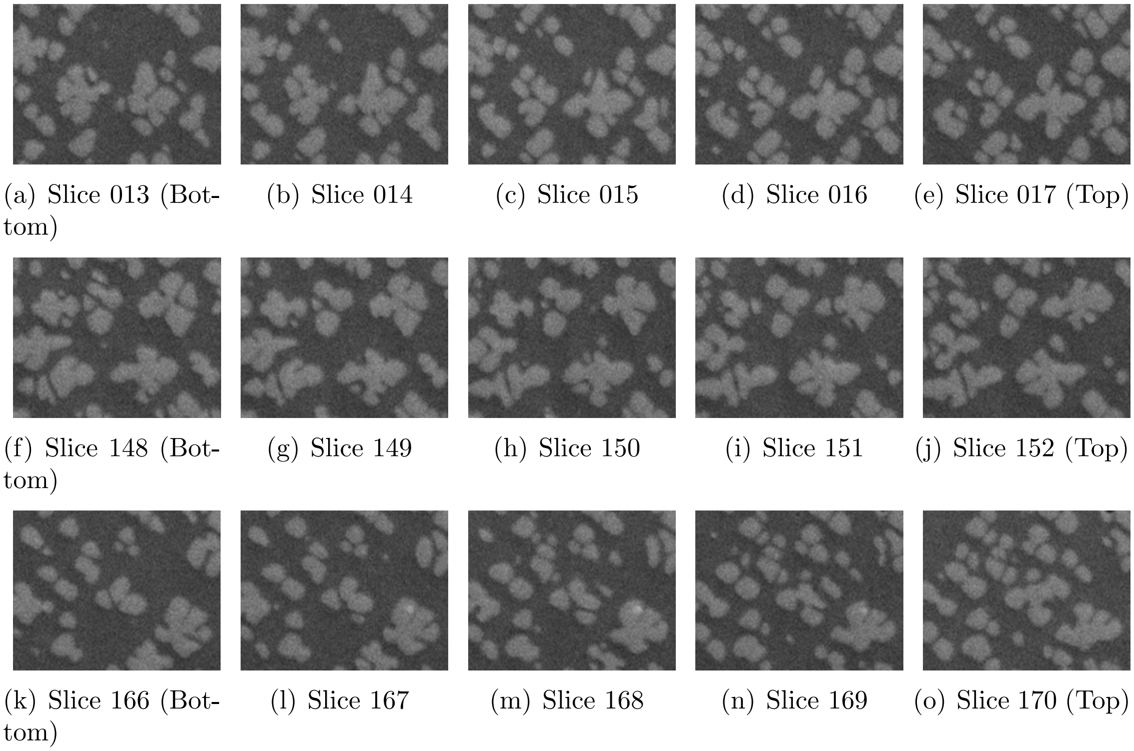
(e) Multi variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.38$, $\delta = 0.5$)

(f) Single variance JDS EM/MPM with MAI/SA (b_0 = 0.9, b_1 = 1.5, $\omega = 0.38$, δ = 0.5, τ = 1.7)

Fig. 2.10. Result images of slice number 171 image of Ni-Al-Cr superalloy image sequence

class 1 objects shrunk much and improves the bridged channel defect. But, there still remain some misclassifications in the boundary, particularly on the denting area (marked with circles in Figure 2.12(f) 2.13(f) and 2.14(f)). Table 2.2 compares the PMP value of each method. And it shows that our proposed method gives the best performance.

We applied our method to the 3D EM/MPM method and got the result images in Figure 2.15 and Figure 2.16. In this experiment, we used the single variance JDS method. The 3D EM/MPM method needs a large amount of memory. Therefore, we apply the method for the three frames and save the result of the middle frame first. And then we move to the next three frames which are one frame shifted from the previous three frames and repeat the same process. For the test image sequence, we use the slice number 1 to 59 of the Ni-Al-Cr superalloy image sequence and the slice number 143 to 188 of the René 88 DT image sequence. In Figure 2.15 and



(a) Slice 013 (Bottom)

(b) Slice 014

(c) Slice 015

(d) Slice 016

(e) Slice 017 (Top)

(f) Slice 148 (Bottom)

(g) Slice 149

(h) Slice 150

(i) Slice 151

(j) Slice 152 (Top)

(k) Slice 166 (Bottom)

(l) Slice 167

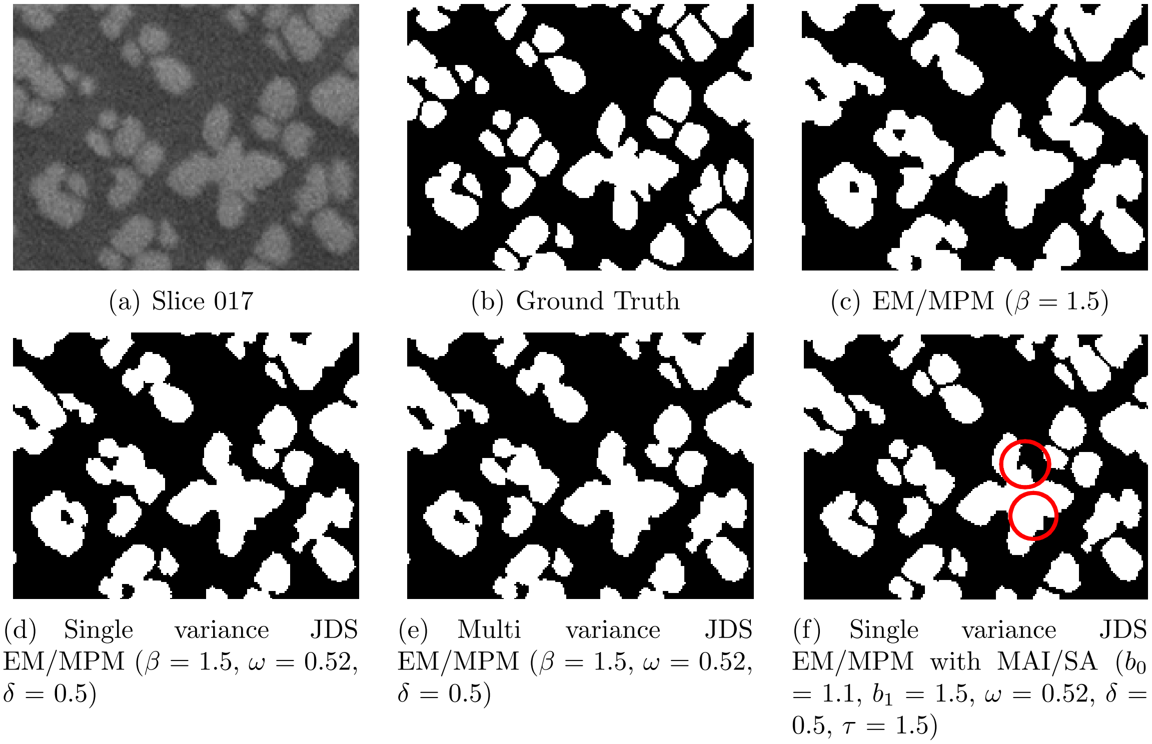
(m) Slice 168

(n) Slice 169

(o) Slice 170 (Top)

Fig. 2.11. Series of René 88 DT images. The light-colored phase is γ' and the gray matrix is γ

Figure 2.16, the first column is the results of the original EM/MPM method, the second column is of our proposed methods applied to the 2D EM/MPM method and the third column is of our proposed methods applied to the 3D EM/MPM method. When we compare the results, especially focused on the regions in the circles, our proposed methods improve the merging problem better than the original EM/MPM method. Especially, when it is applied with 3D EM/MPM method, it shows the best performance in reducing the bridged channel defects. Furthermore, 3D EM/MPM method can improve some jagged boundaries of objects appeared in the circle of Figure 2.16(e). Table 2.3 compares the running time of each method.


(a) Slice 017

(b) Ground Truth

(c) EM/MPM ($\beta = 1.5$)

(d) Single variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.52$, $\delta = 0.5$)

(e) Multi variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.52$, $\delta = 0.5$)

(f) Single variance JDS EM/MPM with MAI/SA (b_0 = 1.1, $b_1 = 1.5$, $\omega = 0.52$, $\delta =$ 0.5, $\tau = 1.5$)

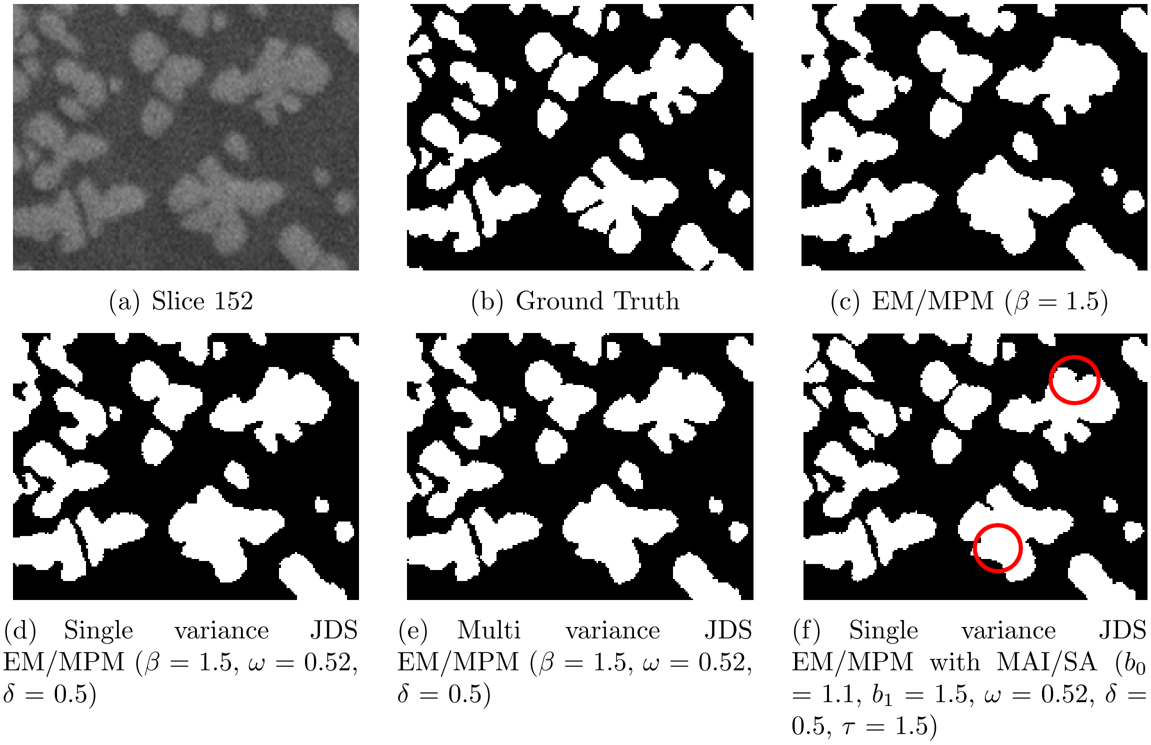
Fig. 2.12. Result images of slice number 017 of René 88 DT image sequence

Table 2.3
Running Time

(sec)	Ni-Al-Cr superalloy	René 88 DT
2D EM/MPM	81	63
2D JDS EM/MPM with MAI/SA	2260	1762
3D JDS EM/MPM with MAI/SA	6638	5175

2.7 Conclusions

First, in this chapter, we proposed the JDS method to reduce the swelling of the object boundary in the segmentation result. This method incorporates the blurring information of the microscope image into the existing segmentation method. To be specific, we proposed two types of the JDS method: one is a single variance method



(a) Slice 152

(b) Ground Truth

(c) EM/MPM ($\beta = 1.5$)

(d) Single variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.52$, $\delta = 0.5$)

(e) Multi variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.52$, $\delta = 0.5$)

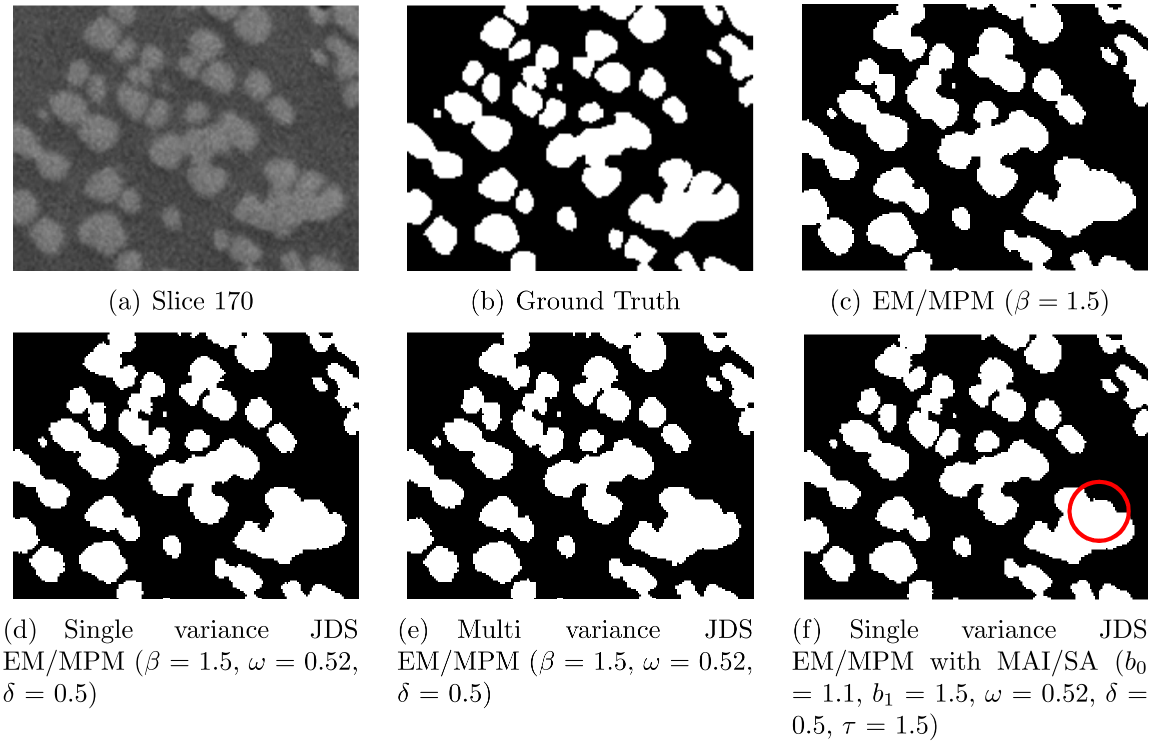
(f) Single variance JDS EM/MPM with MAI/SA ($b_0 = 1.1$, $b_1 = 1.5$, $\omega = 0.52$, $\delta = 0.5$, $\tau = 1.5$)

Fig. 2.13. Result images of slice number 152 of René 88 DT image sequence

and the other is a multi variance method. The multi variance method brings better PMP values than the single variance method. However, using the multi variance method increased the computational complexity. Therefore, if the application pursues the computational simplicity more than the accuracy of the segmentation result, a single variance method would be recommended. The experimental results showed that this JDS method significantly improved the misclassifications of the object boundary.

However, the bridged channels still remain in some areas. To tackle this issue, secondly, we suggested the MAI/SA method as the MAI constraint is designed to limit the excessive merging of large objects during the segmentation process. And this constraint was much effective when it was applied together with the SA scheme.

Lastly, we combined the JDS method and the MAI/SA method with the existing EM/MPM method. The experimental result showed that this combined method reduced the bridged channel defects very effectively. We also performed the 3D seg-



(a) Slice 170

(b) Ground Truth

(c) EM/MPM ($\beta = 1.5$)

(d) Single variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.52$, $\delta = 0.5$)

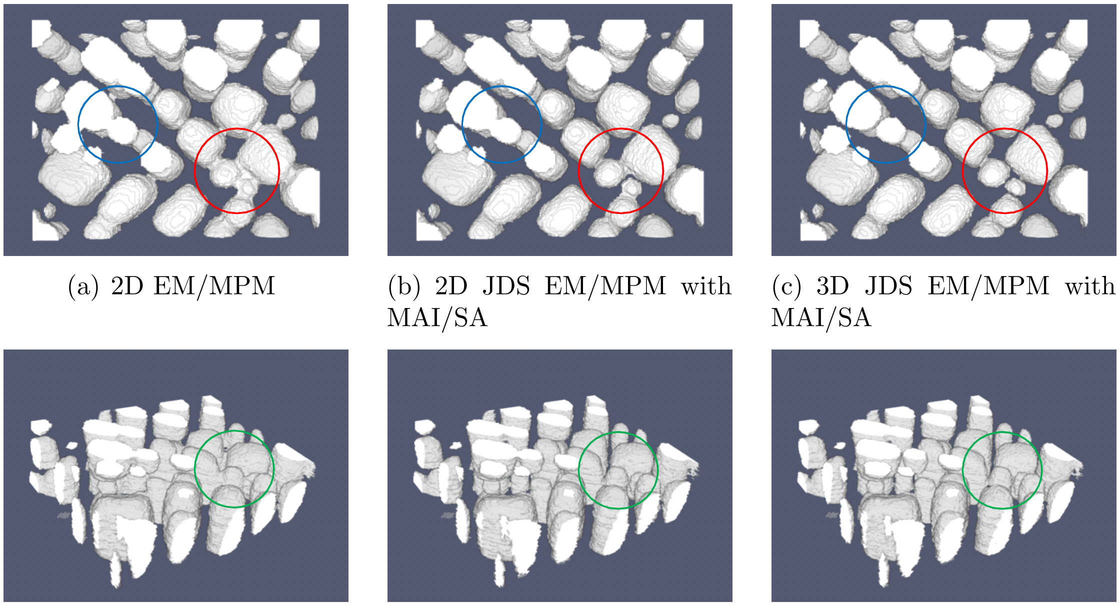
(e) Multi variance JDS EM/MPM ($\beta = 1.5$, $\omega = 0.52$, $\delta = 0.5$)

(f) Single variance JDS EM/MPM with MAI/SA ($b_0 = 1.1$, $b_1 = 1.5$, $\omega = 0.52$, $\delta = 0.5$, $\tau = 1.5$)

Fig. 2.14. Result images of slice number 170 image of René 88 DT image sequence

mentation in the 3D dataset with this proposed method. In this case, the correlation between neighboring slices in the image sequence can be used to improve the boundary precision of the segmentation result. To reduce the memory size in handling the 3D dataset, we made a window of three slices of image and processed the 3D segmentation by moving the window incrementally by one slice. We could see that the 3D JDS EM/MPM with MAI/SA method applied to the 3D dataset cleared the bridged channel problem better than the other methods. Also, we found it improved the jagged boundaries of the objects.

The study on the misclassification issue of the denting areas is to be followed in the future. Also, further study could be done to make more precise blurring image models or to automatically estimate control parameters of the proposed model.



(a) 2D EM/MPM

(b) 2D JDS EM/MPM with MAI/SA

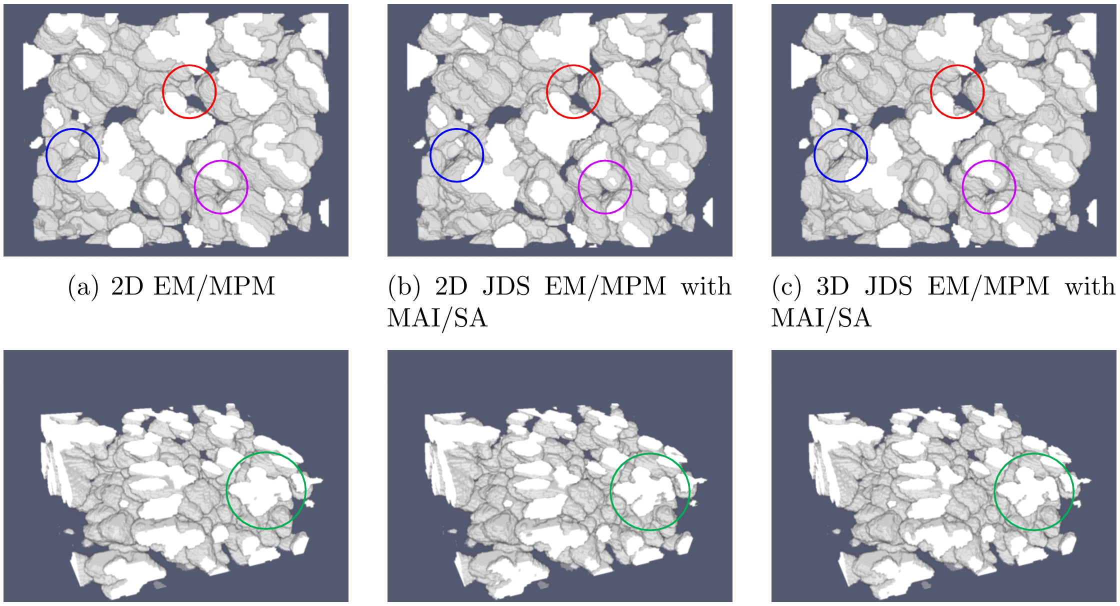
(c) 3D JDS EM/MPM with MAI/SA

(d) 2D EM/MPM ($\beta = 1.5$)

(e) 2D JDS EM/MPM with MAI/SA ($\beta = 1.5$, $\omega = 0.3$, $\delta = 0.1$, $\tau = 1.5$)

(f) 3D JDS EM/MPM with MAI/SA ($\beta = 1.5$, $\omega = 0.3$, $\delta = 0.1$, $\tau = 1.5$)

Fig. 2.15. Result 3D images of Ni-Al-Cr superalloy image sequence



(a) 2D EM/MPM

(b) 2D JDS EM/MPM with MAI/SA

(c) 3D JDS EM/MPM with MAI/SA

(d) 2D EM/MPM ($\beta = 1.5$)

(e) 2D JDS EM/MPM with MAI/SA ($\beta = 1.5$, $\omega = 0.5$, $\delta = 0.5$, $\tau = 1.0$)

(f) 3D JDS EM/MPM with MAI/SA ($\beta = 1.5$, $\omega = 0.5$, $\delta = 0.5$, $\tau = 1.0$)

Fig. 2.16. Result 3D images of René 88 DT image sequence

3. CHANNEL DETECTION IN MICROSCOPE IMAGES OF MATERIALS USING MARKED POINT PROCESS MODELING

3.1 Introduction

In this chapter, we propose two methods to extract channel configurations in materials images. A channel is a narrow gap between two structures. To extract an accurate channel configuration, we need to detect the position, width, length and rotation angle of each channel. In order to detect those parameters, we use the Marked Point Process (MPP) framework. Compared to the conventional Markov random field model, the advantage of the MPP based model is that it can exploit the geometric information of the channels, and their spatial relation.

In order to extract the channels from an image, we design a data energy that reflects the coherence between each channel configuration and the input material image. Then, we apply this data energy together with an interaction energy to the MPP framework. To minimize this energy, we use the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm [29] together with simulated annealing [30]. Once the channel configuration is obtained, this information can be used for further analysis. For instance, we can use it to reduce bridged channel defects in the conventional segmentation of material images [22, 31]. We suggest a methodology for incorporating the channel information with an existing segmentation method to improve boundary precision. As a segmentation method, we use the expectation-maximization/maximization of the posterior marginals (EM/MPM) segmentation algorithm [19], which has been successful in many applications. For experimental validation, we use four materials datasets. The results demonstrate that the proposed method is effective in extracting channels from images, and in improving segmentation results at channel boundaries.

This chapter is organized as follows: Firstly, in Section 3.2, we briefly review the MPP. Our proposed channel modeling methods are presented in Section 3.3. In Section 3.4, the methodology is proposed to improve the result of the existing segmentation method with channel information. In Section 3.5, results of applying our proposed method are shown, with detailed discussion.

3.2 Marked Point Process

A marked point process $\mathbf{X} = \{X_1, \ldots, X_n\}$ defined on a space $S = M \times K$ is a point process, whose realizations are configurations of marked points of the form $\mathbf{x} = \{x_1, \ldots, x_n\} \in \mathcal{C}$. Each marked point consists of random spatial location and random parameters that describe shape of object, such as ellipse, rectangle or line segment. For example, if a microscopic image is composed of randomly located circular particles, we can model this image with the radius and location of each circle. In this case, the ith marked point can be expressed as $X_i = (S_i, N_i)$, where S_i is the location and the mark N_i has the radius of the circle object.

The probability density function of an MPP is given by the Gibbs distribution as

$$p(\mathbf{x}) = \frac{1}{z_{mpp}} \exp(-U(\mathbf{x})) \tag{3.1}$$

where $U(\mathbf{x})$ is the Gibbs energy of a configuration $\mathbf{x}$ and z_{mpp} is a normalizing constant. To describe a density $p(\mathbf{x})$ conditioned on image data $\mathbf{y}$, the Bayesian model is widely used in image processing.

$$p(\mathbf{x}|\mathbf{y}) \propto p_{likelihood}(\mathbf{y}|\mathbf{x}) p_{prior}(\mathbf{x}) \tag{3.2}$$

Then a Gibbs energy can be described as:

$$U(\mathbf{x}|\mathbf{y}) \propto U_D(\mathbf{y}|\mathbf{x}) + U_P(\mathbf{x}) \tag{3.3}$$

where $U_D(\mathbf{y}|\mathbf{x})$ is a data energy and $U_P(\mathbf{x})$ is a prior energy. The term $U_D(\mathbf{y}|\mathbf{x})$ can be further expanded as a sum of individual potentials over objects in a configuration.

$$U_D(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} V_d(\mathbf{y}|x_i) \tag{3.4}$$

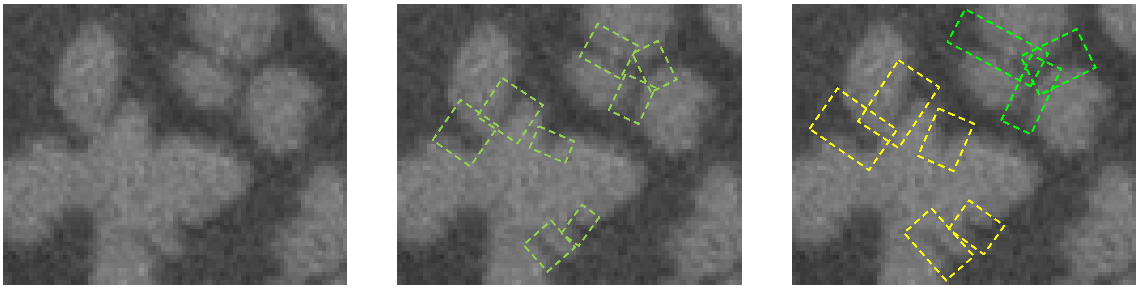
$V_d(\mathbf{y}|x_i)$ is a data potential for an individual object. If $V_d(\mathbf{y}|x_i) \leq 0$, the object x_i is a favorable candidate, since it increases the density $p(\mathbf{x}|\mathbf{y})$. For the prior energy $U_P(\mathbf{x})$, we use only single-pixel and two-pixel clique. Then the prior energy can be expressed as:

$$U_P(\mathbf{x}) = \sum_{i=1}^{n} V_p(x_i) + \lambda_{int} \sum_{x_i \sim x_j} V_{int}(x_i, x_j) \tag{3.5}$$

$V_p(x_i)$ is a prior potential for an individual object. And $V_{int}(x_i, x_j)$ is an interaction potential, where a neighborhood relation $\sim$ is defined between a pair of objects and λ_{int} is a parameter to control the amount of contribution of the interaction potential to the prior energy.

3.3 Proposed Channel Modeling Methods

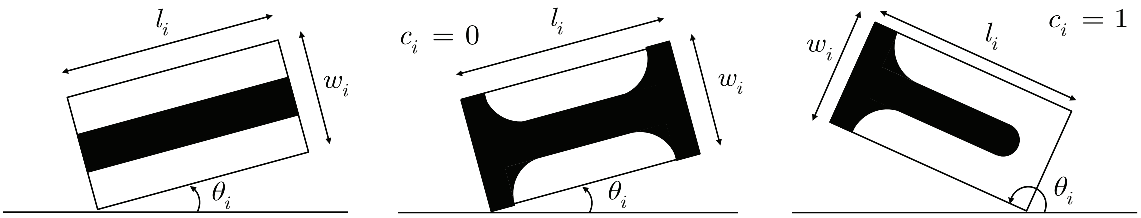
Figure 3.1 shows some examples of channels. To classify these channels further, we can categorize them as simple, necking and denting channels, as shown in Figure 3.1(b) and 3.1(c). While simple and necking channels have two open ends, denting channels have one open and one closed end. Taking a closer look at this figure, we can see these three types of channels have typical shapes. Figure 3.2 represents these typical shapes and some parameters related to the formation of channels. For the ith object, l_i is length, w_i is width and θ_i is rotation angle of the object. A necking



(a) (b) (c)

Fig. 3.1. Channels: (a) Original data (b) Simple channel (c) Necking (marked with green box) and denting (marked with yellow box) channel

channel is different from a simple channel in that it has a special shape at both ends to confine the length of a channel object to the length of the actual channel in the image data. For the necking and denting channels shown in Figure 3.2, as a prior model we simply penalize overlapping of objects [32]. We also define a curved channel model which identifies curved channels with a series of connected simple channels which have a shape as shown in Figure 3.2(a). In this method, we use a Quality Candy model [16] as a prior model to detect channels which have curved shapes. As a data potential in both channel modeling methods, we modify that proposed in [16] which uses Student's t-test. For two datasets D_1 and D_2 the t-test value is calculated as:



(a) Simple channel (b) Necking channel (c) Denting channel

Fig. 3.2. Types of channel

$$t_{test}(D_1, D_2) = \frac{|\bar{D_1} - \bar{D_2}|}{\sqrt{\frac{\tilde{\sigma}^2_{D_1}}{n_{D_1}} + \frac{\tilde{\sigma}^2_{D_2}}{n_{D_2}}}} \tag{3.6}$$

where $\bar{D_1}$ is the sample mean, $\tilde{\sigma}_{D_1}$ is the sample standard deviation and n_{D_1} is the number of data samples in D_1.

3.3.1 Necking and Denting Channel Method

A necking channel mark $n_i = (l_i, w_i, \theta_i)$ is defined on a space $M_0 = [L_{min}, L_{max}] \times [W_{min}, W_{max}] \times [0, \pi]$ and a denting channel mark $n_i = (l_i, w_i, \theta_i)$ is defined on a space $M_1 = [L_{min}, L_{max}] \times [W_{min}, W_{max}] \times [0, 2\pi]$. Finally, a necking and denting channel mark space M can be specified as $M_0 \bigcup M_1$. The ith marked point can be expressed as $x_i = (s_i, n_i)$, where $s_i = (a_i, b_i) \in K$ is the center position of the channel object. For the necking channel, the data potential in (3.4) is given by:

$$V_d(\mathbf{y}|x_i) = \lambda_o V_o(\mathbf{y}|x_i) \tag{3.7}$$

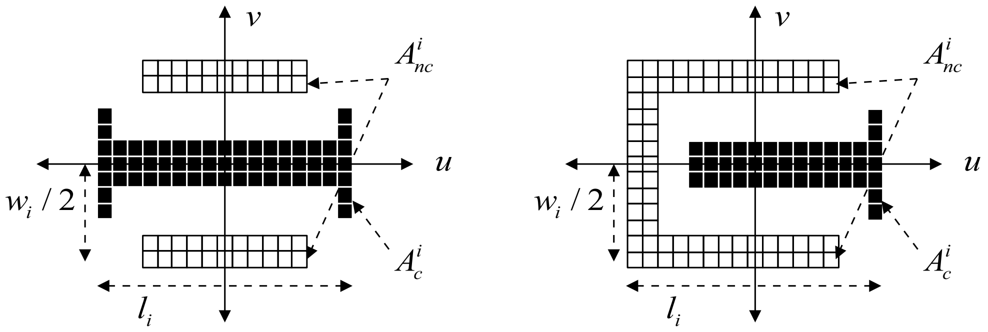
where $V_o(\mathbf{y}|x_i)$ is an object potential which reflects the statistical accordance between the channel model and the image data. This object potential will be explained in detail shortly. λ_o is a positive parameter to control the influence of the object potential. For the denting channel, a second term is added to the data potential, giving:

$$V_d(\mathbf{y}|x_i) = \lambda_o V_o(\mathbf{y}|x_i) + \lambda_c V_c(\mathbf{y}|x_i) \tag{3.8}$$

where $V_c(\mathbf{y}|x_i)$ is a continuity potential. These will be discussed in the following sections. λ_c is a positive parameter to control the influence of the continuity potential.

Object Potential $V_o(\cdot)$

For each object x_i, we expect the grayscale values of pixels in the region A_c^i to be statistically different from those in the region A_{nc}^i in Figure 3.3. We do not consider grayscale values in transition areas between A_{nc}^i and A_c^i. To measure this difference, the t-test is used. Let $R_c^i = \{\tilde{y}_i(u,v)|(u,v) \in A_c^i)\}$ and $R_{nc}^i = \{\tilde{y}_i(u,v)|(u,v) \in A_{nc}^i\}$,



(a) Necking channel (b) Denting channel

Fig. 3.3. Pixel masks of necking and denting channels

where $\tilde{y}(u,v)$ defined on $A_i = \{(u,v)||u| \leq \frac{l_i}{2}, |v| \leq \frac{w_i}{2}\}$ is the image patch from the image data $y(a,b)$ corresponding to the parameters of the object x_i. This can be written as follows:

$$\tilde{y}_i(u,v) = y(a,b) \tag{3.9}$$

$$\begin{bmatrix} a \\ b \\ 1 \end{bmatrix} = T(a_i, b_i)R(\theta_i) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3.10}$$

where $T(\cdot)$ is a translation matrix and $R(\cdot)$ is a rotation matrix. Then, the t-test value of a channel object x_i is calculated as $t_{test}(R_c^i, R_{nc}^i)$ from (3.6). If this value is big, this means the channel marked point is well matched to the image data. However, this t-test term reflects only an overall difference between a channel model and the image data. Therefore, if we exploit more specific characteristics of the channel shape, we can have a more accurate measure of the data fit. The shape of our channel object is symmetric about the u axis. This means that the data should be symmetric also. Therefore, the symmetry error is given by:

$$E_s(x_i, \mathbf{y}) = \frac{1}{|A_c^i \cup A_{nc}^i|} \int\int_{A_c^i \cup A_{nc}^i} |\tilde{y}_i(u,v) - \tilde{y}_i(u,-v)| \mathrm{d}u\mathrm{d}v \tag{3.11}$$

Then, the statistic value reflecting the correspondence between x_i and the image data y is given by:

$$B(\mathbf{y}|x_i) = \frac{t_{test}(R_c^i, R_{nc}^i)}{\max(S_{th}, E_s(x_i, \mathbf{y})/\tilde{m}_i)} \tag{3.12}$$

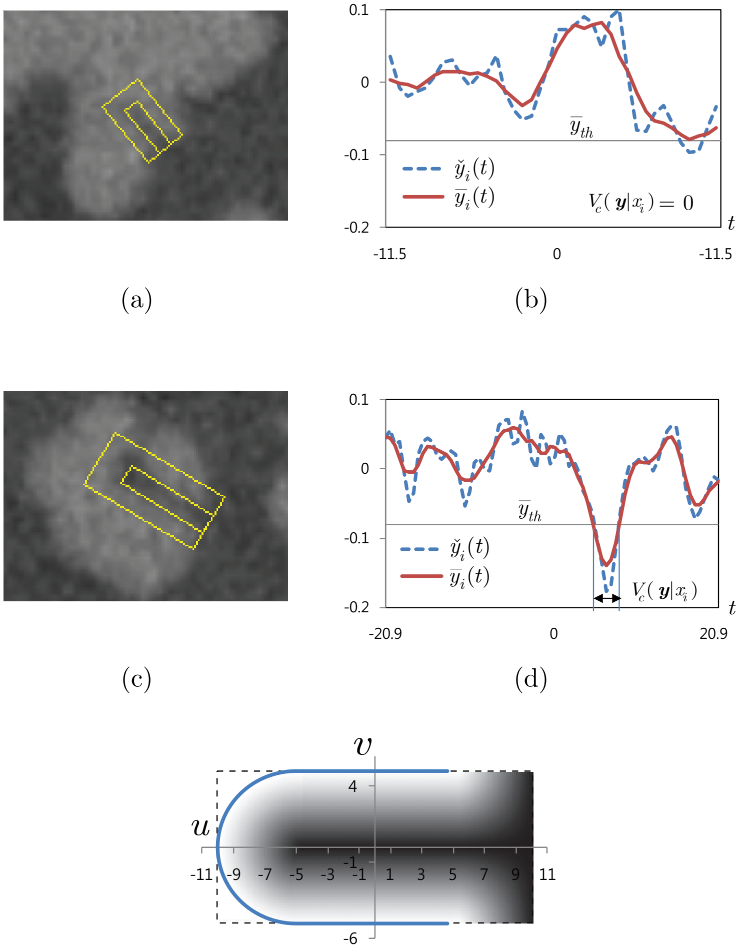
where $\tilde{m}_i$ is an average intensity of image data in $A_c^i \cup A_{nc}^i$ and S_{th} is a control parameter. Finally, the object potential value is given by [12]:

$$V_o(\mathbf{y}|x_i) = \begin{cases} 1 - \frac{B(\mathbf{y}|x_i)}{T} & B(\mathbf{y}|x_i) < T \\ \exp(-\frac{B(\mathbf{y}|x_i)-T}{3B(\mathbf{y}|x_i)}) - 1 & B(\mathbf{y}|x_i) \geq T \end{cases} \tag{3.13}$$

Continuity Potential $V_c(\cdot)$

This potential is designed to avoid misidentifying a necking channel as a denting channel. Actually, the shape of a channel in Figure 3.4(c) does not match with any pixel mask of this method shown in Figure 3.3 but does have two open ends. Therefore, we prefer not to identify this channel as a denting channel based on the definition that a denting channel has one open and one closed end. This continuity potential confines the denting channel to be identified according to its own definition and is used only for the denting channel. To examine if the channel in the image data has one closed end, we evaluate the intensity change of the image data along the path of a parametric function pair $(u(t; l, w), v(t; l, w))$ in the denting channel which has length l and width w. The solid line in Figure 3.4(e) presents an example of this path whose equation is given in Appendix B. First, the function $\tilde{y}_i(t)$ defined on $t \in [-l_i - (\frac{\pi-4}{4})w_i, l_i + (\frac{\pi-4}{4})w_i]$ is constructed as:

$$\breve{y}_i(t) = \tilde{y}_i(u(t\,;\, l_i, w_i), v(t\,;\, l_i, w_i)) \tag{3.14}$$



(a)

(b)

(c)

(d)

(e) $(u_i(t\,;\,20,10), v_i(t\,;\,20,10))$

Fig. 3.4. Continuity potential. (a) Denting channel (b) $\breve{y}_i(t)$ and $\bar{y}_i(t)$ of the denting channel (c) Misidentified channel (d) $\breve{y}_i(t)$ and $\bar{y}_i(t)$ of the misidentified channel (e) Path of a parametric function pair

Then, $\breve{m}_i$ is calculated as:

$$\breve{m}_i = \frac{2}{(\pi - 4)w_i} \int_{-l_i-(\frac{\pi-4}{4})w_i}^{l_i+(\frac{\pi-4}{4})w_i} \breve{y}_i(t)dt \tag{3.15}$$

Then, this $\breve{y}_i(t)$ becomes translated and normalized by $\breve{m}_i$ and low-pass filtered as:

$$\begin{aligned}\hat{y}_i(t) &= \frac{\breve{y}_i(t) - \breve{m}_i}{\breve{m}_i} \\ \bar{y}_i(t) &= h_{LP}(t) * \hat{y}_i(t)\end{aligned} \tag{3.16}$$

where $h_{LP}(t)$ is the impulse response of low-pass filter. Finally, a continuity potential is given as an integration of the length of regions where $\bar{y}_i(t)$ is smaller than the threshold $\bar{y}_{th}$.

$$V_c(\mathbf{y}|x_i) = \int_{-l_i-(\frac{\pi-4}{4})w_i}^{l_i+(\frac{\pi-4}{4})w_i} \mathbb{1}(\bar{y}_i(t) \leq \bar{y}_{th})dt \tag{3.17}$$

Figure 3.4(b) and 3.4(d) show the $\bar{y}_i(t)$ and its continuity potential.

Interaction Potential $V_{int}(\cdot)$

For the prior energy in (3.5), we use an interaction potential and it is given by [33]:

$$V_{int}(x_i, x_j) = exp(\tau_{int}\bar{A}(x_i, x_j)) - 1 \tag{3.18}$$

where τ_{int} is a control parameter and $\bar{A}(x_i, x_j)$ reflects the overlapping amount between x_i and x_j as illustrated in Figure 3.5 and calculated from:

$$\bar{A}(x_i, x_j) = \frac{A(x_i \cap x_j)}{\min(A(x_i), A(x_j))} \tag{3.19}$$

where $A(\cdot)$ is an area function.

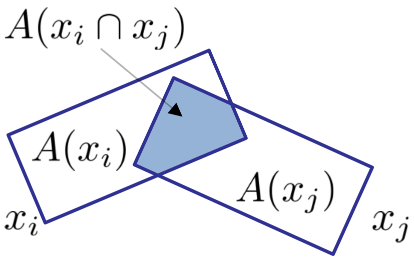


Fig. 3.5. Overlapping area of two channel objects

Optimization

Once $p(\mathbf{x}|\mathbf{y})$ is defined, we can get the most probable configuration $\hat{\mathbf{x}}$ by maximizing $p(\mathbf{x}|\mathbf{y})$, or minimizing $U(\mathbf{x}|\mathbf{y})$.

$$\hat{\mathbf{x}} = arg \min_{\mathbf{x} \in \mathcal{C}} U(\mathbf{x}|\mathbf{y}) \tag{3.20}$$

To simulate an MPP, a Markov Chain is constructed. Green proposed the Reversible Jump Markov Chain Monte Carlo(RJMCMC) method [29] for simulating a Markov chain where the dimension of the state is not fixed. In the case of a simple geometric shape and interaction prior such as non-overlap, the Multiple Birth and Death(MBD) algorithm proposed by Descombes et al. [12] outperforms the RJMCMC method in its execution time. A diffusion algorithm allows the configuration to evolve according to the energy gradient. Interested readers can see [34]. Usually, these optimization methods are applied with simulated annealing [30]. To apply simulated annealing to an optimization method, we can replace $p(\mathbf{x})$ with $p_m(\mathbf{x}) \propto (p(\mathbf{x}))^{\frac{1}{T_m}}$ in (3.1). As a decreasing scheme of temperature, we use a geometric decrease which is given by:

$$T_m = T_0 \cdot \alpha^m \tag{3.21}$$

where $\alpha < 1$. T_0 is an initial temperature and $m \in \mathbb{N}$ is time.

Reversible Jump Markov Chain Monte Carlo

When we simulate distribution $\pi(d\mathbf{x})$ of a configuration $\mathbf{x}$ using the Markov chain, where the dimension of the configuration is not fixed but can vary, the RJMCMC is one of the simulation methods that can deal with this problem using the Metropolis-Hastings paradigm to build a reversible Markov chain. It is composed of several move types m and corresponding proposition kernels $Q_m(\mathbf{x}, d\mathbf{x}')$, which are probabilities to simulate the move m from state $\mathbf{x}$ to $\mathbf{x}'$. Once this move is simulated, this proposal is accepted according to its acceptance ratio α_m. From the detailed balance condition, if $\pi(d\mathbf{x})Q_m(\mathbf{x}, d\mathbf{x}')$ has a finite density with respect to a symmetric measure ξ_m on $\mathcal{C} \times \mathcal{C}$, this ratio is formulated with $\pi(\cdot)$ and $Q_m(\cdot, \cdot)$. One transition of this algorithm is described below:

1. Choose a move type m according to the probability $p_m = Q_m(\mathbf{x}, \mathcal{C})$.

2. Sample a new configuration $\mathbf{x}'$ according to proposition kernel $Q_m(\mathbf{x}, d\mathbf{x}')$ from the current configuration $\mathbf{x}$.

3. Compute a Green's ratio:

$$R_m(\mathbf{x}, \mathbf{x}') = \frac{\pi(d\mathbf{x}')Q_m(\mathbf{x}', d\mathbf{x})}{\pi(d\mathbf{x})Q_m(\mathbf{x}, d\mathbf{x}')} \tag{3.22}$$

4. Accept the move $(\mathbf{x} \rightarrow \mathbf{x}')$ with a probability:

$$\alpha_m = \min(1, R_m(\mathbf{x}, \mathbf{x}')) \tag{3.23}$$

For the convergence of the distribution $\pi(d\mathbf{x})$, the choice of proposition kernels is critical. Some different types of kernels are as below:

- *Birth-and-death kernel* : A birth-and-death kernel allows to add an object to or remove it from a current configuration. Consequently, this kernel changes the dimension of parameters. To add an object, a position is selected according to a point process and the parameters of the object are uniformly chosen from the parameter space. Theoretically, this kernel is sufficient to simulate a marked point process. However, in terms of increasing the convergence speed, it makes the algorithm more efficient to use other types of kernels also.

- *Non-jumping kernel*: A non-jumping kernel allows a small perturbation such as translation, dilation and rotation. This does not involve jumping of a dimension but change the value of parameters.
- *Switching kernel*: A switching kernel allows to change the model of an object. For example, a circle model, described by a position and a radius of objects as a parameter, can be switched to a rectangle model which has a position, width, length and a rotation angle of objects as its parameters. We can see that the number of objects is not changed in this case, but the dimensional jumping is involved. But, in other cases that both models of switching objects have the same number of parameters, the dimension of parameters may not be changed by this kernel.

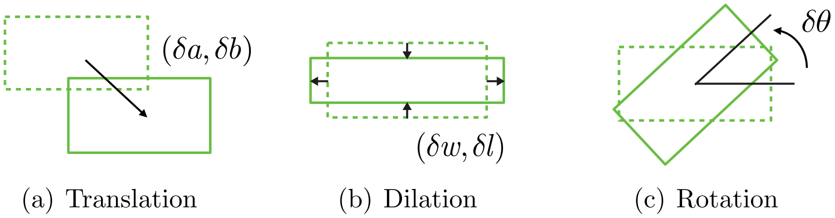
Proposition Kernels

Birth-and-death The Green's ratio of birth for the transition from a current configuration $\mathbf{x}$ to a new configuration $\mathbf{x}'$ is given by

$$R(\mathbf{x}, \mathbf{x}') = \frac{p(\mathbf{x}')}{p(\mathbf{x})} \frac{p_D}{p_B} \frac{\nu(K)}{n(\mathbf{x}) + 1}. \tag{3.24}$$

where $n(\mathbf{x})$ is the number of channels in a configuration $\mathbf{x}$. And $p_B = 1 - p_D$ is the probability of choosing a birth, and p_D is the probability of choosing a death. For the uniform Poisson process, $\nu(K)$ is $\lambda|K|$ and λ is an intensity of Poisson process. And the Green's ratio of death is given by

$$R(\mathbf{x}, \mathbf{x}') = \frac{p(\mathbf{x}')}{p(\mathbf{x})} \frac{p_B}{p_D} \frac{n(\mathbf{x})}{\nu(K)}. \tag{3.25}$$



(a) Translation (b) Dilation (c) Rotation

Fig. 3.6. Non-jumping perturbation

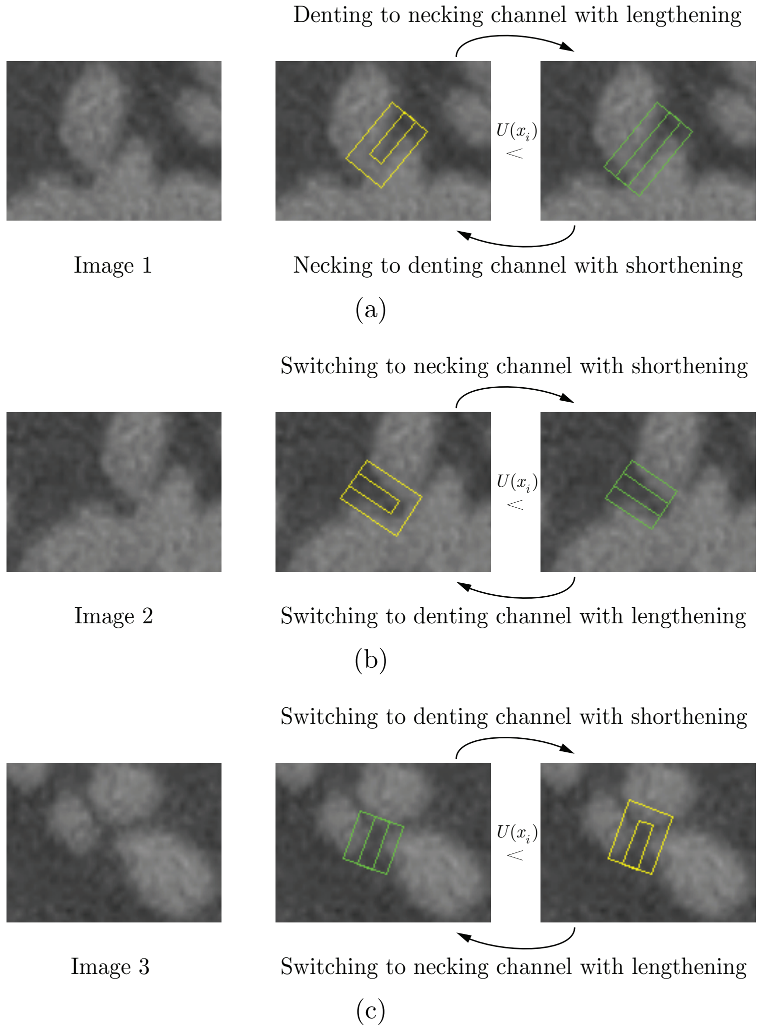
Non-jumping (Translation, Dilation, Rotation) Figure 3.6 shows this type of kernel. Each of these transformations uses parameters that are randomly chosen as follow:

$$\begin{aligned}\delta a &\in [-\Delta a, \Delta a]\\ \delta b &\in [-\Delta b, \Delta b]\\ \delta l &\in [-\Delta l, \Delta l]\\ \delta w &\in [-\Delta w, \Delta w]\\ \delta\theta &\in [-\Delta\theta, \Delta\theta]\end{aligned} \tag{3.26}$$

We set the values of Δa, Δb, Δl and Δw as all 1.5 and the value of $\Delta\theta$ as 0.02. The Green's ratio of these move types is simply given by the following formula:

$$R(\mathbf{x}, \mathbf{x}') = \frac{p(\mathbf{x}')}{p(\mathbf{x})} \tag{3.27}$$

Switching Figure 3.7 shows some channels in the images and two instances of their detected channel objects. In those instances, the left ones are more desirable than the right ones. But the right channels still have attractive potentials even though their potentials are higher than those of left ones. Therefore, to increase convergence speed and improve the quality of final detection results, we need to encourage the right channel objects to be switched to the left ones. Taking a closer look at the figure, we can see that these switchings should accompany translation and dilation as well. Therefore, switching kernels should be carefully designed by taking these


Denting to necking channel with lengthening
$U(x_i)$
$<$
Image 1
Necking to denting channel with shorthening
Switching to necking channel with shorthening
$U(x_i)$
$<$
Image 2
Switching to denting channel with lengthening
Switching to denting channel with shorthening
$U(x_i)$
$<$
Image 3
Switching to necking channel with lengthening

(a)

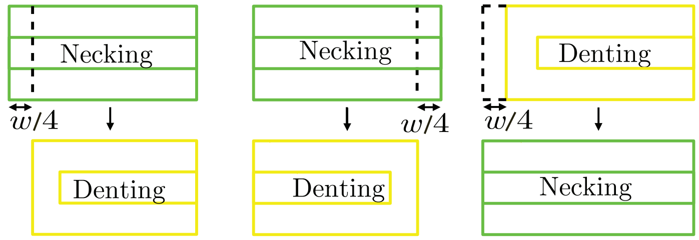
(b)

(c)

Fig. 3.7. Switching with shortening and lengthening

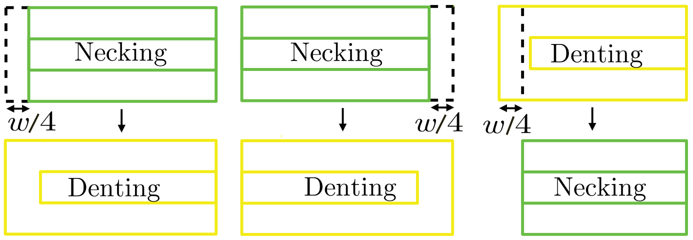
perturbations into account. Considering this translation and dilation, two types of reversible switching can be constructed. The first type is depicted in Figure 3.8 which can be divided further into three subtypes. And the second type is in Figure 3.9 which can be also divided into three subtypes.

For example, let a denting channel $x_i = (a_i, b_i, l_i, w_i, \theta_i)$ be switched to a necking channel $\hat{x}_i$ according to the 'switching type 1' as Figure 3.8(c). The length l_i is



(a) case 1 (b) case 2 (c) case 3

Fig. 3.8. Switching type 1. (a)(b) Necking to denting channel with shortening. (c) Denting to necking channel with lengthening.



(a) case 1 (b) case 2 (c) case 3

Fig. 3.9. Switching type 2. (a)(b) Necking to denting channel with lengthening. (c) Denting to necking channel with shortening.

increased about $\frac{w_i}{4}$ from the closing end and the center position (a_i, b_i) is also changed accordingly as below:

$$\hat{x}_i = \begin{bmatrix} a_i - \frac{w_i}{8}\cos(\theta_n) \\ b_i - \frac{w_i}{8}\sin(\theta_n) \\ l_i + \frac{w_i}{4} \\ w_i \\ \theta_i \end{bmatrix}^{\mathrm{T}} \tag{3.28}$$

Then, the Green's ratio is given by:

$$R(\mathbf{x}, \mathbf{x}') = \frac{p(\mathbf{x}')}{p(\mathbf{x})} \frac{Q_S(\mathbf{x}' \to \mathbf{x})}{Q_S(\mathbf{x} \to \mathbf{x}')} = \frac{p(\mathbf{x}')}{p(\mathbf{x})} \tag{3.29}$$

The computation of the ratio $\frac{Q_S(\mathbf{x}' \to \mathbf{x})}{Q_S(\mathbf{x} \to \mathbf{x}')}$ is detailed in Appendix D.

3.3.2 Curved Channel Method

In this method, the channel segment object $n_i = (l_i, w_i, \theta_i)$ is defined on a space $M = [L_{min}, L_{max}] \times [W_{min}, W_{max}] \times [0, \pi]$, where l_i is the length, w_i is the width and θ_i is the rotation angle of the segment x_i. The channel shape follows a simple channel in Figure 3.2(a). Then, for the area A_c^i and A_{nc}^i in Figure 3.10, the data potential can be calculated with (3.7) and (3.13). For the interaction between objects, we use a Quality Candy model which is widely used in detecting line networks [16].

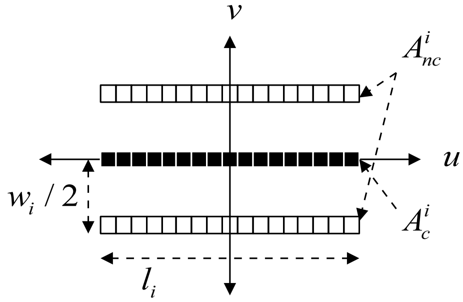


Fig. 3.10. Pixel mask of a simple channel

Quality Candy prior model

In the Quality Candy model, a free segment, a single segment and a double line segment are defined as Figure 3.11. If the endpoints of the two segments are close and the distance is less than ε, then the endpoints are identified as a connected endpoint for the each segment. The free segment does not have any connected end

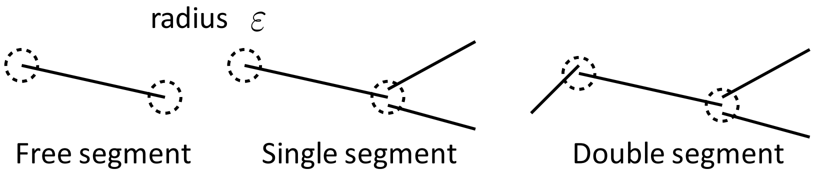


Fig. 3.11. Segment types

points. The single segment has only one connected end point. Finally, the double segment has two connected endpoints. In the original Candy model [35], the internal bad orientation relation and the external bad orientation relation are defined, which penalize undesirable formations between two line segments However, in the Quality Candy model, the potential function $g_{R_c}(x_i, x_j)$ is defined for each pair of connected segments instead of the external bad orientation relation used in the Candy model. This potential can reflect the degree of the undesirable relationship between two connected segments. In addition, the internal bad orientation potential function $g_{R_{io}}(x_i, x_j)$ is defined to penalize the internal bad orientation relationship. R_c is the set of connected segment pairs and R_{io} is the set of segment pairs which have the internal bad orientation relation. Finally, the Gibbs energy of the Quality Candy model is given by [16]:

$$\begin{aligned} U(\mathbf{x}|\mathbf{y}) =& \lambda_o \sum_{i=1}^{n} V_o(\mathbf{y}|x_i) + \omega_f n_f + \omega_s n_s + \omega_d n_d + \omega_c \sum_{<x_i,x_j>\in R_c} g_{R_c}(x_i, x_j) \\ &+ \omega_{eo} \sum_{<x_i,x_j>\in R_{io}} g_{R_{io}}(x_i, x_j) \end{aligned} \tag{3.30}$$

where n_f, n_s and n_d are the number of free segments, of single segments and of double segments respectively, and λ_o, ω_f, ω_s, ω_d, ω_c and ω_{eo} are control parameters. As an optimization method and proposal kernels, we follow the proposals in [16] basically. However, we add one more kernel which dilates the width w_i of a channel object.

3.4 Controlling the Interaction Parameter in EM/MPM Segmentation Algorithm

In EM/MPM segmentation method, the posterior pmf is given as

$$\begin{aligned} & p_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z}|\mathbf{y},\boldsymbol{\theta}) \\ & \quad = \frac{1}{z_{seg} f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \left[\prod_{r=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{z_r}^2}} \right] \cdot \exp\left(-\sum_{r=1}^{N} \frac{(y_r - \mu_{z_r})^2}{2\sigma_{z_r}^2} - \sum_{\{r,s\}\in\mathcal{C}} \beta_{seg} t(z_r, z_s) \right) \end{aligned} \tag{3.31}$$

where $\mathbf{y}$ is an input image, $\mathbf{z}$ is segmented labels and z_r is a label at the position r. And function $t(z_r, z_s)$ is given in (2.7). Here, β_{seg} is an interaction parameter which controls the amount of neighboring pixel merging in the segmentation results. And this parameter is selected uniformly for the entire image. Thus, the result shows some bridged channel defects across channels. To reduce these defects, we propose to control the interaction parameter β_{seg} non-uniformly according to the channel configuration by replacing an interaction parameter β_{seg} with $\tilde{\beta}(z_r, a, b)$:

$$\begin{cases} \beta_{seg} \min_{i=0}^{n} \tilde{\beta}_i(a,b) & z_r = 1 \text{ (object)} \\ \beta_{seg}, & z_r = 0 \text{ (background)} \end{cases} \tag{3.32}$$

And $\tilde{\beta}_i(a,b)$ is constructed from ith channel object and given by:

$$\tilde{\beta}_i(a,b) = \hat{\beta}\left(u, v\,;\, l_i, w_i\right) \tag{3.33}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = R(\theta_i) T(-a_i, -b_i) \begin{bmatrix} a \\ b \\ 1 \end{bmatrix} \tag{3.34}$$

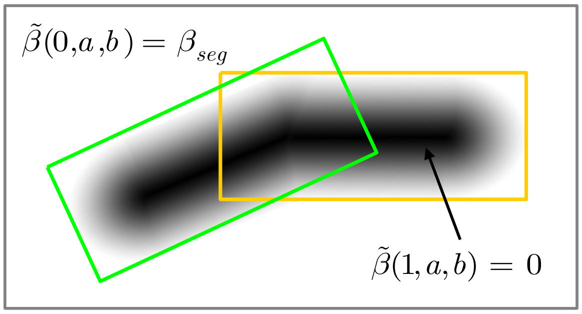

$\tilde{\beta}(0,a,b) = \beta_{seg}$
$\tilde{\beta}(1,a,b) = 0$

Fig. 3.12. Controlling the interaction parameter

And $\hat{\beta}(u, v\,;\, l, w)$ defined on $\mathbb{R}^2$ is given by:

$$\begin{cases} \frac{1-\exp(-\tau_s v^2)}{1-\exp(-\tau_s \frac{w^2}{4})}, & |u| \leq \frac{l}{2} - \frac{w}{2},\ |v| \leq \frac{w}{2} \\ \frac{1-\exp(-\tau_s((u-\frac{w}{2}+\frac{l}{2})^2+v^2))}{1-\exp(-\tau_s \frac{w^2}{4})}, & u < -\frac{l}{2} + \frac{w}{2},\ |v| \leq \frac{w}{2} \\ \frac{1-\exp(-\tau_s((u+\frac{w}{2}-\frac{l}{2})^2+(v^2))}{1-\exp(-\tau_s \frac{w^2}{4})}, & \frac{l}{2} - \frac{w}{2} < u,\ |v| \leq \frac{w}{2} \\ 1, & \text{elsewhere} \end{cases} \tag{3.35}$$

And the amount of separation is controlled by τ_s. And Figure 3.12 illustrates one example of $\tilde{\beta}(z_r, a, b)$. Let the channel configuration be given as rectangles in the figure, then the $\tilde{\beta}(1, a, b)$ gradually becomes zero to the center of channel. This setting of β_s discourages merging of the pixels across these channels.

3.5 Experimental Results

In this section, we present visual and numerical results obtained by using the proposed channel modeling methods and a method controlling interaction parameter. The proposed two channel modeling methods have been tested on four sets of microscope images of materials including a Ni-Al-Cr superalloy image which is shown in Figure 3.13(a) and René88 DT images whose slice numbers are 17, 170 and 1 as shown in Figure 3.14(a), Figure 3.15(a) and Figure 3.16(a). By using the final channel configuration obtained from those methods, we segment the material images which have bridged channel defects in original EM/MPM segmentation results. In the end, we can see that the new segmentation results are much better by successfully reducing these misclassifications caused by the bridged channel defect.

3.5.1 Dataset 1: Ni-Al-Cr Superalloy

The Ni-Al-Cr superalloy material image and its results are shown in Figure 3.17. The resolution of this test image is 194 by 149 pixel. First, we apply the necking and denting channel method. However, there are only necking channels in this test image. Therefore, we use only a necking channel model. In addition, all channels are aligned to two directions in the original image. Therefore, we use this directional information in the result in Figure 3.13(e) to fix θ with 0.2π, 0.7π without using a rotation kernel. Secondly, we perform the curved channel method for this dataset. The results of these two channel detection methods in Figure 3.13(d), Figure 3.13(e) and Figure 3.13(f) show that most of channels are well detected. When we compare those two methods, the curved channel method shows the better performance in convergence speed even though the shape of channels is not a curve.

We use this channel configuration information to reduce the bridged channel defects in the segmentation of material image. By controlling the interaction parameter according to this channel configuration, we can get the improved segmentation results as shown in Figure 3.13(g), Figure 3.13(h) and Figure 3.13(i). In Figure 3.13(c), class 1 (white area) objects of the original EM/MPM segmentation result are so expanded that they have many bridged channel defects. Encouragingly, the results show that our proposed method can eliminate these bridged channel defects effectively. Usually, a percentage of the misclassified pixels(PMP) can be used to evaluate the performance of a segmentation result numerically. The lower PMP value means the better result. As shown in Figure 3.13(h), the best PMP value of the results using our methods is 2.73% while the result using the original EM/MPM segmentation algorithm is 6.59%.

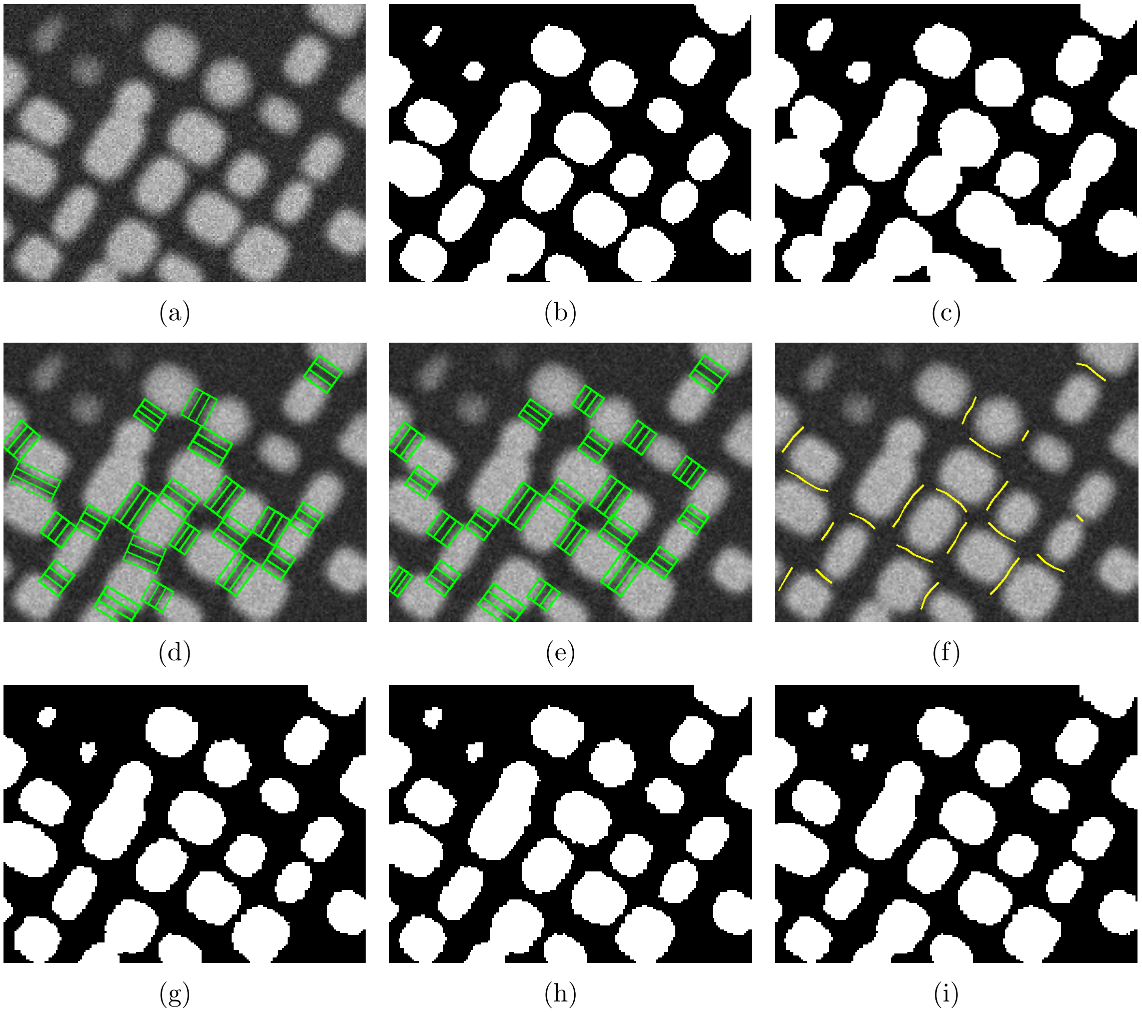


Fig. 3.13. Results of NiAlCr superalloy: (a) test image, (b) ground truth of segmentation, (c) Original EM/MPM (PMP=6.59%), (d) Necking channels (running time 54sec), (e) Necking channels (θ fixed, running time 16sec), (f) curved channels (running time 14sec), (g) EM/MPM after detecting necking channels (PMP=2.74%), (h) EM/MPM after detecting necking channels (θ fixed, PMP = 2.73%), (i) EM/MPM after detecting curved channels (PMP=2.81%), Interaction parameters of all the EM/MPM results are set as 2.7

3.5.2 Dataset 2: René88

The second material image group and their results are shown in Figure 3.14, Figure 3.15 and Figure 3.16. The resolution of all these test images is 194 by 149 pixel. First, we apply the necking and denting channel method. Compare to the Ni-Al-Cr superalloy dataset, the channel size of these test images is much smaller. And their formation is more complicated than the previous material image. Furthermore, they include both necking and denting channels. Despite those hard conditions, the results in Figure 3.14(d) and Figure 3.15(d) are promising that most of channels are successfully detected except some missed or over-detected channels. Especially, the switching kernel works well even though the energy difference of a necking channel and a denting channel is not so big at some channel positions in the image. However, the Figure 3.16(b) shows some misdetection because the original image includes some curved channels. In this case, the curved channel method is more apt to detect those channels. Figure 3.14(e) is a channel detection result when we do not use continuity potential. Compared with Figure 3.14(d) which uses continuity potential, it shows that the continuity potential effectively reduce misidentifying necking channels as denting channels. We also apply this channel information to improve the segmentation result. To do this, as shown in Figure 3.14(h), we control the interaction parameter value in the existing EM/MPM segmentation algorithm according to the channel configuration. In this figure, $\tilde{\beta}_i(a,b)$ is normalized to [0,255] to visualize it. To compare between Figure 3.14(c) and Figure 3.14(f), we can see that many bridged channel defects are reduced. In general, it is hard to reduce the defects of denting channel points. However, our results are encouraging as showing that the proposed method can eliminate these defects effectively.

3.5.3 Dataset 3 and Dataset 4

The third dataset presented in Figure 3.17(a) is a Co-Al-W alloy image. And fourth material images is presented in Figure 3.18(a). The resolution of the dataset 3 is 361 by 378 pixel. And dataset 4 has 142 by 215 pixel resolution. The dataset 3 is difficult to get the precise object boundary through the conventional segmentation method as shown in Figure 3.17(d). However, the results in Figure 3.17(e) and Figure 3.17(f) demonstrate that our channel detection methods are very useful to improve the bridged channel defect in the conventional segmentation results. Comparing the Figure 3.17(e) and Figure 3.17(f), the necking and denting channel method shows

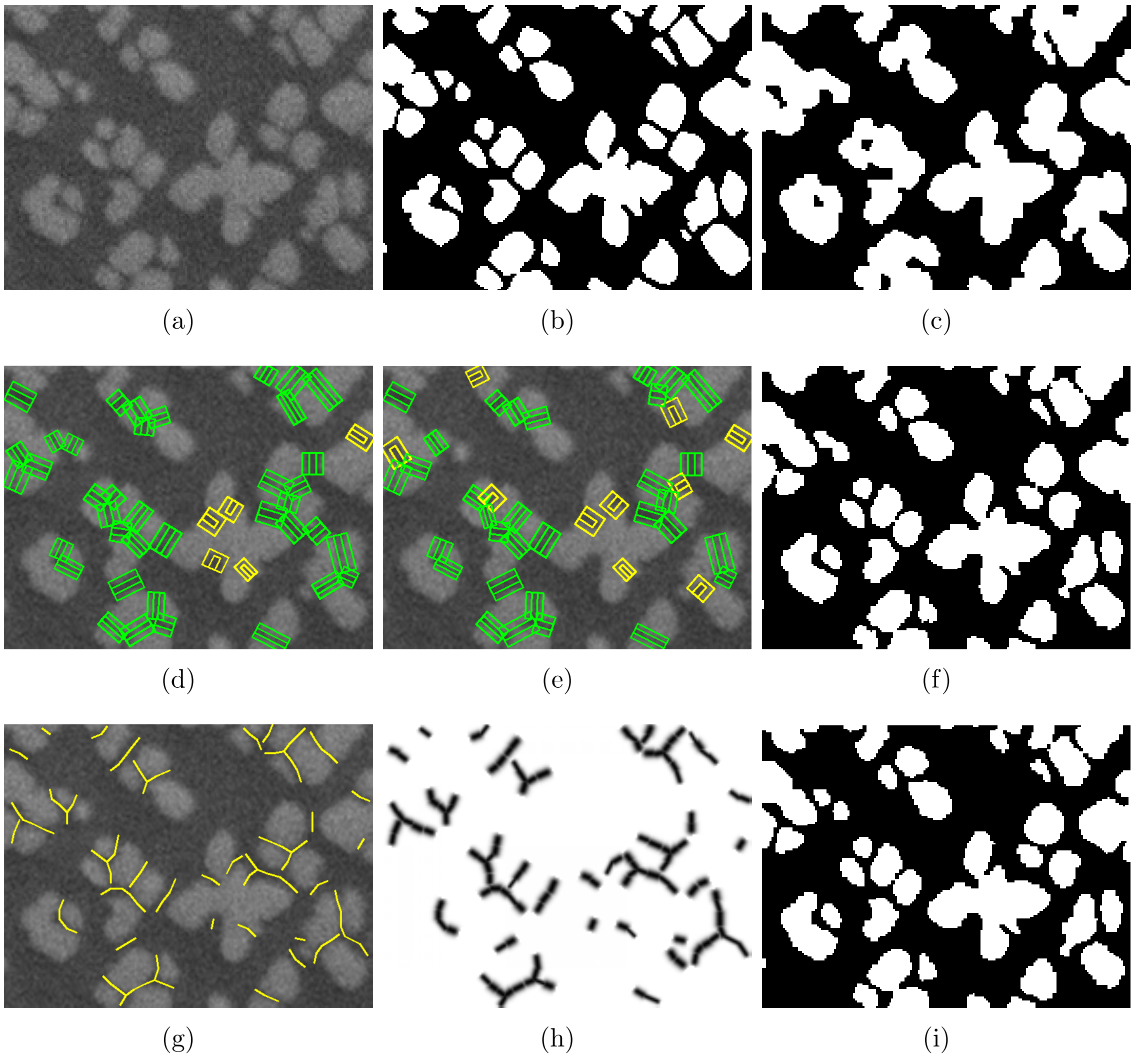


Fig. 3.14. Result images of slice number 17: (a) test image, (b) ground truth of segmentation, (c) Original EM/MPM (PMP=6.21%), (d) Necking and denting channels (running time 86sec), (e) Necking and denting channels without continuity potential (running time 87sec), (f) EM/MPM after detecting necking and denting channels (PMP=4.42%), (g) curved channels (running time 40sec), (h) $\tilde{\beta}(1, a, b)$ of curved channel, (i) EM/MPM after detecting curved channels (PMP=4.44%), Interaction parameters of all the EM/MPM results are set as 2.9

better quality in some object boundaries. We fix θ with three values according to $P(\theta = 0.31\pi) = 0.4$, $P(\theta = 0.77\pi) = 0.4$ and $P(\theta = 0.07\pi) = 0.2$ in executing the necking and denting channel method. The fourth material image has much small and

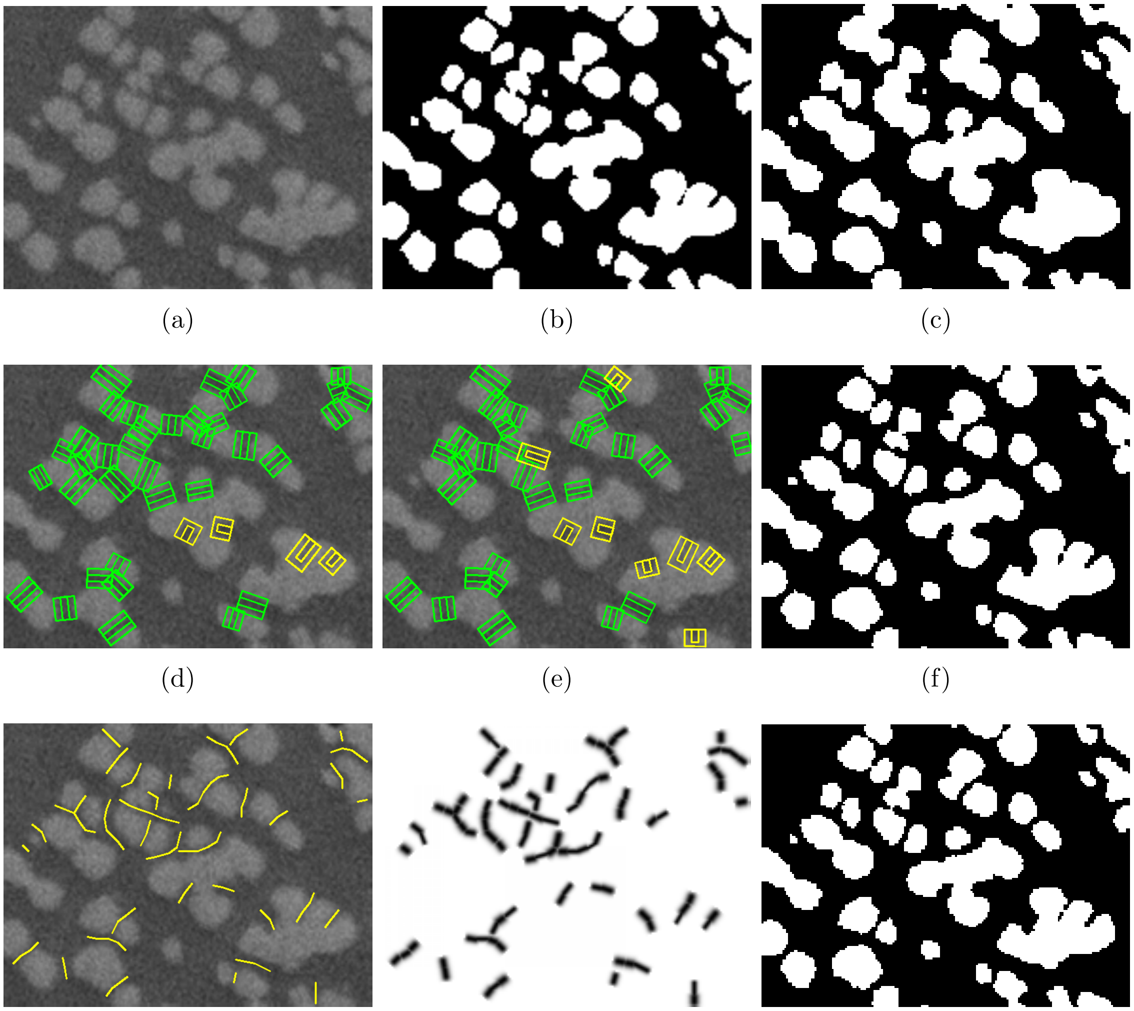
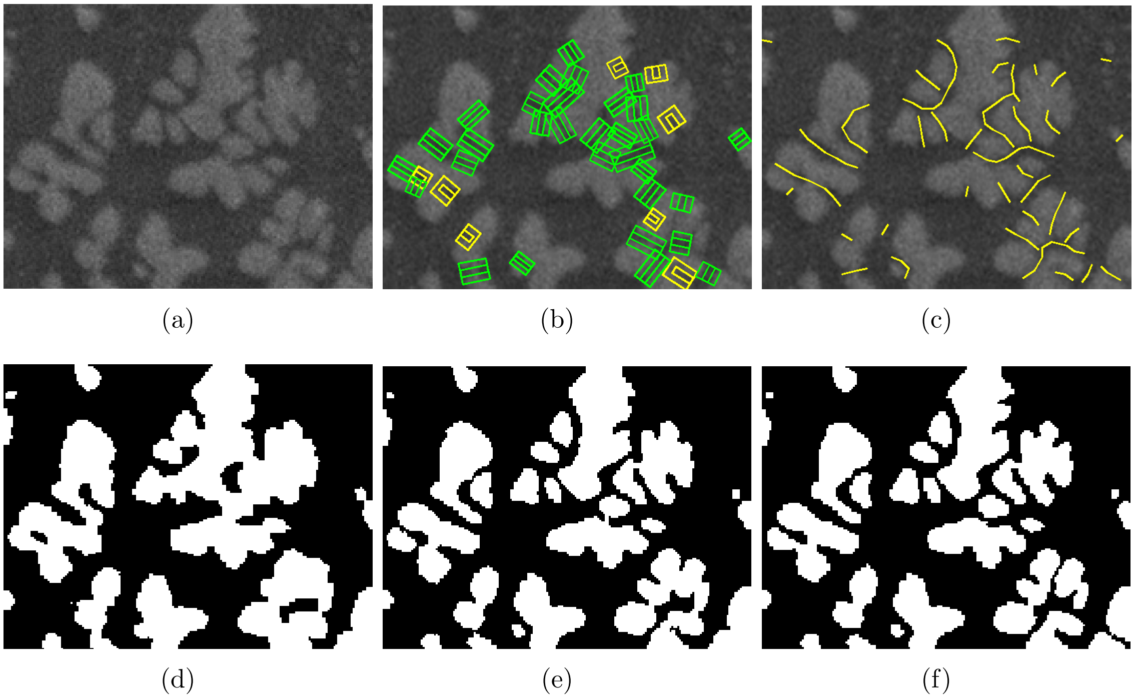


Fig. 3.15. Result images of slice number 170: (a) test image, (b) ground truth of segmentation, (c) Original EM/MPM (PMP=5.52%), (d) Necking and denting channels (running time 87sec), (e) Necking and denting channels without continuity potential (running time 81sec), (f) EM/MPM after detecting necking and denting channels (PMP=3.87%), (g) curved channels (running time 31sec), (h) $\tilde{\beta}(1, a, b)$ of curved channel, (i) EM/MPM after detecting curved channels (PMP=3.91%), Interaction parameters of all the EM/MPM results are set as 2.9

complicated channels. However, the curved channel method successfully detects most of channels.

When we compare overall performance between the necking and denting channel method and the curved channel method, the curved channel method shows better


(a) (b) (c)

(d) (e) (f)

Fig. 3.16. Result images of slice number 1: (a) test image, (b) Necking and denting channels (running time 111sec), (c) curved channels (running time 35sec), (d) Original EM/MPM, (e) EM/MPM after detecting necking and denting channels, (f) EM/MPM after detecting curved channels, Interaction parameters of all the EM/MPM results are set as 2.9

performance in that it can detect curved channels as well as straight channels well and the convergence speed is faster than its counterpart. However, to get reasonable result, the curved channel method need a little effort to tune the many parameters of quality candy model. In addition, the curved method can not discriminate between a necking channel and a denting channel.

3.6 Conclusions

We have presented two channel modeling methods based on the MPP framework to detect channels in microscopic images of materials. One is a curved channel method, and the other is a necking and denting channel method. These methods are designed as follows. Firstly, we defined three types of channel objects which reflect the shape of each channel type. Next, to get a data energy, we calculated the statistical differ-

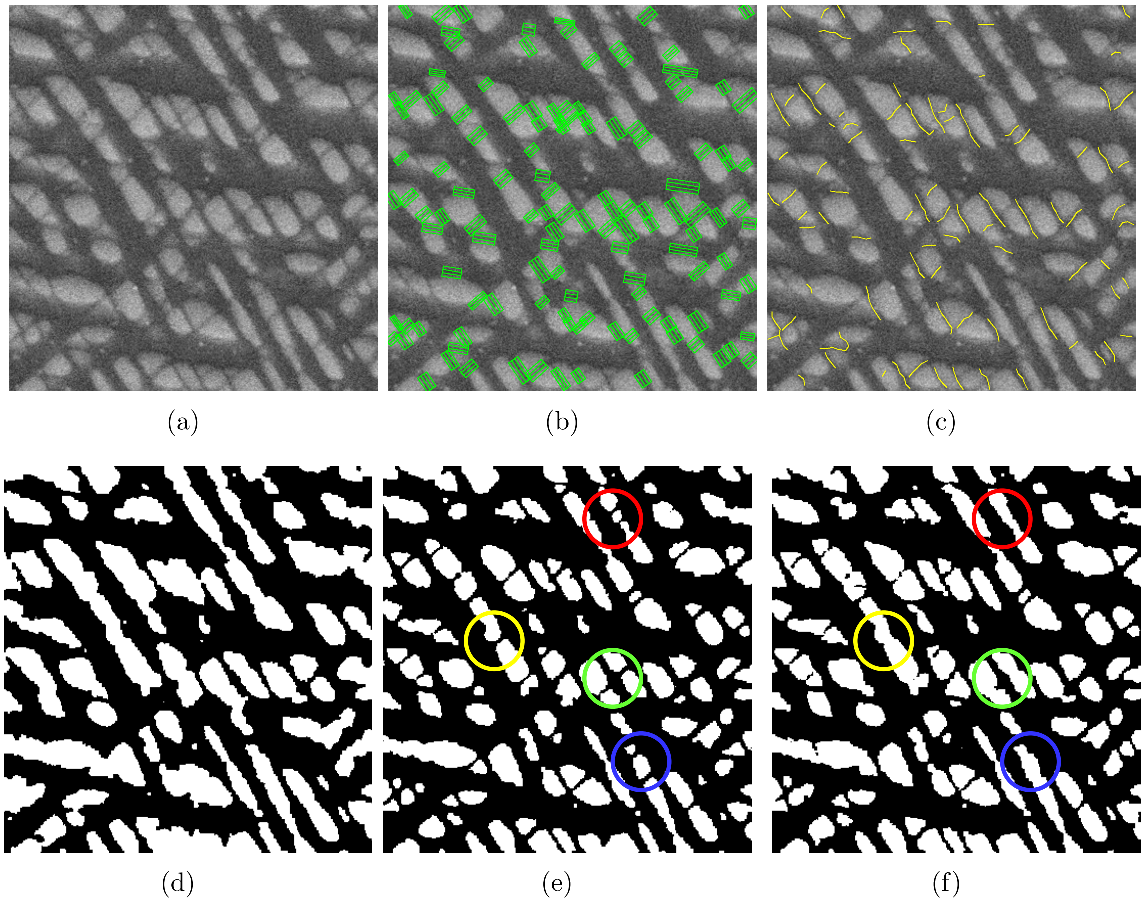
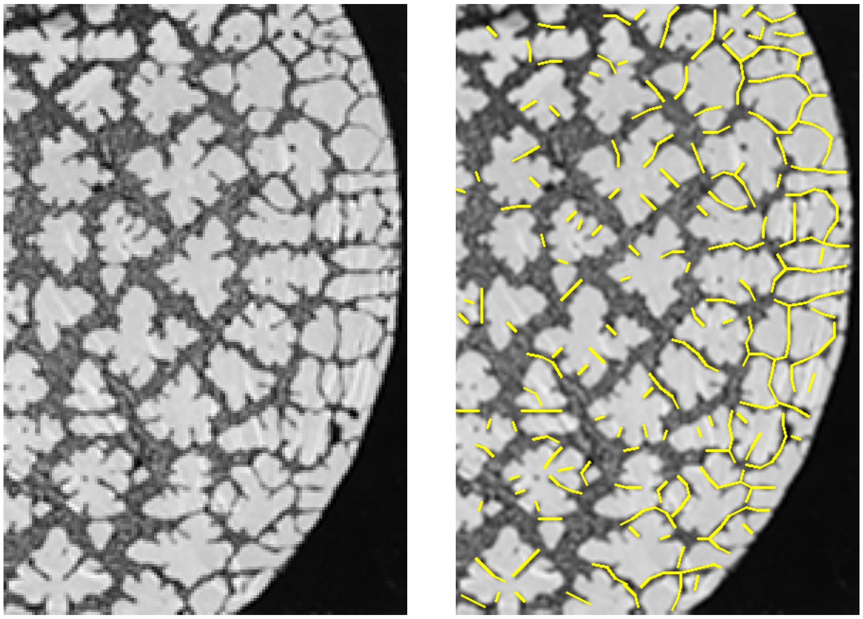


Fig. 3.17. Results of the dataset 3: (a) test image, (b) Necking channels (θ fixed, running time 140sec), (c) curved channels (running time 113sec), (d) Original EM/MPM, (e) EM/MPM after detecting necking channels (θ fixed), (f) EM/MPM after detecting curved channels. Interaction parameters of all the EM/MPM results are set as 2.7

ence between the channel object and the intensity of a channel in the image. Then, Gibbs energy is given as a sum of the data energy and the prior energy. This Gibbs energy is minimized by the RJMCMC with a simulated annealing scheme to find the most probable channel configuration. This channel configuration can be useful in analyzing material images. By using this information, we reduced the bridged channel defects in the segmentation results of material image. In addition, a method of how to control the interaction parameter in the MRF model of the existing EM/MPM segmentation algorithm has been also introduced. Finally, the experimental results demonstrated that both channel modeling methods and the controlling method of the



(a) Test image

(b) Final configuration of channels

Fig. 3.18. Result of the dataset 4: (a) test image, (b) curved channels (running time 235sec)

EM/MPM segmentation algorithm are effective for detecting channels and reducing the segmentation defects.

We can make some suggestions for further research based on implications and limitations of this chapter. First of all, mainly due to the noise in the image, some misidentified channels still remain in the result. To consider that the material images used in our experiment are serial sectioned images of materials which have correlations with their consecutive image slices, the accuracy of channel detection can be increased by reducing the effect of the noise using these correlations. Secondly, in order to increase convergence speed, it is recommendable to use an adaptive cooling schedule in a simulated annealing [36] or a jump-diffusion process in the RJMCMC [37]. Or, for the same purpose, we can use a non-homogeneous birth and death kernel which generates the channels only in the relevant positions so that the computational time can be reduced. Lastly, we can also follow up with the study on reducing the number of tuning parameters.

4. THE MARKED POINT PROCESS FOR 3D DATASETS

4.1 Introduction

To detect a 3D object configuration in a 3D dataset, in this chapter, we propose two methods. We will call these methods the 3D clustering method and the fast 3D fitting method. Both methods are composed of a two-step approach which involves the following procedures: Detectiin a 2D object configuration and detecting 3D objects based on the 2D object configuration. However, the difference of the two methods is in how the 3D objects are detected. The 3D clustering method detects a 3D object by matching 3D objects to the 2D objects as illustrated in Figure 4.1. However, in the fast 3D fitting method, we first construct a birth map from the 2D configuration and then apply a 3D MPP to the original 3D dataset with this birth map. Figure 4.2 represents these steps. The birth map enables us to generate 3D object candidates that are most likely to fit the data. Therefore, we can reduce the computation time of the full 3D MPP method.

In the MPP approach, we define an energy function that reflects prior knowledge about the object model itself and a consistency between the object configuration and the data. Then, this energy function is minimized with an optimization method. As an optimization method, Multiple Birth and Death(MBD) [12] and Reversible jump Markov chain Monte Carlo(RJMCMC) [29] methods are widely used. Recently, Gamal Eldin et al. [38] apply graph cuts to find the optimal configuration. This method is called multiple birth and cut(MBC). In this chapter, we use a MBC method as an optimization method.

To evaluate the performance of this proposed method, two synthetic image sequences, a materials image sequence and a brain tissue image sequence are analyzed with the proposed method. In this experiment, we use superellipsoid and a new elliptical cylinder models as a mark model for the MPP. For 2D objects, a superellipse model is used. The superellipsoid model is illustrated in Figure 4.3.

This chapter is organized as follows: Firstly, in Section 4.2, the proposed 3D clustering and the related optimization method using multiple birth and cut (MBC) are presented. Then the fast 3D fitting method and the related birth map is proposed. In Section 4.4, after our proposed method is tested through a series of experiments, the results are shown with detailed discussion.

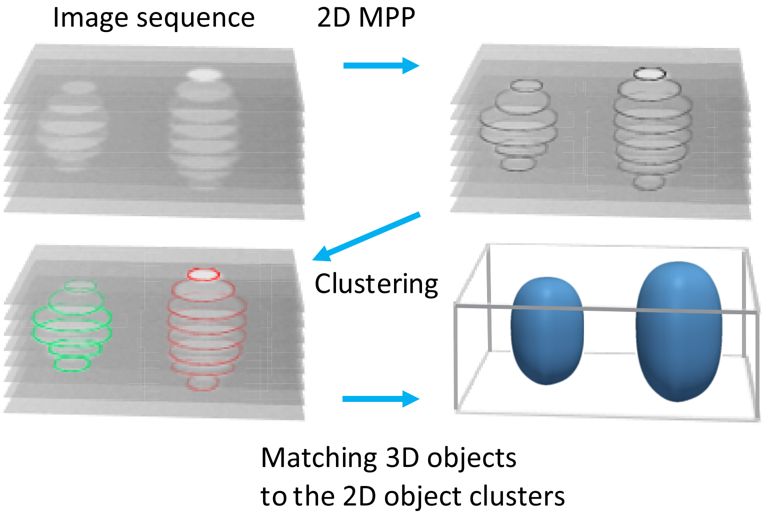


Fig. 4.1. 3D clustering method

4.2 3D Clustering Method

In this chapter, we use a superellipse mark as a 2D object model to identify the cross section of the 3D object. In this case, $w_i = (p_i, a_i, b_i, \theta_i)$ is the i-th superellipse disk object defined on $M \times K$. Here, the center position $p_i = (p_i^x, p_i^y, p_i^z)$ is defined on a 3D lattice $K = [0, X_{max}] \times [0, Y_{max}] \times [0, Z_{max}]$. $a_i \in [a_{min}, a_{max}]$ is the major semi-axis length, $b_i \in [b_{min}, b_{max}]$ is the minor semi-axis length. These disks are parallel to x-y plane and $\theta_i \in [\theta_{min}, \theta_{max}]$ is the orientation of this superellipse object. The direction of rotation is counter-clockwise around z direction. Then, the mark space is given as $M = [a_{min}, a_{max}] \times [b_{min}, b_{max}] \times [\theta_{min}, \theta_{max}]$. The total Gibbs energy $U(\mathbf{w}|\mathbf{y})$,

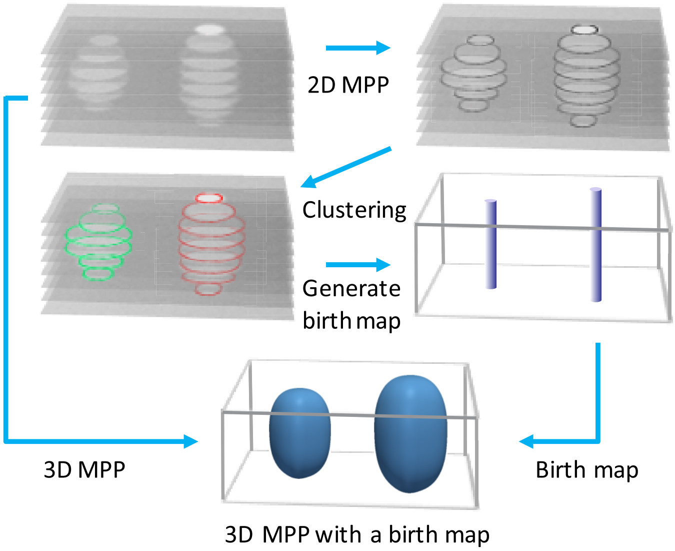


Fig. 4.2. Fast 3D fitting method

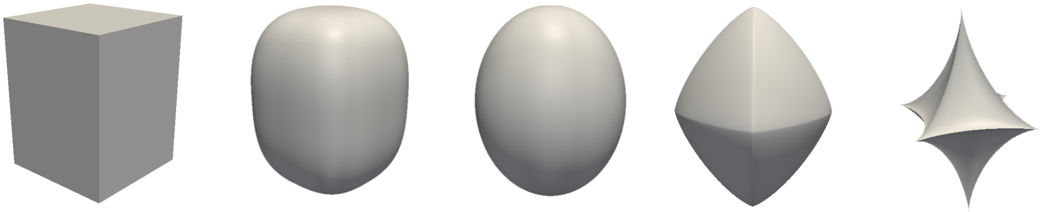


Fig. 4.3. Superellipsoids

which represents the cost of configuration $\mathbf{w} = \{w_1, w_2, \cdots, w_n\}$, can be expressed as:

$$U(\mathbf{w}|\mathbf{y}) = \sum_{i=1}^{n} V_d(\mathbf{y}|w_i) + \lambda \sum_{w_i \sim w_j} V_{in}(w_i, w_j) \tag{4.1}$$

where V_d is a data potential, V_{in} is an interaction potential and λ is a control parameter. $w_i \sim w_j$ is a neighboring relation which means that the area of w_i intersects with the area of w_j and $p_i^z = p_j^z$. As the interaction potential, we use the potential proposed in [32] to penalize the overlapping of two objects.

$$V_{in}(w_i, w_j) = \begin{cases} 0 & O(w_i, w_j) = 0 \\ \infty & O(w_i, w_j) > 0, p_i^z = p_j^z \end{cases} \tag{4.2}$$

where

$$O(w_i, w_j) = \frac{A(w_i \cap w_j)}{\min(A(w_i), A(w_j))} \tag{4.3}$$

As a 3D object model, we use a superellipsoid mark model. $w_i^{3D} = (p_i^{3D}, a_i^{3D}, b_i^{3D}, c_i^{3D},$ $\alpha_i, \beta_i, \gamma_i)$ are superellipsoids located in $p_k^{3D} = (p_i^x, p_i^y, p_i^z) \in K$. $(a_i^{3D}, b_i^{3D}, c_i^{3D}) \in [a_{min}^{3D}, a_{max}^{3D}] \times [b_{min}^{3D}, b_{max}^{3D}] \times [c_{min}^{3D}, c_{max}^{3D}] = L$ are semi-axis lengths. $(\alpha_i, \beta_i, \gamma_i) \in [\alpha_{min}, \alpha_{max}] \times [\beta_{min}, \beta_{max}] \times [\gamma_{min}, \gamma_{max}] = O$ are orientation angles of w_i^{3D}. The mark space is given as $M^{3D} = L \times O$. As a rotation convention, z-y-x extrinsic rotation is used.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{4.4}$$

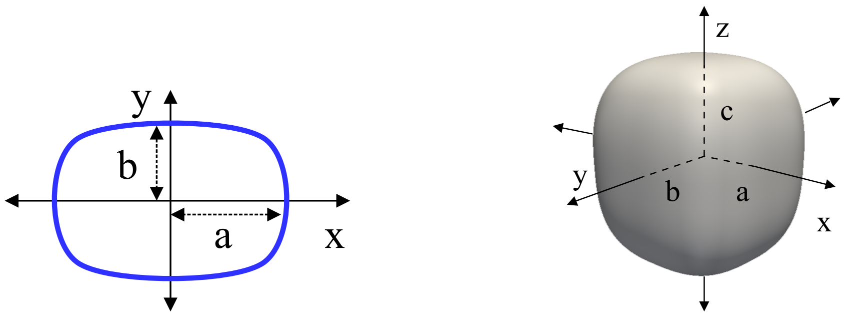
$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{4.5}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

$$\begin{aligned} R &= R_x(\alpha) R_y(\beta) R_z(\gamma) \\ R^{-1} &= R_z(-\gamma) R_y(-\beta) R_x(-\alpha) \end{aligned} \tag{4.7}$$

4.2.1 Superellipse and Superellipsoid

Superellipse



(a) superellipse

(b) superellipsoid

Fig. 4.4. Superellipse and superellipsoid

Let a superellipse for the given marked point w_i be expressed with an equation

$$f_r(x, y; w_i) = 1 \tag{4.8}$$

or an equation with parametric form defined on $u \in [0, 2\pi]$ as

$$\mathbf{s}_r(u; w_i) = (x_r(u; w_i), y_r(u; w_i)). \tag{4.9}$$

When $(p_i^x, p_i^y) = (0, 0)$ and $\theta_i = 0$,

$$f_r(x, y; w_i) = \left|\frac{x}{a_i}\right|^r + \left|\frac{y}{b_i}\right|^r = 1 \tag{4.10}$$

$$\begin{aligned} x_r(u; w_i) &= a_i |\cos(u)|^{\frac{2}{r}} \cdot \operatorname{sgn}(\cos(u)) \\ y_r(u; w_i) &= b_i |\sin(u)|^{\frac{2}{r}} \cdot \operatorname{sgn}(\sin(u)) \end{aligned} \tag{4.11}$$

where

$$\operatorname{sgn}(u) = \begin{cases} -1 & u < 0 \\ 0 & u = 0 \\ 1 & u > 0 \end{cases} \tag{4.12}$$

An outward surface normal is given as

$$\mathbf{n}_r(u; w_i) = \left(\frac{\mathrm{d}y_r(u; w_i)}{\mathrm{d}u}, -\frac{\mathrm{d}x_r(u; w_i)}{\mathrm{d}u} \right) \tag{4.13}$$

And $\bar{\mathbf{n}}_r(u; w_i)$ is the normalized outward surface normal.

Superellipsoid

Let a superellipse for the given marked point w_i^{3D} be expressed with an equation

$$f_{t,r}(x, y, z; w_i^{3D}) = 1 \tag{4.14}$$

or an equation with parametric form defined on $u \in [0, 2\pi]$ and $v \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ given by

$$\mathbf{s}_{t,r}(u, v; w_i^{3D}) = (x_{t,r}(u, v; w_i^{3D}), y_{t,r}(u, v; w_i^{3D}), z_{t,r}(u, v; w_i^{3D})) \tag{4.15}$$

When $(p_i^x, p_i^y, p_i^z) = (0, 0, 0)$ and $(\alpha_i, \beta_i, \gamma_i) = (0, 0, 0)$,

$$f_{t,r}(x, y, z; w_i^{3D}) = \left(\left| \frac{x}{a_i^{3D}} \right|^r + \left| \frac{y}{b_i^{3D}} \right|^r \right)^{\frac{t}{r}} + \left| \frac{z}{c_i^{3D}} \right|^t - 1 = 0 \tag{4.16}$$

$$\begin{aligned} x_{t,r}(u,v;w_i^{3D}) =& a_i^{3D}|\cos(u)|^{\frac{2}{r}} \cdot \mathrm{sgn}(\cos(u)) \\ & \cdot |\cos(v)|^{\frac{2}{t}} \cdot \mathrm{sgn}(\cos(v)) \\ y_{t,r}(u,v;w_i^{3D}) =& b_i^{3D}|\sin(u)|^{\frac{2}{r}} \cdot \mathrm{sgn}(\sin(u)) \\ & \cdot |\cos(v)|^{\frac{2}{t}} \cdot \mathrm{sgn}(\cos(v)) \\ z_{t,r}(u,v;w_i^{3D}) =& c_i^{3D}|\sin(v)|^{\frac{2}{t}} \cdot \mathrm{sgn}(\sin(v)) \end{aligned} \tag{4.17}$$

An outward surface normal is given as

$$\mathbf{n}_{t,r}(u,v;w_i^{3D}) = \frac{\partial \mathbf{s}_{t,r}(u,v;w_i^{3D})}{\partial u} \times \frac{\partial \mathbf{s}_{t,r}(u,v;w_i^{3D})}{\partial v} \tag{4.18}$$

And $\bar{\mathbf{n}}_{t,r}(u,v;w_i^{3D})$ is the normalized outward surface normal.

4.2.2 Data Potential

Data potential $V_d(\mathbf{y}|w_i)$ reflects how much an object w_i is matched to an image $\mathbf{y}$. Usually, a statistic difference in pixel values between outside and inside of object model is used [12, 16]. As a statistic difference, the Bhattacharya distance or the t-test are used. The other way to describe an data potential is using the average of inner product between a surface normal vector and an image data gradient [14, 39]. Gaussian mixture model is also used in tree crown detection [40]. Error between data and reference waveform of object are also used in some applications [7, 41]. In all cases, the data potential should be designed in a way that an object which is matched better should has a lower potential. And, to make an object survive in a final configuration, it is desirable for the object to have a negative value of potential.

In this chapter. the data potential is based on the average of inner product between a normalized surface normal and a normalized image gradient which is faster than other methods in our application. Let function $d_r(u;w_i)$ be defined on $u \in [0, 2\pi]$ as

$$d_r(u;w_i) = \bar{\mathbf{n}}_r(u;w_i) \cdot \frac{\nabla \mathbf{y}(\mathbf{s}_r(u;w_i))}{\sqrt{|\nabla \mathbf{y}(\mathbf{s}_r(u;w_i))|^2 + \varepsilon}} \tag{4.19}$$

where $\bar{\mathbf{n}}_r(u; w_i)$ is a normalized surface normal obtained from (4.13), $\nabla\mathbf{y}(p)$ is an image gradient at position p, $\mathbf{s}_r(u; w_i)$ is outline of an ellipse given by (4.9) and ϵ is a parameter to reduce the error caused by noise on the image. To detect more objects which have regular inner product values along with their boundary, we regularize an average value of $d_r(u; w_i)$ with its variance as below:

$$D_i = \frac{m_i}{v_i + 1} \tag{4.20}$$

where

$$m_i = \frac{1}{2\pi}\int_0^{2\pi} d_r(u; w_i)\mathrm{d}u, \quad v_i = \frac{1}{2\pi}\int_0^{2\pi} (d_r(u; w_i))^2\mathrm{d}u - m_i^2 \tag{4.21}$$

Finally, the data potential $V_d(y|w_i)$ is defined as a likelihood function [39].

$$V_d(\mathbf{y}|w_i) = \min\left(\frac{D_i - T_d}{T_d + 1}, 1\right) \tag{4.22}$$

where T_d is a control parameter.

4.2.3 Optimization

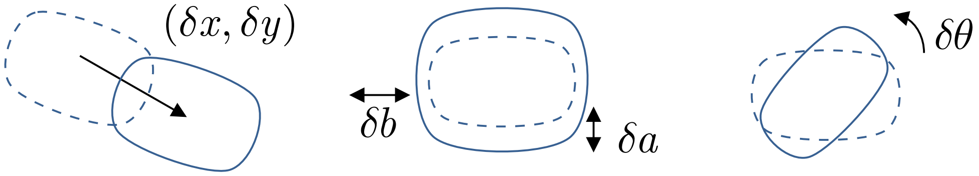
To minimize the Gibbs energy, if it has a simple prior term such as the object non-overlapping, multiple birth and death(MBD) and multiple birth and cut(MBC) can usually be considered [38]. But, in the MBD method, setting the simulated annealing parameters is not straightforward. MBC method does not have these parameters but the convergence speed of the MBC method is slower than the MBD method. However, if a local perturbation(LP) kernel is added to the original MBC method, the convergence speed becomes faster [39, 42]. In addition, to get more benefit from the correlation between the image slices, we propose a birth in a neighborhood kernel to the MBC with LP.

MBC with birth in a neighborhood algorithm, which is used in minimizing a 2D object configuration energy, is given in Algorithm 2. And 3D MBC optimization algorithm, which is used for detecting 3D object configurations, is a direct extension of the 2D MBC algorithm. However, the 3D MBC algorithm does not have a birth in a neighborhood kernel.

Algorithm 2 MBC with birth in a neighborhood

```
Initialization: $n \Leftarrow 0$, Generate $w'$ and $w_{(0)} = w'$
repeat
  Generate random number $r \in [0, 1]$
  if $r < p_1$ then
    Generate $w'$ with a multiple birth kernel
  else
    if $r \geq p_1$ and $r < p_2$ then
      Generate $w'$ with a local perturbation kernel
    else
      Generate $w'$ with a birth in a neighborhood kernel
    end if
  end if
  $w_{(n+1)} \Leftarrow Cut(w_{(n)} \cup w')$
  $n \Leftarrow n + 1$
until converged
```

Local Perturbation Kernel



(a) Translation (b) Dilation (c) Rotation

Fig. 4.5. Non-jumping perturbation

Figure 4.5 shows this type of kernel. Each of these transformations uses the parameters that are randomly chosen as follow:

$$\begin{aligned}\delta x \in& [-\Delta x, \Delta x], \quad \delta y \in [-\Delta y, \Delta y],\\ \delta a \in& [-\Delta a, \Delta a], \quad \delta b \in [-\Delta b, \Delta b],\\ \delta\theta \in& [-\Delta\theta, \Delta\theta]\end{aligned}$$

Birth in a Neighborhood Kernel

First, we randomly choose $\delta z \in \{-1, 1\}$, then new z coordinate of center position is given as $p_i^z + \delta z$. The other parameters for w_i are chosen by applying a local perturbation.

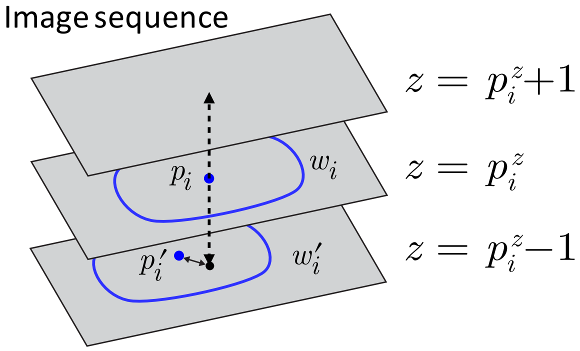


Fig. 4.6. Birth in a neighborhood

Graph Cut

Many vision problems can be interpreted as a pixel-labeling problem. The goal is to find mapping $g : \mathcal{P} \rightarrow \mathcal{L}$ which minimizes some energy function. Vladimir

Kolmogorov et al. [43] presented that a standard form of the energy function of class $\mathcal{F}^2$ shown below can be minimized with graph cuts.

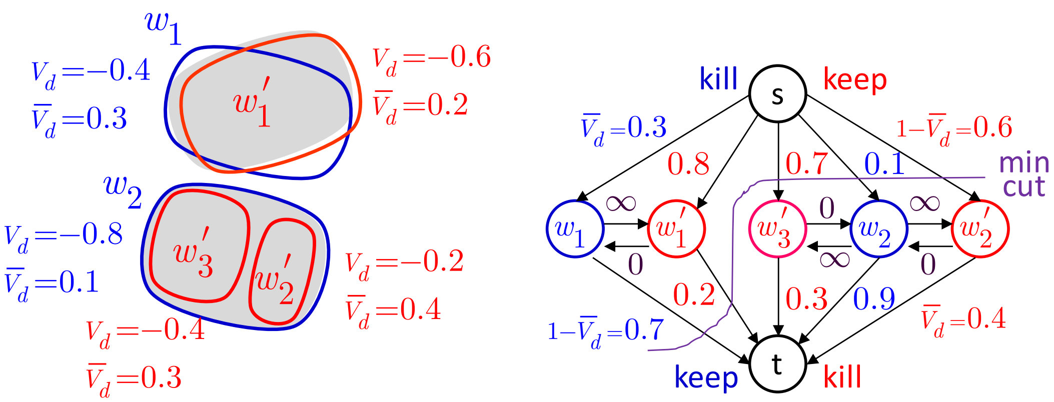
$$E(g) = \sum_{p \in \mathcal{P}} D_p(g_p) + \sum_{p,q \in \mathcal{N}} V_{p,q}(g_p, g_q) \tag{4.23}$$

where $\mathcal{N}$ is a neighborhood system on pixels. $D_p(g_p)$ is the cost of assigning the label g_p to the pixel p. $V_{p,q}(g_p, g_q)$ is the cost of assigning the label (g_p, g_q) to the pixel (p, q). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the directed graph with non-negative edge weights that has two special vertices, the source s and the sink t. An s-t cut is a partition (S, T) which satisfies $s \in S, t \in T, S \cup T = \mathcal{V}, S \cap T = \emptyset$. The cost of s-t cut is the sum of cost of all edges that go from S to T. The problem finding s-t cut which minimizes cost is equivalent to find the maximum flow from source to sink. If the problem is a binary-valued labeling which means $\mathcal{L} = \{0, 1\}$, there are many algorithms which can solve this problem in polynomial time. Let us consider a mapping $v : \mathcal{P} \to \mathcal{V} - \{s, t\}$ and a map $g : \mathcal{V} - \{s, t\} \to \mathcal{L}$ which maps $g(v) = 0$ when $v \in S$ and $g(v) = 1$ when $v \in T$. And, let us construct a graph of energy function $E(g)$ with all vertices including v, s and t and with edges which have weight $W(v_p, s) = D_p(g(v_p) = 0)$, $W(v_p, t) = D_p(g(v_p) = 1)$ or $W(v_p, v_q) = V_{p,q}(g(v_p), g(v_q))$. Then, minimum s-t cut can give the labeling function g that minimizes the energy function. We use the graph cut code developed by Yuri Boykov and Vladimir Kolmogorov [43]. To use the graph cut algorithm, edge weights should be non-negative. Then, the data potential needs to be normalized from $[-1, 1]$ to $[0, 1]$ with equation below [38]:

$$\bar{V}_d(w_i) = \frac{1 + V_d(\mathbf{y}|w_i)}{2} \tag{4.24}$$

Then, set all the objects $w_i \in w_{(n)}$ and $w_i \in w'$ as vertices and set the edges between vertices with weights listed in Table 4.1 and 4.2.

Once an optimum label is acquired, theses labels are used to choose the objects that construct next configuration $w_{(n+1)}$. For an object $w_i \in w_{(n)}$, if $g_{w_i} = 0$ then w_i



(a) Configuration $w_{(n)}$(blue) and w'(red)

(b) Graph

Fig. 4.7. Graph cut optimization

Table 4.1
Data term

	$g_{w_i} = 0$	$g_{w_i} = 0$
$w_i \in w_{(n)}$	$\bar{V}_d(w_i)$	$1 - \bar{V}_d(w_i)$
$w_i \in w'$	$1 - \bar{V}_d(w_i)$	$\bar{V}_d(w_i)$

Table 4.2
Interaction term

$(g_{w_i}, g_{w'_j})$	
$(0,0)$	0
$(0,1)$	∞
$(1,0)$	0
$(1,1)$	0

should be killed otherwise keep. In case of an object $w_i \in w'$, label interpretations are inverted.

4.2.4 Clustering and 3D Object Matching

To construct 3D objects from the configuration of 2D objects, it is necessary to cluster those 2D objects in related groups first. The 2D objects w_i and w_j are interlayer neighbors denoted as $w_i \sim_l w_j$, if and only if they satify following relations:

$$\begin{aligned} &O(w_i, w_j) = \frac{A(w_i \cap w_j)}{\min(A(w_i), A(w_j))} \geq T_c, \\ &|p_i^{xy} - p_j^{xy}| \leq T_p, \\ &|p_i^z - p_j^z| \leq n_z \end{aligned} \tag{4.25}$$

where $p_i^{xy} = (p_i^x, p_i^y)$, $A(\cdot)$ is an area function and $A(w_i \cap w_j)$ is an overlapping area function which counts only on x and y coordinates, not necessarily z. And $n_z \in \mathbb{N}$ is a small number greater than 1. We set this value as 2 in our experiment. If $w_i \sim_l w_j$, w_i and w_j should be in the same set C_k related with a 3D object. From these clustered object sets, we can get 3D object $w_k^{3D} = (p_k, a_k, b_k, c_k, \alpha_k, \beta_k, \gamma_k)$ by a 3D object matching. Let s_m be the data points on the outline of objects in C_k, and $\hat{C}_k$ be the set of these points. Then, the 3D object matching is an optimization problem to find optimal parameters which satisfy:

$$\arg\min_{w_k^{3D}} \sum_m \left(f_{t,r}(s_m; w_k^{3D}) - 1\right)^2 \tag{4.26}$$

where $f_{t,r}(\cdot)$ is given in (4.16). To solve this optimization, we use Levenberg-Marquardt algorithm [44]. To obtain reliable results, it is essential to set the initial values of these parameters carefully. We set these initial vales as below:

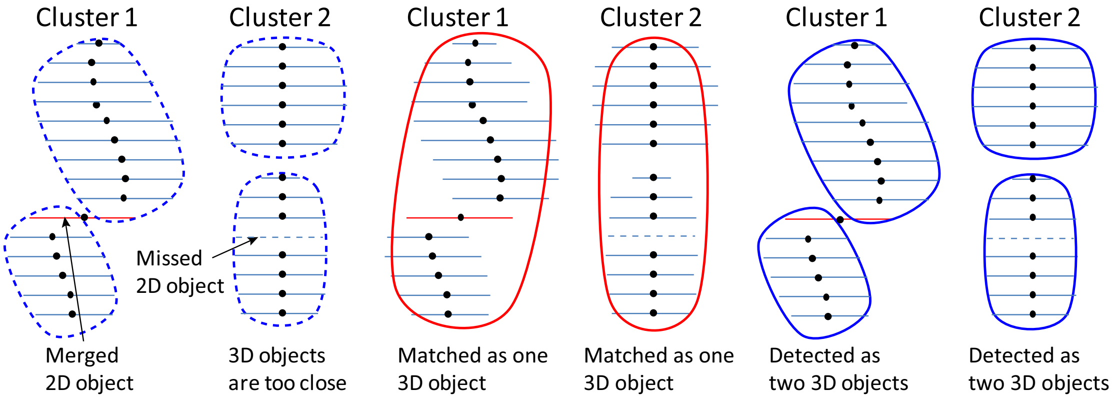
- (p_k^x, p_k^y, p_k^z) : A centroid of data in $\hat{C}_k$.
- (a_k, b_k) : $\bar{a}_i$ and $\bar{b}_i$, averages of major and minor axis length of $\forall w_i \in C_k$.
- c_k : A half value of the number of objects in the cluster
- γ_k : An average of θ_i of $\forall w_i \in C_k$.

- (α_k, β_k) : An angle of line in K which is fitted to the p_i of $\forall w_i \in C_k$(Ordinary Least Squares is used).

4.3 Fast 3D Fitting Method

Clustering errors can cause large errors in the final 3D object configuration in the 3D clustering method. This 3D clustering method also can not guarantee non-overlapping of 3D objects with each other. Most common errors in 2D detection, which cause the clustering errors, are a missed object and a merged object. The missed object brings out the separation which makes the objects classified to different clusters while those are supposed to be classified in the same cluster. The merged object combines the clusters which are supposed to be different clusters. There are some examples of the misclustering in Figure 4.8(a). This figure illustrates the side view of a 2D object detection and related clustering result on image slices. Solid lines are 2D objects and dotted ellipses are cross-sectional outline shapes of original 3D objects we expect to detect. Dots in the middle of lines are center positions of 2D objects. A cluster 1 has merging errors originated from the merged 2D objects which are depicted as a red line in the figure. A cluster 2 also has a merging error. However, this error occurs because the original 3D objects are too close. For the bottom 3D object of cluster 2, one object is missed in the 2D object detection. To avoid a separation in a clustering step, we can set n_z value bigger than 1. In this case, this setting might cause other problems of merging as the figure illustrated. Like this, there is a trade-off between merging and separation in the clustering step according to the setting of n_z and T_p values in Equation (4.26). Figure 4.8(b) shows related matching errors for these clustering errors when using the 3D clustering method.

To overcome these problems, the fast 3D fitting method uses a 3D MPP method in detecting a 3D object configuration. A 3D MPP method does not have these problems because it uses image data itself to detect 3D objects whereas a 3D clustering method uses detected 2D objects. However, the 3D MPP method has computational complexity issue. This complexity is proportional to the volume of the space for birth. To reduce the computation time, a non-uniform birth process is used in many MPP applications. In MBD method, the birth map is used which enables the birth step to generate more candidates on more probable sites [12]. Lacoste et al. [16] also applied inhomogeneous birth and death kernel. To detect line segments, they generated segments according to the probability map which was off-line computed in advance by fixing the length as minimum value and changing midpoint and orientation. Likewise, to improve the computational complexity of the 3D MPP method, the fast 3D fitting method generates 3D objects inhomogeneously in the birth step according to the birth map calculated from the clustered 2D object configuration.



(a) Misdetection and clustering errors

(b) 3D clustering method

(c) Fast 3D fitting

Fig. 4.8. Clustering and fitting error

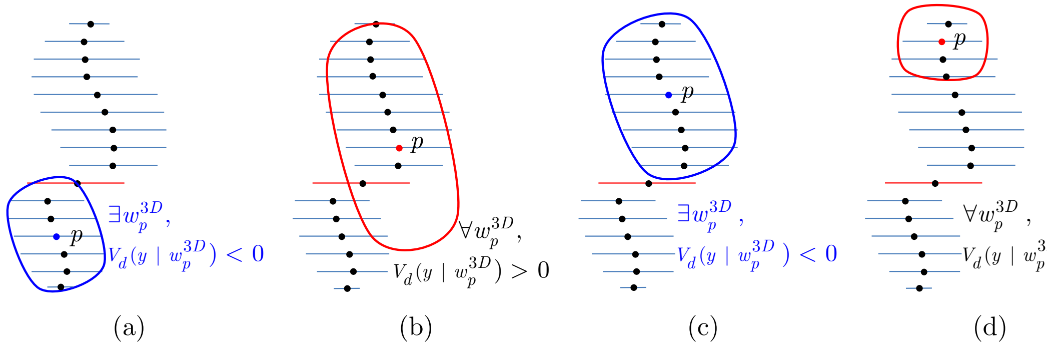
4.3.1 Center Position Map

We propose more 3D object candidates at the positions where objects are supposed to be existed in high probability. The resulting configuration from the 2D object detection can be used to predict the existence of 3D objects. If there is no error in 2D object detection and clustering, the 3D objects are highly probable to exist near the center position of the 2D object which is located in the middle of objects in the same cluster. But, if there is a merging error in clustering, this assumption is no more valid. However, even in this case, 3D objects are still have high chance to be near the center position of the most of 2D objects in the cluster. Figure 4.9 illustrates this merging example. The cluster is supposed to be separated as two clusters but is clustered as one. To get 3D objects compatible with image data, we need to generate candidates near every center position of all the 2D objects in the cluster. There are well fitted objects at the positions p such as the one illustrated as blue dots in Figure 4.9(a) and 4.9(c), whereas we can hardly get good candidates at the positions p in Figure 4.9(b) and 4.9(d). In these figures, negative potentials mean that the objects are well fitted to the image data.

Let the i-th 2D ellipse belong to a cluster C_k be $w_{k,i} = (p_{k,i}, a_{k,i}, b_{k,i}, \theta_{k,i}) \in C_k$, where $k \in 1, \cdots, N_c$ and $i \in 1, \cdots, N_k$. For $p \in K$, $(k_p, i_p) = \arg\min_{(k,i)} d_c(p, p_{k,i})$ and $\hat{d}_c(p) = d_c(p, p_{k_p,i_p})$. Here, $d_c(p, p_{k,i}) = \sqrt{(x - p^x_{k,i})^2 + (y - p^y_{k,i})^2 + W_z(z - p^z_{k,i})^2}$ and W_z is a control parameter. We can generate a 3D object candidate $w^{3D}_p = (p, a^{3D}_p, b^{3D}_p, c^{3D}_p, \alpha_p, \beta_p, \gamma_p)$ inhomogeneously according to the probability

$$B(p) = \frac{\exp(-\hat{d}_c(p)/2\sigma^2)}{\sum_{s\in K} \exp(-\hat{d}_c(s)/2\sigma^2)} \tag{4.27}$$

where σ is a control parameter. Here, $(p, a^{3D}_p, b^{3D}_p, c^{3D}_p)$ and $(\alpha_p, \beta_p, \gamma_p)$ are sampled from semi-axis length map and orientation map respectively. Those maps will be discussed in the following sections.



p
$\exists w^{3D}_p$,
$V_d(y \mid w^{3D}_p) < 0$
(a)
p
$\forall w^{3D}_p$,
$V_d(y \mid w^{3D}_p) > 0$
(b)
p
$\exists w^{3D}_p$,
$V_d(y \mid w^{3D}_p) < 0$
(c)
p
$\forall w^{3D}_p$,
$V_d(y \mid w^{3D}_p) > 0$
(d)

Fig. 4.9. Birth map (position)

4.3.2 Orientation Map

Let ϕ be the tilt angle of an object from z axis as in Figure 4.11(a). Then it can be calculated from rotation angles α and β of the object as $\cos(\phi) = \cos(\alpha)\cos(\beta)$. If the maximum tilt angle is as below,

$$\phi_{max} \ll \tan^{-1} \frac{c_{min}}{\sqrt{a_{max}^2 + b_{max}^2}} \tag{4.28}$$

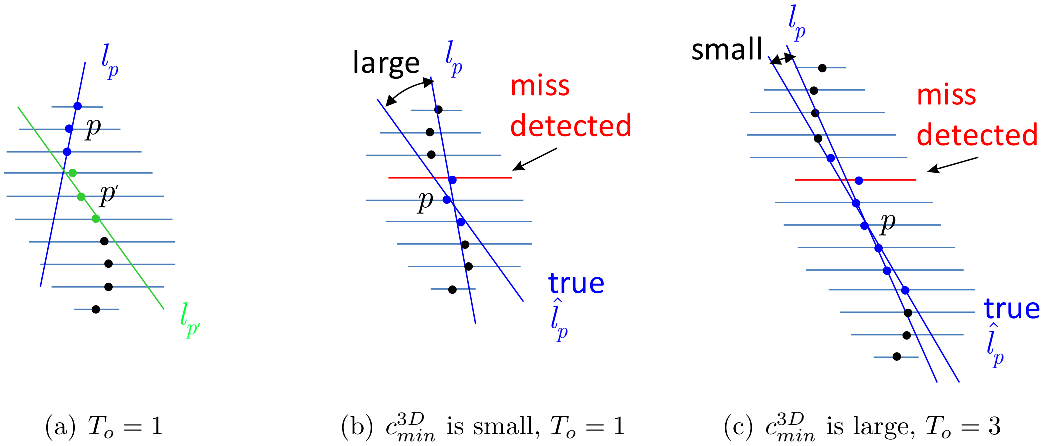
then we can sample the preferable rotation angles α_p and β_p from the formation of center positions of 2D superellipses. Let l_p be a line fitted with these center positions $\{p_{k_p,i} | \, |p^z_{k_p,i} - p^z_{k_p,i_p}| \leq T_o\}$ and $\mathbf{v} = (v_x, v_y, v_z)$ be the direction vector of the line l_p when $v_z \geq 0$. Where T_o is a control parameter which will be discussed in detail shortly. Then, $\alpha_{l_p}, \beta_{l_p}$ are the rotation angles of $\mathbf{v}$ from a $\mathbf{e}_z = (0, 0, 1)$ direction around x and y axes.

$$\mathbf{v} = R_x(\alpha_{l_p}) R_y(\beta_{l_p}) \mathbf{e}_z \tag{4.29}$$

Then we can sample α_p and β_p from normal distributions and γ_p from a uniform distribution.

$$\alpha_p \sim \mathcal{N}(\alpha_{l_p}, \sigma_o^2), \; \beta_p \sim \mathcal{N}(\beta_{l_p}, \sigma_o^2), \; \gamma_p \sim \mathcal{U}(\gamma_{min}, \gamma_{max}) \tag{4.30}$$

When we choose the parameter σ_o, we need to consider the following trade-off. If $\alpha_{l_p}, \beta_{l_p}$ values are predicted near the true rotation angles, we can increase a convergence speed by reducing the value of this parameter. However, if the prediction is much off the true orientation, the computation time might take rather longer for the small value of the parameter. Figure 4.10(b) and 4.10(c) show some misdetections of 2D objects which cause poor prediction of $\alpha_{l_p}, \beta_{l_p}$. To reduce the effect of errors to the prediction, we can increase the number of objects which is used in the line fitting. The parameter T_o can control the number of objects and restricted by the maximum semi-axis length c^{3D}_{min}. If c^{3D}_{min} is big, we can much reduce computation time through the orientation prediction.



(a) $T_o = 1$ (b) c^{3D}_{min} is small, $T_o = 1$ (c) c^{3D}_{min} is large, $T_o = 3$

Fig. 4.10. Birth map (orientation)

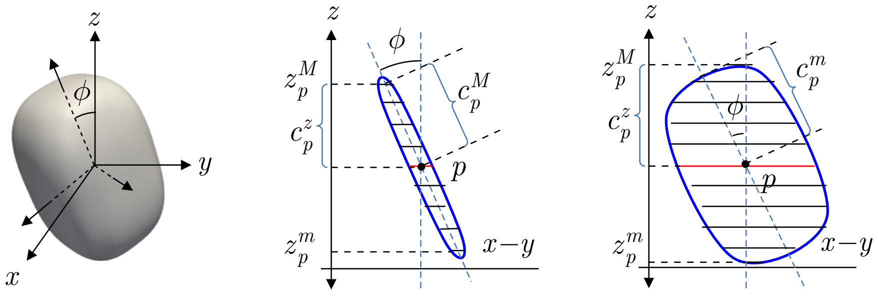
4.3.3 Semi-axis Length Map

We can predict preferable semi-axis length a^{3D}_p and b^{3D}_p of the proposed object w^{3D}_p at position p from tilt angle ϕ_{k_p} and semi-axis length a_{k_p,i_p} and b_{k_p,i_p} of a 2D object. The possible maximum value comes when the longer 2D semi-axis length direction is perpendicular to the tilt direction of the 3D object, and the possible minimum value comes when the shorter one has same direction with the tilt direction as depicted in Figure 4.12. Therefore, a^{3D}_p and b^{3D}_p are supposed to be between r^m_p and r^M_p.

$$r^m_p = \min(a_{k_p,i_p}, b_{k_p,i_p})\cos(\phi_p), \quad r^M_p = \max(a_{k_p,i_p}, b_{k_p,i_p}) \tag{4.31}$$

Finally, we can sample a^{3D}_p and b^{3D}_p from normal distributions as below:

$$a^{3D}_p, b^{3D}_p \sim \mathcal{N}\left(\frac{r^M_p + r^m_p}{2}, \left(\frac{r^M_p - r^m_p + \varepsilon_r}{2}\sigma_r\right)^2\right) \tag{4.32}$$



(a) tilt angle ϕ (b) c_p^M (c) c_p^m

Fig. 4.11. Birth map (semi-axis length c_p^{3D})

where ε_r is a control parameter to avoid the variance term become zero. We set this value with 2. σ_r is a parameter to control the variance. In case of semi-axis length c_p^{3D}, we can predict preferable length from tilt angle ϕ_{k_p} and the number of objects N_{k_p} in a cluster C_{k_p}. First, we calculate the minimum z position and the maximum z position in a cluster C_{k_p}.

$$z_p^m = \min_{w_{k_p,i} \in C_{k_p}} z_{k_p,i}, \quad z_p^M = \max_{w_{k_p,i} \in C_{k_p}} z_{k_p,i} \tag{4.33}$$

If the tilt angle ϕ_{k_p} is zero, the object w_p^{3D} needs to have semi-axis length c_p^{3D} as near c_p^z as below:

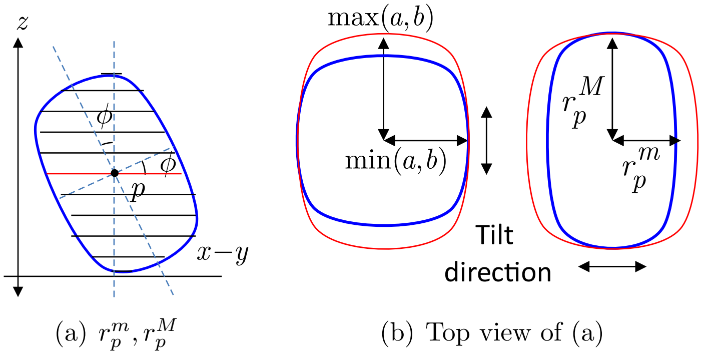
$$c_p^z = \min(z - z_p^m, z_p^M - z) \tag{4.34}$$

If the tilt angle ϕ_{k_p} is none-zero, it is much probable that the length is between c_p^m and c_p^M as depicted in Figure 4.11, where c_p^m and c_p^M are given as follow (see Appendix D):

$$c_p^m = \frac{c_p^z}{\cos(\phi_p)} - \sqrt{(a_p^{3D})^2 + (b_p^{3D})^2}\tan(\phi_p), \quad c_p^M = \frac{c_p^z}{\cos(\phi_p)} \tag{4.35}$$

Finally, we can sample c_p^{3D} from normal distributions as below:

$$c_p^{3D} \sim \mathcal{N}\left(\frac{c_p^M + c_p^m}{2}, \left(\frac{c_p^M - c_p^m + \varepsilon_c}{2}\sigma_c\right)^2\right) \tag{4.36}$$



(a) r_p^m, r_p^M (b) Top view of (a)

Fig. 4.12. Birth map (semi-axis length a_p^{3D} and b_p^{3D})

4.4 Experimental Results

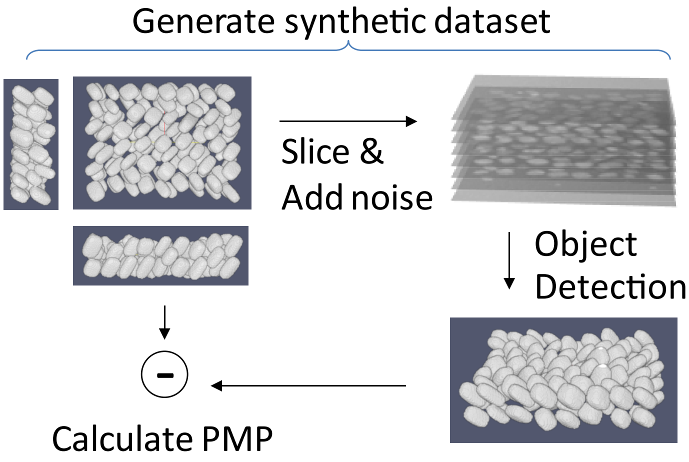


Fig. 4.13. Synthetic dataset generation and detection

For test datasets, we use four sets of image sequences. Two sets are synthetic image sequences, the third one is a materials microscopic image sequence and the last one is a microscope image sequence of a brain tissue. For the first 3D dataset, we generate an ellipsoid object configuration using a superellipsoid MPP with parameter $t = r = 2$. And an image sequence is generated using the Gaussian model with different means and variances for inside and outside of the ellipsoids. Then, we get a sequence of 2D image slices (194 × 149, 60 slices). To these images, we apply a blurring filter and add the Gaussian noise with σ_n. For the second dataset, we generate an image sequence (194 × 149, 60 slices) containing superellipsoids with parameter $t = r = 3$. This dataset is much difficult to process with a 3D clustering method and fast 3D fitting method. Because the object has a large tilt angle, the cross section of a superellipsoid object does not fit well to the superellipse model. For these two data sequences, we know the ground truth of a 3D object configuration. To evaluate the final quality of results numerically, we calculate the percentage of misclassified pixels(PMP) between the ground truth and the results of three methods: a 3D clustering method, a fast 3D fitting method and a full 3D MPP method. The lower PMP value means the better result. On the other hand, for the third test dataset, we use a Ni-Al-Cr superalloy microscope image sequences(194 × 149, 59 slices). Mark spaces of these three test dataset are listed in Table 4.3.

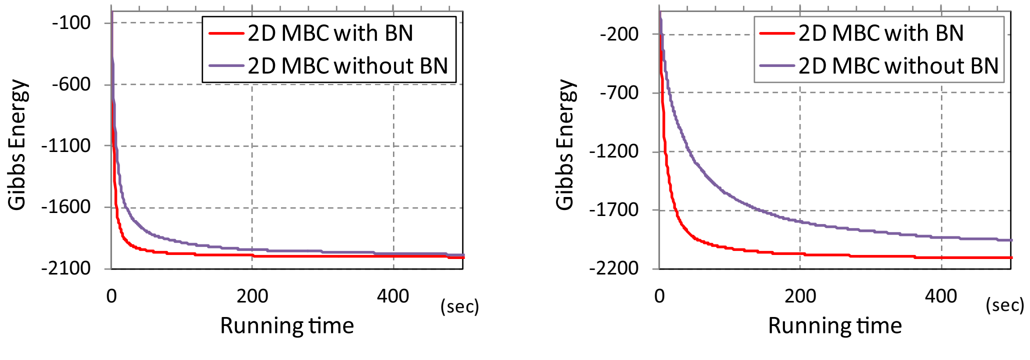
Section 4.4.1 compares the results of 2D MBC using a birth in a neighborhood(BN) kernel with the results without using the kernel. In Section 4.4.2, the result comparing a 3D clustering method and fast 3D fitting method is demonstrated. Section 4.4.3 shows that the fast 3D fitting method can significantly reduce the computation time in a 3D object detection compared to the full 3D MPP method. In Section 4.4.4, we apply our method to brain tissue 3D dataset. In our experiments, we use a machine equipped with Intel i7 2.40GHz CPU.

Table 4.3
Mark space

	axis length (a^{3D}, b^{3D}, c^{3D})	orientation (α, β, γ)
dataset 1	$[5, 7] \times [5, 7] \times [5, 7]$	$[-0.52, 0.52] \times [-0.52, 0.52] \times [0, 1.57]$
dataset 2	$[3, 13] \times [3, 13] \times [10, 15]$	$[-0.79, -0.26] \times [-0.79, 0.79] \times [0.26, 1.52]$
dataset 3	$[2, 23] \times [2, 23] \times [5, 30]$	$[-0.52, 0.52] \times [-0.52, 0.52] \times [0.59, 0.98]$

4.4.1 Birth in a Neighborhood(BN)

Both in a 3D clustering method and in a fast 3D fitting method, a 2D MPP method is used first to detect 2D objects which are then clustered and used to find a 3D object configuration. In the 2D MPP method, we use a MBC as an optimization method to find the optimum configuration. To increase the optimization speed, we propose to use an additional BN kernel to the existing MBC method. For the dataset 1 and 2, we perform the MBC method with a BN kernel and compare the result to the one without using the kernel. Figure 4.14(a) shows the total energy change according to a running time for the dataset 1. Figure 4.14(b) is for the dataset 2. These results show that the BN kernel can accelerate the 2D object detection. The reason that the improvement in convergence speed for the dataset 1 is not as big as that of dataset 2 is that the average of semi-axis length c^{3D} of objects in dataset 1 is shorter than that of dataset 2.



(a) Synthetic dataset 1 (ellipsoid)

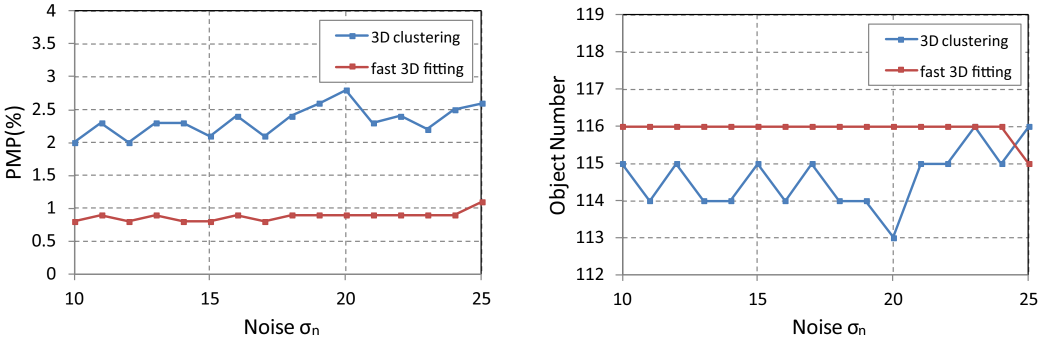
(b) Synthetic dataset 2 (superellipsoid)

Fig. 4.14. 2D MPP results with vs. without birth in a neighborhood(BN)

4.4.2 3D clustering vs. fast 3D fitting

We generate several sets of the dataset 2 by changing a noise level σ_n from 20 to 25. Then, we apply a 3D clustering method and a fast 3D fitting method to those datasets. Figure 4.15(a) shows changes of PMP value according to the noise level. We observe that a fast 3D fitting method shows better results in a sense that overall PMP values are lower and a variation is also smaller than 3D clustering method. In the results of a 3D clustering method, this variation mainly comes from object merging errors. When we see the graph in Figure 4.15(b), a fast 3D fitting method is detecting all the objects in the dataset whereas a 3D clustering method is missing some objects. In this dataset, true number of objects are 116. From the result images, we observe that most of these misdetections are merging errors. Some areas, which we are interested in, are displayed in Figure 4.16 and Figure 4.17. In these figures, the first row is the 2D MPP result and the second and the third rows are displaying cross-sectional outlines of detected 3D objects of a 3D clustering and of a fast 3D fitting method respectively. Figure 4.16(b) and Figure 4.17(b) show some object merging errors in a 3D clustering method. This object merging error is inevitable in a 3D clustering method when merging errors already exist in a clustering step. Figure 4.16(a) and Figure 4.17(a) show these clustering errors. However, our proposed fast 3D fitting method can avoid this problem as displayed in Figure 4.16(c) and Figure 4.17(c) even though it takes longer time.

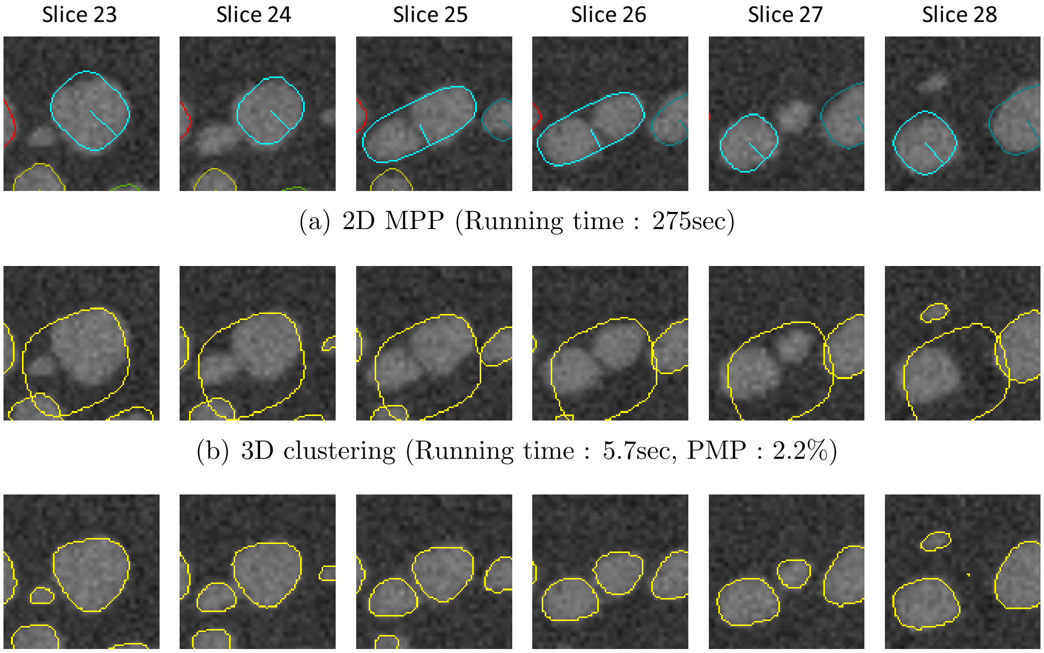
Figure 4.18 compares the whole cross-sectional images of the 25th result image slice for the three methods. In the result of a 3D clustering method(Figure 4.18(c)), we can find an object overlapping problem (marked with an arrow) beside merging errors. This is another inevitable problem of a 3D clustering method. There is no such problems in the results of a fast 3D fitting(Figure 4.18(d)) and a full 3D MPP methods (Figure 4.18(b)) because they penalize an object overlapping with an interaction term in their objective energy functions. These figures also show that the boundary precision of the result of a fast 3D fitting method is better than those of a 3D



(a) PMP vs. Noise level

(b) Object number vs. Noise level

Fig. 4.15. 3D clustering method vs. fast 3D fitting method (synthetic dataset 2)



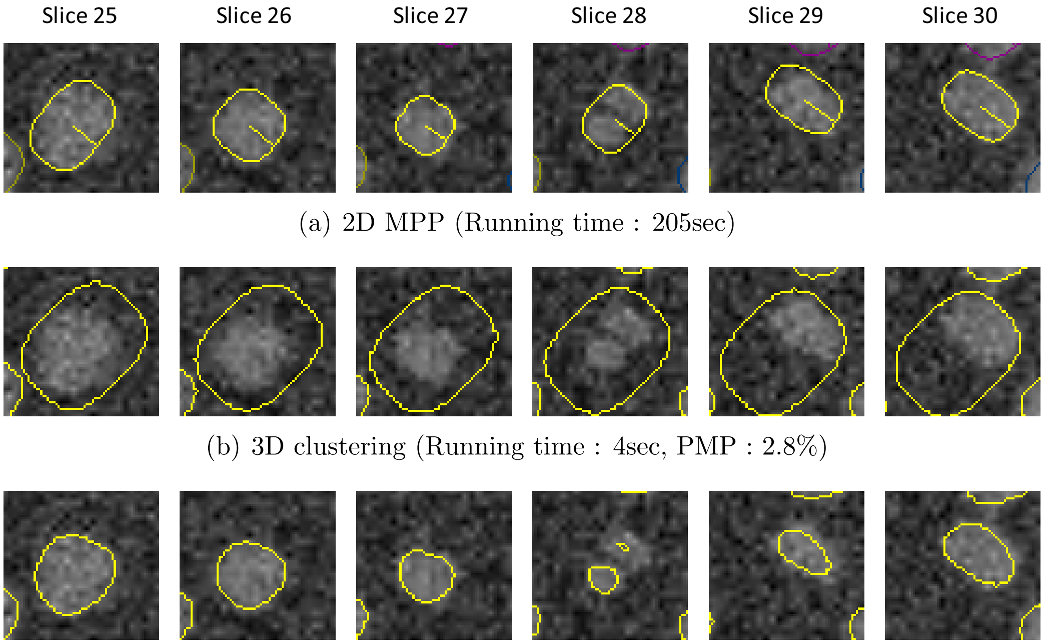
(a) 2D MPP (Running time : 275sec)

(b) 3D clustering (Running time : 5.7sec, PMP : 2.2%)

(c) Fast 3D fitting (Running time : 2121sec, PMP : 1.1%)

Fig. 4.16. The effect of error in 2D MPP and clustering : Synthetic dataset 2 with noise σ_n = 10 (Only area of interest are displayed from the whole test images)

clustering method and a full 3D MPP method. Figure 4.19 shows 3D reconstructed object configurations of the three methods. We can observe the merged object in the result of a 3D clustering method (marked with red, green and blue circles). The total execution times and PMP values are mentioned in Table 4.4. Here, ‘Birth map’ means a running time for generating the birth map from the clustered 2D objects.



(a) 2D MPP (Running time : 205sec)

(b) 3D clustering (Running time : 4sec, PMP : 2.8%)

(c) Fast 3D fitting (Running time : 6126sec, PMP : 1.0%)

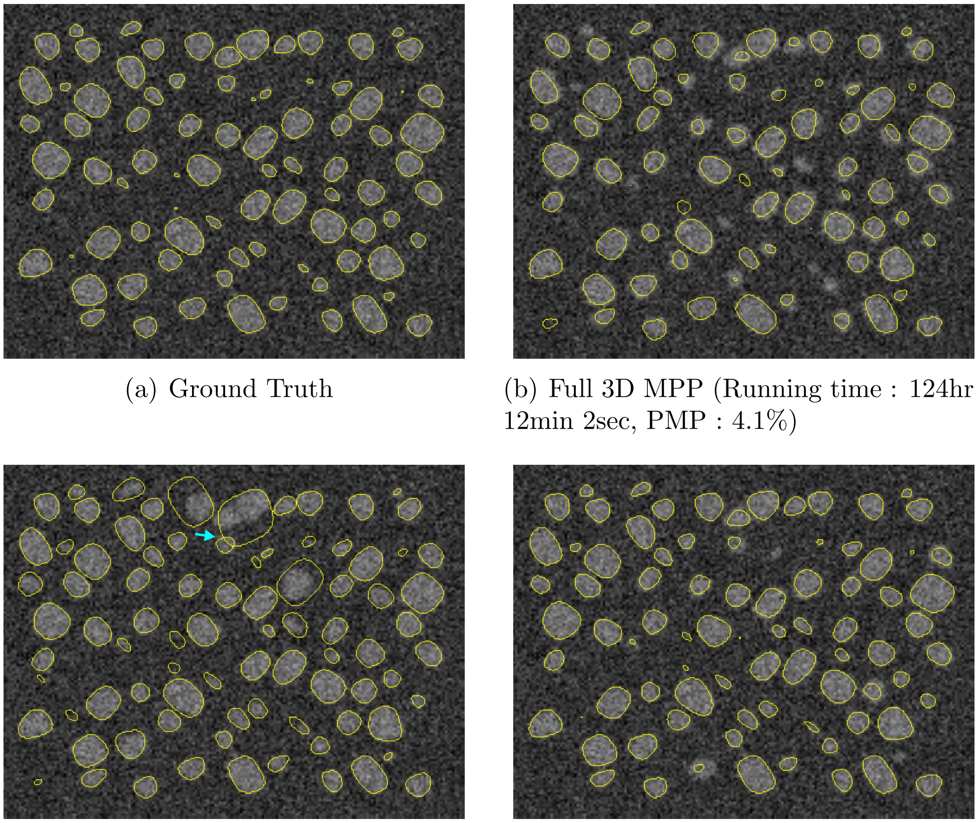
Fig. 4.17. The effect of error in 2D MPP and clustering : Synthetic dataset 2 with noise σ_n = 20 (Only area of interest are displayed from the whole test images)

Table 4.4
Running time and PMP for dataset 2 (σ_n = 20)

(sec)	3D clustering	Fast 3D fitting	Full 3D MPP
2D MPP	212	212	not used
Clustering	0.1	0.1	not used
3D obj. matching	4	not used	not used
Birth map	not used	26	not used
3D MPP	not used	6100	447122
Total	216	6338	447122
PMP	2.8%	1.0%	4.1%

4.4.3 Fast 3D fitting vs. full 3D MPP

In this section, we compare the performance of a fast 3D fitting method and a full 3D MPP method. First, to evaluate the improvement in convergence speed when using a birth map described in section 4.3, we apply three versions of the birth map: position map only, position and semi-axis length map and all three maps together(fast



(a) Ground Truth

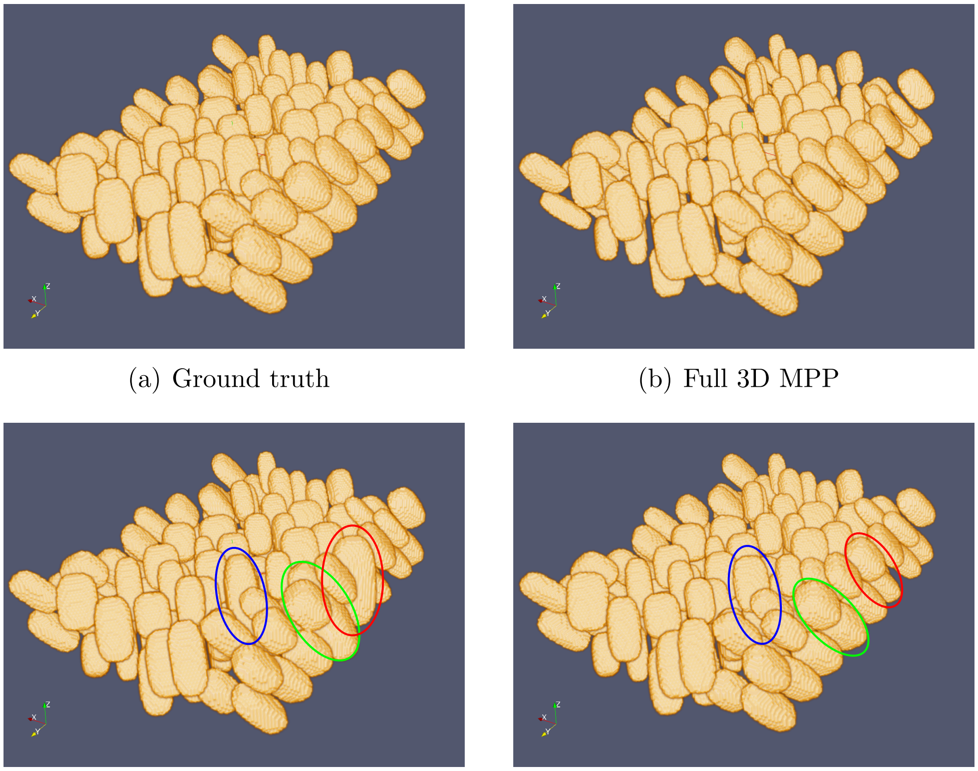
(b) Full 3D MPP (Running time : 124hr 12min 2sec, PMP : 4.1%)

(c) 3D clustering (Running time : 3min 36sec , PMP : 2.8%)

(d) Fast 3D fitting (Running time : 1hr 45min 38sec, PMP 1.0%)

Fig. 4.18. Results of Synthetic dataset 2 (noise $\sigma_n = 20$): The 25th cross sectional images of a ground truth of the dataset and final configurations of the three methods.

3D fitting). Also, we apply a 3D MPP method which does not use any birth map. Graphs in the first column in Figure 4.20 compare the three versions of the birth map. Graphs in the second column display the convergence of a fast 3D fitting method and a full 3D MPP method. The first row is for dataset 1. The second row is for dataset 2. And, the third row is for the Ni-Al-Cr superalloy dataset. These graphs demonstrate that the fast 3D fitting method can significantly reduce the computation time compare to the full 3D MPP method. In detail, for all the three datasets, the position map contributes most improvement. The semi-axis length map contributes for a dataset 2 and a Ni-Al-Cr superalloy dataset because those datasets have a wide range of semi-axis length (see Table 4.3). An orientation map does not contribute much for a



(a) Ground truth

(b) Full 3D MPP

(c) 3D clustering

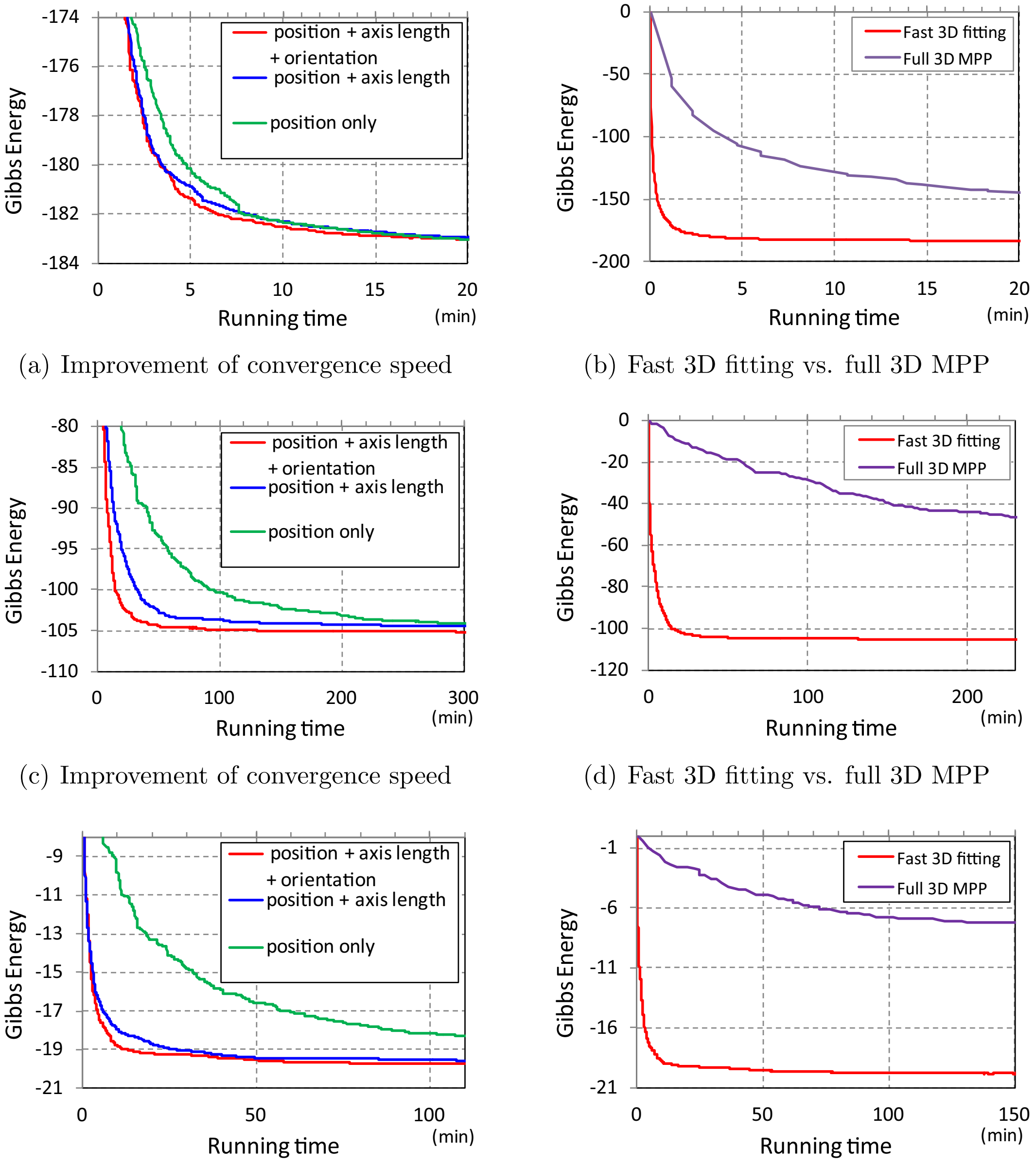
(d) Fast 3D fitting

Fig. 4.19. Results of Synthetic dataset 2 (noise $\sigma_n = 20$): 3D reconstructed images of a ground truth and results of the three methods

dataset 1 because the minimum semi-axis length is short as 5. An Ni-Al-Cr superalloy dataset also can not get much benefit from the orientation map. The reason is that there are some mispredictions of an orientation for the border touching objects.

Figure 4.21 compares the whole cross sectional images of the 16th result image slice for the three methods. Figure 4.22 shows 3D reconstructed object configurations of the three methods. The total execution times and PMP values are listed in Table 4.5.

Figure 4.23 compares the whole cross sectional images of the 38th result image slice for the three methods. The result image of 3D clustering method shows an objects overlapping (marked with an arrow). Figure 4.24 displays 3D reconstructed object configurations of the three methods. The total execution times are listed in Table 4.6.



(a) Improvement of convergence speed

(b) Fast 3D fitting vs. full 3D MPP

(c) Improvement of convergence speed

(d) Fast 3D fitting vs. full 3D MPP

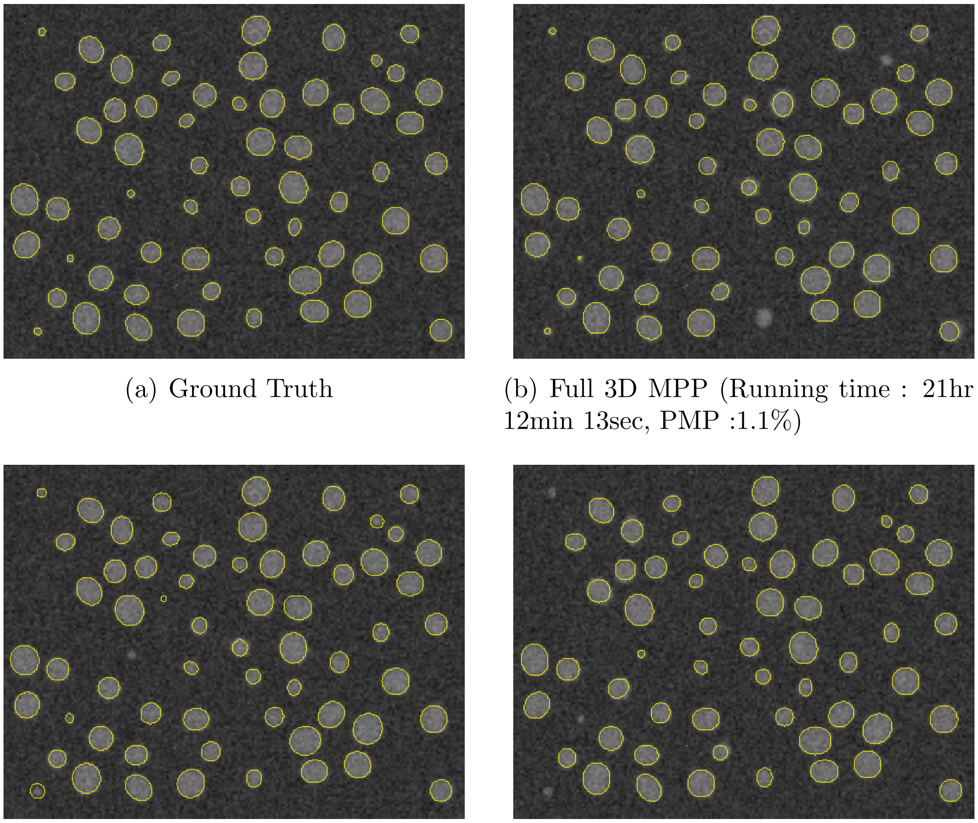
(e) Improvement of convergence speed

(f) Fast 3D fitting vs. full 3D MPP

Fig. 4.20. Fast 3D fitting vs. full 3D MPP : The first row(synthetic dataset 1), the second row(synthetic dataset 2) and the third row(Ni-Al-Cr superalloy)

4.4.4 Brain Tissue Image Dataset

Figure 4.25(a) shows a microscope image of the mouse brain tissue which is a small portion of a full 3D dataset. We selected the slice number from 962 to 1031



(a) Ground Truth

(b) Full 3D MPP (Running time : 21hr 12min 13sec, PMP :1.1%)

(c) 3D clustering (Running time : 38sec, PMP : 1%)

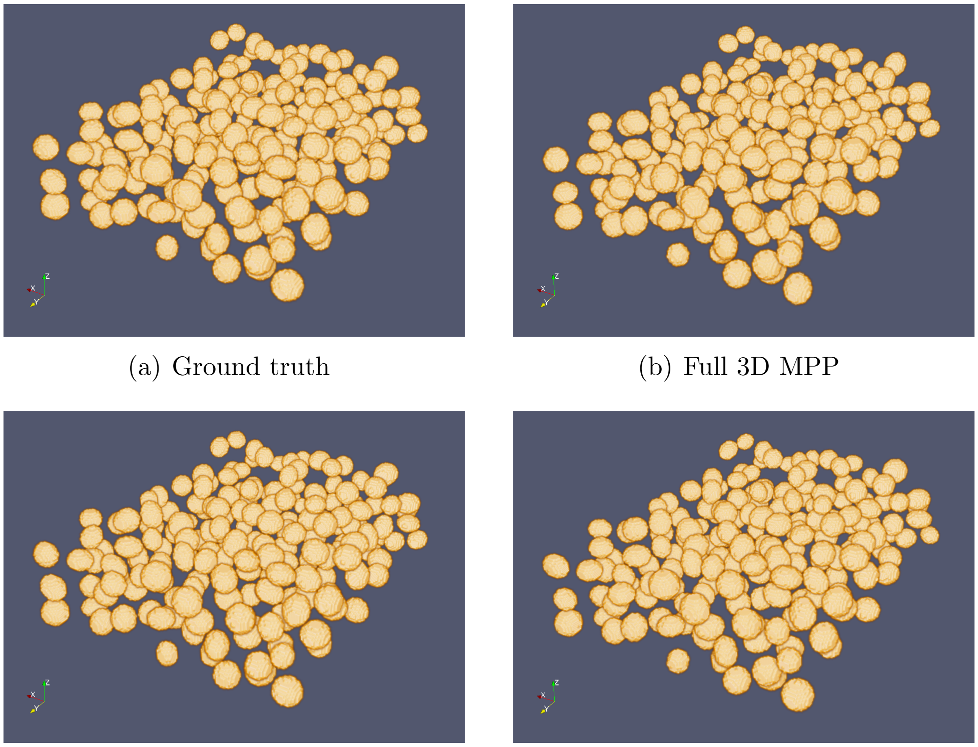
(d) Fast 3D fitting (Running time : 11min 25sec, PMP 0.5%)

Fig. 4.21. Results of Synthetic dataset 1 (noise $\sigma_n = 10$): The 16th cross sectional image of a ground truth of the dataset and final configurations of the three methods.

Table 4.5
Running time and PMP for dataset 1 ($\sigma_n = 10$)

(sec)	3D clustering	Fast 3D fitting	Full 3D MPP
2D MPP	32	32	not used
Clustering	0.1	0.1	not used
3D obj. matching	6	not used	not used
Birth map	not used	29	not used
3D MPP	not used	624	76333
Total	38	685	76333
PMP	1.0%	0.5%	1.1%

in this dataset and cropped each image with a resolution of 168 X 208. From this 3D dataset, we need to detect blood vessels and neurons as illustrated in Figure



(a) Ground truth

(b) Full 3D MPP

(c) 3D clustering

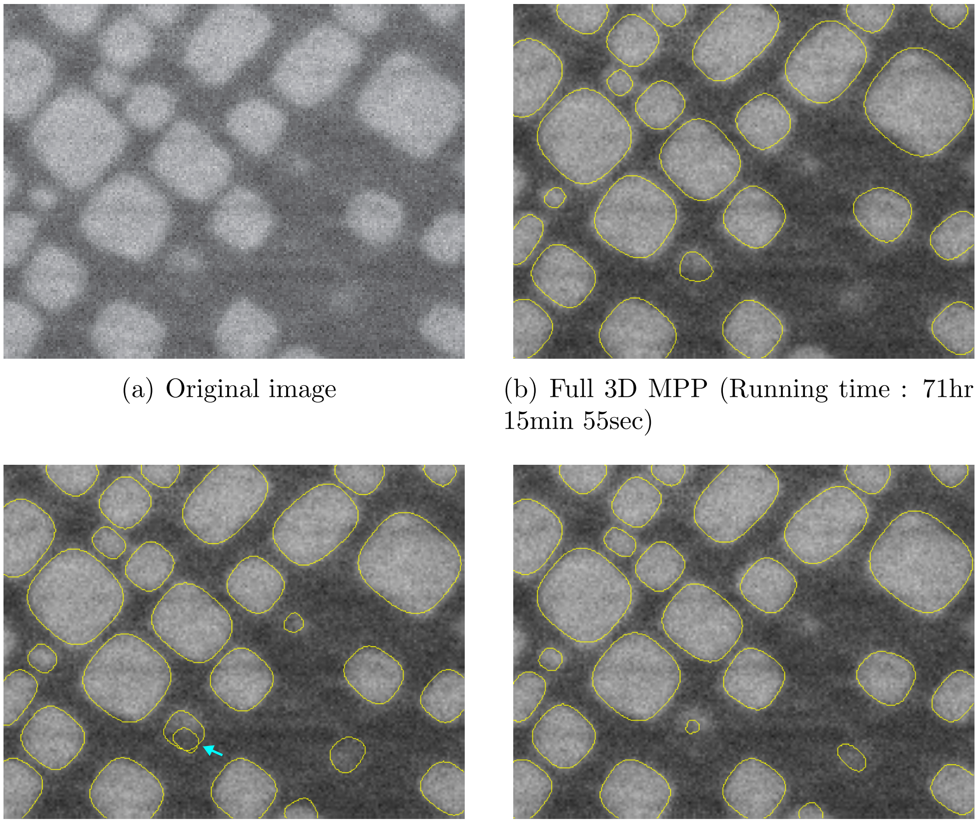
(d) Fast 3D fitting

Fig. 4.22. Results of Synthetic dataset 1 (noise $\sigma_n = 10$): 3D reconstructed images of a ground truth and results of the three methods

Table 4.6
Running time of Ni-Al-Cr dataset

(sec)	3D clustering	Fast 3D fitting	Full 3D MPP
2D MPP	162	162	not used
Clustering	0.1	0.1	not used
3D obj. matching	3	not used	not used
Birth map	not used	14	not used
3D MPP	not used	7160	256495
Total	165	7336	256495

4.25(a). First, for the blood vessel detection, we apply the full 3D MPP method for the elliptical cylinder model. During this step, we fixed the length of the cylinder as 3 to reduce computation time. From this elliptical cylinder configuration, we clustered the cylinders into some groups according to a clustering rule. The rule is, if the center position of two objects is close enough, they belong to the same group. After



(a) Original image

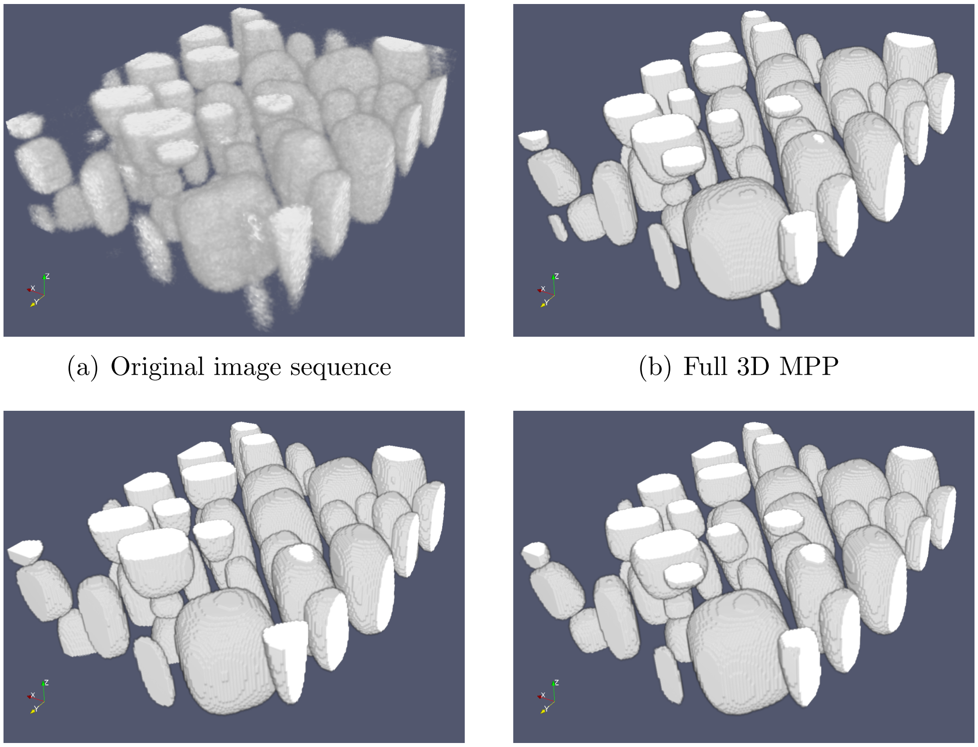
(b) Full 3D MPP (Running time : 71hr 15min 55sec)

(c) 3D clustering (Running time : 2min 45sec)

(d) Fast 3D fitting (Running time : 2hr 2min 16sec)

Fig. 4.23. Results of Ni-Al-Cr superalloy: The first row and the second row are the 38th cross sectional image of a original dataset of the dataset and final configurations of the three methods.

clustering, we identified blood vessel clusters. If the number of the elements in a cluster is big enough, it is estimated as a blood vessel cluster. To detect the neurons, we try the three ellipsoid detection methods(3D clustering, fast 3D fitting and full 3D MPP). As we intend to detect only the neurons, we avoid generating the ellipsoids near the center position of each elliptical cylinder in the blood vessel cluster. Figure 4.25 shows the results of this work. In this application, we focus on detecting neurons. If we need to detect the formation of the blood vessels more precisely, developing a 3D Candy model would be helpful.



(a) Original image sequence

(b) Full 3D MPP

(c) 3D clustering

(d) Fast 3D fitting

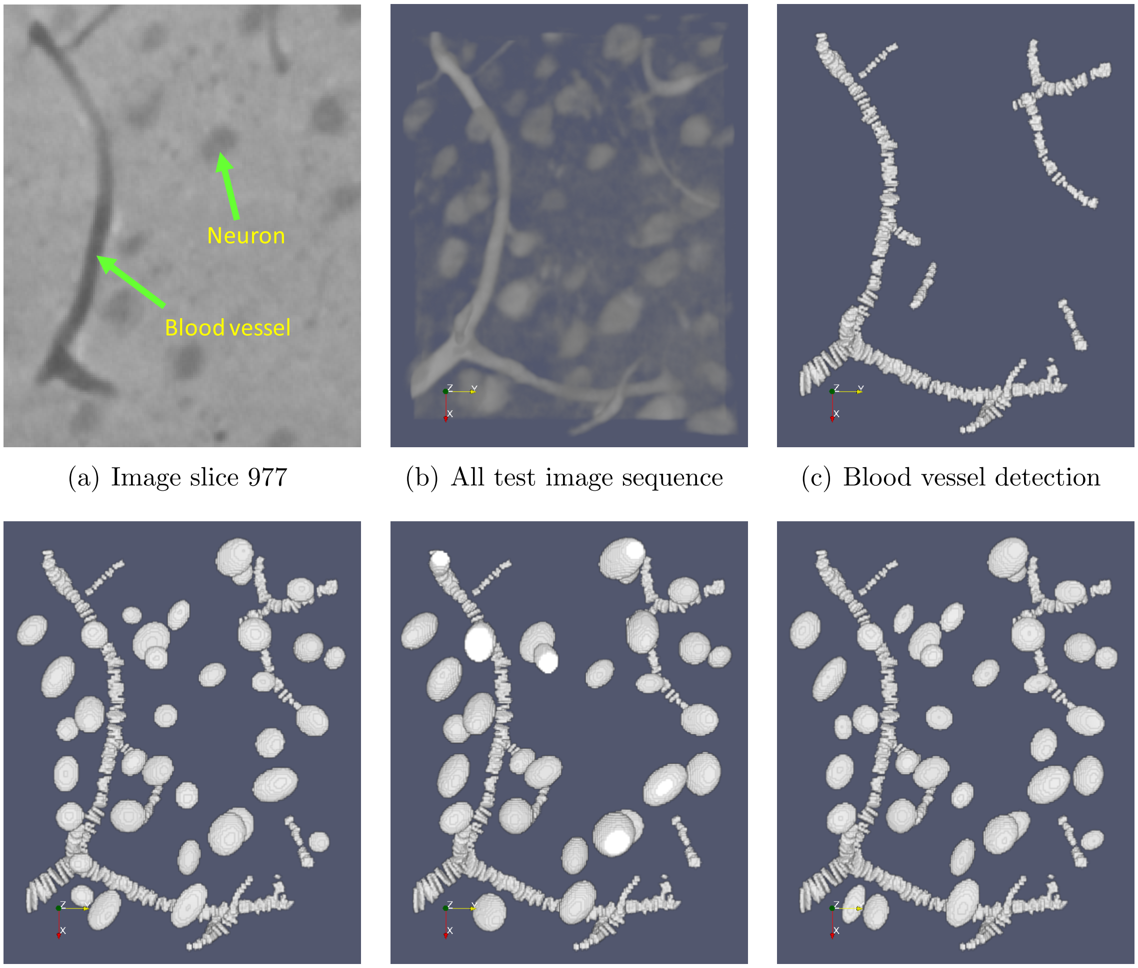
Fig. 4.24. Results of Ni-Al-Cr superalloy: 3D reconstructed images of a original dataset and results of the three methods

Table 4.7
Running time of detecting neuron in brain tissue

(sec)	3D clustering	Fast 3D fitting	Full 3D MPP
Blood vessel Detec.	4998	4998	4998
2D MPP	20	20	not used
Clustering	0.03	0.03	not used
3D obj. matching	1.4	not used	not used
Birth map	not used	11	not used
3D MPP	not used	4648	27635
Total	5020	9678	32633

4.5 Conclusions

In this chapter, we propose two alternative 3D object detection approaches. One is a 3D clustering method and the other is a fast 3D fitting method. Both methods consist of two steps: performing a 2D MPP first and constructing 3D objects from



(a) Image slice 977

(b) All test image sequence

(c) Blood vessel detection

(d) Full 3D MPP (Obj num : 35)

(e) 3D clustering (Obj num : 29)

(f) Fast 3D fitting (Obj num : 35)

Fig. 4.25. Results of brain tissue images: 3D reconstructed images of a original dataset and results of the three methods

the clustered data of the 2D MPP result. But the difference between two methods is how to construct 3D objects from the clustered 2D objects. The former applies a 3D object matching to the clustered 2D objects while the latter applies a 3D MPP method with inhomogeneous birth in order to reduce computation time. And this inhomogeneous birth can be implemented with a birth map which is extracted from the 2D object configuration. Among these two methods and an existing full 3D MPP method, a 3D clustering method is the fastest. However, an intrinsic problem of the 3D clustering method is in that even small misdetection in a 2D MPP and clustering process can have a significant influence on the final result. A fast 3D fitting can partly overcome this problem and is quite faster than a full 3D MPP method. To evaluate the performance of our proposed methods, two synthetic datasets, one material dataset and one biomedical image dataset are used. In this experiment, we used a superellipsoid model as a mark model for MPP. Especially, three cases of superellipsoid are used. Experimental results show that these new methods can significantly reduce the computational complexity of detecting 3D objects in a 3D image dataset. But proposed two methods has the same limitation that tilt angles of 3D objects should be small enough. This limitation can be a good future research topic. And, because we used only three types of an object model in this chapter, more object models can also be further studied.

APPENDICES

A. DERIVATION OF THE EM UPDATE

The EM algorithm is an iterative procedure. At each iteration, the expectation step and the maximization step are performed. In the expectation step the following function is computed.

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) =& E[\log f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] \\ &+ E[\log p_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta})|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] \end{aligned} \tag{A.1}$$

In the maximization step, we can estimate $\boldsymbol{\theta}(p)$ which maximize (A.1). In (A.1), the second term does not depend on $\boldsymbol{\theta}(p)$ because $\mathbf{X}$ does not depend on $\boldsymbol{\theta}(p)$. Therefore, by substituting (2.19) into (A.1) and differentiating with parameter μ_k we can get

$$\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1))}{\partial \mu_k} = \frac{\partial}{\partial \mu_k} E[\log f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})|\mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)]$$

$$= \frac{\partial}{\partial \mu_k} \sum_{r=1}^{N} E\left[- \frac{(y_r - \sum_{s=-D}^{D} h_s \mu_{x_{r-s}})^2}{2\sigma^2(r)} \middle| \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1) \right]$$

$$
\begin{aligned}
&= \frac{\partial}{\partial \mu_k} \sum_{r=1}^{N} E\Bigg[-\sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L} \delta(x_{r+D}=l_1) \\
&\qquad \delta(x_{r+D-1}=l_2)\cdots\delta(x_{r-D}=l_{2D+1}) \\
&\qquad \cdot \left(\frac{(y_r-(h_{-D}\mu_{l_1}+h_{-D+1}\mu_{l_2}+\cdots+h_D\mu_{l_{2D+1}}))^2}{2v(l_1,l_2,\cdots,l_{D+1},\cdots,l_{2D+1})} \right) \\
&\qquad \Bigg| \mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)\Bigg], \\
&= -\frac{\partial}{\partial \mu_k} \sum_{r=1}^{N}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L} \\
&\qquad \Bigg(\frac{(y_r-(h_{-D}\mu_{l_1}+h_{-D+1}\mu_{l_2}+\cdots+h_D\mu_{l_{2D+1}}))^2}{2v(l_1,l_2,\cdots,l_{D+1},\cdots,l_{2D+1})} \\
&\qquad \cdot E[\delta(x_{r+D}=l_1)\delta(x_{r+D-1}=l_2)\cdots\delta(x_{r-D}=l_{2D+1}) \\
&\qquad |\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)]\Bigg) \\
&= -\frac{\partial}{\partial \mu_k} \sum_{r=1}^{N}\sum_{l_1=1}^{L}\sum_{l_2=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L} \\
&\qquad \Bigg(\frac{(y_r-(h_{-D}\mu_{l_1}+h_{-D+1}\mu_{l_2}+\cdots+h_D\mu_{l_{2D+1}}))^2}{2v(l_1,l_2,\cdots,l_{D+1},\cdots,l_{2D+1})} \\
&\qquad \cdot p_{X_{r+D},X_{r+D-1},\cdots,X_{r-D}|\mathbf{Y}}(x_{r+D}=l_1,x_{r+D-1}=l_2, \\
&\qquad \cdots,x_{r-D}=l_{2D+1}|\mathbf{y},\boldsymbol{\theta}(p-1))\Bigg) \\
&= a_{k,1}\mu_1+a_{k,2}\mu_2+\cdots+a_{k,L}\mu_L-b_k=0 \qquad \text{(A.2)}
\end{aligned}
$$

Similarly, by differentiating with parameter σ_k we can get

$$\frac{\partial Q(\boldsymbol{\theta},\boldsymbol{\theta}(p-1))}{\partial \sigma_k} = \frac{\partial}{\partial \sigma_k} E[\log f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})|\mathbf{Y}=\mathbf{y},\boldsymbol{\theta}(p-1)]$$

$$= \frac{\partial}{\partial \sigma_k} \sum_{r=1}^{N} E\Bigg[- \frac{(y_r - \sum_{s=-D}^{D} h_s \mu_{x_{r-s}})^2}{2\sigma^2(r)} - \frac{1}{2} \log \sigma^2(r) \Bigg| \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1) \Bigg]$$

$$= \frac{\partial}{\partial \sigma_k} \sum_{r=1}^{N} E\Bigg[\sum_{l_1=1}^{L} \sum_{l_2=1}^{L} \cdots \sum_{l_{2D+1}=1}^{L} \delta(x_{r+D} = l_1) \delta(x_{r+D-1} = l_2) \cdots \delta(x_{r-D} = l_{2D+1}) \cdot \Bigg(- \frac{(y_r - (h_{-D}\mu_{l_1} + h_{-D+1}\mu_{l_2} + \cdots + h_D \mu_{l_{2D+1}}))^2}{2v(l_1, \cdots, l_{2D+1})} - \frac{1}{2} \log v(l_1, \cdots, l_{2D+1}) \Bigg) \Bigg| \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1) \Bigg]$$

$$= \frac{\partial}{\partial \sigma_k} \sum_{r=1}^{N} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} \cdots \sum_{l_{2D+1}=1}^{L} \Bigg\{ \Bigg(- \frac{(y_r - (h_{-D}\mu_{l_1} + \cdots + h_D \mu_{l_{2D+1}}))^2}{2v(l_1, \cdots, l_{2D+1})} - \frac{1}{2} \log v(l_1, \cdots, l_{2D+1}) \Bigg) \cdot E[\delta(x_{r+D} = l_1) \delta(x_{r+D-1} = l_2) \cdots \delta(x_{r-D} = l_{2D+1}) | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] \Bigg\}$$

$$= \frac{\partial}{\partial \sigma_k} \sum_{r=1}^{N} \sum_{l_1=1}^{L} \sum_{l_2=1}^{L} \cdots \sum_{l_{2D+1}=1}^{L} \Bigg\{ \Bigg(- \frac{(y_r - (h_{-D}\mu_{l_1} + \cdots + h_D \mu_{l_{2D+1}}))^2}{2v(l_1, \cdots, l_{2D+1})} - \frac{1}{2} \log v(l_1, \cdots, l_{2D+1}) \Bigg) \cdot p_{X_{r+D}, X_{r+D-1}, \cdots, X_{r-D} | \mathbf{Y}}(x_{r+D} = l_1, x_{r+D-1} = l_2, \cdots, x_{r-D} = l_{2D+1} | \mathbf{y}, \boldsymbol{\theta}(p-1)) \Bigg\} \tag{A.3}$$

If we use the single variance form $v(l_1, \cdots, l_{2D+1}) = \sigma_k^2$, this becomes

$$\begin{aligned}\frac{\partial Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1))}{\partial \sigma_k} &= \sum_{r=1}^{N} \sum_{l_1=1}^{L} \cdots \sum_{l_D=1}^{L} \sum_{l_{D+2}=1}^{L} \cdots \sum_{l_{2D+1}=1}^{L} \\ &\Bigg\{ \left(\frac{(y_r - (h_{-D}\mu_{l_1} + \cdots + h_D \mu_{l_{2D+1}}))^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right) \\ &\cdot p_{X_{r+D}, \cdots, X_r, \cdots, X_{r-D}|\mathbf{Y}}(x_{r+D} = l_1, \cdots, x_r = k, \cdots, \\ &x_{r-D} = l_{2D+1} | \mathbf{y}, \boldsymbol{\theta}(p-1)) \Bigg\} \\ &= 0 \end{aligned} \tag{A.4}$$

Therefore,

$$\begin{aligned} &\sum_{r=1}^{N} \sum_{l_1=1}^{L} \cdots \sum_{l_D=1}^{L} \sum_{l_{D+2}=1}^{L} \cdots \sum_{l_{2D+1}=1}^{L} \\ &\Big(y_r - (h_{-D}\mu_{l_1} + h_{-D+1}\mu_{l_2} + \cdots + h_D \mu_{l_{2D+1}}) \Big)^2 \\ &\cdot p_{X_{r+D}, \cdots, X_r, \cdots, X_{r-D}|\mathbf{Y}}(x_{r+D} = l_1, \cdots, x_r = k, \cdots, \\ &x_{r-D} = l_{2D+1} | \mathbf{y}, \boldsymbol{\theta}(p-1)) \\ &- \sigma_k^2 \sum_{r=1}^{N} \sum_{l_1=1}^{L} \cdots \sum_{l_D=1}^{L} \sum_{l_{D+2}=1}^{L} \cdots \sum_{l_{2D+1}=1}^{L} \\ &p_{X_{r+D}, \cdots, X_r, \cdots, X_{r-D}|\mathbf{Y}}(x_{r+D} = l_1, \cdots, x_r = k, \cdots, \\ &x_{r-D} = l_{2D+1} | \mathbf{y}, \boldsymbol{\theta}(p-1)) = 0 \end{aligned} \tag{A.5}$$

Let

$$\begin{aligned}N_k = &\sum_{r=1}^{N}\sum_{l_1=1}^{L}\cdots\sum_{l_D=1}^{L}\sum_{l_{D+2}=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L} \\ &\quad p_{X_{r+D},\cdots,X_r,\cdots,X_{r-D}|\mathbf{Y}}(x_{r+D} = l_1, \cdots, x_r = k, \cdots, \\ &\qquad x_{r-D} = l_{2D+1}|\mathbf{y}, \boldsymbol{\theta}(p-1)) \\ = &\sum_{r=1}^{N} p_{X_r|\mathbf{Y}}(k|\mathbf{y}, \boldsymbol{\theta}(p-1))\end{aligned} \tag{A.6}$$

then from (A.5) we can get

$$\begin{aligned}\sigma_k^2 = \frac{1}{N_k}\Bigg\{ &\sum_{r=1}^{N}\sum_{l_1=1}^{L}\cdots\sum_{l_D=1}^{L}\sum_{l_{D+2}=1}^{L}\cdots\sum_{l_{2D+1}=1}^{L} \\ &\Big(y_r - (h_{-D}\mu_{l_1} + \cdots + h_{-1}\mu_{l_D} + h_0\mu_k + \\ &\qquad h_1\mu_{l_{D+2}} + \cdots + h_D\mu_{l_{2D+1}})\Big)^2 \\ &\cdot p_{X_{r+D},\cdots,X_r,\cdots,X_{r-D}|\mathbf{Y}}(x_{r+D} = l_1, \cdots, x_r = k, \\ &\qquad \cdots, x_{r-D} = l_{2D+1}|\mathbf{y}, \boldsymbol{\theta}(p-1))\Bigg\}\end{aligned} \tag{A.7}$$

B. PARAMETRIC PATH FOR CONTINUITY POTENTIAL CALCULATION

$$u(t\,;\,l,w)=\begin{cases}-t-\frac{l}{2}-(\frac{\pi-2}{4})w & t\in T_1\\ (1-\cos(\frac{2t}{w}))\frac{w}{2}-\frac{l}{2} & t\in T_2\\ t-\frac{l}{2}-(\frac{\pi-2}{4})w & t\in T_3\end{cases} \tag{B.1}$$

$$v(t\,;\,l,w)=\begin{cases}-\frac{w}{2} & t\in T_1\\ \frac{w}{2}\sin(\frac{2t}{w}) & t\in T_2\\ \frac{w}{2} & t\in T_3\end{cases} \tag{B.2}$$

where

$$\begin{aligned}T_1&=\{t\in R|-l-(\frac{\pi-4}{4})w\leq t<-\frac{\pi}{4}w\}\\ T_2&=\{t\in R|-\frac{\pi}{4}w\leq t<\frac{\pi}{4}w\}\\ T_3&=\{t\in R|\frac{\pi}{4}w\leq t\leq l+(\frac{\pi-4}{4})w\}.\end{aligned} \tag{B.3}$$

C. THE RATIO OF KERNELS FOR THE SWITCHING KERNEL

Let an object x_i of type m be switched to an object $\hat{x}_i$ of type n such that the current configuration $\mathbf{x}$ is perturbed into the configuration $\mathbf{x}' = \mathbf{x} - \{x_i\} \bigcup \hat{x}_i$. And if the number of parameter of x_i is different from that of $\hat{x}_i$, we need to make them be equal with auxiliary variables as $(\hat{x}_i, v_{nm})$ and (x_i, u_{mn}). Then the ratio of kernel is expressed by:

$$\frac{Q_S(\mathbf{x}' \to \mathbf{x})}{Q_S(\mathbf{x} \to \mathbf{x}')} = \frac{J_{nm}\varphi_{nm}(v_{nm})}{J_{mn}\varphi_{mn}(u_{mn})} \left| \frac{\partial \Psi_{mn}(x_i, u_{mn})}{\partial(x_i, u_{mn})} \right| \tag{C.1}$$

where $\Psi_{mn}(x_i, u_{mn})$ is a mapping between $(\hat{x}_i, v_{nm})$ and (x_i, u_{mn}) given by:

$$(\hat{x}_i, , v_{nm}) = \Psi(x_i, u_{mn}) \tag{C.2}$$

In our switching kernel, the number of parameters of both types is equal. Therefore, (C.1) becomes:

$$\frac{Q_S(\mathbf{x}' \to \mathbf{x})}{Q_S(\mathbf{x} \to \mathbf{x}')} = \frac{J_{nm}}{J_{mn}} \left| \frac{\partial \Psi_{mn}(x_i)}{\partial x_i} \right| \tag{C.3}$$

Now, let a necking channel object $x_i = (a_i, b_i, l_i, w_i, \theta_i)$ be switched to denting channel object according to switching type 1. And, let the probability to choose switching from a necking channel to a denting channel be J_{ND} and denting to necking be J_{DN}. In our case, a denting and a necking channel are equiprobable($J_{DN} = J_{ND}$). Again, for the switching of type 1, let the probability to choose switching from a necking channel to a denting channel be J'_{ND} and denting to necking be J'_{DN}. We want switching type 1 and 2 are equiprobable. Therefore, $J'_{ND} = \frac{J_{ND}}{2}$, $J'_{DN} = \frac{J_{DN}}{2}$ and $\frac{J'_{DN}}{J'_{ND}} = \frac{J_{DN}}{J_{ND}} = 1$. Then, (C.3) becomes:

$$\frac{Q_S(\mathbf{x}' \to \mathbf{x})}{Q_S(\mathbf{x} \to \mathbf{x}')} = \left| \frac{\partial \Psi_{ND}(x_i)}{\partial x_i} \right| \tag{C.4}$$

where, for the 'case 1' of 'switching type 1' in Figure 3.8(a),

$$\hat{x}_i = \Psi_{ND}(x_i) = \begin{bmatrix} a_i + \frac{w_i}{8}\cos(\theta_n) \\ b_i + \frac{w_i}{8}\sin(\theta_n) \\ l_i - \frac{w_i}{4} \\ w_i \\ \theta_i \end{bmatrix}^{\mathrm{T}} . \tag{C.5}$$

And,

$$\begin{aligned} &\left| \frac{\partial \Psi_{ND}(x_i)}{\partial x_i} \right| = \\ &\begin{vmatrix} 1 & 0 & 0 & \frac{\cos(\theta_i)}{8} & -\frac{w_i}{8}\sin(\theta_i) \\ 0 & 1 & 0 & \frac{\sin(\theta_i)}{8} & \frac{w_i}{8}\cos(\theta_i) \\ 0 & 0 & 1 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \\ &= 1. \end{aligned} \tag{C.6}$$

For the 'case 2',

$$\hat{x}_i = \Psi_{ND}(x_i) = \begin{bmatrix} a_i - \frac{w_i}{8}\cos(\theta_n) \\ b_i - \frac{w_i}{8}\sin(\theta_n) \\ l_i - \frac{w_i}{4} \\ w_i \\ \theta_i + \pi \end{bmatrix}^{\mathrm{T}}. \tag{C.7}$$

And,

$$\begin{aligned} \left|\frac{\partial \Psi_{ND}(x_i)}{\partial x_i}\right| &= \\ &\begin{vmatrix} 1 & 0 & 0 & -\frac{\cos(\theta_i)}{8} & \frac{w_i}{8}\sin(\theta_i) \\ 0 & 1 & 0 & -\frac{\sin(\theta_i)}{8} & -\frac{w_i}{8}\cos(\theta_i) \\ 0 & 0 & 1 & -\frac{1}{4} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \\ &= 1. \end{aligned} \tag{C.8}$$

Therefore, in both cases,

$$\frac{Q_S(\mathbf{x}' \to \mathbf{x})}{Q_S(\mathbf{x} \to \mathbf{x}')} = 1. \tag{C.9}$$

And then, let us consider 'case 3' of 'switching type 1'.

When $\theta_i \leq \pi$,

$$\hat{x}_i = \Psi_{DN}(x_i) = \begin{bmatrix} a_i - \frac{w_i}{8}\cos(\theta_n) \\ b_i - \frac{w_i}{8}\sin(\theta_n) \\ l_i + \frac{w_i}{4} \\ w_i \\ \theta_i \end{bmatrix}^{\mathrm{T}}. \tag{C.10}$$

Or, when $\theta_i > \pi$,

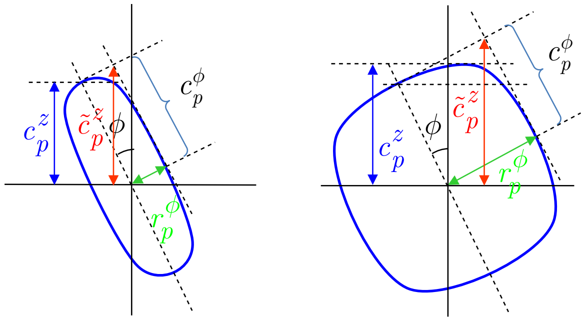
$$\hat{x}_i = \Psi_{DN}(x_i) = \begin{bmatrix} a_i - \frac{w_i}{8}\cos(\theta_n) \\ b_i - \frac{w_i}{8}\sin(\theta_n) \\ l_i + \frac{w_i}{4} \\ w_i \\ \theta_i - \pi \end{bmatrix}^{\mathrm{T}}. \tag{C.11}$$

In both cases,

$$\begin{aligned} &\frac{Q_S(\mathbf{x}' \to \mathbf{x})}{Q_S(\mathbf{x} \to \mathbf{x}')} = \left| \frac{\partial \Psi_{DN}(x_i)}{\partial x_i} \right| = \\ &\begin{vmatrix} 1 & 0 & 0 & -\frac{\cos(\theta_i)}{8} & \frac{w_i}{8}\sin(\theta_i) \\ 0 & 1 & 0 & -\frac{\sin(\theta_i)}{8} & -\frac{w_i}{8}\cos(\theta_i) \\ 0 & 0 & 1 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \\ &= 1. \end{aligned} \tag{C.12}$$

Similarly, The ratio of the kernels of switching type 2 is also equal to 1.

D. DERIVATION OF THE RANGE OF SEMI-AXIS LENGTH C



(a) (b)

Fig. D.1. Derivation of c_p^m and c_p^M

Figure D.1 illustrates side view of 3D object when the tilt angle is ϕ_p. The vertical direction is z direction and horizontal line is $x-y$ plain. From this figure, by geometric calculation, we can derive some relations as below:

$$c_p^\phi = \frac{\tilde{c}_p^z}{\cos(\phi_p)} - r_p^\phi \tan(\phi_p) \tag{D.1}$$

$$c_p^z \leq \tilde{c}_p^z \leq c_p^z + r_p^\phi \sin(\phi_p) \tag{D.2}$$

From (D.1) and (D.2), we can get

$$c_p^\phi \leq \frac{c_p^z + r_p^\phi \sin(\phi_p)}{\cos(\phi_p)} - r_p^\phi \tan(\phi_p) = \frac{c_p^z}{\cos(\phi_p)} \tag{D.3}$$

Therefore, the maximum value of c_p^{ϕ} is given by

$$c_p^M = \frac{c_p^z}{\cos(\phi_p)} \tag{D.4}$$

From (D.1) and (D.2), we can also get

$$c_p^{\phi} \geq \frac{c_p^z}{\cos(\phi_p)} - r_p^{\phi} \tan(\phi_p) \tag{D.5}$$

Here,

$$r_p^{\phi} \leq \sqrt{(a_{max}^{3D})^2 + (b_{max}^{3D})^2} \tag{D.6}$$

then

$$c_p^{\phi} \geq \frac{c_p^z}{\cos(\phi_p)} - \sqrt{(a_{max}^{3D})^2 + (b_{max}^{3D})^2}\tan(\phi_p) \tag{D.7}$$

Finally, the minimum value of c_p^{ϕ} is given by

$$c_p^m = \frac{c_p^z}{\cos(\phi_p)} - \sqrt{(a_{max}^{3D})^2 + (b_{max}^{3D})^2}\tan(\phi_p) \tag{D.8}$$

LIST OF REFERENCES

LIST OF REFERENCES

[1] D. Dimiduk, "Microstructure-Property-Design relationships in the simulation era: An introduction," in *Computational Methods for Microstructure-Property Relationships* (S. Ghosh and D. Dimiduk, eds.), Springer, 2011.

[2] H. C. Chuang, L. M. Huffman, M. L. Comer, J. P. Simmons, and I. Pollak, "An automated segmentation for nickel-based superalloy," in *IEEE International Conference on Image Processing*, pp. 2280–2283, Oct. 2008.

[3] B. Maruyama, J. E. Spowart, D. J. Hooper, H. M. Mullens, A. M. Druma, C. Druma, and M. K. Alam, "A new technique for obtaining three-dimensional structures in pitch-based carbon foams," *Scripta Materialia*, vol. 54, no. 9, pp. 1709–1713, 2006.

[4] H. Panahi, M. Kobchenko, F. Renard, A. Mazzini, J. Scheibert, D. K. Dysthe, B. Jamtveit, A. Malthe-Sørenssen, and P. Meakin, "A 4D synchrotron X-ray tomography study of the formation of hydrocarbon migration pathways in heated organic-rich shale," *ArXiv e-prints*, Jan. 2014.

[5] L. Wojnar, *Image analysis. Applications in materials engineering.* CRC Press, 1999.

[6] L. Duval, M. Moreaud, C. Couprie, D. Jeulin, H. Talbot, and J. Angulo, "Image processing for materials characterization: Issues, challenges and opportunities," in *IEEE International Conference on Image Processing*, pp. 4862–4866, Oct. 2014.

[7] D. W. Kim and M. L. Comer, "Channel detection in microscope images of materials using marked point process modeling," in *IEEE International Conference on Image Processing*, pp. 3054–3058, Sep. 2015.

[8] H. Zhao, M. L. Comer, and M. D. Graef, "A unified Markov random field / marked point process image model and its application to computational materials," in *IEEE International Conference on Image Processing*, pp. 1713–1716, Oct. 2014.

[9] Y. Leng, *Materials Characterization- Introduction to Microscopic and Spectroscopic Methods.* Wiley-VCH, 2013.

[10] M. A. Groeber, "Digital representation of materials grain structure," in *Computational Methods for Microstructure-Property Relationships* (S. Ghosh and D. Dimiduk, eds.), Springer, 2011.

[11] A. J. Baddeley and M. N. M. Van Lieshout, "Stochastic geometry models in high-level vision," *Journal of Applied Statistics*, vol. 20, no. 5-6, pp. 231–256, 1993.

[12] X. Descombes, R. Minlos, and E. Zhizhina, "Object extraction using a stochastic birth-and-death dynamics in continuum," *Journal of Mathematical Imaging and Vision*, vol. 33, no. 3, pp. 347–359, 2009.

[13] M. Ortner, X. Descombes, and J. Zerubia, "Building outline extraction from digital elevation models using marked point processes," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 107–132, 2007.

[14] M. S. Kulikova, I. H. Jermyn, X. Descombes, E. Zhizhina, and J. Zerubia, "Extraction of arbitrarily shaped objects using stochastic multiple birth-and-death dynamics and active contours," in *SPIE, Computational Imaging VIII*, Feb. 2010.

[15] M. Kulikova, I. Jermyn, X. Descombes, J. Zerubia, and E. Zhizhina, "A marked point process model with strong prior shape information for the extraction of multiple, arbitrarily-shaped objects," in *International Conference on Signal Image Technology and Internet Based Systems*, pp. 180–186, Nov. 2009.

[16] C. Lacoste, X. Descombes, and J. Zerubia, "Point processes for unsupervised line network extraction in remote sensing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1568–1579, 2005.

[17] D. Chai, W. Forstner, and F. Lafarge, "Recovering line-networks in images by junction-point processes," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1894–1901, June 2013.

[18] Á. Utasi and C. Benedek, "A 3-D marked point process model for multi-view people detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3385–3392, June 2011.

[19] M. L. Comer and E. Delp, "The EM/MPM algorithm for segmentation of textured images: analysis and further experimental results," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1731–1744, 2000.

[20] L. Reimer, *Scanning Electron Microscopy: Physics of Image Formation and Microanalysis.* Springer-Verlag, 1998.

[21] G. Wilkening and L. Koenders, *Nanoscale Calibration Standards And Methods: dimensional and related measurements in the micro-and nanometer range.* Wiley-VCH, 2005.

[22] D. W. Kim and M. L. Comer, "Segmentation of materials images using 3D electron interaction modeling," in *SPIE, Computational Imaging XI*, vol. 8657, Feb. 2013.

[23] J. M. Hammersley and P. Clifford, "Markov field on finite graphs and lattices," *unpublished*, 1971.

[24] R. Kinderman and J. L. Snell, *Markov Random Fields and Their Applications.* American Mathematical Society, 1980.

[25] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society B*, vol. 36, pp. 192–236, 1974.

[26] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 76–89, 1987.

[27] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, 1952.

[28] M. Uchic, M. Groeber, D. Dimiduk, and J. P. Simmons, "3D microstructural characterization of nickel superalloys via serial-sectioning using a dual beam fibsem," *Scripta Materialia*, vol. 55, no. 1, pp. 23–28, 2006.

[29] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *SCIENCE*, vol. 220, no. 4598, pp. 671–680, 1983.

[31] D. W. Kim and M. L. Comer, "Joint deconvolution/segmentation of microscope images of materials," in *IEEE Statistical Signal Processing Workshop*, pp. 688–691, Aug. 2012.

[32] S. Descamps, X. Descombes, A. Bechet, and J. Zerubia, "Automatic flamingo detection using a multiple birth and death process," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1113–1116, March 2008.

[33] F. Lafarge, G. Gimel'farb, and X. Descombes, "Geometric feature extraction by a multimarked point process," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1597–1609, 2010.

[34] X. Descombes, *Stochastic geometry for image analysis.* ISTE Ltd, 2012.

[35] R. Stoica, X. Descombes, and J. Zerubia, "A Gibbs point process for road extraction from remotely sensed images," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 121–136, 2004.

[36] G. Perrin, X. Descombes, and J. Zerubia, "Adaptive simulated annealing for energy minimization problem in a marked point process application," in *Energy Minimization Methods in Computer Vision and Pattern Recognition* (A. Rangarajan, B. Vemuri, and A. Yuille, eds.), vol. 3757 of *Lecture Notes in Computer Science*, pp. 3–17, Springer Berlin Heidelberg, 2005.

[37] U. Grenander and M. I. Miller, "Representations of knowledge in complex systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 56, no. 4, pp. 549–603, 1994.

[38] A. Gamal Eldin, X. Descombes, G. Charpiat, and J. Zerubia, "Multiple birth and cut algorithm for multiple object detection," *Journal of Multimedia Processing and Technologies*, 2012.

[39] E. Soubiès, P. Weiss, and X. Descombes, "A 3D segmentation algorithm for ellipsoidal shapes. application to nuclei extraction.," in *International Conference on Pattern Recognition Applications and Methods*, pp. 97–105, Feb. 2013.

[40] G. Perrin, X. Descombes, and J. Zerubia, "Point processes in forestry : an application to tree crown detection," Research Report 5544, INRIA, France, 2003.

[41] C. Mallet, F. Lafarge, F. Bretar, U. Soergel, and C. Heipke, “Lidar waveform modeling using a marked point process,” in *IEEE International Conference on Image Processing*, pp. 1713–1716, Nov. 2009.

[42] A. Gamal Eldin, G. Charpiat, X. Descombes, and J. Zerubia, “An efficient optimizer for simple point process models,” in *SPIE, Computational Imaging XI*, vol. 8657, Feb. 2013.

[43] V. Kolmogorov and R. Zabin, “What energy functions can be minimized via graph cuts?,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 147–159, Feb. 2004.

[44] J. J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory,” in *Numerical Analysis* (G. Watson, ed.), vol. 630 of *Lecture Notes in Mathematics*, pp. 105–116, Springer Berlin Heidelberg, 1978.

VITA

VITA

Dae Woo Kim was born in Yeoncheon, South Korea. He earned his B.S. and M.Eng. degrees in Electrical Engineering from Yonsei University, Seoul, South Korea, in 1995 and 1997, respectively. From 1997 until 2009, he was a Senior researcher at the Corporate Research Division in LG Electronics Inc., Seoul, South Korea. He is currently pursuing the Ph.D. degree under the supervision of Professor Mary L. Comer in Electrical and Computer Engineering at Purdue University, West Lafayette, IN. His research at Purdue mainly focused on segmentation and feature detection of microscope images of materal. His research interests include texture-based segmentation of images, image and video compression, video analysis, signal processing, and computer vision. He is a student member of the IEEE.