January 2016

COMBINING MARKOV RANDOM FIELD AND MARKED POINT PROCESS FOR MICROSCOPY IMAGE MODELING

Huixi Zhao
Purdue University

Graduate School Form 30
(Revised 08/14)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Huixi Zhao

Entitled Combining Markov Random Field and Marked Point Process for Microscopy Image Modeling

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

MARY L. COMER

CHARLES A. BOUMAN

EDWARD J. DELP

MARK R. BELL

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 on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): MARY L. COMER

Approved by: V. Balakrishnan 08/11/2016

Head of the Department Graduate Program Date

COMBINING MARKOV RANDOM FIELD AND MARKED POINT PROCESS FOR MICROSCOPY IMAGE MODELING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Zhao Huixi

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2016

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

This research is supported by the Air Force Office of Scientific Research, MURI contract # FA9550-12-1-0458.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Huixi, Zhao Ph.D., Purdue University, December 2016. Combining Markov Random Field and Marked Point Process for Microscopy Image Modeling. Major Professor: Mary L. Comer.

In many microscopy image analysis applications, it is of critical importance to address both pixel-level segmentation and object-level feature extraction. While Markov random field (MRF) models are useful for imposing local constraints at the pixel level, they have limited capability for imposing global constraints. Marked Point Process (MPP) models incorporate global information, such as shape, as a prior, but local constraints, such as pixel-wise interaction, are not easily modeled.

To address the problem, we first propose a hybrid MRF/MPP model to incorporate both local and global constraints within one single energy function for image analysis. Optimization using this model is performed using simulation schemes, including reversible jump Markov chain Monte Carlo (RJ MCMC) and multiple birth and death algorithms.

Secondly, we propose a two-pass multiple birth and death algorithm. In the death step of the original multiple birth and death algorithm, objects which are killed in later stages might affect the accuracy of the death rate of objects processed earlier, especially for adjacent object pairs, where two objects have close interaction. In our algorithm, we add a rebirth step after the death step to solve this problem.

Finally, we propose a joint MRF/MPP model. Unlike the hybrid model, where the MRF is interpreted as an energy term within the MPP framework, this model combines the MRF and the MPP into a joint probability distribution. We show experiments to demonstrate the comparison of this model and the hybrid model.

1. INTRODUCTION

The Markov random field (MRF) and Marked point process (MPP) are two powerful image modeling tools. In the MRF model, the Markov property, represented by a Gibbs distribution, can model pixel-wise interactions. Its effectiveness in imposing local smoothness constraints is well-suited for image analysis. However, it has limitations for imposing global constraints. For example, the geometric information of an image cannot be conveniently modeled using a MRF.

On the other hand, the MPP model assumes that objects are distributed following a random process consisting of points randomly located in space. Global constraints can be incorporated more easily into the MPP model, where in addition to variables representing the random locations of objects, a set of variables (marks) are assigned to describe features of the objects. Since the MPP model is very useful for modeling the randomness of the locations and numbers of the target objects, it has been used in various object detection applications [1, 2]. The drawback of this object-level representation lies in its difficulty in modeling local constraints, such as pixel-wise interactions. As a result, in most cases the boundary information of the target objects can not be precisely described. Furthermore, the accuracy of the object detection will decline if the object boundaries do not follow exactly the geometric model.

In some image analysis applications, both pixel-level and object-level representations are of critical importance. For example, in microscopy images of material systems, object-level microstructural features such as particle size and shape are essential for identifying structure property correlations; pixel-level segmentation can provide information to help characterize material properties that are determined by the activity at interfaces between regions with different physical or chemical compositions. Unfortunately, neither MPP nor MRF alone is adequate to model this hierarchical two-level information.

We propose two models to perform both pixel classification (segmentation) and object detection/identification. Specifically, the dissertation is organized as follows:

In Chapter 2, we introduce some general MPP models from the literature and apply them to extract microstructure features from material images. We also describe two optimization methods for the MPP model: multiple birth and death simulation and reversible jump Markov chain Monte Carlo simulation [1,3,4].

In Chapter 3, we propose a modified multiple birth and death algorithm with a two-pass death step. Such a modified algorithm serves as a faster alternative to the original multiple birth and death algorithm, which can be used as an optimization approach for both the MPP model and the hybrid MRF/MPP model.

In Chapter 4, we propose a hybrid MRF/MPP model, where the geometric properties of objects (global constraints) and pixel-wise interactions (local constraints) are integrated into one energy function. We perform optimization with this model using Monte Carlo simulation. Specifically, we employ multiple birth and death simulation and reversible jump Markov chain Monte Carlo simulation for optimization for object detection and segmentation.

In Chapter 5,we propose a joint MRF/MPP model. In this model, a conditional posterior distribution of a joint object/segmentation configuration given an observed image is proposed. This model differs from the hybrid MRF/MPP model in that, first, this model combines the MRF and the MPP as a joint model, while the hybrid model interprets MRF as a segmentation potential, which is embedded in the MPP framework; and second, the relation between the object field and segmentation is modeled as part of the prior distribution in our joint model. The joint object/segmentation configuration is obtained by maximizing the posterior distribution. An alternating minimization algorithm is used for optimization with this model.

2. MARKED POINT PROCESS MODELS FOR MICROSCOPE IMAGES

Marked point process (MPP) modeling [5–9] is a stochastic approach for object detection. A *point process* model is very useful for modeling the random locations of objects in an image. A set of random variables (or marks) can then be associated with each point, or object, to describe properties of that object, leading to a *marked point process.*

Consider S the image lattice, which is a compact subset of $\mathbb{R}^2$. A configuration of n objects in an image is a finite unordered set of points $\{S_1, S_2, ..., S_n\} \subset S$ with each S_i representing the random location of one object. For each object S_i, there is a mark M_i. The mark for an object contains random variables describing the object. Note that we will use the term "mark" to refer to both the collection of random variables M_i and each random variable in M_i itself. A marked object is defined as $W_i = \{S_i, M_i\} \in W$, where $W \subset S \times M$ is the random marked object field. Taking the rectangle model as an example, let M be the mark space, $M = [a_{min}, a_{max}] \times [b_{min}, b_{max}] \times [0, \pi]$, where the mark consists of two axis lengths and one orientation. Let Ω_W denote the space of all possible realizations of W. Then $w = (w_1, ..., w_n) \in \Omega_W$ is a possible object configuration.

We introduce a non-homogeneous Gibbs process on the configuration space. Let y denote the observed image. The density of the marked point process is:

$$f(w|y) = \frac{1}{z_{mpp}} exp\{-V_{mpp}(w|y)\} \tag{2.1}$$

where z_{mpp} is the normalizing constant. The energy function $V_{mpp}(w|y)$ is divided into two types of energy: the data energy $V^d_{mpp}(y|w)$ which describes how well object w fits image y and the prior energy $V^p_{mpp}(w)$, describing the interactions between objects. Thus, the most likely configuration corresponds to a *maximum a posteriori* MAP estimation of w:

$$\hat{w} = \underset{w \in W}{\arg\max} f(w|y) = \underset{w \in W}{\arg\min} [V_{mpp}^{d}(y|w) + V_{mpp}^{p}(w)] \tag{2.2}$$

Usually, this MAP estimation is performed by a stochastic sampling scheme.

Since the number and locations of objects in images are often random, the MPP is very suitable for object modeling. Recently, several MPP models have been shown to be quite effective for object detection. Among them, Descombes et al. [1] proposed an ellipse model to detect flamingos in remote sensing images. Perrin et al. [10, 11] used the ellipse model for tree crown detection in remote sensing images. Avenel et al. [12] used the ellipse model for breast cancer detection on medical images. Dong et al. [13] used the ellipse model for rolling leukocytes detection in medical images. Craciun et al. [14] also used the ellipse model for boat detection in harbors on high resolution optical images. Yongtao et al. [15] and Gomes et al. [16] used a disk model for tree detection from airborne / mobile laser scanning or LiDAR point cloud data. Neeraj et al. [17] also used the disk model for Nuclei detection in microscopy images. Kulikova et al. [18] included the active contour as an energy term in the MPP for tree crown detection. Kulikova et al. [19] also used the active contour for nuclei extraction from histopathological images. Kim et al. [20] proposed a channel model to detect channels in microscope images of materials. Ortner et al. [21, 22] and Benedek et al. [23] proposed a rectangle model to represent buildings in remote sensing images. Borcs and Benedek [24] also used the rectangle model for vehicle detection in aerial images. Borcs and Benedek [25] further advanced a L-2 MPP model which included a prior term for groups of objects for vehicle detection. Stoica et al. [26] and Lacoste et al. [2] proposed the “Candy model” to represent roads in aerial images as line segments. Batool et al. [27] also used the Candy model to detect wrinkles in aging human faces. Dengfeng et al. [28] modeled the joint points of connecting line segments as objects to detect line structures in general images. Keresztes et al. [29] proposed a parabola model for fault detection in 2D sections of seismic blocks. Weina et al. [30] proposed a mixture of Bernoulli shape prototypes which are learned from training sequences to detect humans in video sequences. Utasi et al. [31, 32] extended this model to a 3-D model for multi-view people detection. Sreetama et al. [33] proposed a sphere model for Neurite tracing in medical imaging. Soubies et al. [34] proposed a 3D ellipsoid model for nuclei extraction. Craciun et al. [35] proposed a spatio-temporal marked point process for detection and tracking of moving objects.

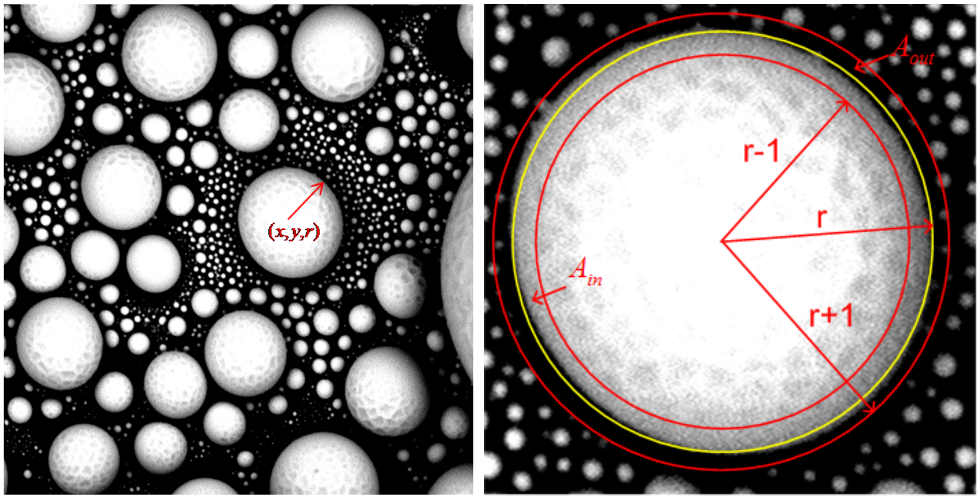
In this chapter, we apply marked point process to Tin ball images and NiCrAl images to extract material particles. We also represent the multiple birth and death algorithm to get the MAP estimation of marked object configuration in each example.

2.1 Tin image example

The geometric shape of Tin balls can be approximately interpreted as sphere. On 2D images, we use circle model as the marked object to describe each Tin ball. As in Figure 2.1(a), $(x, y) \in S$ represents the location of the object while $r \in M$ represents the radius, which is the only mark of the circle model. We assume for this work the radius is uniformly distributed between r_{min} and r_{max} value. Thus a circle object w_i is fully described as $w_i = \{(x_i, y_i), r_i\}$.

Let $D_{w_i} \subset S$ be the pixel site region projected by w_i onto the image lattice. We define the neighborhood system in (2.2) as: w_i and w_j are neighbors, denoted $w_i \sim w_j$, if D_{w_i} intersects D_{w_j}.

For a given object w from the marked object field, $V^d_{mpp}(y|w)$ is defined as the data energy, which describes how well the object fits the observed image. We expect gray values of pixels outside of the Tin ball particle boundary to be statistically different from those within the boundary. Hence, for each w with marks $m = \{r\} \in M$, we associate two auxiliary objects w^α with marks $m^\alpha = \{r + 1\}$ and w^β with marks $m^\beta = \{r - 1\}$ at the same location as w. Then the inside region and outside region of



(a) Tin ball image

(b) Circle model

Fig. 2.1. Tin ball image and the circle model

w are defined as $A_{in}(w) = D_{w^{\alpha}} - D_w$ and $A_{out}(w) = D_w - D_{w^{\beta}}$ respectively (Figure 2.1 (b)).

We define function $\gamma^w_{in}(s)$ and $\gamma^w_{out}(s)$ as:

$$\begin{aligned} \gamma^w_{in}(s) &= \begin{cases} 1 & 20 < I(s) < 150 \\ 0 & otherwise \end{cases} \quad \text{for} \quad s \in A_{in}(w) \\ \gamma^w_{out}(s) &= \begin{cases} 1 & I(s) < 5 \\ 0 & otherwise \end{cases} \quad \text{for} \quad s \in A_{out}(w) \\ \gamma_{in}(w) &= \sum_{s \in A_{in}(w)} \gamma^w_{in}(s) \qquad \gamma_{out}(w) = \sum_{s \in A_{out}(w)} \gamma^w_{out}(s) \end{aligned} \tag{2.3}$$

here, $I(s)$ is the gray value of pixel s. Since there is not much noise in the Tin ball image, background pixels usually have gray values less than 5. Thus $\gamma_{out}(w)$ collect all the background pixels in $A_{out}(w)$. Likewise, in $A_{in}(w)$, $\gamma_{in}(w)$ collect pixels corresponding to Tin ball boundary. As a consequence, $\gamma_{in}(w)$ and $\gamma_{out}(w)$ provide information on how well $A_{in}(w)$ fits the Tin ball boundary and $A_{out}(w)$ fits the background.

Then we calculate $V^d_{mpp}(y|w)$ as:

$$V^d_{mpp}(y|w) = 1 - \frac{\Gamma(y|w)}{T} \tag{2.4}$$

where

$$\Gamma(y|w) = \frac{\gamma_{in}(w)}{|A_{in}(w)|} + \frac{\gamma_{out}(w)}{|A_{out}(w)|} \tag{2.5}$$

here T is the threshold, which is set as 1 in this application.

Inspired by the Strauss Model [36], we use an overlapping penalizer as a prior to describe the spatial interaction between objects. As in [37], the overlapping penalizer $R(w_i, w_j)$ is defined as the normalized intersection area, which is illustrated in Figure 2.2:

$$R(w_i, w_j) = \frac{D_{w_i} \cap D_{w_j}}{\min(D_{w_i}, D_{w_j})} \tag{2.6}$$

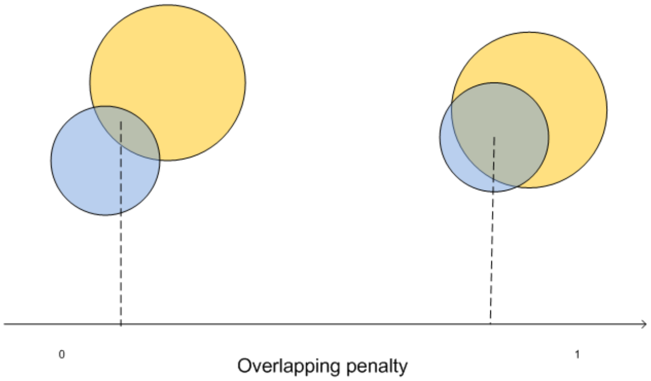


Fig. 2.2. Overlapping penalty for intersected objects

Then the prior energy $V^p_{mpp}(w)$ is defined as:

$$V^p_{mpp}(w_i, w_j) = \begin{cases} R(w_i, w_j) & if\ R(w_i, w_j)\ < Th_{overlapping} \\ \infty & otherwise \end{cases} \tag{2.7}$$

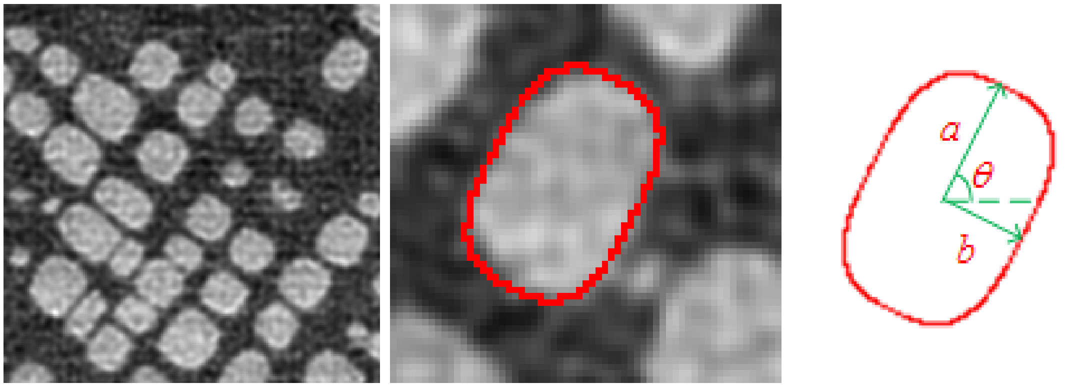
Here we assign zero probability for any overlapping ratio greater than $Th_{overlapping}$.

2.2 NiCrAl image example

In our application, a superellipse model is used as the object model to describe the geometric shape of a NiCrAl particle [38]. The equation for a superellipse is given by:

$$\left|\frac{x}{a}\right|^n + \left|\frac{y}{b}\right|^n = 1 \tag{2.8}$$

where a,b are the major and minor axes and n is a parameter to control the shape of the curve. The value $n = 3$ best fit the shape of NiCrAl particle [38]. We also introduce parameter θ to control the supperellispe orientation. Consequently, the mark space associated with the object space is defined as $M = [a_{\min}, a_{\max}] \times [b_{\min}, b_{\max}] \times [0, \pi]$. A NiCrAl image and the superellipse model are shown in Figure 2.3.

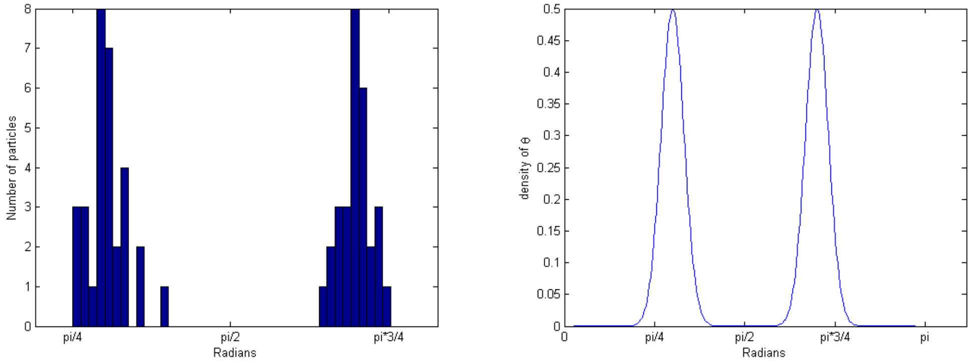


(a) NiCrAl image (b) superellipse model (c) marks of the model

Fig. 2.3. NiCrAl image and the superellipse model

We assume that a and b are uniformly distributed between the minimum and maximum value. Theoretically, orientation of NiCrAl particle has only two values [38], however, in realization there is usually some small fluctuation in the orientation, as in Figure 2.4(a). As a consequence, θ is assumed to follow a mixture of two Gaussian distributions with mean values $\frac{3}{10}\pi$ and $\frac{7}{10}\pi$. The variance is set as 0.1^2 for both Gaussians. The density of θ is illustrated in Figure 2.4(b).

Similar to what we did in the Tin ball example, for each object w, we use the statistical differences of pixel gray value between object's inside and outside region to



(a) histogram of particle orientations

(b) density of θ

Fig. 2.4. orientation of NiCrAl particles and the density of θ

define the data energy $V^d_{mpp}(w)$. Hence, for object w with marks $m = \{a, b, \theta\} \in M$, we associate two auxiliary objects w^α with marks $m^\alpha = \{a+1, b+1, \theta\}$ and w^β with marks $m^\beta = \{a-1, b-1, \theta\}$ at the same location as w. Then the inside region and outside region of w are defined as $A_{in}(w) = D_{w^\alpha} - D_w$ and $A_{out}(w) = D_w - D_{w^\beta}$ respectively.

We assume pixel gray values in $A_{in}(w)$ and $A_{out}(w)$ follow Gaussian distributions. This corresponds to a Gaussian noise observation model. We define the Bhattacharya distance [39] $B(y|w)$ between $A_{in}(w)$ and $A_{out}(w)$ as:

$$B(y|w) = \frac{(\mu_{in} - \mu_{out})^2}{4\sqrt{\sigma^2_{in} + \sigma^2_{out}}} - \frac{1}{2} \log \frac{2\sigma_{in}\sigma_{out}}{\sigma^2_{in} + \sigma^2_{out}} \tag{2.9}$$

where (μ_{in}, σ_{in}) and $(\mu_{out}, \sigma_{out})$ are Gaussian parameters for $A_{in}(w)$ and $A_{out}(w)$. Then the data energy $V^d_{mpp}(y|w)$ is defined as a likelihood function, which is calculated from $B(y|w)$, as in [1]:

$$V^d_{mpp}(y|w) = \begin{cases} 1 - \frac{B(y|w)}{T} & B(y|w) < T \\ \exp(-\frac{B(y|w)-T}{3B(y|w)}) - 1 & B(y|w) \geq T \end{cases} \tag{2.10}$$

Figure 2.5 shows a plot of the data energy as a function of the distance between the inside and outside region of w.

We use the overlapping penalizer in (2.7) to describe the spatial interaction between objects. Then $V^p_{mpp}(w)$ is calculated as (2.8).

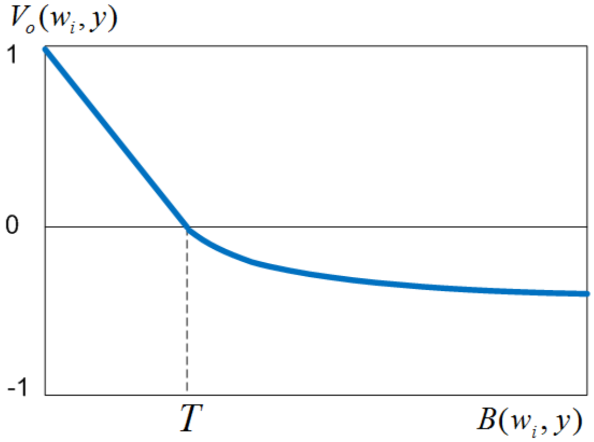


Fig. 2.5. Plot of $V^d_{mpp}(y|w)$

2.3 Optimization method

Our optimization goal is to find an object configuration that minimizes the energy function $V_{mpp}(w|y)$ and that has the minimum number of objects among all such configurations. In general there is no closed-form solution to obtain such an optimized configuration w, so we have to resort to iterative approaches.

2.3.1 Reversible jump Markov chain Monte Carlo algorithm

RJ MCMC [3, 4] is an extension of the Metropolis-Hastings algorithm, which allows simulation of a multidimensional system. A sampler is proposed to simulate a discrete time Markov Chain with stationary distribution π on the configuration space, where π is the density function whose mass is entirely concentrated on the set of configurations that minimize the energy function $V_{mpp}(w|y)$. Different kernels are introduced to allow perturbations and dimension changes in the configuration space, which is important since the number of objects in an image is random. Each kernel performs a state transition from w to w' according to a probability $Q_m(w \to w')$. The transition is accepted according to the Green ratio to meet the detailed balance equations, which ensures the process converges to π. The Green ratio is calculated as [3]:

$$R(w, w') = min\left(1, \frac{Q_m(w' \to w))exp(-V_{mpp}(w'|y))}{Q_m(w \to w'))exp(-V_{mpp}(w|y))}\right) \tag{2.11}$$

Three types of kernel are explored to perform state transitions:

- *Birth and death kernels*: These kernels will add or remove an object from the current configuration w. The corresponding mark is sampled according to specific distributions. The increase (birth kernel) and decrease (death kernel) in configuration dimension theoretically ensure the process visits the whole configuration space. However, this type of kernel alone usually leads to slow convergence. To speed up, two other kernels are used.

- *Perturbation kernel*: This kernel involves changes of mark for an object in the current configuration. The mark is updated according to specific distributions. The dimension of the configuration space remains the same.

- *Switch kernel*: This kernel allows an object of a certain type to change to another type. The dimension of the configuration space remains the same. However, since the object type changes, the corresponding mark is also modified.

Usually RJ MCMC is embedded within a simulated annealing [40, 41] framework, which replaces $V_{mpp}(w|y)$ by $V_{mpp}(w|y)/T_t$. A logarithmic decrease of T_t to 0 theoretically ensures the process converges to a global optimum [42, 43].

With the benefit of the switch kernel, RJ MCMC is very suitable for models including different types of objects.

2.3.2 Multiple Birth and Death algorithm

Although different kernels are proposed to speed up the process, RJ MCMC only treats one object at each iteration, which results in slow convergence speed. Descombes et al. [1] proposed a multiple birth and death algorithm to parallelize state transitions for simple object interaction models such as the overlapping penalty prior. It has been demonstrated to have faster convergence compared to RJ MCMC. Generally, the steps of the multiple birth and death algorithm are summarized as:

Initialization: start with an empty configuration $w = \phi$, set the birth rate $b\_rate = b_0$, the inverse temperature $\alpha = \alpha_0$ and the discretization step $\sigma = \sigma_0$.

- Birth step: visit all pixels on the image lattice in raster order. For each pixel at s, if no object is associated with it in the current configuration w, add w_i with probability σb_rate.

  a) if birth is chosen at s: $w = w \cup w_i$, sample each mark from its corresponding distribution.

  b) calculate $V^d_{mpp}(w_i)$, assign w_i to s.

- Death step: sort all elements in current configuration w by decreasing values of $V^d_{mpp}(w_i)$. For each w_i taken in this order, compute the death rate as:

$$d^1_{w_i} = \frac{\sigma \exp(-\alpha(V_{mpp}(w/\{w_i\}|y) - V_{mpp}(w|y)))}{1 + \sigma \exp(-\alpha(V_{mpp}(w/\{w_i\}|y) - V_{mpp}(w|y)))}$$

  Delete w_i from w with probability d_{w_i}.

- Convergence test: If and only if all the objects that are born during the birth step are killed during the death step, terminate the process. Otherwise, Let $\alpha = \alpha \times F$ and $\sigma = \sigma/F$ and go back to birth step, where F is the cooling factor and go back to the birth step.

Although the multiple birth and death algorithm is faster than RJ MCMC, it is limited to simple object interaction priors such as the overlapping penalizer. Its application to more sophisticated priors, such as the one in the Candy model, still remains to be explored [1].

2.4 Experimental results

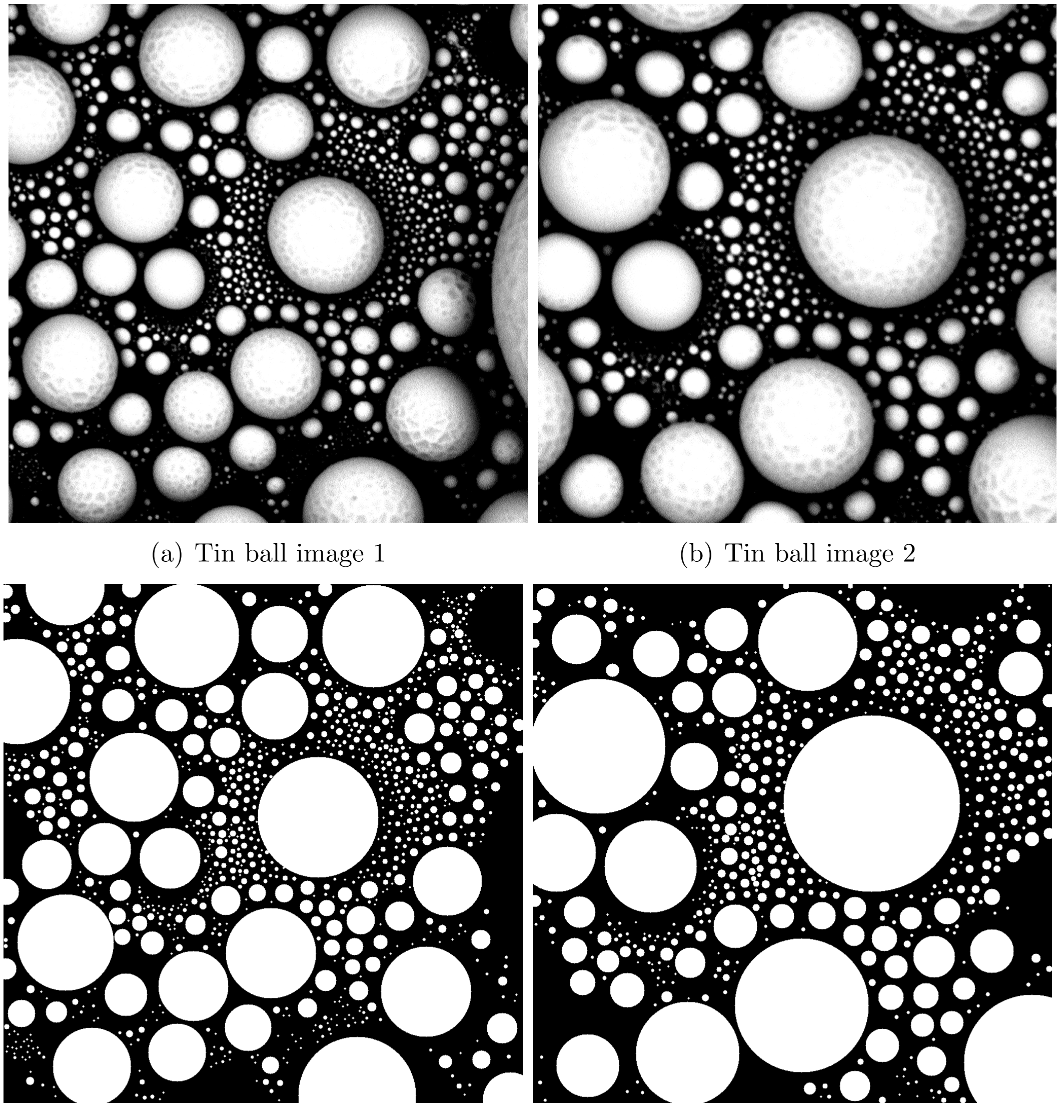
We apply marked point process to two examples: Tin ball images and NiCrAl images. The objective is to identify all the particles in each example. To examine the performance of our algorithm, we calculate the missed detection rate and false detection rate. The missed detection rate is defined as the ratio of the number of missed detecting objects to the total number of objects. The false detection rate is defined as the ratio of the number of falsely detecting objects to the total number of objects. Along the boundaries of the images in either example, some particles (Tin balls or NiCrAl particles) only show a small portion of themselves. Since they do not geometrically fit our model (circle model or supperellipse model), we do not consider them as effective objects in the images. Thus they are not taken into account for computing the missed detection rate and false detection rate.

In the Tin ball example, we set r_{min} as 2 and r_{max} as 200. $Th_{overlapping}$ is set at 0.2. T in (2.4) is set as 1. We apply marked point process to two Tin ball images with size 1024 × 1024. The algorithm is implemented in C and it takes about 30 minutes to get the result for each image on a i5 2.67 Ghz computer. Original images and results are shown in Figure 2.6. Both missed detection rate and false detection rate are shown in Table 2.1.

Table 2.1
Missed detection rate and false detection rate for Tin ball images

Image	Missed detection rate	False detection rate
Tin ball image 1	1.18%	0.04%
Tin ball image 2	1.39%	0.03%

Frome table 2.1, we can see our algorithm performs fairly good in identifying Tin balls. After further investigation, we find all the missed Tin balls are of small



(a) Tin ball image 1

(b) Tin ball image 2

(c) MPP result for Tin ball image 1

(d) MPP result for Tin ball image 2

Fig. 2.6. MPP result for Tin ball images

size with $r \leq 4$. Furthermore, all the missed Tin balls look dim in the background with average pixel gray value less than 30, while pixel gray values for background are around 5. For the false detection part, it is because some Tin balls are under the shadow of others, thus do not look like circle model any more. Both cases can be found in Figure 2.8(a-b).

In the NiCrAl particle example, we set $Th_{overlapping} = 0.2$. a_{min} and b_{min} are set as 5, while a_{max} and b_{max} are 25. T is set as 20. Two Tin NiCrAl images with size 744 × 645 are used to test our algorithm. For each image, the computing time is about 60 minutes on a i5 2.67 Ghz computer. Original images and results are shown in Figure 2.7. Both missed detection rate and false detection rate are shown in Table 2.2.

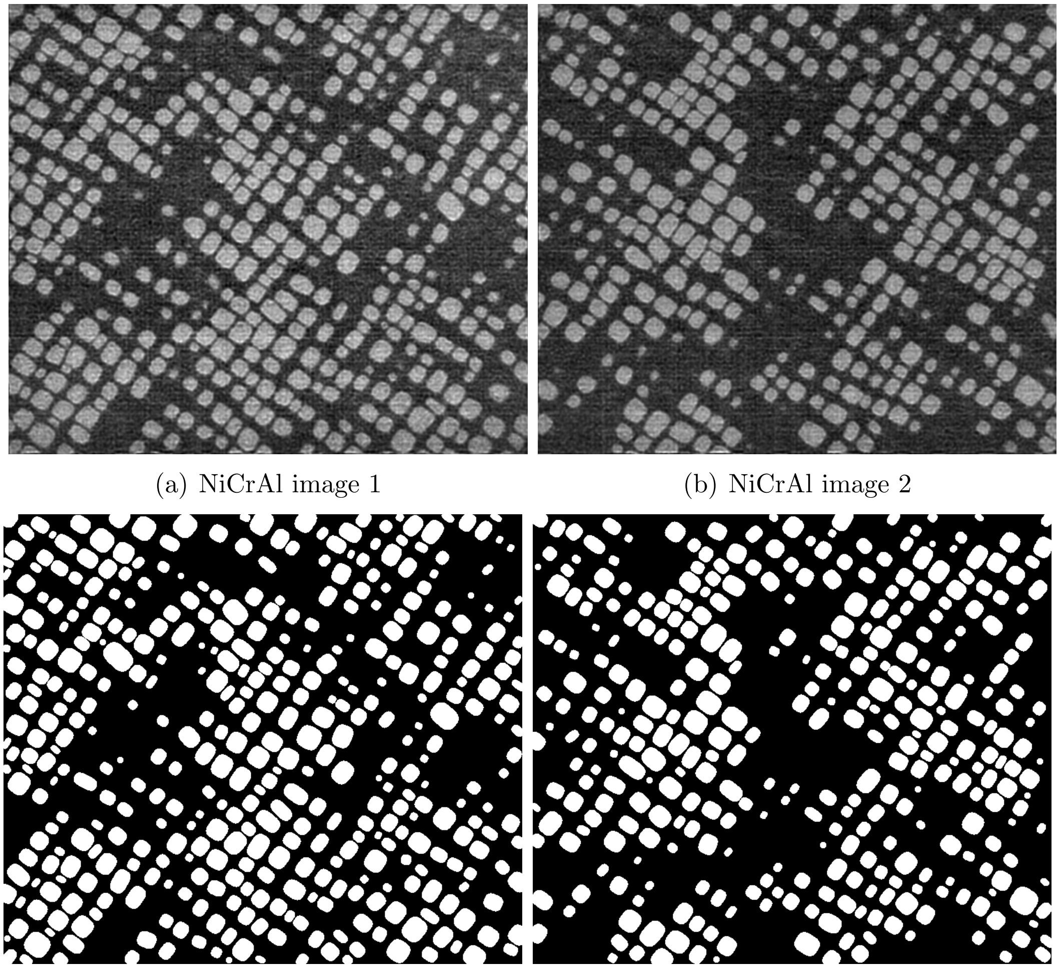
Table 2.2
Missed detection rate and false detection rate for NiCrAl images

Image	Missed detection rate	False detection rate
NiCrAl image 1	0.25%	0.25%
NiCrAl image 2	1.17%	0.30%

Similar as the Tin ball example, all the missed particles are of small size with major axis less than 6. In addition, the missed and falsely detected particles have complicated boundary information, which can not be well represented by the superellipse model. One example of the false detection is shown in Figure 2.8 (c) and (d).

2.5 Conclusion

In this chapter, we introduce the marked point process and the corresponding optimization methods: RJ MCMC and the multiple birth and death algorithm. We also apply MPP to Tin ball images and NiCrAl images to identify the material particles. The results show that MPP can effectively impose the geometric constraints on material particles, especially when the shape of particles can be well described by a mathematical model. It is very convenient to use MPP to address the interaction between particles as well. However, we should note that when the particle's shape is too complicated to be simply described by the object model, the MPP detection accuracy declines.



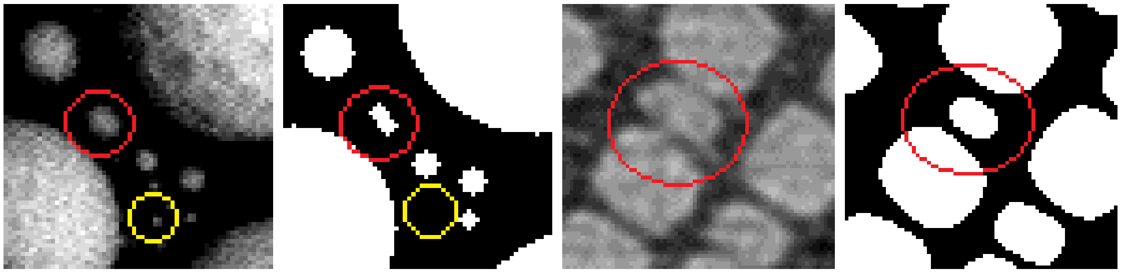
(a) NiCrAl image 1

(b) NiCrAl image 2

(c) MPP result for NiCrAl image 1

(d) MPP result for NiCrAl image 2

Fig. 2.7. MPP result for NiCrAl images



(a) Tin ball image (b) Tin MPP result (c) NiCrAl image (d) NiCrAl MPP result

Fig. 2.8. Examples of missed detection and false detection. In (a), the Tin ball circled in red looks more like an ellipse than a circle. It is identified as two overlaped objects in (b). The Tin ball circled in yellow is too small ($r = 2$) and dim, which is missed detected in (b). In (c), the circled NiCrAl particle is falsely detected because its complicated shape cannot be well modeled by the supperellipse model.

3. A TWO-PASS MULTIPLE BIRTH AND DEATH ALGORITHM

Multiple birth and death algorithm is an effective optimization method for the marked point process, which is widely used for object detection in various applications such as remote sensing and microscopy image analysis. However, in the death step, the objects which are killed in later process might affect the death rate of the objects treated earlier, especially for adjacent object pairs, where two objects have close interaction. In practice, this problem will lead to slow convergence and inaccurate result. In this chapter, we proposed a two-pass algorithm to solve this problem. We present experimental results to show our method has faster convergence speed and better performance than the original multiple birth and death algorithm.

3.1 Introduction

The multiple birth and death (MBD) algorithm, proposed in [1], is a popular optimization method for marked point process with simple object interaction priors such as the overlapping penalizer. Although [44, 45] proposed a multiple birth and cut algorithm to reduce the number of parameters in MBD, it has slower convergence speed and guarantees a local minimum. The general steps of MBD algorithm have been introduced in chapter 2. In the original MBD algorithm, the death step processes each object in the object configuration in descending order of the data term, in which each object is removed from the configuration based on the corresponding death rate. As the objects are handled one by one, the object configuration is dynamically updated accordingly. Since each object is treated just once, the death step is considered as a one-pass method.

For each object w_i, the death rate is calculated from the energy loss if it is removed from the object configuration at that time, which include both the data term and the prior term. The prior term describes the interaction between w_i and its neighbor objects in the current configuration. However, if its neighbor objects are killed in the later process of the death step, the prior term of w_i at this moment might not be reliable.

Although the MBD algorithm will theoretically converge to a minimum within the simulated annealing framework, in practice we find this problem will pose a hazard on the convergence speed and the accuracy of the result. In some applications, the process cannot converge within a reasonable time. For simplicity, in the death step, among all the objects sorted in descending order of the data term, we call such an object, which has interactions with the objects ahead of it and gets itself killed at its turn, a "phantom" object.

In this chapter, we advance a two-pass death step to solve the negative effect of the "phantom" objects. In Section 3.2, we describe this two-pass mechanism. In Section 3.3, we show the experiment results. We draw conclusion in Section 3.4.

3.2 A two-pass death step

We propose a two-pass method to find the "phantom" objects. The negative effects by the "phantom" objects are removed in the second pass. The modified MBD algorithm with the two-pass death step is as follows:

Initialization: start with an empty configuration $w = \phi$, set the birth rate $b\_rate = b_0$, the inverse temperature $\alpha = \alpha_0$ and the discretization step $\sigma = \sigma_0$.

- Birth step: visit all pixels on the image lattice in raster order. For each pixel at s, if no object is associated with it in the current configuration w, add w_i with probability σb_rate.

  a) if birth is chosen at s: $w = w \cup w_i$, sample each mark from its corresponding distribution.

b) calculate $V^d_{mpp}(w_i)$, assign w_i to s.

- Death step: Sort all objects in w in descending order of $V^d_{mpp}(w_i)$.

  a) first pass: Let $w' = \{w'_1, w'_2, ..., w'_n\}$ denote a copy of w, where $w'_i = w_i, \forall w_i \in w$. For every object w'_i taken in descending order of $V^d_{mpp}(w'_i)$, the corresponding death rate is calculated as:

  $$d^1_{w'_i} = \frac{\sigma \exp(-\alpha(V_{mpp}(w'/\{w'_i\}|y) - V_{mpp}(w'|y)))}{1 + \sigma \exp(-\alpha(V_{mpp}(w'/\{w'_i\}|y) - V_{mpp}(w'|y)))}$$

  Kill w'_i with probability $d^1_{w'_i}$. If death is chosen for w'_i, $w' = w'/\{w'_i\}$.

  b) second pass: sort all objects that are killed in the first pass in ascending order of $V^d_{mpp}(w'_i)$. For every object w'_i taken in this order, recalculate its death rate $d^2_{w'_i}$ based on the current object configuration w'. If $d^2_{w'_i}$ is different from $d^1_{w'_i}$, give rebirth to w'_i with probability $1 - d^2_{w'_i}$. If rebirth is chosen for w'_i

  1) find the "phantom" object w'_j, whose interaction with w'_i causes the difference between $d^1_{w'_i}$ and $d^2_{w'_i}$. In w, move w_j in front of w_i.

  2) $w' = w' \cup w'_i$.

  c) For every object w_i taken in the current order of w, calculate the death rate as:

  $$d_{w_i} = \frac{\sigma \exp(-\alpha(V_{mpp}(w/\{w_i\}|y) - V_{mpp}(w|y)))}{1 + \sigma \exp(-\alpha(V_{mpp}(w/\{w_i\}|y) - V_{mpp}(w|y)))}$$

  Kill w_i with probability d_{w_i}. If death is chosen for w_i, $w = w/\{w_i\}$.

- Convergence test: If and only if all the objects that are born during the birth step are killed during the death step, terminate the process. Otherwise, Let $\alpha = \alpha \times F$ and $\sigma = \sigma/F$ and go back to birth step, where F is the cooling factor and go back to the birth step.

The difference between this algorithm and the original MBD algorithm is the order we process the objects in the death step. In the death step of the original MBD algorithm, each object is processed in the order of decreasing values of the data term. In our algorithm, with the help of the second pass, we find the “phantom” objects and move them in front of the objects whose death rates might be negatively affected by these “phantom” objects. In this way, the “phantom” objects are processed earlier and the hazard brought by them can be avoided. Since the order of the death step only affects the convergence speed [44], our two-pass algorithm converges.

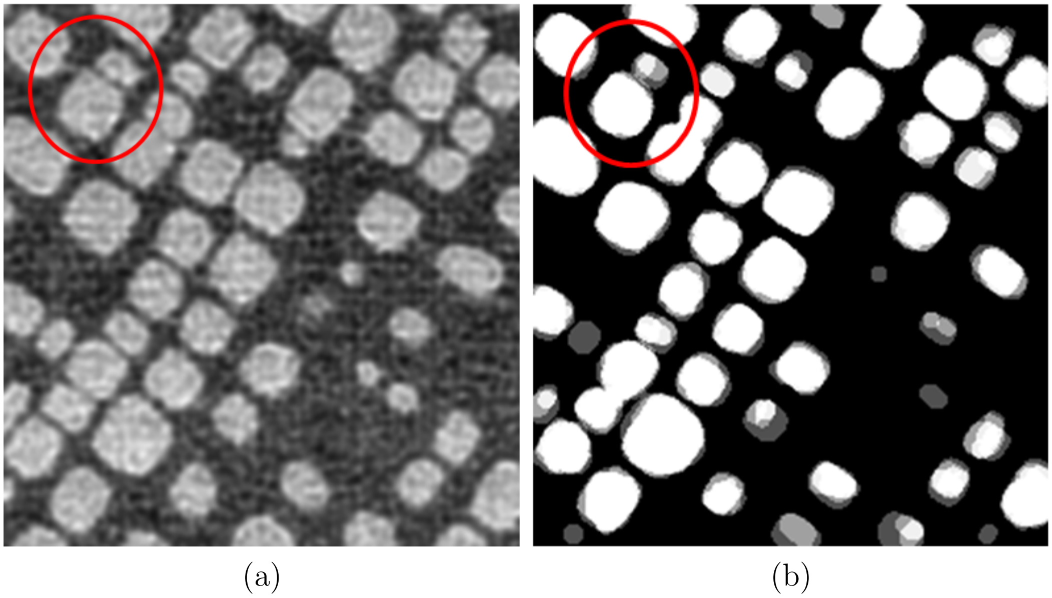


Fig. 3.1. (a) NiCrAl image (b) the result of the birth step.

We take the NiCrAl image as an example. We use the same model as in chapter 1, which use a superellipse model to define the data term and an overlapping penalizer as the prior term. We only focus on the two adjacent NiCrAl particles (circled in red) in Figure 3.2(a). The result of the birth step is presented in Figure 3.2(b), where objects are shown in grey. Brighter areas are the overlapping regions between objects. For the two particles of interest, there are two objects born for the upper particle and three objects for the lower particle. For clarity, we use different colors to represent these five objects. The green and purple ones are the objects for the upper-right particle while the pink, blue and yellow objects are for the lower-left particle.

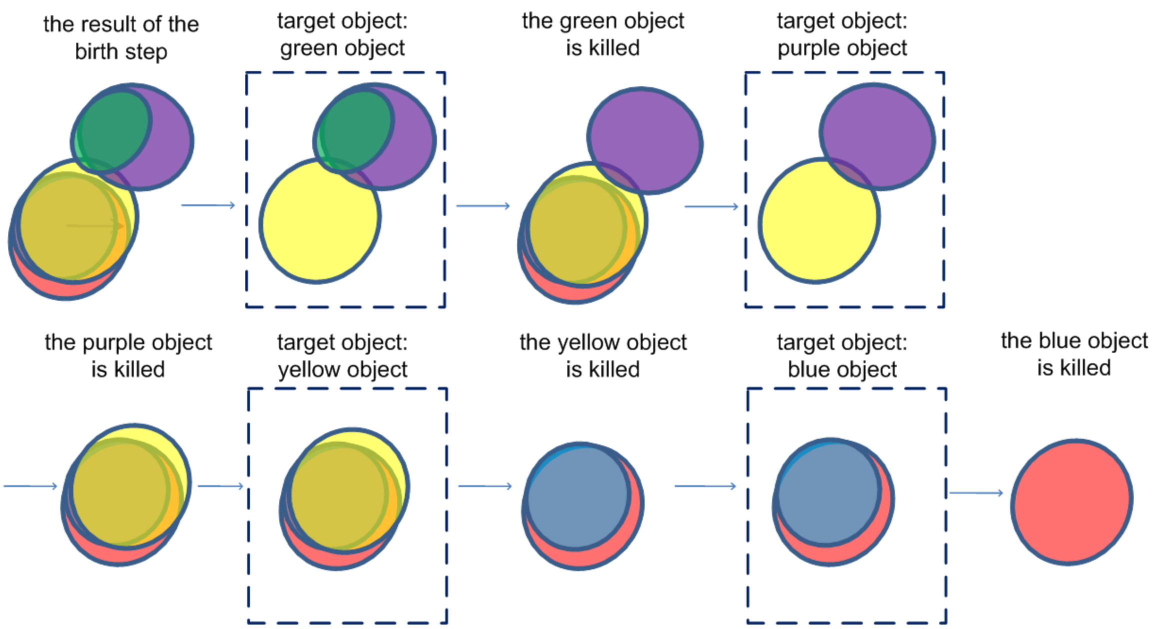

the result of the birth step
target object: green object
the green object is killed
target object: purple object
the purple object is killed
target object: yellow object
the yellow object is killed
target object: blue object
the blue object is killed

Fig. 3.2. The original death step for the two particles of interest

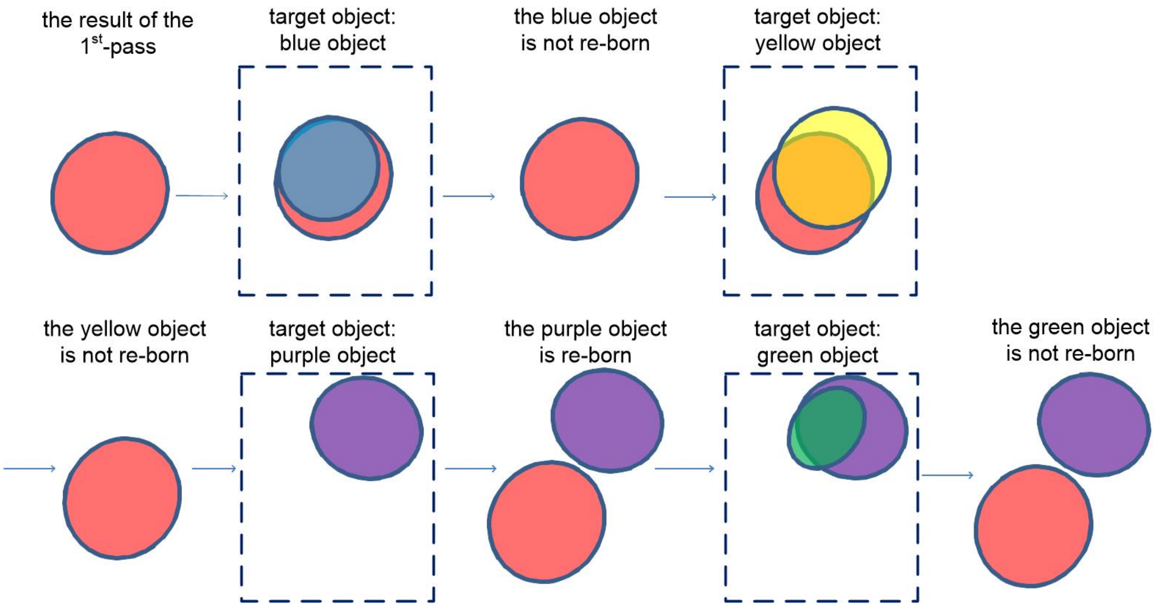

the result of the 1st-pass
target object: blue object
the blue object is not re-born
target object: yellow object
the yellow object is not re-born
target object: purple object
the purple object is re-born
target object: green object
the green object is not re-born

Fig. 3.3. The second pass for the two particles of interest

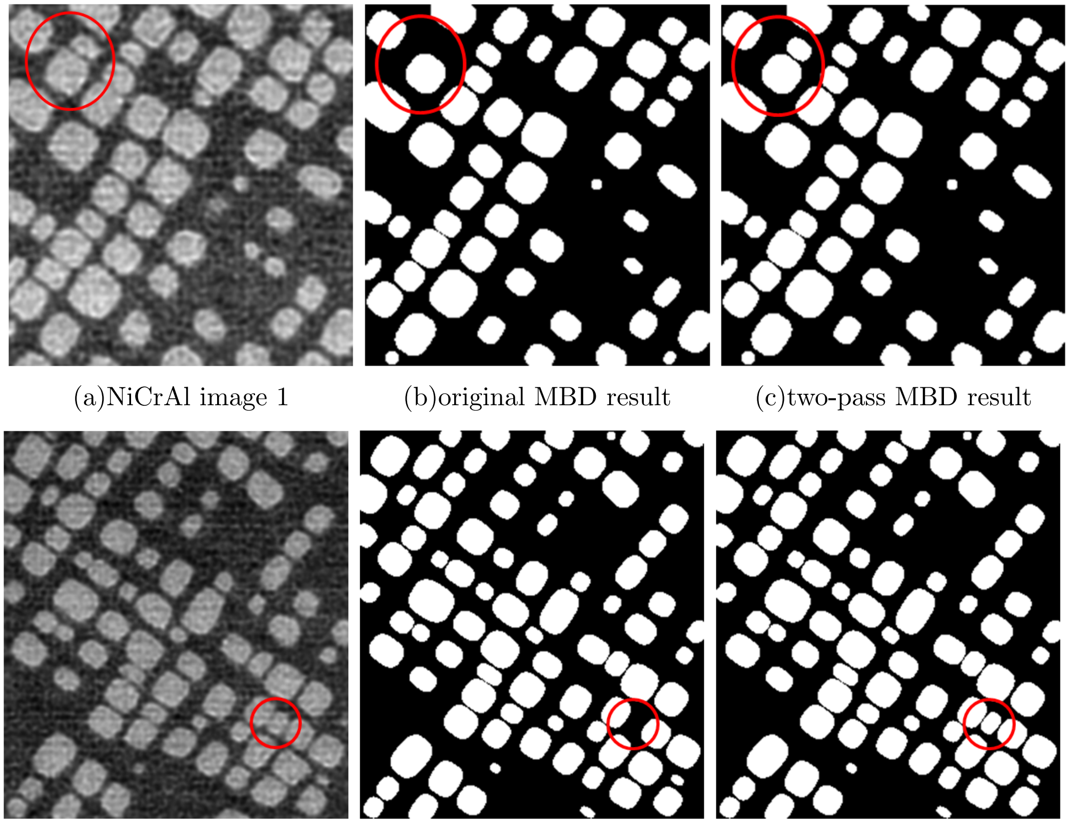
Figure 3.2 shows the process of the original death step. At each step, the object to be processed is called the target object. The dashed block shows the interaction of the target object with its neighbor objects. Based on the decreasing values of the data term, the order we process these 5 objects is green, purple, yellow, blue and pink. The green object is firstly killed because of both high data term and prior term (large overlapping with the purple object and the yellow object). The second target object is the purple one. Because of the overlapping with the yellow object, the high prior term results in a high death rate, even if the data term is relatively low. The purple object is killed accordingly. Later, the yellow and blue objects are also killed. Because the yellow object is killed later, the prior term we calculated for the purple object is not reliable. Here the yellow object is a so-called "phantom" object.

In our two-pass method, we process the objects in w' the same way as in the original death step. Figure 3.3 shows the second pass for processing the objects in w'. It should be noted every object here is a copy of the object in w. The blue object, which is the last one killed in the first pass, is firstly processed in the second pass. Its death rate remains the same as that in the first pass. So it's not given rebirth. The yellow object is not added back according to the low rebirth rate brought by its large overlapping with the pink object. We give rebirth to the purple object according to its high rebirth rate. We add it back to w' and find the yellow object as the "phantom" object for the purple object. As a result, in w we move the yellow object in front of the purple object. The green object is not given rebirth because of the low rebirth rate.

Then we process the objects in w based on the current order, which is green, yellow, purple, blue and pink.

3.3 Experiments

We test our algorithm on 10 NiCrAl alloy microscopy images. For comparison, we apply both the original MBD algorithm and our two-pass method. The results



(a)NiCrAl image 1 (b)original MBD result (c)two-pass MBD result

(d)NiCrAl image 2 (e)original MBD result (f)two-pass MBD result

Fig. 3.4. Results of the original multiple birth and death algorithm and the two-pass multiple birth and death algorithm for NiCrAl images

show that in every image our method performs at least as good as the original MBD algorithm. For images containing particle pairs, where two particles are very close to each other, our method shows better performance and faster convergence. Figure 3.4 shows two comparison results for such case, where both images contain particle pairs (circled in red). The results at iteration 127 are shown for both the original MBD algorithm and our algorithm. In each case, the original algorithm failed to detect the object circled in red.

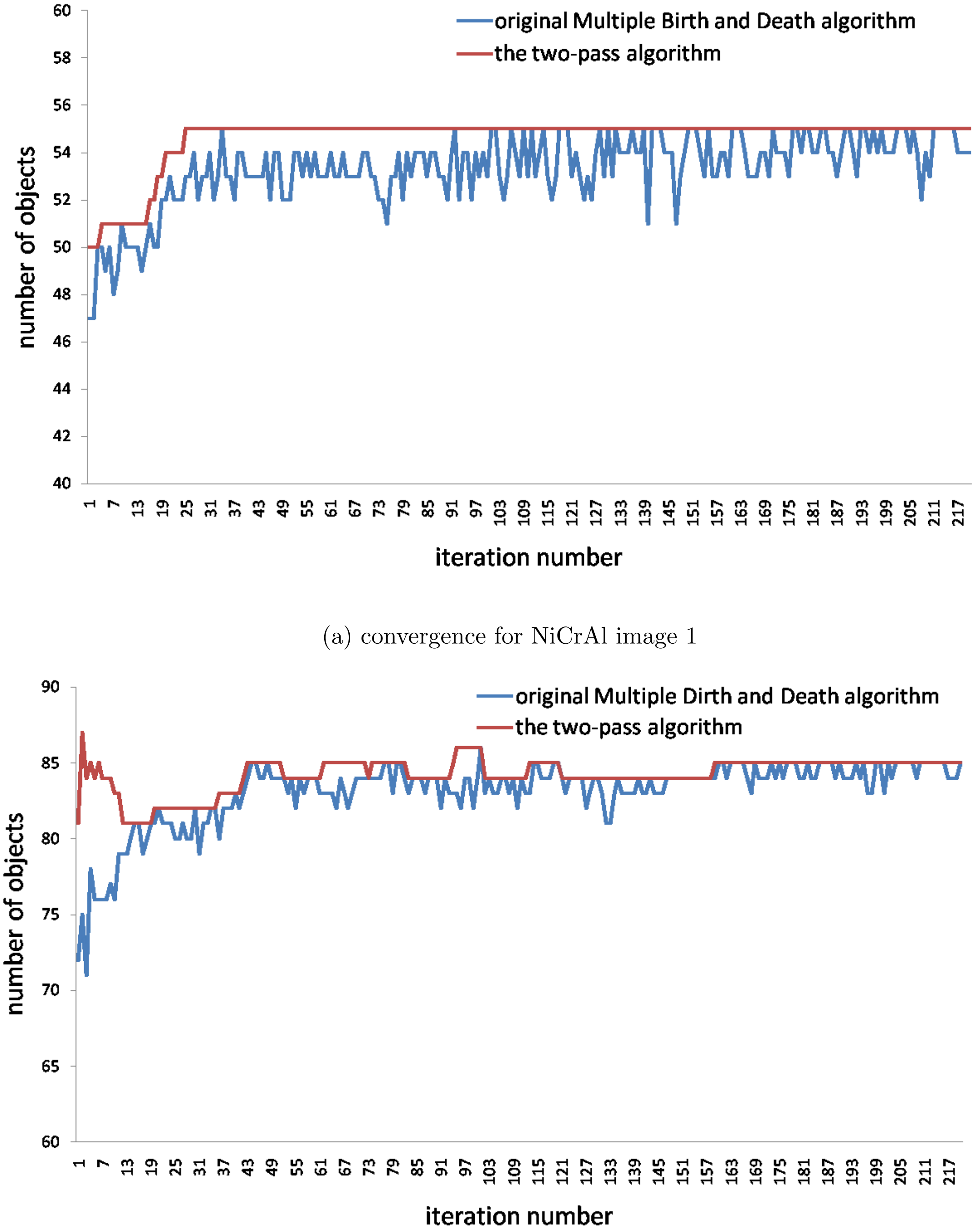
Figure 3.5 shows the convergence of both algorithms for these two examples. For Figure 3.5(a), our method converges to the global minimum (55 objects) at iteration

25, while the original method does not converge even at iteration 219. For Figure 3.5(b), our method converges to the global minimum (85 objects) at the 156th iteration and the original method fails to converge at iteration 219.

3.4 Conclusion

In the original death step of the multiple birth and death algorithm, the "phantom" objects bring negative effect on the death rate calculation on objects prior to them, leading to slow convergence. This problem will get severe in adjacent object pairs, where two objects have close interaction with each other.

In this chapter, we proposed a two-pass multiple birth and death algorithm to solve this problem. Experiments show our proposed method has faster convergence speed and more accurate results compared to the original multiple birth and death algorithm.



(a) convergence for NiCrAl image 1

(b)convergence for NiCrAl image 2

Fig. 3.5. Convergence comparison of the original multiple birth and death algorithm and the two-pass multiple birth and death algorithm for NiCrAl images

4. A HYBRID MARKOV RANDOM FIELD/MARKED POINT PROCESS MODEL FOR MICROSCOPE IMAGES

4.1 Introduction

Stochastic models have long served as powerful tools for image analysis. The Markov Random Field (MRF) has become perhaps the most commonly used model for a wide variety of applications. The MRF represents a Gibbs distribution in terms of localized conditional probabilities satisfying a Markov property. The well-known Ising and Potts MRF models [40,46] have a corresponding interpretation in statistical mechanics. Thus, many physical phenomena are well modeled by an MRF. One problem with the MRF model, however, is that it is difficult to incorporate certain types of global information into the model, since the MRF is defined in terms of local interactions. Such global information not only models high-level features as object shape, size, etc, but also can help to improve the accuracy of the segmentation, especially in noisy or low contrast images.

A multiplelayer MRF [47–51], or larger cliques system [52–54] can be used for higher-level modeling, but these approaches still model the number and locations of objects as deterministic. [55] introduced an object shape priori into MRF, where a layered pictorial structure is used as a guidance for pixel-level segmentation. Such pictorial structures can be trained from a large library of exemplars. This approach mainly focuses on the segmentation of a single object or a small number of objects. However, in some applications in the field of remote sensing, microscopy material and medical imaging, where a large number of objects are randomly located in the image, the approaches mentioned above are not flexible enough to represent the randomness of the number and locations of objects, as well as the interactions between neighboring objects.

An alternative to the MRF for image modeling is the marked point process. Marked point process (MPP) models have been proposed for modeling image objects [5–7]. A *point process* model is very useful for modeling the random locations of objects in an image. A set of random variables (or marks) can then be associated with each point, or object, to describe properties of that object, leading to a *marked point process.* Since the number and locations of objects in images are often random, the MPP is often more appropriate than the traditional MRF for object modeling.

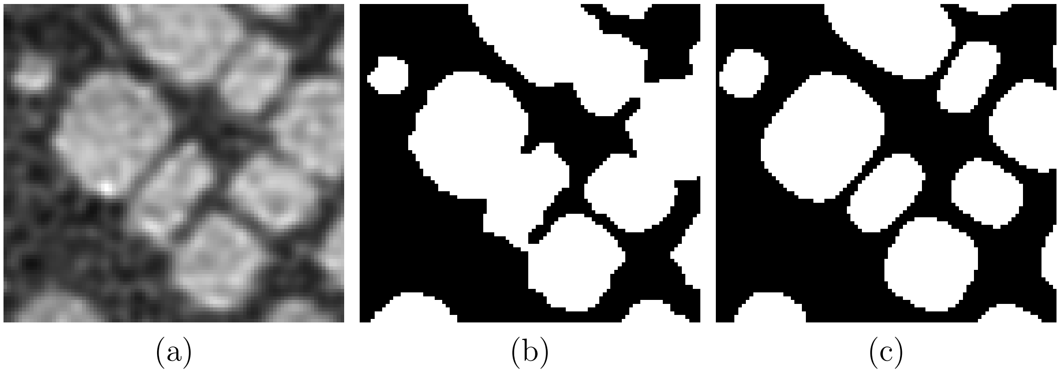


Fig. 4.1. (a) NiCrAl image (b) MRF result (graph cuts) (c) MPP result (superellipse model with multiple birth and death algorithm).

A problem with the MPP is that local constraints, such as pixel-wise interactions, are difficult to model by an MPP. This drawback not only leads to imprecise boundary descriptions, but also can impair the accuracy of object detection, especially when object boundaries do not follow exactly the geometric model. Limitations of both the MRF and the MPP are illustrated in Figure 4.1. Figure 4.1(a) shows a noisy, low-contrast electron microscope image of a NiCrAl alloy. The MRF-based segmentation (Figure 4.1(b)) is poor because of the low contrast between particles. Without global constraints, boundaries between objects merge due to noise and the low contrast. The MPP result (Figure 4.1(c)) is problematic because boundaries of particles are imprecise due to the imperfect fit of the superellipse model.

In many microscopy image analysis applications, it is of critical importance to address both pixel-level segmentation and object-level feature extraction. For example, in microscopy material systems,high-level microstructural features such as particle size and shape are important for characterizing structure-property correlations, but precise boundary localization is also of critical importance, because many properties of materials are determined by the activity at interfaces between regions having different physical or chemical compositions. Pixel-level image information, such as segmentations, are not sufficient alone, nor are object-level representations, to represent these microstructural features [56].

There have been methods proposed to exploit both models together for the purpose of better segmentation and object detection. In [57], a top-down and bottom-up approach was proposed to combine the MRF and the MPP. The bottom-up step corresponds to the MRF based segmentation, while the top-down step represents the MPP based object detection. Each step alternatively takes advantage of the other one. Then two steps are iteratively performed until convergence. In [58], the alternation between these two steps are replaced by a fusion/decision step that merges the results from both steps into one classification. However, these interactive schemes do not incorporate two models into one framework, nor do they jointly consider the segmentation and the object fitting.

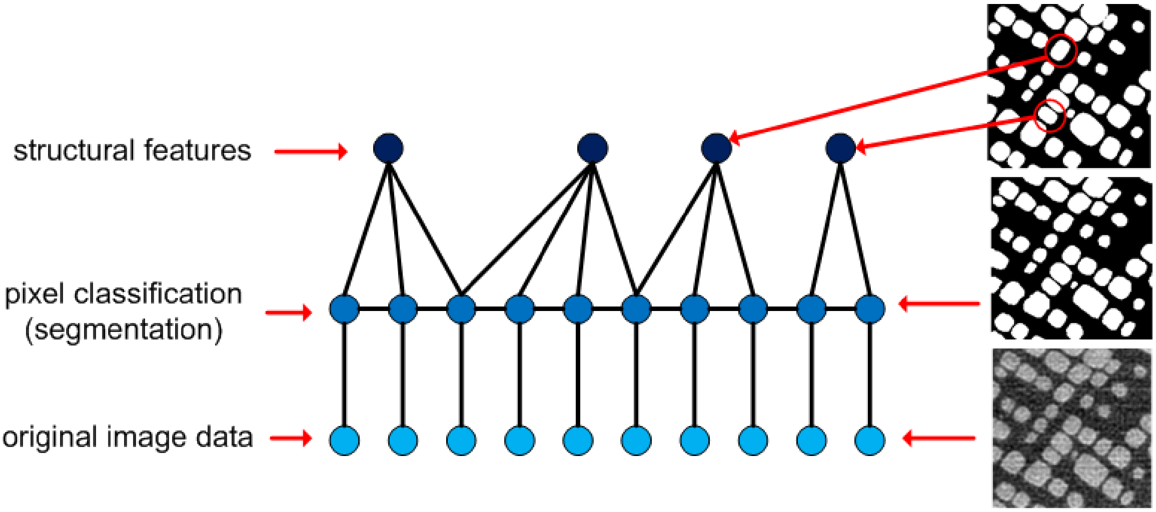


Fig. 4.2. A three-layer representation of image data.

We propose such a model to perform both pixel classification (segmentation) and object detection/identification. A three-layer representation is proposed, as shown in Figure 4.2. The bottom layer is the measured image data, where each node represents a pixel. The middle layer is the segmentation, which classifies each pixel of the image to one of a set of possible classes. The top layer describes the structural features, where each node represents a marked object, such as a grain or a particle in microscopy images of materials. Unlike traditional schemes which try to impose local constraints at the middle layer and global constraints at the top layer separately, our approach incorporates both constraints into one hybrid framework. In this way, we utilize both the geometric properties of objects (global constraints) and pixel-wise interactions (local constraints) to optimize these two layers simultaneously.

We perform optimization with our model using Monte Carlo simulation. Specifically, we employ two different approaches to optimization for object detection and segmentation: multiple birth and death simulation and reversible jump Markov chain Monte Carlo simulation [1, 3, 4]. Simulation is an important tool for optimizing our model. This is because the density function of a MPP cannot be defined with respect to the Lebesgue measure, since the process is represented as an unordered set of points. Instead, we follow [7], and define our density function with respect to the Lebesgue-Poisson measure. Because the density function for our model is not defined with respect to the Lebesgue measure, traditional optimization techniques that rely on gradients, such as gradient descent type approaches, cannot be used.

One note on terminology: Technically speaking, since our MPP model is represented by a Gibbs distribution, it can also be considered to be a Markov random field, according to the Hammersley-Clifford Theorem. However, we distinguish the two models in this paper based on whether the spatial locations for the process—pixel lattice sites for the MRF, unordered points in $\mathbb{R}^n$ for the MPP—are modeled as random variables (as in the MPP) or not (as in the MRF). We believe this distinction is consistent with the literature.

In Section 4.2, we describe our hybrid model and its use in joint object detection and segmentation. Experimental results are shown in Section 4.3. We conclude in Section 4.4.

4.2 A hybrid MRF/MPP model

Let $Y = (Y_1, ..., Y_N)$ denote the observed image and $X = (X_1, ..., X_N)$ the segmented image, both defined on a 2D lattice S, where N is the number of pixels and the lattice sites are ordered from 1 to N. Both X and Y are random fields. Particular sample realizations of Y and X will be denoted $y = (y_1, ..., y_{_N})$ and $x = (x_1, ..., x_{_N})$. A configuration of n objects in an image is a finite unordered set of points $\{S_1, S_2, ..., S_n\} \subset S$ with each S_i representing the random location of one object. For each object S_i, there is a mark M_i. The mark for an object contains random variables describing the object. Note that we will use the term "mark" to refer to both the collection of random variables M_i and each random variable in M_i itself. A marked object is defined as $W_i = \{S_i, M_i\} \in W$, where $W \subset S \times M$ is the random marked object field. Taking the rectangle model as an example, let M be the mark space, $M = [a_{min}, a_{max}] \times [b_{min}, b_{max}] \times [0, \pi]$, where the mark consists of two axis lengths and one orientation. Let Ω_W denote the space of all possible realizations of W. Then $w = (w_1, ..., w_n) \in \Omega_W$ is a possible object configuration.

For each object w_i, we define an auxiliary object w_i^γ, an enlarged version of w_i which covers a slightly larger image region, so that pixels both inside and outside the object boundary are included. For example, for a rectangle object w_i with mark $m = (a, b, \theta)$, the corresponding w_i^γ might have mark $m^\gamma = (a + 2, b + 2, \theta)$. Let $D_{w_i^\gamma} \subset S$ be the image region projected onto the lattice S by w_i^γ. Then the segmentation of object w_i is defined as x_{w_i}, which is the segmentation x restricted to the region $D_{w_i^\gamma}$.

Each object of interest is represented in our hybrid MRF/MPP model by a marked object and its corresponding segmentation. A hybrid model for a NiCrAl particle is illustrated in Figure 4.3.

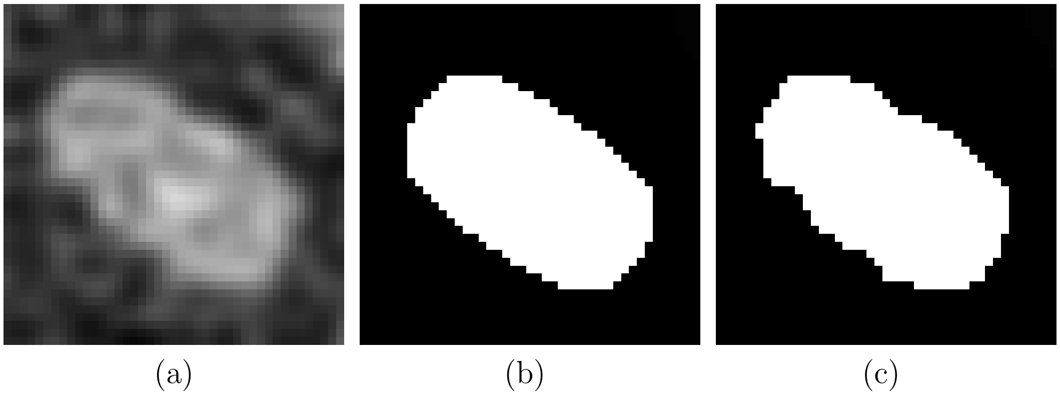


Fig. 4.3. An object of interest is represented by two parts: an object part and a segmentation part. (a) a NiCrAl particle (b) object part w_i in white (c) segmentation part x_{w_i} in white.

We introduce a non-homogeneous Gibbs process on the configuration space Ω_W, with conditional probability density function:

$$f(w|y) = \frac{1}{Z} exp \{-V(w|y)\} \tag{4.1}$$

where Z is a normalizing constant. Note that since the object locations form a nonordered set of points, the density function $f(w|y)$ cannot be defined with respect to the Lebesgue measure. We assume the Poisson-Lebesgue measure instead, as in [7]. The neighborhood system with regard to this process is defined as: w_i and w_j are neighbors, denoted $w_i \sim w_j$, if $D_{w_i} \cap D_{w_j}$ is not empty. The energy function for the Gibbs distribution is

$$V(w|y) = \sum_{w_i \in W} (V_o(w_i, y) + V_s(x_{w_i}^{MAP})) + \sum_{w_i \sim w_j} V_p(w_i, w_j) \tag{4.2}$$

where $V_o(w_i, y)$ is defined as the object potential, an energy potential describing how well object w_i fits the observed image; $V_s(x_{w_i}^{MAP})$ is defined as the segmentation potential, an energy potential based on the likelihood of the MAP segmentation corresponding to object w_i; and $V_p(w_i, w_j)$ is an object interaction prior.

We next specify potentials $V_o(w_i, y)$, $V_s(x_{w_i}^{MAP})$ and $V_p(w_i, w_j)$ for our model.

4.3 Object potential for observed image

The object potential $V_o(w_i, y)$ describes how a given object w_i fits the image data. The potential is based on a shape model to impose geometric constraints. Often a statistical difference in intensity values between the outside and the inside of an object is used to define the object potential. The object potential will satisfy the following two criteria [59]:

- An object which fits the image data better has lower potential.
- The best-fitting objects have negative potentials.

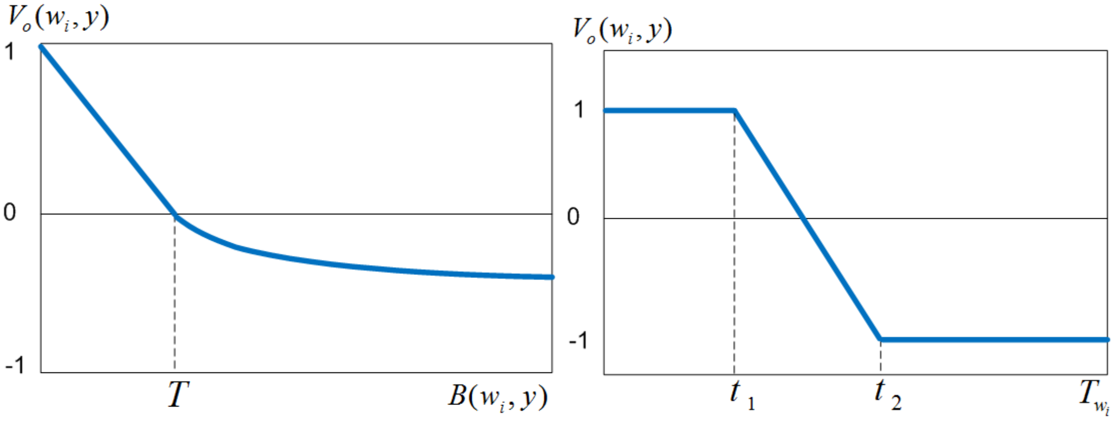
We use two different MPP models in this chapter. The first, a superellipse model, proposed in [60], is given by the equation:

$$g(x, y) = \left|\frac{x}{a}\right|^n + \left|\frac{y}{b}\right|^n = 1 \tag{4.3}$$

The superellipse model has been shown to fit NiCrAl particles well [60]. The model has marks that include major and minor axis lengths a and b, respectively, orientation θ and parameter n, which controls the shape of the superellipse model. We assume that pixel values outside of the particle boundary are statistically different from those within the boundary. For each w_i with mark $m_i = (a, b, \theta, n)$, we associate two auxiliary objects w_i^α with mark $m_i^\alpha = (a + 1, b + 1, \theta, n)$ and w_i^β with mark $m_i^\beta = (a - 1, b - 1, \theta, n)$, at the same location as w_i [1]. Then the inside region and outside region of w_i are defined as $A_{in}(w_i) = D_{w_i^\alpha} - D_{w_i}$ and $A_{out}(w_i) = D_{w_i} - D_{w_i^\beta}$ respectively (Figure 4.5(a)).

Pixel values in $A_{in}(w_i)$ and $A_{out}(w_i)$ are assumed to follow Gaussian distributions. This corresponds to a Gaussian noise observation model. As in [39], we use the Bhattacharyya distance to distinguish objects from background regions. The Bhattacharyya distance $B(w_i, y)$ between $A_{in}(w_i)$ and $A_{out}(w_i)$ is given by:

$$B(w_i, y) = \frac{(\mu_{in} - \mu_{out})^2}{4\sqrt{\sigma_{in}^2 + \sigma_{out}^2}} - \frac{1}{2}\log\frac{2\sigma_{in}\sigma_{out}}{\sigma_{in}^2 + \sigma_{out}^2} \tag{4.4}$$



(a)

(b)

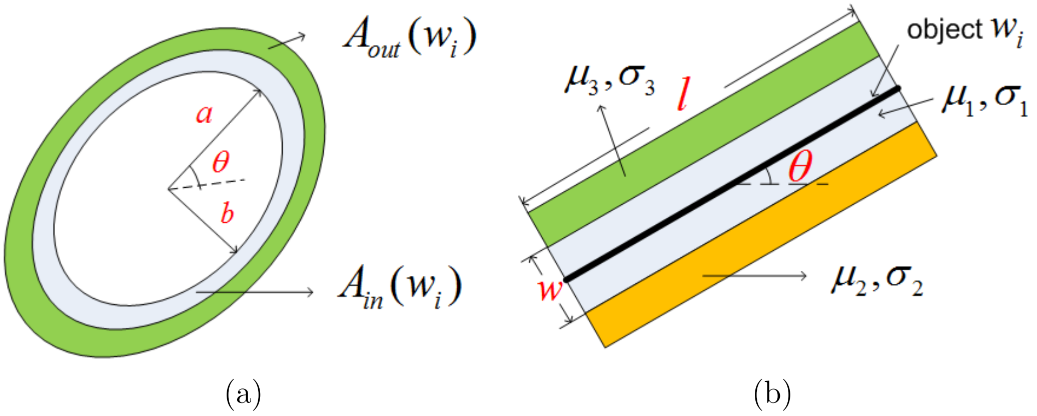
Fig. 4.4. (a) plot of the object potential $V_o(w_i, y)$ versus the Bhattacharyya distance $B(w_i, y)$ (b) plot of the object potential $V_o(w_i, y)$ versus the Student's t-test T_{w_i}.

where (μ_{in}, σ_{in}) and $(\mu_{out}, \sigma_{out})$ are Gaussian parameters for $A_{in}(w_i)$ and $A_{out}(w_i)$ [39]. Then the object potential $V_o(w_i, y)$ is defined as a likelihood function, which is calculated from $B(w_i, y)$, as in [1]:

$$V_o(w_i, y) = \begin{cases} 1 - \frac{B(w_i,y)}{T} & B(w_i, y) < T \\ \exp(-\frac{B(w_i,y)-T}{3B(w_i,y)}) - 1 & B(w_i, y) \geq T \end{cases} \tag{4.5}$$

Figure 4.4(a) shows a plot of the object potential as a function of the Bhattacharyya distance between the inside and outside region of w_i.

The second object model we will use is a line segment model. We propose a Candy model, modified from that used in [2] to model line networks. A rectangle model is used to represent each line segment, with mark $m = (l, \omega, \theta)$, corresponding



(a)

(b)

Fig. 4.5. Inside and outside regions for object w_i for (a) superellipse model and (b) line segment model.

to the length, width and orientation. Similar to the superellipse model, the inside and outside regions are assumed to follow Gaussian distributions with mean and variance (μ_1, σ_1), (μ_2, σ_2) and (μ_3, σ_3) respectively (Figure 4.5(b)). In [2], the mean difference T_{w_i} is calculated as the minimum of the Student's t-test values between the inside and outside regions on either side:

$$T_{w_i} = min(T^1_{w_i}(l, \omega, \theta), T^2_{w_i}(l, \omega, \theta)) \tag{4.6}$$

where

$$T^k_{w_i}(l, \omega, \theta) = \frac{|\mu_1 - \mu_k|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_k^2}{n_k}}} \quad k \in \{2, 3\} \tag{4.7}$$

If the two outside regions are homogeneous, we can assume they follow the same Gaussian model, so $(\mu_2, \sigma_2) = (\mu_3, \sigma_3)$. The object potential $V_o(w_i, y)$ is a function of T_{w_i}, as in [2]:

$$V_o(w_i, y) = \begin{cases} 1 & if\ T_{w_i} < t_1 \\ 1 - 2\frac{T_{w_i} - t_1}{t_2 - t_1} & if\ t_2 \leqslant T_{w_i} \leqslant t_1 \\ -1 & if\ t_2 < T_{w_i} \end{cases} \tag{4.8}$$

The object potential $V_o(w_i, y)$ for the Candy model is plotted in Figure 4.4(b).

In the original Candy model, three types of line segments are defined: free segment, single segment and double segment [2]. As in Figure 4.6(a-c), the free segment has two free ending points. The single segment has only one ending point connected, and the double segment has two ending points connected. Two ending points are said to be connected if they are within a distance less than ϵ of each other. The introduction of ϵ effectively helps to establish the interactions between nearby line segments, not only the line segments that are totally connected. However, one problem is that the results usually have many discontinuities between lines. Although [2] proposed a "Quality Candy" model to alleviate this situation, the problem is not completely solved. An alternative approach is to restrict ϵ to one pixel, where ending points within a small distance (larger than 1) would no long be considered connected. However, we consider this too harsh a constraint, because line segments in Figure 4.6(b-c) would be considered free segments, so the single or double segment properties would not be captured for them.

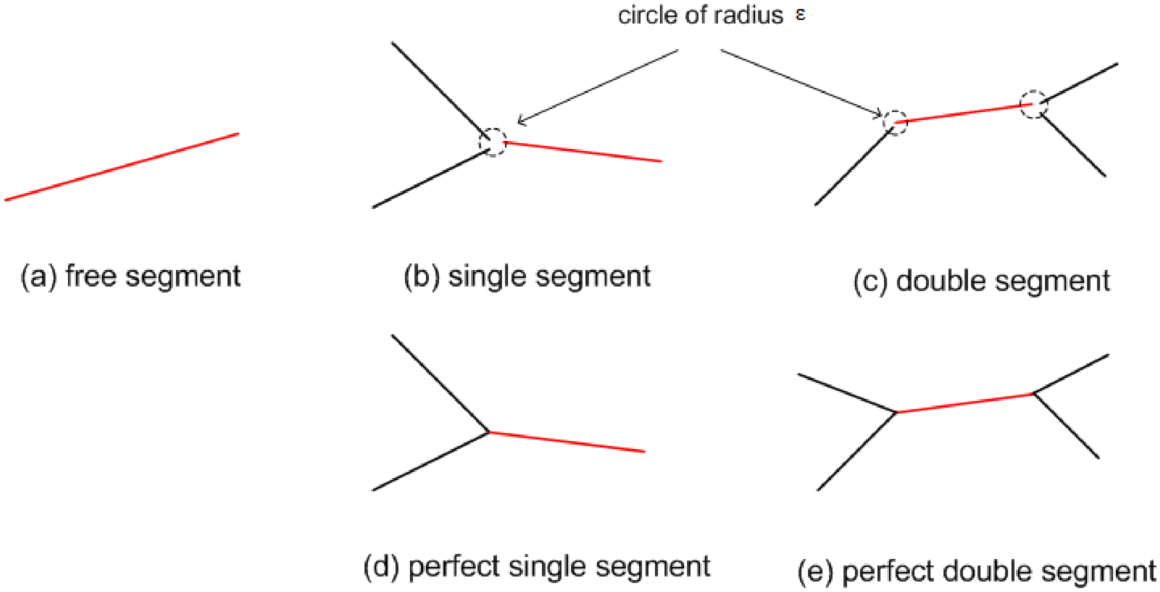


Fig. 4.6. Five types of line segments.

In this chapter, we introduce too more line segments: perfect single segment and perfect double segment, as in Figure 4.6(d,e). A segment is defined as a perfect single segment if one of its ending points is at the same location as the ending points of other segments. As well, a perfect double segment has two ending points at the same location as other segments' ending points. We encourage single and double segments to transform to perfect single and double segments, as described in the object interaction prior. In this way, we preserve the advantage of capturing interactions within a distance ϵ as well as solving the problem of discontinuities in lines.

It should be noted that both the Bhattacharyya distance and Student's t-test are statistical approaches to measuring the difference between two data sets. Following [39] and [2], we opt to use the Bhattacharyya distance for the superellipse model and Student's t-test for the Candy model. However, we believe both methods would be adequate for our model. For example, [61] used the Bhattacharyya distance for the Candy model. The advantages of each approach for our model, together with other statistical difference test approaches, could be explored in future research.

4.4 Segmentation potential for observed image

The object potential imposes geometric constraints, but often the shape of an object does not exactly fit the model. The object model is not sufficient to localize complicated boundary information precisely. As a result, the accuracy of the object detection and the precision of the segmentation can both decline. Therefore, a segmentation potential is sought as a local constraint to better fit the object. The segmentation potential is the part of the model that is based on an MRF.

For an object w_i, the segmentation potential, denoted $V_s(x^{MAP}_{w_i,\lambda})$, incorporates local constraints using an MRF. In particular, $V_s(x^{MAP}_{w_i,\lambda})$ describes how well the optimal maximum a posteriori (MAP) segmentation that best fits the object w_i also fits the image data. $x^{MAP}_{w_i,\lambda}$ is the MAP segmentation for object w_i, where we make the assumption that the segmentation associated with a particular w_i with mark m_i

depends only on the image data covered by w_i^γ with mark m_i^γ, where w_i and w_i^γ are centered at the same location. Thus,

$$x_{w_{i,\lambda}}^{MAP} = \arg\max_{x_{w_{i,\lambda}}} p(x_{w_{i,\lambda}}|\hat{y}(w_i)) \tag{4.9}$$

where $\widehat{y}(w_i) \triangleq \left\{ y_s | s \in D_{w_i^\gamma} \right\}$ and $\lambda = (\lambda_1, \lambda_2)$ is a weighting parameter. $p(x_{w_{i,\lambda}}|\widehat{y}(w_i))$ is the posterior probability of segmentation $x_{w_{i,\lambda}}$ given image data $\widehat{y}(w_i)$. It is equivalent to obtain $x_{w_{i,\lambda}}^{MAP}$ by minimizing an energy function $G(x_{w_{i,\lambda}})$ over all possible segmentations of object w_i with parameter λ, where $G(x_{w_{i,\lambda}}) \triangleq -\ln(p(\widehat{y}(w_i)|x_{w_{i,\lambda}})) - \ln(p(x_{w_{i,\lambda}}))$. Then the segmentation potential is defined as:

$$V_s(x_{w_{i,\lambda}}^{MAP}) \triangleq \frac{G(x_{w_{i,\lambda}}^{MAP}) - G(x_{w_{i,0}}^{MAP})}{|D_{w_i}|} \tag{4.10}$$

where we use the notation:

$$\begin{aligned} G(x_{w_{i,\lambda}}) &= G_{w_i}^1(x_{w_i}) + G_{w_i}^2(x_{w_{i,\lambda_1,\lambda_2}}) \\ G_{w_i}^1(x_{w_i}) &= -\ln(p(\widehat{y}(w_i)|x_{w_i}) \\ G_{w_i}^2(x_{w_{i,\lambda_1,\lambda_2}}) &= -\ln(p(x_{w_{i,\lambda_1,\lambda_2}})) \end{aligned} \tag{4.11}$$

The value of $G_{w_i}^1$ at pixel s, is defined $G_{w_i}^1(x_s)$, using a Gaussian model:

$$G_{w_i}^1(x_s) = \sqrt{2\pi}\sigma_{x_s} + \frac{(y_s - \mu_{x_s})^2}{2\sigma_{x_s}^2} \tag{4.12}$$

Note that we use the notation $G_{w_i}^1$ as a cost for both an object and a single pixel classification. The meaning should be clear from context. The parameters μ_{x_s},$\sigma_{x_s}^2$ are the mean and variance of the observed Gaussian intensity for class x_s, which we estimate using a preprocessing method such as K-means.

For a given w_i, we assume pixels in D_{w_i} are more likely to belong to the object while pixels in $D_{w_i^\gamma} - D_{w_i}$ are more likely to belong to the background. As a consequence,

$G^2_{w_i}(x_{w_i,\lambda_1,\lambda_2})$ is defined as a combination of a shape prior and a nonhomogeneous Ising model, letting

$$G^2_{w_i}(x_{w_i,\lambda_1,\lambda_2}) = \lambda_1 \delta_{w_i}(s, x_s) + \sum_{p,q \in D_{w_i^\gamma}} \beta_{p,q}[1 - \delta(x_p, x_q)] \tag{4.13}$$

where

$$\delta_{w_i}(s, x_s) = \begin{cases} -1 & if\ s \in D_{w_i}\ and\ x_s \in foreground \\ -1 & if\ s \in D_{w_i^\gamma} - D_{w_i}\ and\ x_s \in background \\ 1 & otherwise \end{cases} \tag{4.14}$$

$$\beta_{p,q} = \begin{cases} \beta & if\ p, q \in D_{w_i}\ or\ p, q \in D_{w_i^\gamma} - D_{w_i} \\ \beta - \lambda_2 & otherwise \end{cases} \tag{4.15}$$

and

$$\delta(x_p, x_q) = \begin{cases} 1 & if\ x_p = x_q \\ 0 & otherwise \end{cases} \tag{4.16}$$

The term $\delta_{w_i}(s, x_s)$ encourages the segmentation to fit the object w_i and λ_1 is a positive parameter. When $s \in D_{w_i}$ and $x_s \in foreground$ or $s \in D^\gamma_{w_i} - D_{w_i}$ and $x_s \in background$, the pixel label x_s complies with the assumed foreground or background status based on the object w_i, and we subtract an additional positive energy term λ_1 from $G_{w_i}(x_{w_i,\lambda})$. Otherwise, a penalty energy term λ_1 is added.

$\beta_{p,q}$ is a location-adaptive interaction parameter, with $0 \leqslant \lambda_2 \leqslant \beta$. In this way, we encourage smoothness in the segmentation within foreground and background regions while discouraging smoothing across region boundaries (Figure 4.7).

We minimize $G(x_{w_i,\lambda})$ using graph cuts [62]. The paramters λ_1, λ_2 control the degree of global constraints on the segmentation, while $x_{w_i,0}$ is the MRF-based segmentation with no global constraints applied.

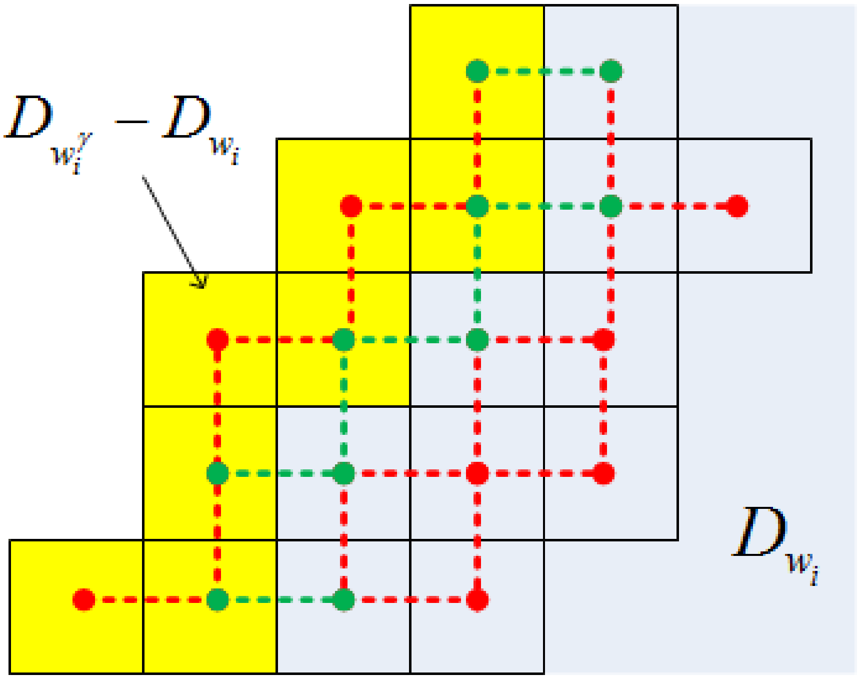


Fig. 4.7. Illustration of $G^2_{w_i}(x_{w_i})$. $D_{w_i^\gamma} - D_{w_i}$ is shown in yellow while D_{w_i} is shown in light blue. In the four neighborhood system, red dashed links between pixel sites correspond to $\beta_{p,q} = \beta$ and green dashed links correspond to $\beta_{p,q} = \beta - \lambda_2$.

$G(x_{w_{i,\lambda}})$ measures how well a segmentation constrained by w_i fits the image data. Particularly, for a pixel s, when the pixel label complies with the assumed background/foreground, the energy difference between $G(x_{w_{i,\lambda}})$ and $G(x_{w_{i,0}})$ will decrease between 0 and λ_1. On the contrary, when the pixel label does not comply with the assumed background/foreground, the energy difference will increase between 0 and λ_1. Both cases are illustrated within the graph cuts framework (Figure 4.8,4.9). For a given w_i, the minimum of $G(x_{w_{i,\lambda}})$ also corresponds to the best choice of mark for the specific object.

The term $V_s(x^{MAP}_{w_{i,\lambda}})$ can be viewed as the energy of the most likely segmentation constrained by geometric properties relative to that of the most likely segmentation with no geometric constraints, normalized by the size of the object. It should be noted that both object potential and segmentation potential try to choose a mark

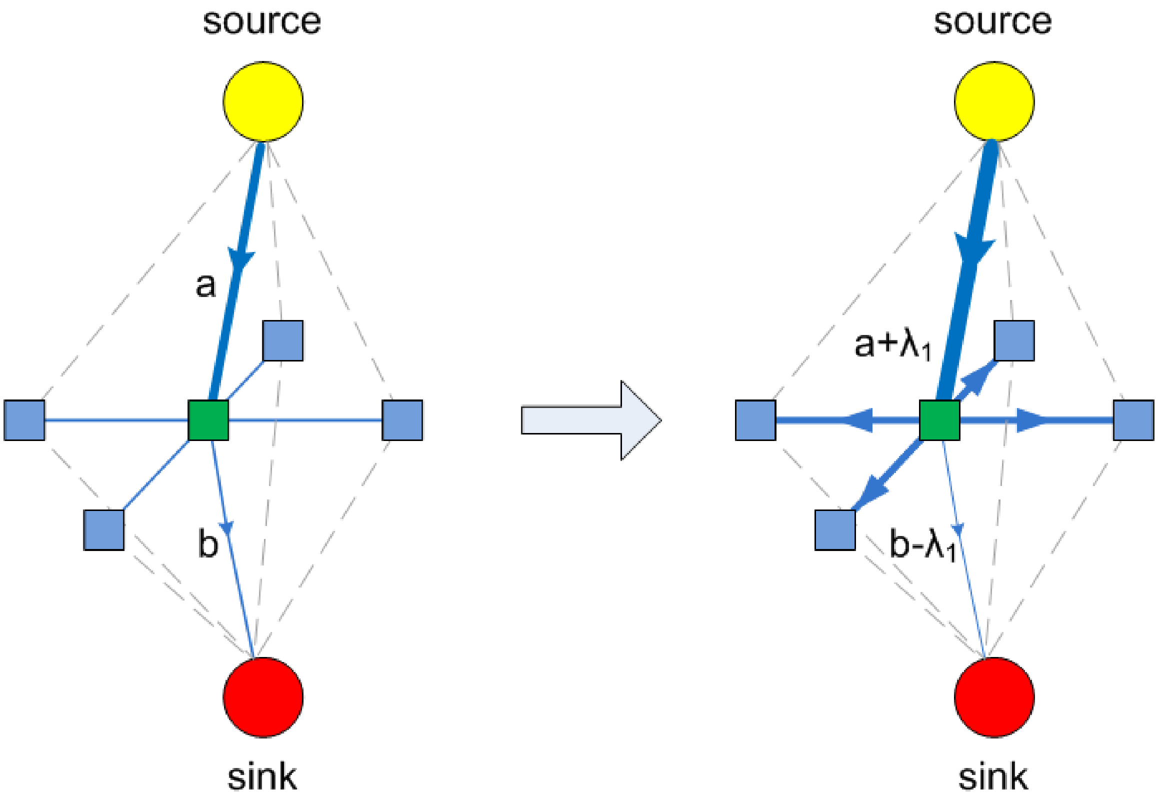


Fig. 4.8. The left figure shows a graph for the green pixel. Its neighbour pixels are colored in blue. Source and sink nodes represent the background and foreground, respectively. a, b are the probabilities of the pixel belonging to background or foreground. When the pixel's label is background, the link between the pixel and the sink gets cut. If we also assume the green pixel's label is background (right figure), we add λ_1 into a while subtracting λ_1 from b. When we optimize this graph, the link between the pixel and the sink will still get cut, with an energy loss of λ_1 for that link. While the added λ_1 in the link between the pixel and the source can flow to other pixels via neighborhood links, the total energy will decrease between $[0, \lambda_1]$.

for a specific object, based on the MPP model and the MRF model, respectively. For example, when we use the superellipse model for a NiCrAl particle with mark $m = (a, b, \theta)$, the different marks chosen by the two potentials are illustrated in Figure 4.10.

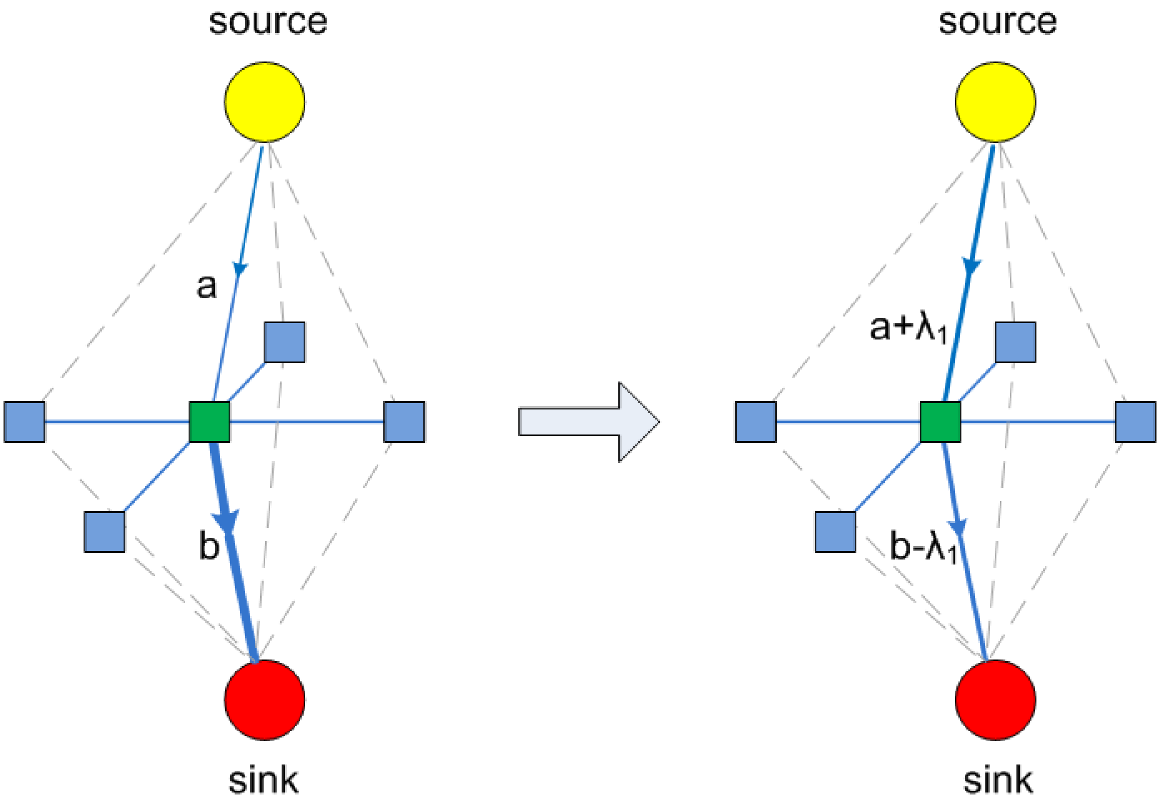


Fig. 4.9. When the pixel's label is foreground, the link between the pixel and the source gets cut. If we assume the green pixel's label is background (right figure), we add λ_1 into a while subtracting λ_1 from b. If λ_1 is chosen relatively small, after the optimization, the link between the pixel and the source will still get cut, with an energy increase of λ_1 for that link. While we subtract λ_1 in the link between the pixel and the sink, the energy flow coming from other pixels via neighborhood links can decrease between $[0, \lambda_1]$. The total energy will increase between $[0, \lambda_1]$.

4.5 Object interaction prior

For an arbitrary object pair w_i and w_j, the object interaction prior is denoted $V_p(w_i, w_j)$, which describes the interactions between these two objects. The object interaction prior characterizes interactions between objects, by penalizing or favoring particular object behaviors through potentials in the energy function. A prior that is widely used in many applications is the overlapping penalizer [1, 23, 37], which

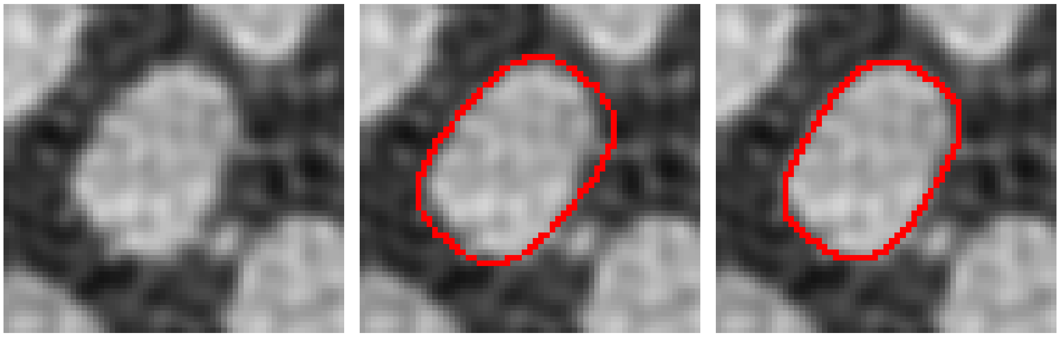


Fig. 4.10. (a) a NiCrAl particle (b) mark minimizing the object potential, $m = (21, 11, 0.21\pi)$ (c) mark minimizing the segmentation potential with $\lambda = (2, 0.4)$, $m = (20, 10, 0.25\pi)$.

discourages spatial overlap for two adjacent objects. As in [37], the overlapping penalizer $R(w_i, w_j)$ is defined as the normalized intersection area:

$$R(w_i, w_j) = \frac{|D_{w_i} \cap D_{w_j}|}{\min(|D_{w_i}|, |D_{w_j}|)} \tag{4.17}$$

Then the object interaction prior is defined as:

$$V_p(w_i, w_j) = \begin{cases} R(w_i, w_j) & if\ R(w_i, w_j)\ < T_{overlap} \\ \infty & otherwise \end{cases} \tag{4.18}$$

Here zero probability is assigned for any overlapping ratio greater than threshold $T_{overlap}$.

More sophisticated object interaction priors can be designed to model more complicated interaction patterns between objects. For example, in [24] an alignment term to favor parallel objects in certain directions in vehicle detection applications was proposed. The Candy model includes orientation terms to model patterns for crossing and connecting lines, which is used to extract line networks in remote sensing [2].

We define the object interaction prior for our modified Candy model as follows:

$$\begin{aligned} V_p(w_i, w_j) = \eta_0 n + \eta_1 n_s + \eta_2 n_d + \eta_3 n_{ps} + \eta_4 n_{pd} \\ + \eta_5 \sum g_{io}(w_i, w_j) + \eta_6 \sum g_{eo}(w_i, w_j) \end{aligned} \tag{4.19}$$

where n is the number of total line segments. n_s, n_d, n_{ps}, n_{pd} are the number of single, double, perfect single and perfect double line segments respectively; η_i are constant weights; g_{io} and g_{eo} are potential functions for internal and external "bad" orientation object pairs respectively, whose definition can be found in [2]. Rules for setting η_i are described in [2]. In order to encourage single and double segments to transform to perfect single and perfect double segments, we set $\eta_3 < \eta_1$, $\eta_4 < \eta_2$.

4.6 Parameter setting

The parameter λ, which affects the value of the segmentation potential, is introduced in our hybrid model. This parameter controls to what extent the geometric constraint is imposed on the segmentation for each object. When $\lambda = (0, 0)$, the segmentation potential is zero and the hybrid model regresses to the original MPP model. Since there is no geometric shape constraint for $\lambda = (0, 0)$, the corresponding segmentation is the same as the MRF-based MAP segmentation. As we increase λ_1, the geometric constraint has more impact on the segmentation. The segmentations for two adjacent NiCrAl particles with different values of λ_1 are illustrated in Figure 4.11.

λ_1 plays an important role in balancing the effect of geometric constraints on the segmentation. The parameter λ_2 effectively controls the amount of smoothing across object boundaries. Since we have not developed an algorithm for optimizing this parameter, we empirically set $\lambda_1 = \frac{1}{5} min\{\delta_{x_1}, \delta_{x_2}\}$ and λ_2 as $\frac{1}{2}\beta$ to preserve the boundary accuracy. The setting has shown good performance in experimental results.

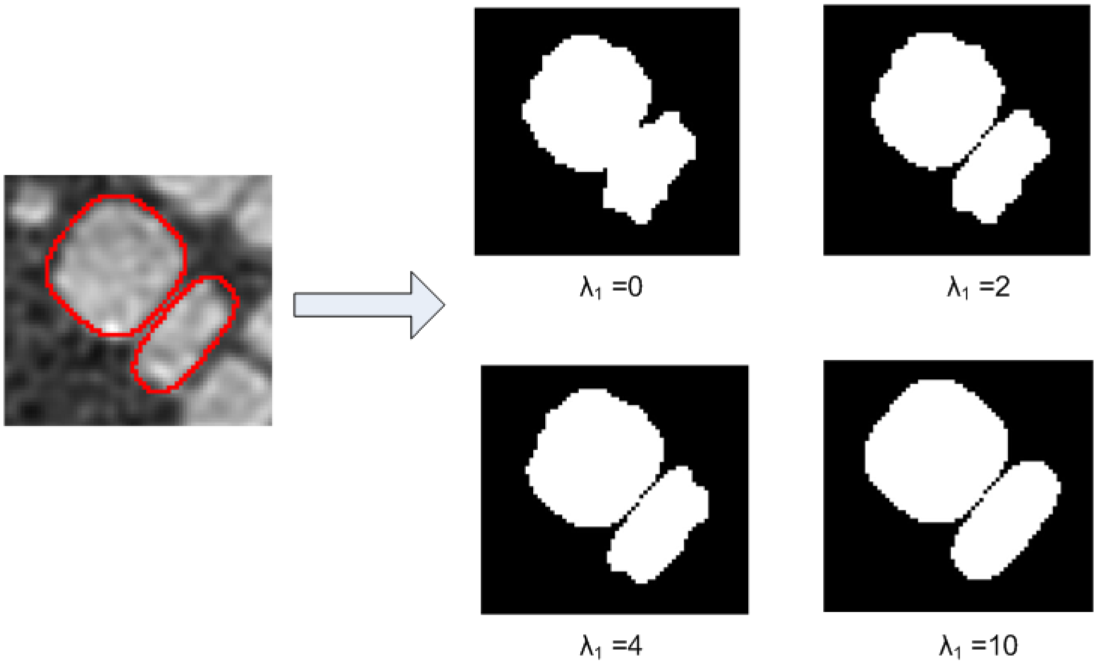


Fig. 4.11. Segmentations with different choices of λ_1. Note that when λ_1 is large enough ($\lambda_1 = 10$ in this case), the segmentation is overwhelmed by the geometric constraint.

4.7 Optimization method

Our optimization goal is to find an object configuration that minimizes the energy function $V(w|y)$ and that has the minimum number of objects among all such configurations. For each object w_i in the object configuration w, the corresponding MRF-based MAP segmentation $x^{MAP}_{w_i,\lambda}$ is obtained in order to compute the segmentation potential $V_s(x^{MAP}_{w_i,\lambda})$, by minimizing $G(x_{w_i,\lambda})$. As in the original MPP model, in general there is no closed-form solution to obtain such an optimized configuration w. Both the RJ MCMC and multiple birth and death algorithms can be used with our hybrid model.

4.8 Experiments

The proposed algorithm has been tested on three image data sets: NiCrAl particles, silicate and carbon foam images. We perform both object extraction and segmentation. We also apply MPP-only object detection and the graph cuts algorithm for comparison. Software from [62] is used for graph cuts.

4.8.1 NiCrAl images

Figure 4.12(a) presents a NiCrAl alloy microscopy image with dimension 744×645. The main goal for this application is to identify each NiCrAl particle and obtain its corresponding segmentation. The superellipse model has been shown to model well the geometric shape of NiCrAl particles [38]. Since the value $n = 3$ best fits the shape of NiCrAl particles [38], we use that value for n.

The axis lengths a,b are sampled uniformly between the minimum and maximum values. Theoretically, in the NiCrAl alloy, there are only two possible orientations [38]. However, in reality there is usually some small fluctuation in the orientation for each NiCrAl particle. As a result, we assume θ is distributed following a mixture of two Gaussians with mean values $\frac{1}{4}\pi$ and $\frac{3}{4}\pi$. The variance is set as 0.1^2 for both Gaussians.

We use the overlapping penalizer in Section 4.2 as a prior to describe the spatial interaction between objects. The object potential is defined as in Section 4.2. We use the two-pass multiple birth and death (MBD) algorithm for optimization. For the MBD algorithm, we set $\lambda = (2, 0.4)$ under the criterion in Section 4.2, and the threshold T is set as 25 following [1]. We follow the guidelines in [1] to set the parameters for multiple birth and death algorithm as $b_0 = 0.8, \alpha_0 = 10, \sigma_0 = 0.9$ and the cooling factor as $1/0.98$. The results are shown in Figure 4.12.

We use ten different images with dimension 744×645 to verify the performance of our algorithm. Note that since we do not have enough ground truth images to effectively train our model, we use parameter values that were determined experimentally for our tests. In the object part, we use the missed detection rate (M.D.R.) and false

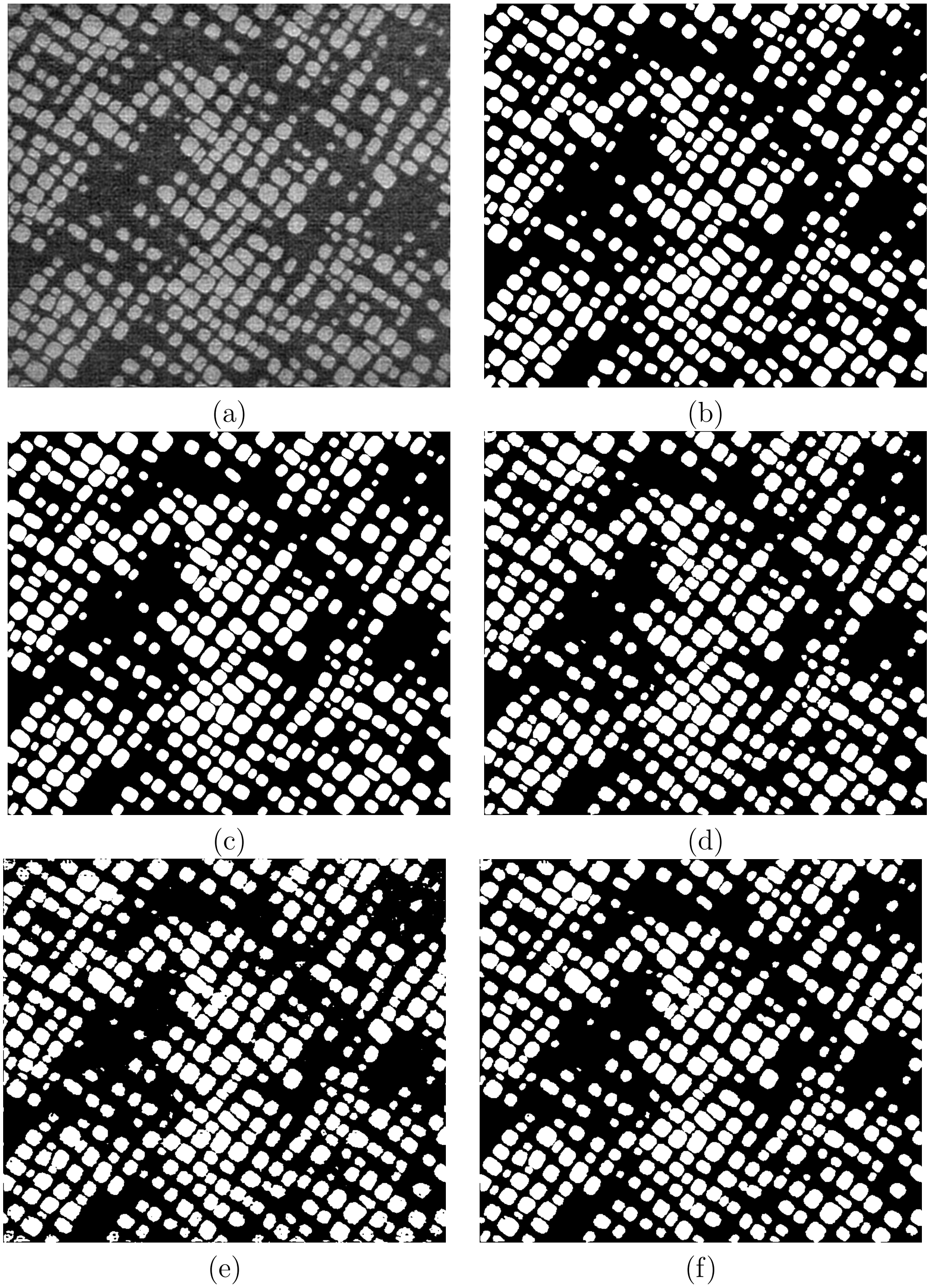


Fig. 4.12. (a) original NiCrAl alloy image (b) MPP only result with the MBD algorithm (c) proposed model (object part) with the MBD algorithm (d) hand drawn segmentation (e)graph cuts result (f) proposed model (segmentation part) with the MBD algorithm.

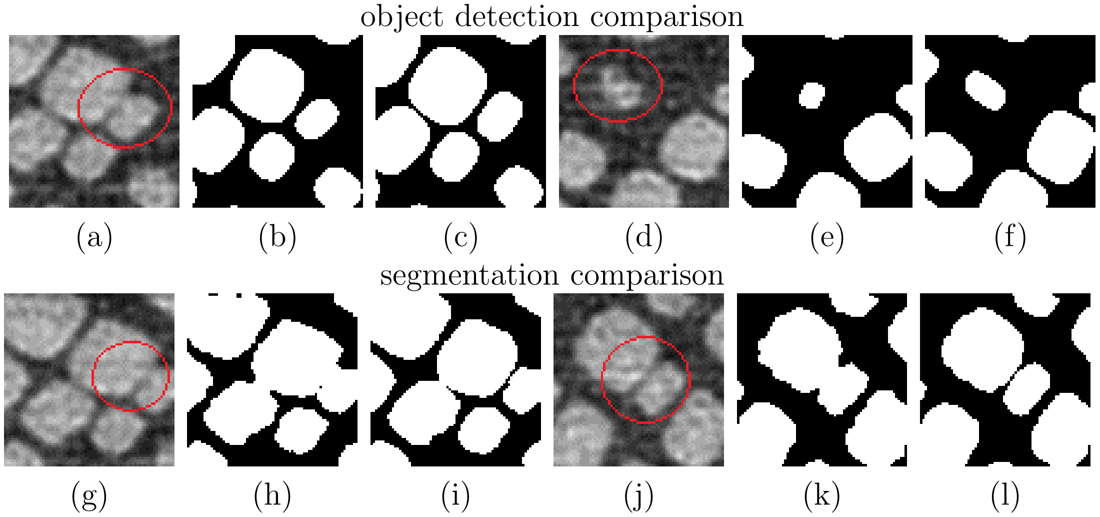


Fig. 4.13. comparison of object detection and segmentation based on the MBD algorithm (a)crop 1(b) MPP-only result for crop 1 (c) proposed model result for crop 1 (object part) (d) crop 2 (e)MPP-only result for crop 2 (f)proposed model result for crop 2 (object part) (g) crop 3 (h) graph cuts result for crop 3 (i) proposed model result for crop 3 (segmentation part) (j) crop 4 (k) graph cuts result for crop 4 (l) proposed model result for crop 4.

detection rate (F.D.R.) defined in Chapter 1 to verify the detection performance. Averages over the ten images of the missed and false detection rates for the MPP model and the combined model are shown in Table 4.1. The result shows our hybrid model has better performance in the accuracy of marks (orientation, axes lengths) compared with the original MPP model (Figure 4.13 first row: two crops of Figure 4.12(a)).

Table 4.1
Missed and false detection rates for NiCrAl images

	MBD	
	M.D.R.	F.D.R.
MPP model	0.83%	3.25%
Hybrid model (object part)	0.83%	0.89%

Table 4.2
Type I and Type II errors for NiCrAl images

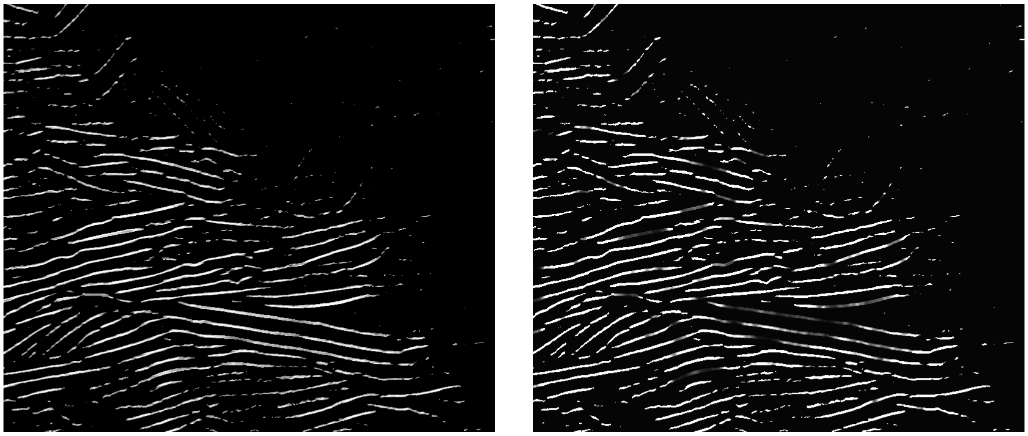
	Type I	Type II
Graph cuts	3.46%	1.78%
Hybrid model (segmentation part)	0.68%	0.91%

In the segmentation part, averages over the ten images of both Type I and Type II errors, which are defined as the ratio of misclassification of background and foreground pixels to the number of pixels of the corresponding class, respectively, are listed in Table 4.2. The result shows our algorithm performs better than graph cuts in both Type I and Type II errors. As in Figure 4.13, second row, (two cropped areas from Figure 4.12(a)), MRF based graph cuts method tends to misclassify the background pixels lying between closely adjacent objects. In our algorithm, this merging effect has been greatly reduced with the help of global constraints.

4.8.2 Silicate image

In this application, we aim to extract the silicate structures and obtain the corresponding segmentation of the image in Figure 4.15 (a), with dimension 804 × 699. The modified Candy model is used and each silicate is modeled as a connected set of line segments. The object potential and segmentation potential for each line segment are defined the same as in Section 4.2. We follow [2] to construct the interactions between line segments as the geometric prior, which consists of an internal "bad" orientation and an external "bad" orientation penalizers. RJ MCMC algorithm is used as the optimization method.

The length and width for the line segments are sampled uniformly between the minimum and maximum values. The orientation is sampled uniformly between $[0, 2\pi]$. We set $\lambda = (3, 1)$. We follow the guidelines in [2] to set up the parameters for the Candy model and RJ MCMC. Since the calculation of the object and segmentation potential are time-consuming, similarly to [2] we precalculate these two potentials



(a)

(b)

Fig. 4.14. illustration of (a) object potentials and (b) segmentation potentials for object with marks ($l = 5, w = 4, \theta = \pi/30$) positioned at every pixel site.

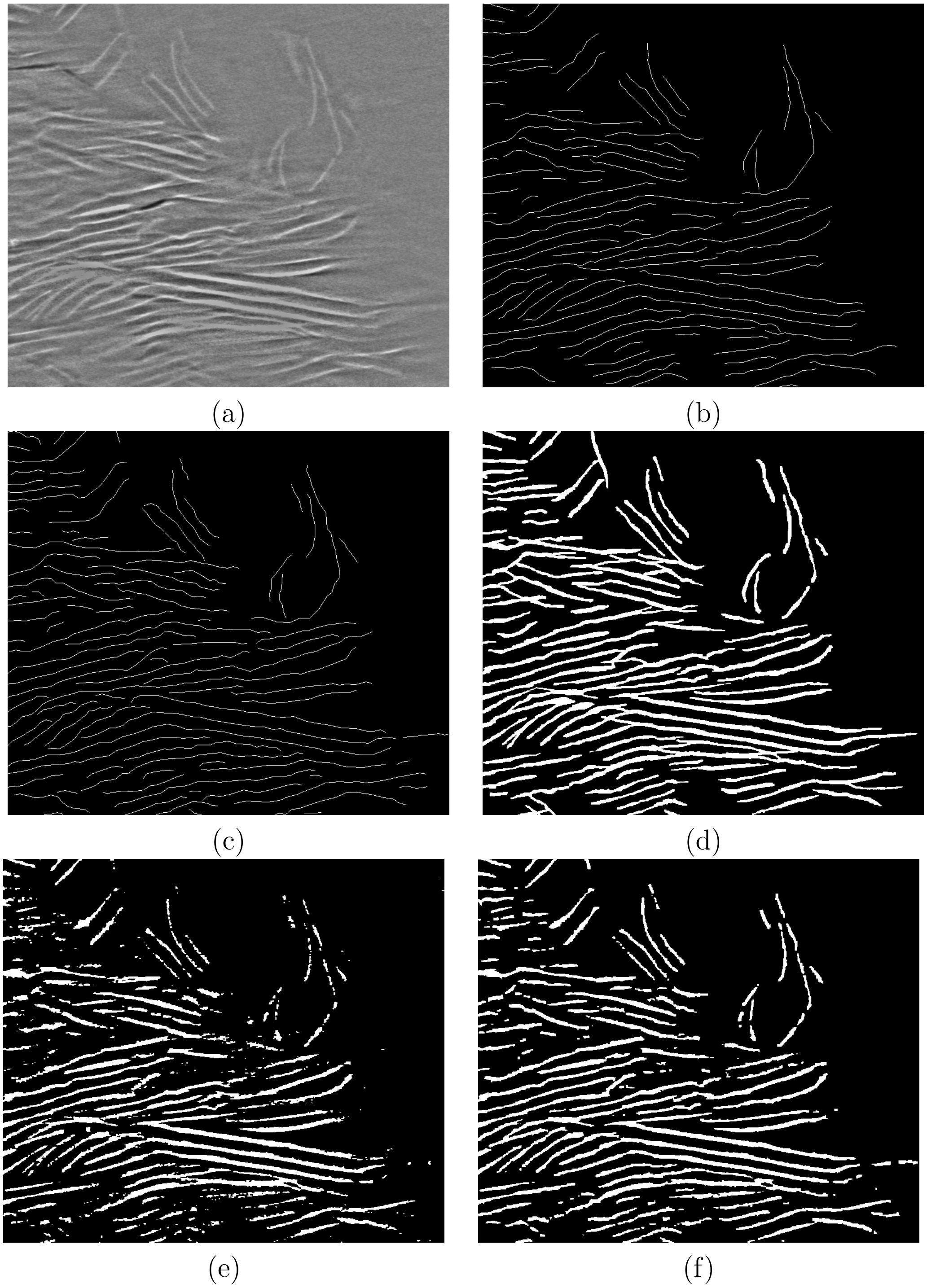


Fig. 4.15. (a) original silicate image (b) MPP only result with the RJ MCMC algorithm (c) proposed model (object part) with the RJ MCMC algorithm (d) hand drawn segmentation (e) graph cuts result (f) proposed model (segmentation part) with the RJ MCMC algorithm.

to speed up the procedure. At every pixel site s, we position a line segment with minimum length l_{min} for a fixed number of orientations N_o and widths N_ω. We calculate the object and segmentation potentials for every orientation $\theta_k, k = 1, ..., N_o$ and width $\omega_{i,i=1,...,N_\omega}$ and compute the potentials for site s:

$$\begin{aligned} V^o_{\omega_i,\theta_k}(s) &= V_o(y|w_s(l_{min}, \omega_i, \theta_k)) \\ V^s_{\omega_i,\theta_k}(s) &= V_s(x^{MAP}_{w_s(l_{min},\omega_i,\theta_k)}) \end{aligned} \tag{4.20}$$

where $w_s(l_{min}, \omega_i, \theta_k)$ is the object with marks $(l_{min}, \omega_i, \theta_k)$ at site s. An illustration of $V^o_{4,\pi/30}(s)$ and $V^s_{4,\pi/30}(s)$ for all pixels with $l_{min} = 5$ is shown in Figure 4.14. Low potentials are colored in white while high potentials are in black.

Then the object potentials and segmentation potentials for an arbitrary object with marks (l, ω, θ) are approximated by:

$$V_o(w, y) = \frac{1}{|w|} \sum_{p \in D_w} V^o_{\omega,\theta}(p) \;\; V_s(x^{MAP}_w) = \frac{1}{|w|} \sum_{p \in D_w} V^s_{\omega,\theta}(p) \tag{4.21}$$

Table 4.3
Missed and false detection rates for Silicate image

	RJ MCMC	
	M.D.R.	F.D.R.
MPP model	20.73%	3.79%
Hybrid model (object part)	10.36%	3.58%

Table 4.4
Type I and Type II errors for Silicate images

	Type I	Type II
Graph cuts	35.43%	1.01%
Hybrid model (segmentation part)	24.60%	1.30%

The results are shown in Figure 4.15. In the object part, we use the missed detection rate (M.D.R.) and the false detection rate (F.D.R.), which are defined as the ratio of the length of omitted and overdetected line segments to the reference length, respectively, to verify the performance. Both rates are listed in table 4.3. For the segmentation part, Type I and Type II errors are listed in table 4.4. The results show that with the benefit of segmentation potential, our model has lower missed detection rate compared to the original MPP model. The corresponding segmentation performs better in Type I error compared with graph cuts method, although slightly worse in Type II error.

4.8.3 Carbon foam images

A Carbon foam microscopy image with dimension 627×700 is presented in Figure 4.16(a). There are some dark and light grey diagonal lines on the image, which are artifacts from the acquisition process. These artifacts will cause adverse effects on processing such as segmentation and further analysis. Our goal is to detect these artifacts and inpaint the corresponding pixels.

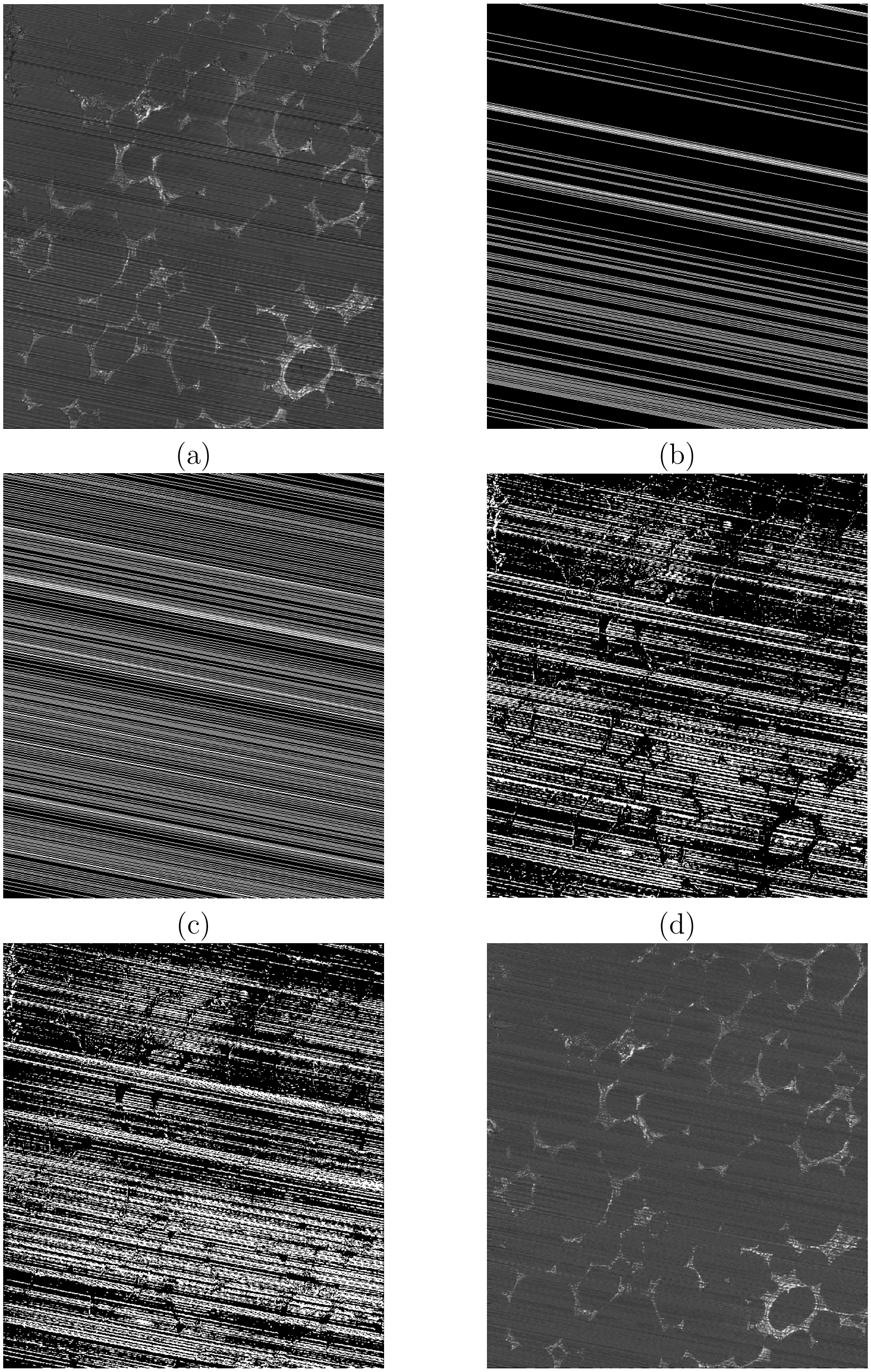
We use the line segment model to represent the diagonal lines. Since for each image, the diagonal lines have the same angle, the orientation for the model is predefined as a constant. The length of each diagonal line is calculated as the distance between the upper-left end and the lower-right end. The marks for this model are w and v. w is the width of the line, which is uniformly sampled between $[1, 4]$. $v \in 0, 1$ is the line tag, where $v = 0$ represents a dark grey line and $v = 1$ represents a light grey line.

We still use the overlapping penalizer in Section 4.2 as the object interaction prior. We set the threshold $T_{overlap} = 0$ to forbid any overlapping between two line segment models. The object potential is defined as in Section 4.2. We use the multiple birth and death algorithm as the optimization method. We set $\lambda = (3, 0.5)$. The parameters for multiple birth and death algorithm are still set as $b_0 = 0.8, \alpha_0 = 10, \sigma_0 = 0.9$ and the cooling factor as 1/0.98. The results are shown in Figure 4.16.

In the object part, with the help of the segmentation potential, many more diagonal lines are successfully detected compared to the original MPP model. In the segmentation part, our hybrid model shows better performance in preserving the line continuity. The false detection is also reduced compared with the graph cuts based MRF model. We inpaint each detected diagonal line with pixels outside the boundary of the corresponding segmentation (Figure 4.16). We further use the MRF based EM-MPM method [63] to segment both the original image (Figure 4.16(a)) and the inpainted image (Figure 4.16(f)). The results are shown as Figure 4.16(g) and Figure 4.16(h). The results show that the artifacts are much alleviated compared with the original image.

4.9 Conclusion

In this chapter, we proposed a hybrid model to analyze images at both the pixel level and the object level. Our model combines the MRF and MPP into one hybrid framework. Therefore, both local constraints and global constraints contribute to pixel-based segmentation and object-based identification. This model is adapted with general optimization methods such as deterministic approaches like ICM algorithm, or simulation schemes like RJ MCMC and multiple birth and death algorithms. The experimental results show that in the object identification part, with the help of local constraints, this hybrid model performs more accurately than the original MPP in selecting object positions and marks. In the segmentation part, constrained by the geometric shape, our hybrid model also has better performance than graph cuts with the original MRF model. In spite of the computational complexity, parallelization methods such as [64], can be explored as a speed-up to make our model more applicable to large image data sets.



(a)

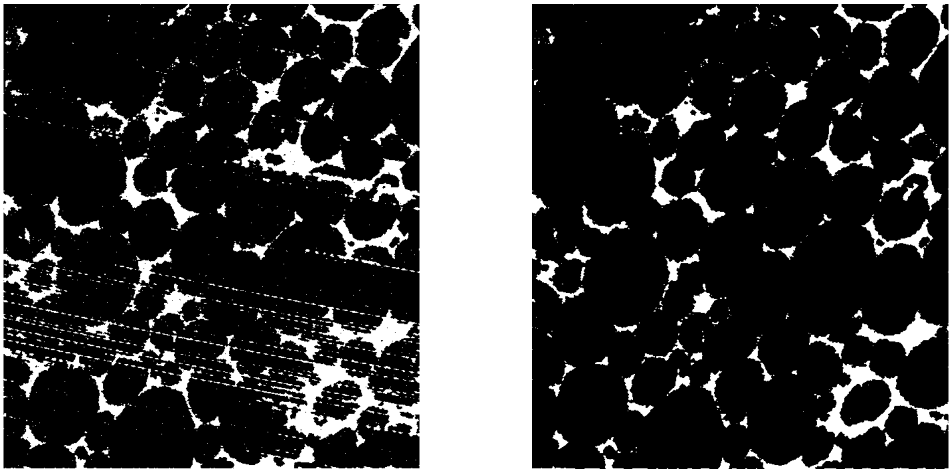
(b)

(c)

(d)

(e)

(f)



(g) (h)

Fig. 4.16. (a) original carbon foam image (b) MPP only result (c) proposed alg. (object part) (d) graph cuts result (e) proposed alg. (segmentation part) (f) inpainted result (g) MRF-based segmentation for original image (h) MRF-based segmentation for inpainted image.

5. A JOINT MARKOV RANDOM FIELD/MARKED POINT PROCESS MODEL FOR MICROSCOPE IMAGES

5.1 Introduction

In Chapter 4, we proposed a hybrid MRF/MPP model. In that model, we incorporate the MRF model as an energy potential into the original MPP model. Thus, the data fitting of an object consists of both an object potential, which imposes the global constraints, and a segmentation potential, which imposes the local constraints. However, such a segmentation potential involves a MAP optimization on the image region that covers an object, which is usually time-consuming, with the computational cost increasing nonlinearly as the object size increases.

In this chapter, we propose a joint MRF/MPP model, where a posterior probability of a joint object/segmentation configuration given the observed image is proposed. According to Bayes' rule, this posterior probability is proportional to the product of a prior and a likelihood. Unlike MRF or MPP, where the likelihood is defined as a function describing how well the observed image fits the segmentation or object field, our joint model interprets the likelihood as a data-fitting on both segmentation and object fields. In addition, the prior in our model not only integrates the interactions between neighboring pixel pairs (as in the MRF) and object pairs (as in the MPP), but also incorporates a term describes the relations between the segmentation and object field, which favors the consistency between them. This model combines the MRF and the MPP as a joint model, while the hybrid model interprets MRF as a segmentation potential, which is embedded in the MPP framework. The relation between the object field and segmentation is modeled as a prior in our joint model. The joint object/segmentation configuration is obtained by maximizing the posterior probability. The alternating minimization algorithm is adopted, alternatively optimizing the posterior with respect to the segmentation or object configuration while keeping the other one fixed.

In Section 5.2, we describe our model and the optimization method. We show the experimental results and draw the conclusion in Section 5.3 and Section 5.4, respectively.

5.2 The joint MRF/MPP model

Let S denote a 2D image lattice and N the number of pixels, where the lattice sites are ordered from 1 to N. We define random fields Y and X on S, where $Y = (Y_1, ..., Y_N)$ is denoted the observed image and $X = (X_1, ..., X_N)$ the segmented image. The segmented image distinguishes each pixel as either an object pixel or a non-object pixel. A pixel is characterized as an object pixel if it belongs to an object or a non-object pixel if it does not belong to any object. $y = (y_1, ..., y_{_N})$ and $x = (x_1, ..., x_{_N})$ are particular sample realizations of Y and X respectively. A marked object (point) is defined as $W_i = \{S_i, M_i\}$, which is characterized by its random location S_i and a set of random variables (marks) M_i describing the object. Taking the ellipse model as an example, the mark space M can be defined as $M = [a_{min}, a_{max}] \times [b_{min}, b_{max}] \times [0, \pi]$, consisting of two axis lengths and one orientation. Note that we will use the term “mark” to refer to both the collection of random variables M_i and each random variable in M_i itself. Then $W \subset S \times M$ is defined as the random marked object field. Let Ω_W denote the space of all possible realizations of W. Then $w = (w_1, ..., w_n) \in \Omega_W$ is a possible object configuration, which is a finite unordered set of marked objects in image S. We define the density function $p(w, x|y)$ such that the posterior probability of W, X given Y is:

$$P(W \in A, X = x|Y = y) = \int_{w \in A} p(w, x|y) d\lambda(w) \tag{5.1}$$

where A is a set of possible realizations of the object field and $\lambda(\cdot)$ is the Lebesgue measure.

According to the Bayes' rule, we have:

$$p(w,x|y) = \frac{f(y|w,x)p(w,x)}{f(y)} \tag{5.2}$$

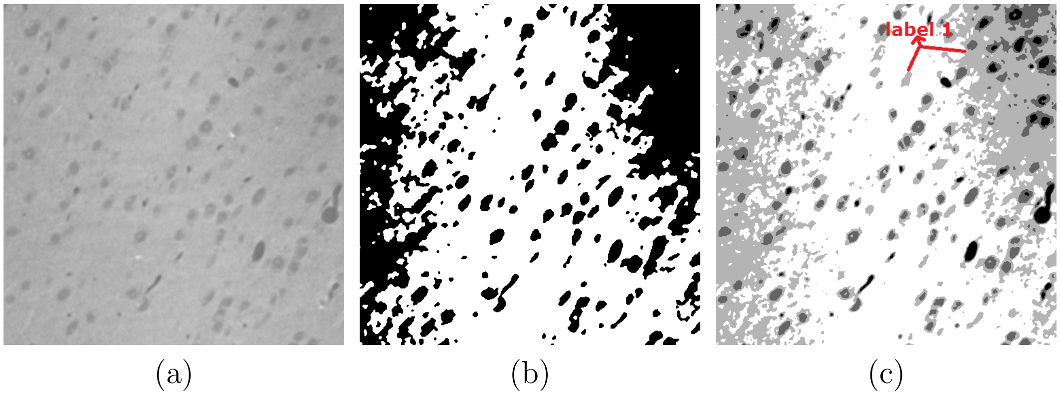
$f(y|w,x)$ is the conditional probability density function of y given w,x. $p(w,x)$ is a joint probability function such that $P(W \in A, X = x) = \int_{w \in A} p(w,x) d\lambda(w)$. $f(y)$ is the probability density function of y, which does not depend on w or x.

5.2.1 Model for observed image

In an MPP, the targeted object is usually described by a shape model, such as the rectangle, disk or ellipse. The associated marks are used to characterize the geometric information of the model, such as the axes length, orientation, etc.

In microscopy imaging, because of the physical characteristics of the image acquisition system [38], we sometimes have a non-uniform illumination across the image; the image is generally spatially non-stationary. As in Figure 5.1(a) - a microscope neuron image from a mouse brain, the mild illumination variation phenomenon results in intensity inhomogeneities in both the neuron particles and background, which brings difficulties on intensity based segmentation. Figure 5.1(b) shows the graph cuts segmentation with two labels (one label representing the object regions and the other representing the non-object regions), where a lot of pixels are misclassified. We can alleviate this problem by imposing more labels. However, in this case, the one to one correspondence between the labels and the object, non-object regions is gone. For example, in Figure 5.1(c), where 4 labels are used for segmentation, pixels belonging to label 1 can be either object pixels or non-object pixels.

In our joint model, we include the label as a mark to capture the intensity variation of targeted objects. Denote $L = \{l_0, l_1 ..., l_n\}$ the set of labels, which consists of finite number of labels n. Each label l_i is described as a Gaussian model with mean and standard deviation $(\mu_{l_i}, \sigma_{l_i})$. Such parameters can be determined by user or estimated using a preprocessing method such as K-means [65], or an Expectation-



(a) (b) (c)

Fig. 5.1. (a) Neuron image (b) graph cuts segmentation with 2 labels (c) graph cuts segmentation with 4 labels.

Maximization (EM) approach in [63]. n is set large enough to capture the objects, usually chosen empirically by user according to the severity of illumination variation of the application. Take Figure 5.1 as an example, we consider 2 labels cannot capture all the objects, but $n = 4$ is enough. If an object has mark l, the pixels belonging to it are also labeled with l. Thus, if an ellipse model has mark $m = \{\theta, a, b\}$ in the original MPP model, including one orientation and two axis lengths, in our model, it will have mark $m = \{\theta, a, b, l\}$.

Let f denote the set of labels that appear inside at least one object in the image; b denote the set of labels that appear as non-object pixels in the image, where $f \subset L$ and $b \subset L$. To model the illumination variation, both the object regions and non-object regions are represented with several labels. It should be noted that f and b do not have to be disjoint. As the case in Figure 5.1(c), where labels are numbered in ascending order of the intensity that represents the label, $f = \{l_1, l_2, l_3\}$ and $b = \{l_0, l_1, l_2\}$. Let x_s denote the segmentation for pixel s. We classify s as an object pixel if $x_s = l$, where $l \in f$ and $l \notin b$; or if $x_s = l$, where $l \in f \cap b$ and s belongs to an object. Otherwise, we classify s as a non-object pixel. It should be noted that another approach for the illumination change is to model the object pixels as a

Gaussian Mixture Model. However, we have not found a way to incorporate GMM into our joint model.

The model for the observed image is characterized by a likelihood function describing how well an observed image y fits the object field and segmentation w, x. We assume the random variables $(Y_1, Y_2, ..., Y_N)$ are conditionally independent given the segmentation X and the object field W. Consider an object realization $w = (w_1, w_2, ..., w_n)$. For each object w_i, we define two auxiliary objects w_i^τ and w_i^γ. w_i^τ is a shrunk version of w_i covering a slightly smaller image region, while w_i^γ is an enlarged version of w_i which covers a slightly larger image region [60]. w_i^τ and w_i^γ are determined by the model we use. For example, an ellipse model w_i with mark $m = \{\theta, a, b, l\}$, will have $w_i^\tau = \{\theta, a-1, b-1, l\}$ and $w_i^\gamma = \{\theta, a+1, b+1, l\}$; while in the Candy model [2], where a line segment model w_i is used with mark $m = \{\theta, a, b, l\}$, representing the orientation, width, length and label, we set the corresponding $w_i^\tau = \{\theta, a-2, b, l\}$ and $w_i^\gamma = \{\theta, a+2, b, l\}$.

Let **Silhouette** $\boldsymbol{S}_w$ denote the projection of w onto the pixel lattice upon which Y and X are defined. Then we divide the image into four regions: the object inner regions $\boldsymbol{S}_{w^\tau}$, the object inside boundary regions $\boldsymbol{S}_w/\boldsymbol{S}_{w^\tau}$, the object outside boundary regions $\boldsymbol{S}_{w^\gamma}/\boldsymbol{S}_w$ and the remaining regions $\boldsymbol{S}/\boldsymbol{S}_{w^\gamma}$. As an example, the four regions for an ellipse model are illustrated in Figure 5.2(a).

Let $G_{x_s}(y_s; \mu_{x_s}, \sigma_{x_s})$ denote the Gaussian density function with mean μ_{x_s} and standard deviation σ_{x_s}, where y_s is the value of y at pixel s, and x_s is the value of x at pixel s.

In the object inner regions $\boldsymbol{S}_{w^\tau}$, pixels are assumed object pixels and follow the same Gaussian distribution characterized by the label of the corresponding object. Specifically, for an object w_i with label l_{w_i}, the pixels are modeled as:

$$g_1(y, w_i, x) = \prod_{s \in \boldsymbol{S}_{w_i^\tau}} G_{x_s}(y_s; \mu_{x_s = l_{w_i}}, \sigma_{x_s = l_{w_i}}) \tag{5.3}$$

In the object inside boundary regions $\boldsymbol{S}_w/\boldsymbol{S}_{w^\tau}$ and outside boundary regions $\boldsymbol{S}_{w^\gamma}/\boldsymbol{S}_w$, we allow pixels to be classified as either object pixels or non-object pixels. This is because in most cases the geometric model does not follow exactly the object boundaries. Figure 5.2(b) shows a NiCrAl particle and Figure 5.2(c) shows its segmentation. In Figure 5.2(d), a superellipse model [60] is used to fit the particle. Since the model cannot perfectly match the boundary, some non-object pixels are inside the model while some object pixels are outside the model. For an object w_i with label l_{w_i}, the corresponding pixels are modeled as:

$$\begin{aligned} g_2(y, w_i, x) = &\prod_{s \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^\tau}} G_{x_s}(y_s; \mu_{x_s}, \sigma_{x_s}) \\ &\prod_{s \in \boldsymbol{S}_{w_i^\gamma}/\boldsymbol{S}_{w_i}} G_{x_s}(y_s; \mu_{x_s}, \sigma_{x_s}) \end{aligned} \tag{5.4}$$

If s is classified as an object pixel, $x_s = l_{w_i}$; If s is classified as a non-object pixel, we allow x_s as any label in b except for l_{w_i}, if $l_{w_i} \in b$. It should be noted that although we model the pixels in both the inside and outside boundary regions with the same distribution and label choices, we distinguish them in $g_2(w_i)$. This is because we encourage consistency between the object field and segmentation. Specifically, we encourage pixels inside the boundary regions as object pixels and outside the boundary regions as non-object pixels, which is described in the joint prior.

In the remaining regions $\boldsymbol{S}/\boldsymbol{S}_{w^\gamma}$, pixels are allowed to be classified as either object pixels or non-object pixels:

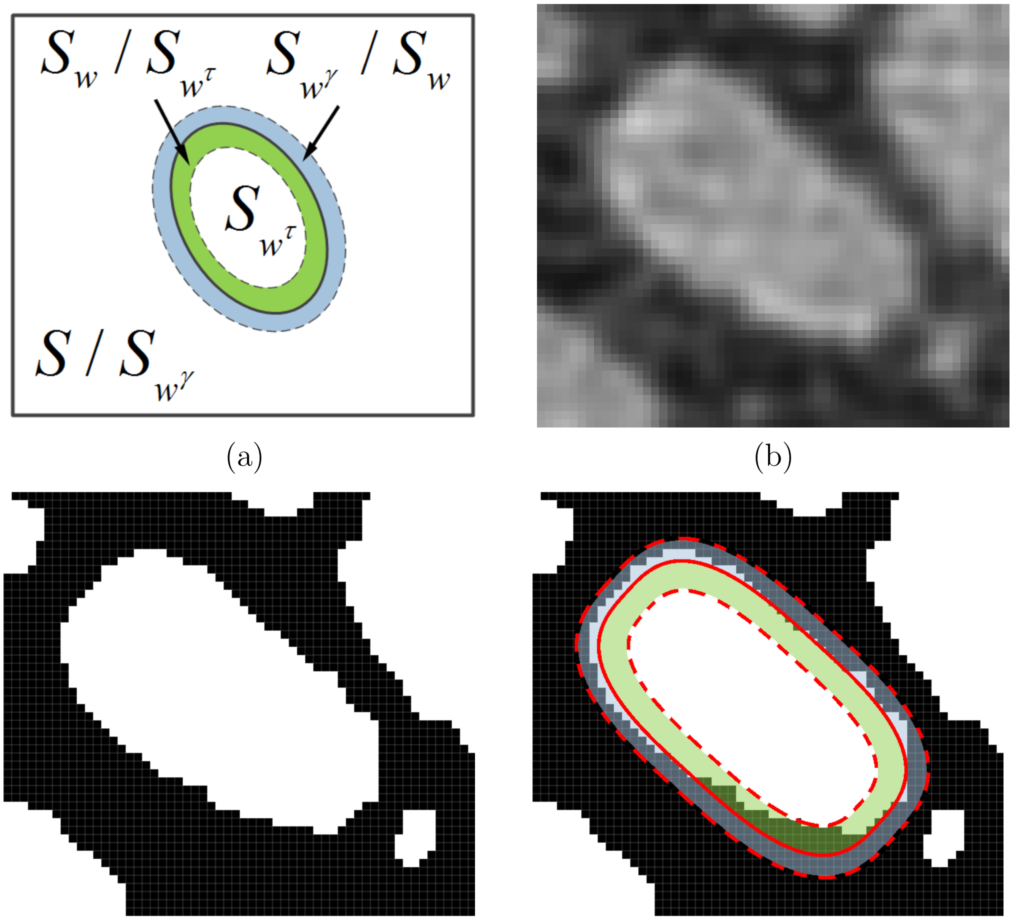
$$g_3(y, w, x) = \prod_{s \in \boldsymbol{S}/\boldsymbol{S}_{w^\gamma}} G_{x_s}(y_s; \mu_{x_s}, \sigma_{x_s}) \tag{5.5}$$

where $x_s \in L$ can be arbitrarily chosen among the label space. If $x_s \in f$ and $x_s \notin b$, we classify s as an object pixel, modeling the scenario where object regions cannot be characterized by an object. Otherwise, we classify s as a non-object pixel.

A special case is that if $l_s \in f \cap b$ and it is not characterized by an object, we classify it as an non-object pixel.

Then the image model is assumed to follow the product of the above Gaussian distributions:

$$f(y|w,x) = \prod_{w_i \in w} [g_1(y,w_i,x) g_2(y,w_i,x)] \cdot g_3(y,w,x) \tag{5.6}$$



(a) (b)

(c) (d)

Fig. 5.2. (a) Four regions for an ellipse model (b) NiCrAl image (c) the corresponding segmentation (d)the superellipse model for the NiCrAl particle with inside and outside boundary regions highlighted.

5.2.2 Joint prior

$p(w, x)$ is a joint probability mass function describing the interactions between neighboring pixel pairs, object pairs and the relation between the object field and segmentation.

The pixel neighborhood system for the image lattice $\boldsymbol{S}$ is defined the same as that in an MRF. The corresponding collection of cliques is denoted $\mathcal{C}$, where $\{r, s\} \in \mathcal{C}$ is a pixel pair satisfying the neighboring relationship. Specifically, the 8-neighborhood system [66] is used. The object neighborhood system is defined the same as that in [1]: w_i and w_j are neighbors, denoted $w_i \sim w_j$, if the objects satisfy a neighbor relationship that depends on the specific MPP model used. Such a relationship defines the object pairwise interactions. A commonly used relationship is: $w_i \sim w_j$ if and only if $d(w_i, w_j) < r$, where $d(.,.)$ is the Euclidean distance and r is the interaction threshold. In the candy model [2], a more complicated relationship is defined.

In addition to establishing the neighborhood systems for the pixels and objects, which describe the interactions within the label field and object field respectively, we characterize the relation between the object field and segmentation. We encourage consistency between the object field and the segmentation. Specifically, for a pixel s, if it belongs to a certain object, we encourage it to be classified as an object pixel; if it does not belong to any object, we encourage it to be classified as a non-object pixel. Then, we define the joint prior as:

$$\begin{aligned} p(w, x) = \frac{1}{Z}\text{exp}\{- \sum_{\{r,s\} \in \mathcal{C}} \beta_1(1 - \delta(x_r, x_s)) - \sum_{w_i \sim w_j} \beta_2 t_2(w_i, w_j) \\ - \sum_{w_i \in w} [\sum_{s \in S_{w_i}/S_{w_i^\tau}} \alpha_1 t_1(x_s, l_{w_i}) + \sum_{s \in S_{w_i^\gamma}/S_{w_i}} \alpha_2 \delta(x_s, l_{w_i})]\} \end{aligned} \tag{5.7}$$

where

$$t_1(a, b) = \begin{cases} -1 & \text{if } a = b \\ 1 & \text{if } a \neq b \end{cases} \tag{5.8}$$

Z is the normalizing constant, and $t_2(w_i, w_j)$ is a function describing the interaction of neighboring object pairs. $\beta_1 > 0$ and $\beta_2 > 0$ are the interaction parameters for the pixel pair and object pair. $\alpha_i > 0, i = \{1, 2\}$ are the weight parameters, which impose a cost when the segmentation and object field are consistent or not.

Specifically, given an object w_i, for pixel $s \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^\tau}$,if $x_s \neq l_{w_i}$, which means the segmentation does not comply with the label of w_i, an increase of α_1 will be imposed; for pixel $s \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^\tau}$,if $x_s = l_{w_i}$, where the segmentation complies with the object label, a decrease of α_1 will be imposed; for pixel $s \in \boldsymbol{S}_{w_i^\gamma}/\boldsymbol{S}_{w_i}$, if $x_s = l_{w_i}$, where the segmentation does not comply with the non-object status based on the object outside boundary regions, an increase of α_2 will be imposed.

5.2.3 Optimization approach

Our target is to find the (w, x) that maximizes the posterior probability $p(w, x|y)$. Since $f(y)$ does not depend on w or x, it is equivalent to obtain a (w, x) that minimizes the energy function $V(w, x) = V_d(w, x) + V_p(w, x)$, where $V_d(w, x) = -ln(f(y|w, x))$ and $V_p(w, x) = -ln(p(w, x))$.

Minimizing over both w and x simultaneously is very difficult. Instead, we minimize $V(w, x)$ with respect to w while keeping x fixed. Then we alternate the minimization with respect to x while keeping w fixed. In other words, rather than solving the original minimization problem over two variables, we solve a sequence of minimization problems over only one variable. Such an approach, usually referred to as the alternating minimization algorithm [67], is summarized as:

Algorithm 1: Minimization of the posterior probability

```
Initialization:
$w^0 = \emptyset$, $x^0 = \underset{x}{\text{argmin}} V_d(w^0, x) + V_p(w^0, x)$;
for $k = 1, ..., n$ do
    $w^{k+1} = \underset{w}{\text{argmin}}(V_d(w, x^k) + V_p(w, x^k))$;
    $x^{k+1} = \underset{x}{\text{argmin}}(V_d(w^{k+1}, x) + V_p(w^{k+1}, x))$;
```

where w^k and x^k represent the object configuration and segmentation at iteration k.

Optimization over w

At iteration k, the object configuration that minimizes $V(w, x^{k-1})$ is denoted $w^k = \underset{w}{\operatorname{argmin}}(V_d(w, x^{k-1}) + V_p(w, x^{k-1}))$.

For an object w_i with label l_{w_i}, given the current object configuration w, where $w_i \notin w$, the energy change from adding candidate object w_i to the current object field is calculated for four different cases, as follows:

- Case 1: Pixel $s \in \boldsymbol{S}_{w_i^{\tau}}, x_s^{k-1} \neq l_{w_i}$. As in (5.3), we label pixels in the inner regions with l_{w_i}. The corresponding energy change is:

$$V_a^1(x_s^{k-1} \neq l_{w_i}) = E_{\to l_{w_i}}(s) \tag{5.9}$$

where

$$\begin{aligned} E_{\to l_{w_i}}(s) = {} & \frac{(y_s - \mu_{l_{w_i}})^2}{2\sigma^2_{l_{w_i}}} + ln\sqrt{2\pi\sigma^2_{l_{w_i}}} - \frac{(y_s - \mu_{x_s^{k-1}})^2}{2\sigma^2_{x_s^{k-1}}} \\ & - ln\sqrt{2\pi\sigma^2_{x_s^{k-1}}} + \sum_{\{r,s\}\in\mathcal{C}} \beta_1(\delta(x_r, x_s^{k-1}) - \delta(x_r, l_{w_i})) \end{aligned} \tag{5.10}$$

from the model for $p(w, x|y)$.

- Case 2: Pixel $s \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^{\tau}}$, $x_s^{k-1} = l_{w_i}$. The corresponding energy change is:

$$V_a^2(x_s^{k-1} = l_{w_i}) = -\alpha_1 \tag{5.11}$$

- Case 3: Pixel $s \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^{\tau}}$, $x_s^{k-1} \neq l_{w_i}$. According to (5.4), it may remain at its current label as a non-object pixel, or be re-labeled to l_{w_i} as an object pixel, which will not be determined until the following segmentation optimization step. However, in the segmentation optimization step, a pixel is usually classified with the label that yields the minimum energy. Thus, we consider it reasonable to calculate the energy change here as the minimum over these two possibilities:

$$V_a^3(x_s^{k-1} \neq l_{w_i}) = min[E_{\rightarrow l_{w_i}}(s), \alpha_1] \tag{5.12}$$

- Case 4: Pixel $s \in \boldsymbol{S}_{w_i^\gamma}/\boldsymbol{S}_{w_i}$, $x_s^{k-1} = l_{w_i}$. Similarly, since it may be classified as a non-object pixel in the following segmentation optimization step, the energy change is calculated as:

$$V_a^4(x_s^{k-1} = l_{w_i}) = min[E_{l_{w_i}\rightarrow}(s), \alpha_2] \tag{5.13}$$

where

$$\begin{aligned} E_{l_{w_i}\rightarrow}(s) = \frac{(y_s - \mu_{l^\circ})^2}{2\sigma_{l^\circ}^2} + ln\sqrt{2\pi\sigma_{l^\circ}^2} - \frac{(y_s - \mu_{l_{w_i}})^2}{2\sigma_{l_{w_i}}^2} \\ - ln\sqrt{2\pi\sigma_{l_{w_i}}^2} + \sum_{\{r,s\}\in\mathcal{C}} \beta_1(\delta(x_r, l_{w_i}) - \delta(x_r, l^\circ)) \end{aligned} \tag{5.14}$$

from the model for $p(w, x|y)$.

$l^\circ \in b/\{l_{w_i}\}$ is the non-object label with minimum mean difference with l_{w_i}, where $|\mu_{l^\circ} - \mu_{x_s^{k-1}}| < |\mu_{l_k} - \mu_{x_s^{k-1}}|, \forall l_k \in b/\{l_{w_i}, l^\circ\}$.

Thus, the energy change from adding w_i to w is calculated as the energy change corresponding to the four cases above, normalized by the size of w_i, together with the energy change from the object interaction:

$$\begin{aligned} V^a_{w_i} = &\sum_{\substack{w_j \in w \\ w_i \sim w_j}} \beta_2 t_2(w_i, w_j) + \frac{1}{|S_{w_i}|}[\sum_{s \in \boldsymbol{S}_{w_i^{\mathcal{T}}}} V^1_a(x^{k-1}_s \neq l_{w_i}) \\ &+ \sum_{s \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^{\mathcal{T}}}} (V^2_a(x^{k-1}_s = l_{w_i}) + V^3_a(x^{k-1}_s \neq l_{w_i})) \\ &+ \sum_{s \in \boldsymbol{S}_{w_i^{\gamma}}/\boldsymbol{S}_{w_i}} V^4_a(x^{k-1}_s = l_{w_i})] \end{aligned} \tag{5.15}$$

Likewise, we can calculate the energy change from removing an object $w_i \in w$ from w. With the energy change from adding or deleting an object, the Green ratio [3] can be calculated. Then we can obtain the optimized object configuration w^k by a stochastic simulation approach, such as RJ MCMC or a multiple birth and death algorithm [1].

optimization over x

At iteration k, let $x^k = \underset{x}{\operatorname{argmin}}(V_d(w^k, x) + V_p(w^k, x))$ denote the segmentation that minimizes $V(w^k, x)$. This corresponds to a markov random field.

Since w^k is fixed, x^k can be obtained by the graph cuts algorithm [62, 68]. Either α expansion or $\alpha - \beta$ swap method is compatible with our model for multiple label segmentation. Specifically, in the $\alpha - \beta$ swap method, for any two labels $l_{\mathbb{A}}, l_{\mathbb{B}} \in L$, the weights of the edges between a pixel p and the two labels $w_1(p, l_{\mathbb{A}}), w_1(p, l_{\mathbb{B}})$, the weight of the edge between two neighboring pixels $w_0(p, q)$ are listed in Table 5.1.

$\boldsymbol{S}_l = \{p \in \boldsymbol{S} | x_p = l\}$ is a subset of pixels classified with label l. $D_p(l)$, usually defined as a data term, describes how well a pixel p fits label l. From $p(w, x|y)$, it is calculated as:

Table 5.1
Edge weights of $\alpha - \beta$ graph

edge	weight	for
$w_0(p,q)$	β_1	$\{p,q\} \in \mathcal{C}$ $p, q \notin \boldsymbol{S}_{l_{\mathbb{A}}}$ or $\boldsymbol{S}_{l_{\mathbb{B}}}$
$w_1(p,l_{\mathbb{A}})$	$D_p(l_{\mathbb{A}}) + \sum_{\substack{q \in \mathcal{C}_p \\ q \notin \boldsymbol{S}_{l_{\mathbb{A}}} \\ q \notin \boldsymbol{S}_{l_{\mathbb{B}}}}} \beta_1$	$p \notin \boldsymbol{S}_{l_{\mathbb{A}}}$ or $\boldsymbol{S}_{l_{\mathbb{B}}}$
$w_1(p,l_{\mathbb{B}})$	$D_p(l_{\mathbb{B}}) + \sum_{\substack{q \in \mathcal{C}_p \\ q \notin \boldsymbol{S}_{l_{\mathbb{A}}} \\ q \notin \boldsymbol{S}_{l_{\mathbb{B}}}}} \beta_1$	$p \notin \boldsymbol{S}_{l_{\mathbb{A}}}$ or $\boldsymbol{S}_{l_{\mathbb{B}}}$

$$D_p(l) = \begin{cases} \frac{(y_s-\mu_l)^2}{2\sigma_l^2} + ln\sqrt{2\pi\sigma_l^2} & p \in \boldsymbol{S}/\boldsymbol{S}_{w^\gamma} \\ \frac{(y_s-\mu_l)^2}{2\sigma_l^2} + ln\sqrt{2\pi\sigma_l^2} & p \in \boldsymbol{S}_{w_i^\tau}, l = l_{w_i} \\ \infty & p \in \boldsymbol{S}_{w_i^\tau}, l \neq l_{w_i} \\ \frac{(y_s-\mu_l)^2}{2\sigma_l^2} + ln\sqrt{2\pi\sigma_l^2} - \alpha_1 & p \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^\tau}, l = l_{w_i} \\ \frac{(y_s-\mu_l)^2}{2\sigma_l^2} + ln\sqrt{2\pi\sigma_l^2} + \alpha_1 & p \in \boldsymbol{S}_{w_i}/\boldsymbol{S}_{w_i^\tau}, l \neq l_{w_i} \\ \frac{(y_s-\mu_l)^2}{2\sigma_l^2} + ln\sqrt{2\pi\sigma_l^2} + \alpha_2 & p \in \boldsymbol{S}_{w_i^\gamma}/\boldsymbol{S}_{w_i}, l = l_{w_i} \\ \frac{(y_s-\mu_l)^2}{2\sigma_l^2} + ln\sqrt{2\pi\sigma_l^2} & p \in \boldsymbol{S}_{w_i^\gamma}/\boldsymbol{S}_{w_i}, l \neq l_{w_i} \end{cases} \tag{5.16}$$

5.2.4 Parameter setting

In (5.7), β_1 is the smoothness parameter to encourage neighboring pixels having the same label, which can be set following a deterministic approach in [69] or a training approach in [70]. β_2, the interaction parameter of neighboring objects, can be set deterministically or estimated by an Expectation-Maximization approach [71].

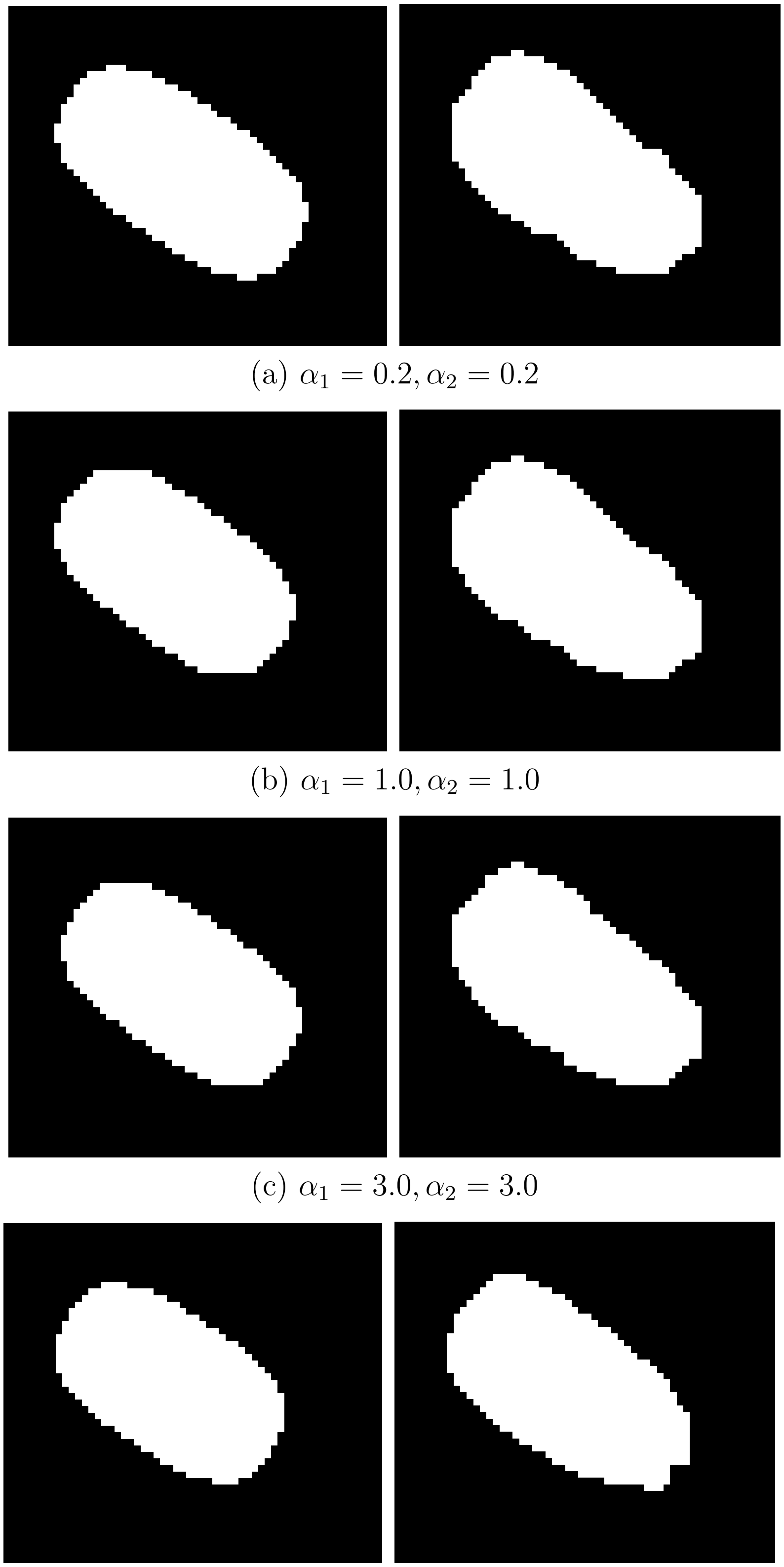
α_1, α_2 are weighting parameters imposing costs when the label field and object field are consistent or not. Specifically, in the object inside boundary region $\boldsymbol{S}_w/\boldsymbol{S}_{w^\tau}$, if pixel's label is consistent with the object field, a decrease energy of α_1 will be imposed; otherwise, an increase energy of α_1 will be imposed. In the object outside boundary regions $\boldsymbol{S}_{w^\gamma}/\boldsymbol{S}_w$, if pixel's label is not consistent with the object field, an increase energy of α_2 will be imposed. These two parameters affect not only the segmentation, but also the object detection result. Figure 5.3 shows the results for the NiCrAl particle in Figure 5.2(b) with different choices of α_1, α_2. In Figure 5.3, as we increase the values of α_1, α_2, the segmentation is influenced more by the geometric model (supperellipse model in this case), while the object detection accuracy also declines.

We have not developed an algorithm to optimize α_1, α_2. Empirically, we consider $[\frac{1}{10} ln\sigma_{min}, \frac{1}{2} ln\sigma_{min}]$ a safe range to set α_1, α_2, where σ_{min} is the minimum standard deviation of all labels. Experiments show good performance with this setting.

5.3 Experimental results

We apply our algorithm to four microscopy image sets: NiCrAl alloy, neuron, Silicate and wood images. Both object detection and segmentation are performed.

In the object detection part w, we use the missed detection rate (M.D.R.) and false detection rate (F.D.R.) to verify the performance.We also applied the MPP-only algorithm, our previous algorithm zhao 2014 [72] for comparison. In the segmentation part x, we use the Type I and Type II errors for verification. We applied the graph cuts segmentation and our hybrid model (zhao 2014) for comparison.



(a) $\alpha_1 = 0.2, \alpha_2 = 0.2$

(b) $\alpha_1 = 1.0, \alpha_2 = 1.0$

(c) $\alpha_1 = 3.0, \alpha_2 = 3.0$

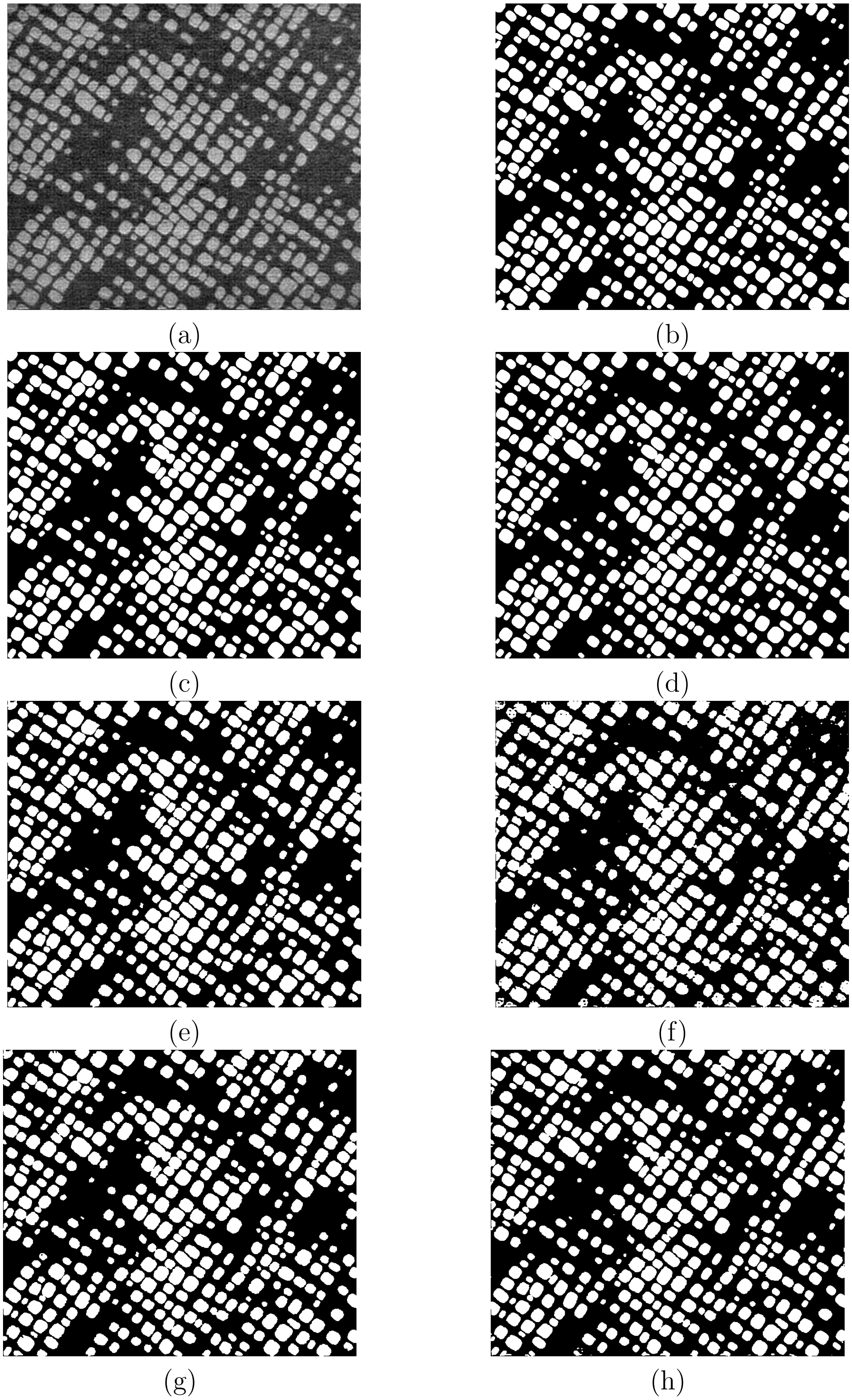
(d) $\alpha_1 = 10.0, \alpha_2 = 10.0$

Fig. 5.3. Results of Figure 2(b) with different parameter setting. For each result, left is the object detection part, right is the segmentation part.

Figure 5.4(a) presents a NiCrAl alloy microscopy image with dimension 744 × 645. The target for this application is to detect each NiCrAl particle and obtain its corresponding segmentation. We classify the pixels into 2 labels (l_0, l_1) and each label is characterized by a Gaussian model with intensity mean and variance (μ_i, δ_i^2), where $i \in \{0, 1\}$. The K-means is used to obtain the mean and variance of each label. We use K-means in the following applications as well. We define $f = \{l_1\}$ and $b = \{l_0\}$. We use the superellipse model to describe the geometric shape of the NiCrAl particle, with marks (a, b, θ, l). The long and short axes (a, b) are uniformly sampled between the minimum and maximum values. Following [72], the orientation θ is sampled from a mixture of two Gaussians with mean values $\frac{1}{4}\pi$ and $\frac{3}{4}\pi$. The variance is set as 0.1^2 for both Gaussians. l is sampled from f, which is set as l_1 in this application. We set $\alpha_1 = 1.2, \alpha_2 = 1.0$ according to Section II. $t_2(w_i, w_j)$ is set the same as the overlapping penalizer in [1]. We deterministically set $\beta_1 = \beta_2 = 1$. The two-pass multiple birth and death algorithm and graph cuts are used for optimization over w and x, respectively. The results are shown in Figure 5.4.

We verify the performance over 10 images. The average of miss and false detection rates, as well as the Type II and Type III errors, are listed in Table 5.2 and 5.3. In the object detection part, our algorithm has similar performance with zhao 2014. Both algorithms perform better than the original MPP algorithm in the accuracy of marks (orientation, axes lengths). In the segmentation part, our algorithm and zhao 2014 have similar performance as well. The MRF based graph cuts method tends to misclassify the non-object pixels lying between closely adjacent objects. Such merging effect will negatively affect the identification of activities between neighboring particles. In our algorithm and zhao 2014, this merging effect has been greatly reduced with the help of global constraints.

Figure 5.5(a) presents a microscope neuron image with dimension of 501×501 from a mouse brain slice, which consists of neuron particles and blood vessels. The target is to detect all neuron particles and blood vessels whose cross-sections form an ellipse shape, and obtain the corresponding segmentations. Since the illumination varies

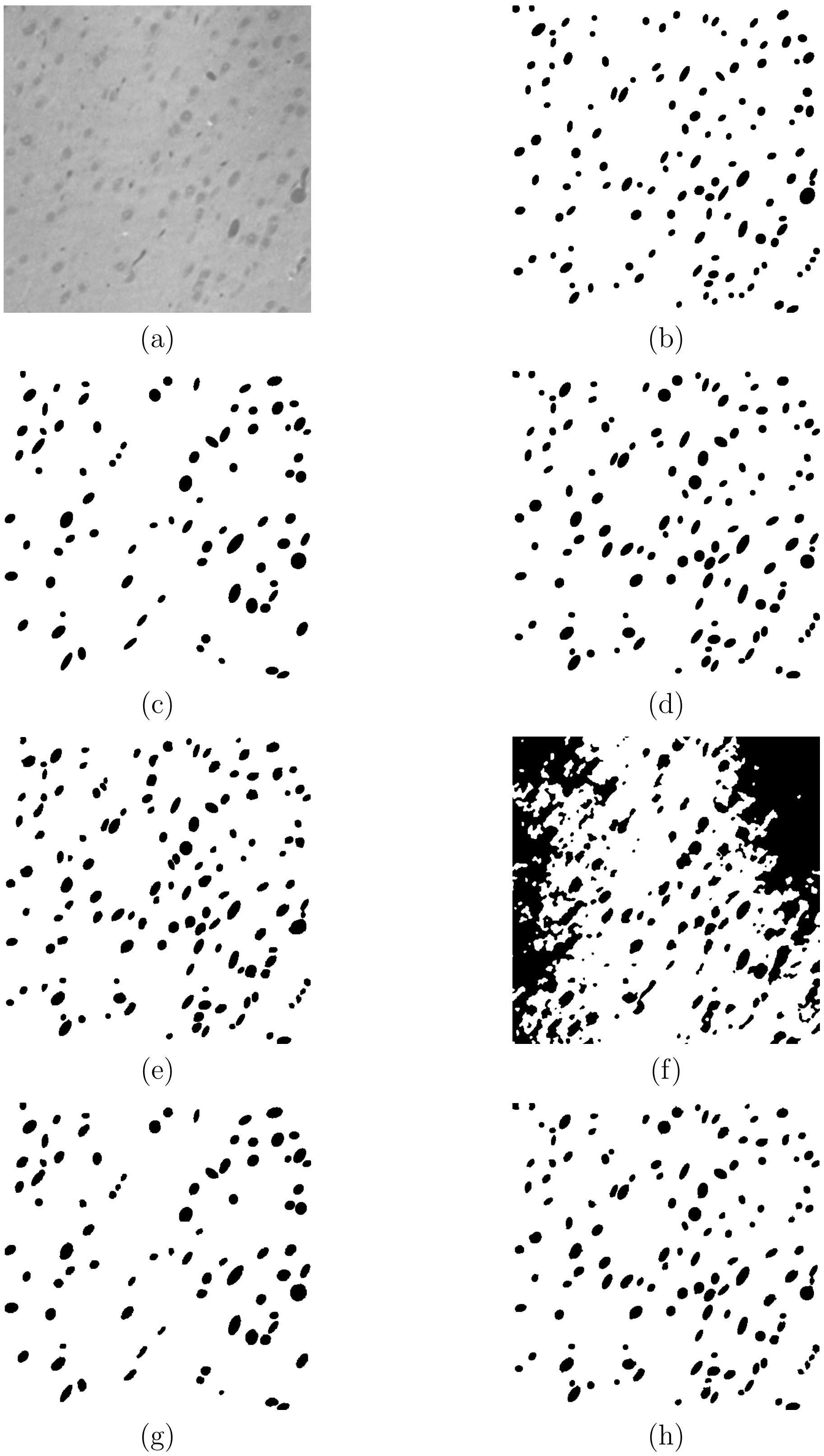


(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5.4. (a) NiCrAl image (b) MPP only result (c) zhao 2014 alg. (object detection part) (d) proposed alg.(object detection part) (e)hand drawn segmentation (f) MRF based Graph cuts result (g) zhao 2014 alg. (segmentation part) (h) proposed alg.(segmentation part) across the image, we classify pixels into 4 labels and each label is characterized by a Gaussian model with intensity mean and variance (μ_i, δ_i^2). The labels are numbered in descending order of intensity mean, where l_0 has the highest mean and l_5 the lowest. We define $f = \{l_1, l_2, l_3\}$ and $b = \{l_0, l_1, l_2\}$. We use the ellipse model to describe the geometric shape, with marks (a, b, θ, l). The long and short axes (a, b) are uniformly sampled between the minimum and maximum values. The orientation θ is uniformly sampled between $[0, 2\pi]$. l is uniformly sample from f. We set $\alpha_1 = \alpha_2 = 0.3$. $t_2(w_i, w_j)$ is set as the overlapping penalizer. We deterministically set $\beta_1 = \beta_2 = 1$. The two-pass multiple birth and death algorithm and graph cuts are used for optimization over w and x, respectively. The results are shown in Figure 5.5.

We verify the performances over 40 images. The comparisons of the object detection and segmentation are listed in Table 5.2 and 5.3. In the object detection part, our algorithm performs better than both mpp model and zhao 2014. It should be noted that zhao 2014 does not take into account the label as a mark and assumes pixels inside all objects follow the same Gaussian model. Such an assumption no longer holds in this illumination variant application. The segmentation potential derived from this assumption even impairs the detection of objects in some regions of the image, resulting in more missed detection rate than the original MPP model. In the segmentation part, our model also has better performance than the other two methods.

Figure 5.6(a) shows a silicate image with dimension 804×699. We aim to extract the silicate structures and obtain the corresponding segmentation. 2 labels (l_0, l_1) are used for pixel classification and we define $f = \{l_1\}$ and $b = \{l_0\}$. We use a modified Candy model [72] to model each silicate as a connected set of line segments. Each line segment has marks (a, b, θ, l). The width and length (a, b) are uniformly sampled between the minimum and maximum values. The orientation θ is uniformly sampled between $[0, 2\pi]$. l is sampled from f, which is set as l_1 in this application. $t_2(w_i, w_j)$ is set as the object interaction prior in [72]. We set $\alpha_1 = 4.0, \alpha_2 = 3.0$. We



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5.5. (a) Neuron image (b) MPP result (c) zhao 2014 alg. (object detection part) (d) proposed alg.(object detection part) (e)hand drawn segmentation (f) Graph cuts results (g) zhao 2014 alg. (segmentation part) (h) proposed alg.(segmentation part)

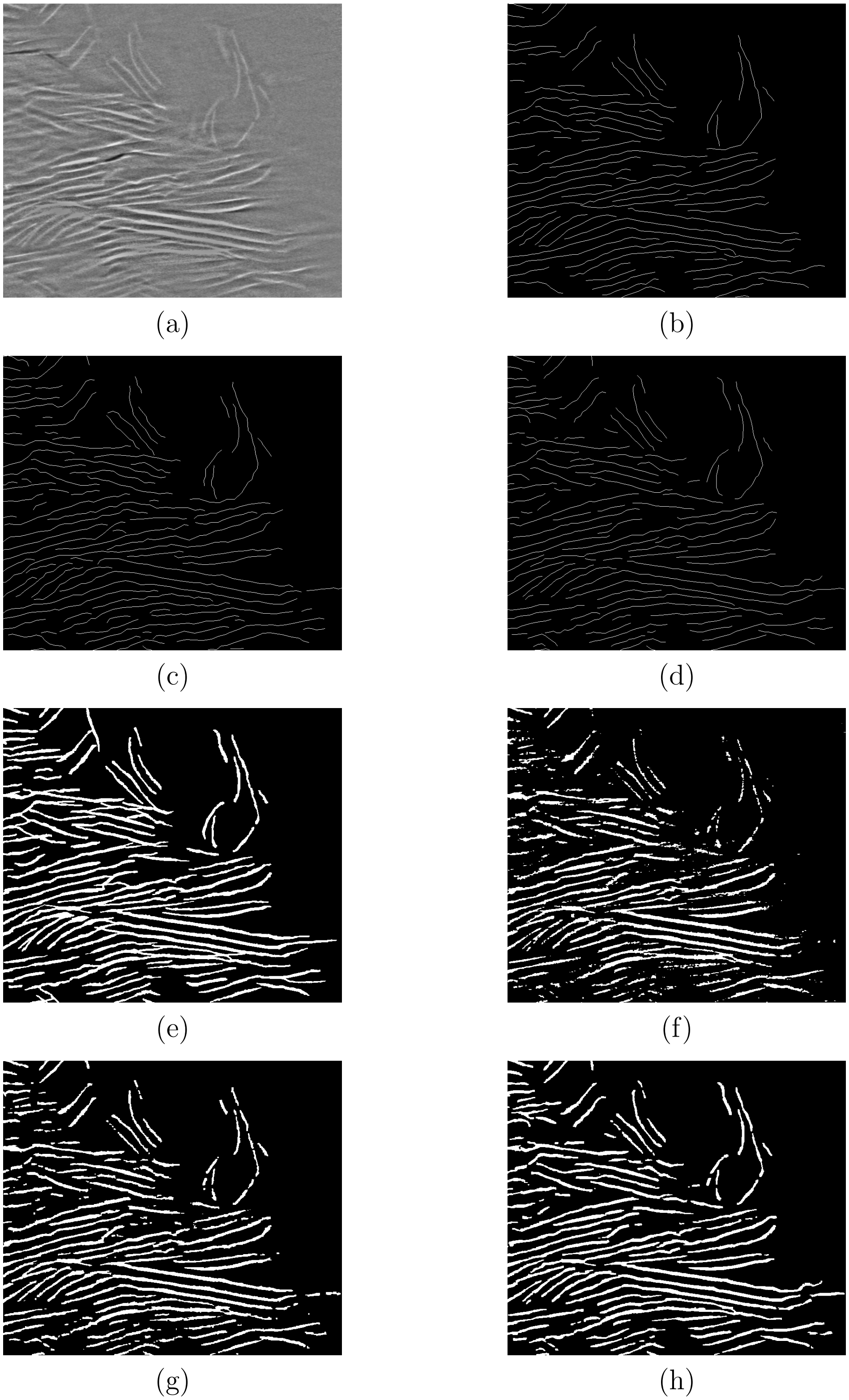
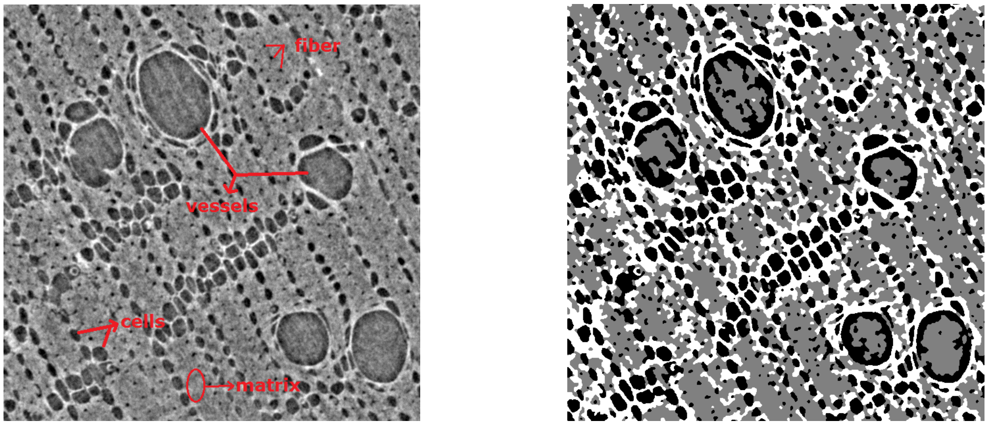


Fig. 5.6. (a) Silicate image (b) MPP only result (c) zhao 2014 alg. (object detection part) (d) proposed alg.(object detection part) (e)hand drawn segmentation (f) Graph cuts results (g) zhao 2014 alg. (segmentation part) (h) proposed alg.(segmentation part)



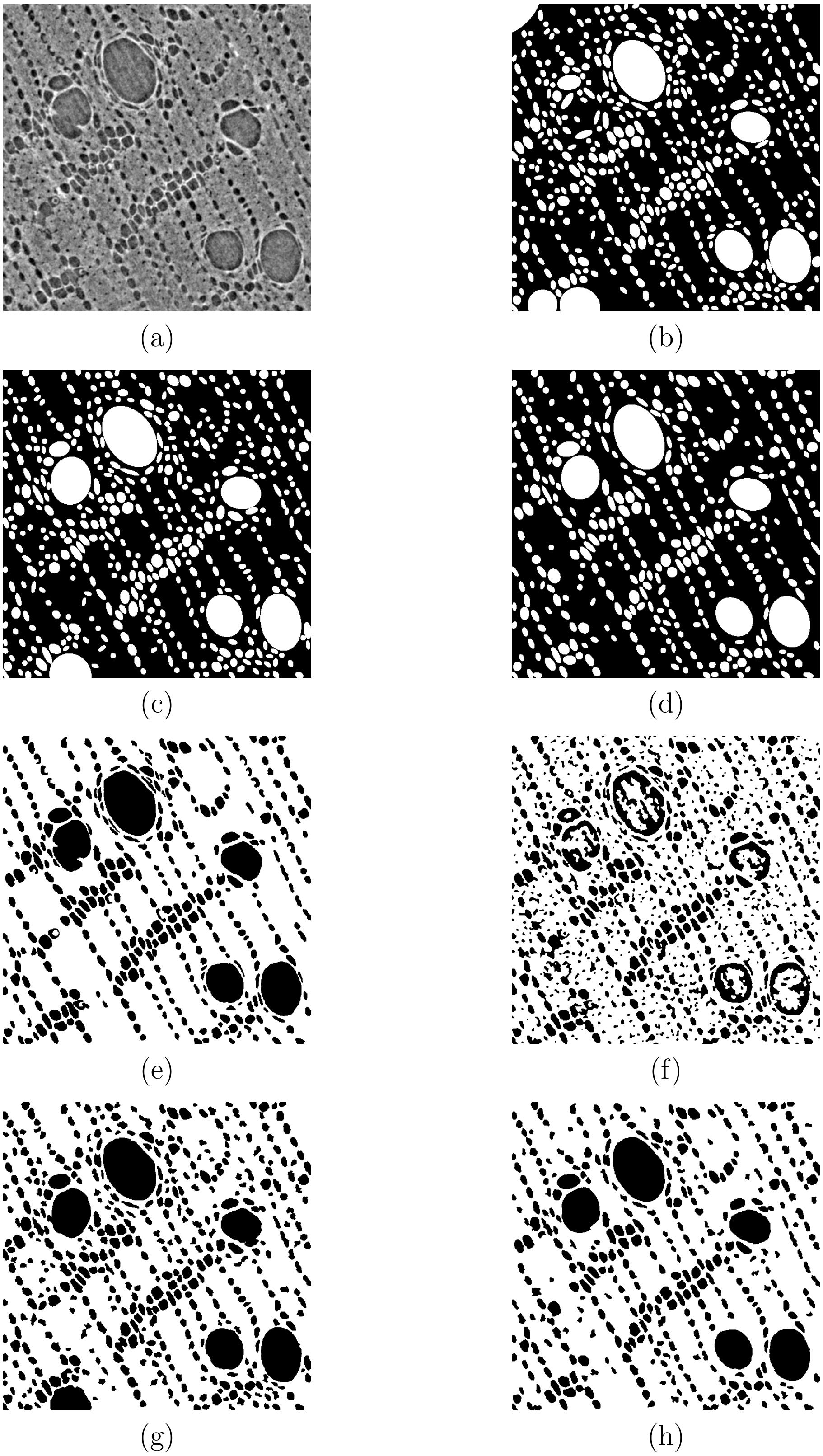
(a) (b)

Fig. 5.7. (a) Wood image (b) graph cuts classification with three labels

deterministically set $\beta_1 = \beta_2 = 1$. The RJ MCMC algorithm and graph cuts are used for optimization over w and x, respectively. The results are shown in Figure 5.6.

As shown in Table 5.2 and 5.3, in the object detection part, zhao 2014 and our proposed algorithm have similar results. In the segmentation part, our proposed algorithm has better accuracy and more natural boundaries segmentation with less artifacts than zhao 2014. One of the reason is that in zhao 2014, the location-adaptive interaction parameter $\beta_{p,q}$ can help to calculate the segmentation potential [72], it also imposes artifacts along the segmentation boundary, especially when the object boundary does not fit the segmentation perfectly, which is the case in this application. Our algorithm has a consistent interaction parameter across the whole image, thus has better accuracy in the boundaries segmentation.

Figure 5.7(a) shows a 634×634 microscopy image of a wood sample, which consists of wood vessels (represented by largest ellipses), parenchyma cells (represented by smaller ellipses), fibers and matrix. The size and shape information of vessels and cells, which store water and food respectively, is important for further analysis of the wood. We aim to detect the vessels and cells and obtain the segmentation. We use three labels for pixel classification due to the intensity inhomogeneity in both the object and non-object regions (Figure 5.7(b)). The labels are numbered


(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 5.8. (a) Wood image (b) MPP only result (c) zhao 2014 alg. (object detection part) (d) proposed alg.(object detection part) (e)hand drawn segmentation (f) Graph cuts results (g) zhao 2014 alg. (segmentation part) (h) proposed alg.(segmentation part)

Table 5.2
Object detection: missed and false detection rates

	NiCrAl		Neuron	
	M.D.R.	F.D.R.	M.D.R.	F.D.R.
MPP	0.83%	3.25%	5.27%	6.66%
zhao 2014	0.83%	0.89%	9.76%	5.36%
proposed alg.	0.80%	0.90%	2.17%	3.19%
	Silicate		Wood	
	M.D.R.	F.D.R.	M.D.R.	F.D.R.
MPP	20.73%	3.79%	1.97%	38.13%
zhao 2014	10.36%	3.58%	1.69%	25.99%
proposed alg.	10.74%	2.44%	5.37%	11.58%

Table 5.3
Segmentation: Type I and Type II errors

	NiCrAl		Neurons	
	Type I	Type II	Type I	Type II
Graph cuts	3.46%	1.78%	80.88%	87.43%
zhao 2014	0.68%	0.91%	2.22%	26.82%
proposed alg.	0.67%	0.78%	0.28%	6.11%
	Silicate		Wood	
	Type I	Type II	Type I	Type II
Graph cuts	35.43%	1.01%	7.33%	17.44%
zhao 2014	24.60%	1.30%	7.59%	6.33%
proposed alg.	13.96%	0.57%	1.67%	7.04%

Table 5.4
Average processing time per image

	NiCrAl	Neurons	Silicate	Wood
zhao 2014	7′37″	2′46″	17′22″	31′42″
proposed alg.	3′14″	1′16″	8′01″	10′01″

in descending order of intensity mean, where l_0 has the highest mean and l_2 the lowest. We define $f = \{l_1, l_2\}$ and $b = \{l_0, l_1, l_2\}$. The ellipse model is adopted and both axes (a, b) are uniformly sampled between the minimum and maximum values. The orientation θ is uniformly sampled between $[0, 2\pi]$. l is uniformly sample from f. We set $\alpha_1 = 1.4, \alpha_2 = 1.2$. $t_2(w_i, w_j)$ is set as the overlapping penalizer. We deterministically set $\beta_1 = \beta_2 = 1$. The two-pass multiple birth and death algorithm and graph cuts are used for optimization over w and x, respectively. The results are shown in Figure 5.8.

In the object detection (Table 5.2), our algorithm shows better performance than the original MPP and zhao 2014 in the false detection rate, while the latter two falsely detected many fibers as parenchyma cells. The original MPP model missed a vessel on the upper left which does not quite fit an ellipse shape, while both our algorithm and zhao 2014 successfully detected it (Figure 5.8(b-d)). Our algorithm performs worse in the missed detection rate. Most missed objects are the cells around the vessels, which look more like a rectangle. In the segmentation part (Table 5.3), our algorithm performs better than the other two models in Type I error, a slightly wore than zhao 2014 in Type II error.

The experiments are performed on a computer with 10-core Intel Xeon-E5 processors. Results show that the processing speed of our algorithm is at least 2 times faster than zhao 2014. This is because zhao 2014 needs to calculate the segmentation potential in the calculation of the energy change for adding or deleting an object, which involves a MAP optimization and is usually time consuming. The computational cost increases nonlinearly as the size of object increases. As in the case of the wood example, where the size of vessels is large, zhao 2014 is 3 times slower than our algorithm. In our algorithm, according to Section 5.2, the calculation of such energy change is easier and time saving. Average processing time per image is listed in Table 5.4 for both algorithms.

5.4 Conclusion

In this chapter, we proposed a joint MRF/MPP model for microscopy image analysis at both pixel and object level. The object field and segmentation are integrated into one model, where local and global constraints are both used for object detection and segmentation. The alternating minimization algorithm is adopted as the inference method. Our model differs from zhao 2014 in (1). our model combines MRF and MPP as a joint model, while zhao 2014 interprets MRF as a segmentation potential, which is still embedded in the MPP framework; (2). the relation between the object field and segmentation is modeled as a prior in our joint model. Experiments show our method has similar or better accuracy and faster processing speed than zhao 2014.

LIST OF REFERENCES

LIST OF REFERENCES

[1] R. M. X. Descombes and E. Zhizhina, "Object extraction using a stochastic birth-and-death dynamics in continuum," *JMIV*, vol. 33, pp. 347–359, 2009.

[2] X. D. Caroline Lacoste and J. Zerubia, "Point processes for unsupervised line network extraction in remote sensing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27(10), pp. 1568–1579, 2005.

[3] P. Green, "Reversible jump mcmc computation and bayesian model determination," *Biometrika*, pp. 711–732, 1995.

[4] L. G. X. Descombes, R. Stoica and J. Zerubia, "A rjmcmc algorithm for object processes in image processing," *Monte Carlo Methods Applicat.*, pp. 149–156, 2001.

[5] A. J. Baddeley and M. N. M. V. Lieshout, "Stochastic geometry models in high-level vision," *Journal of Applied Statistics*, vol. 20, pp. 231–256, 1993.

[6] A. J. Geyer and J. Møller, "Simulation procedures and likelihood inference for spatial point processes," *Scandinavian Journal of Statistics*, vol. 21, pp. 3590–373, 1994.

[7] X. Descombes and J. Zerubia, "Marked point process in image analysis," *IEEE Signal Processing Magazine*, vol. 19(5), pp. 77–84, 2002.

[8] J. Møller and R. Waagepetersen, *Statistical Inference and Simulation for Spatial Point Processes*. Boca Raton: CRC Press, 2004.

[9] e. Descombes, Xavier, *Stochastic geometry for image analysis*. John Wiley and Sons, 2013.

[10] X. D. Perrin, Guillaume and J. Zerubia, "A marked point process model for tree crown extraction in plantations," *Image Processing (ICIP) IEEE International Conference on*, vol. 1, pp. I–661, 2005.

[11] X. D. Perrin, Guillaume and J. Zerubia, "2d and 3d vegetation resource parameters assessment using marked point processes," *International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 1–4, 2006.

[12] C. Avenel and M. S. Kulikova, "Marked point processes with simple and complex shape objects for cell nuclei extraction from breast cancer h&e images," *SPIE Medical Imaging. International Society for Optics and Photonics*, 2013.

[13] G. Dong and S. Acton, "Detection of rolling leukocytes by marked point processes," *Journal of Electronic Imaging*, 2013.

[14] P. Craciun and J. Zerubia, "Unsupervised marked point process model for boat extraction in harbors from high resolution optical remotely sensed images," *Image Processing (ICIP) IEEE International Conference on*, pp. 4122–4125, 2013.

[15] H. G. C. W. Yu Yongtao, Jie Li and M. Cheng, "A marked point process for automated tree detection from mobile laser scanning point cloud data," *Computer Vision in Remote Sensing (CVRS), IEEE International Conference on*, pp. 140–145, 2012.

[16] M. F. Gomes and P. Maillard, "Integration of marked point processes and template matching for the identification of individual tree crowns in an urban and a wooded savanna environment in brazil," *SPIE Remote Sensing. International Society for Optics and Photonics*, 2014.

[17] K. W. D. Neeraj J. Gadgil, Paul Salama and E. J. Delp, "Nuclei segmentation of fluorescence microscopy images based on midpoint analysis and marked point process," *IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*, pp. 37–40, 2016.

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

[19] A. V. L. R. Kulikova, Maria and D. Racoceanu, "Nuclei extraction from histopathological images using a marked point process approach," *SPIE Medical Imaging. International Society for Optics and Photonics*, 2012.

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

[21] X. D. Mathias Ortner and J. Zerubia, "Building outline extraction from digital elevation models using marked point processes," *International Journal of Computer Vision*, pp. 107–132, 2007.

[22] X. D. Mathias Ortner and J. Zerubia, "A marked point process of rectangles and segments for automatic analysis of digital elevation models," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 30(1), pp. 105–119, 2008.

[23] X. D. Csaba Benedek and J. Zerubia, "Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34(1), pp. 33–50, 2012.

[24] A. Borcs and C. Benedek, "A marked point process model for vehicle detection in aerial lidar point clouds," *ISPRS Congress*, pp. 93–98, 2012.

[25] A. Borcs and C. Benedek, "Extraction of vehicle groups in airborne lidar point clouds with two-level point processes," *Geoscience and Remote Sensing, IEEE Transactions on*, pp. 1475–1489, 2015.

[26] X. D. R. Stoica and J. Zerubia, "Road extraction in remotely sensed images using a stochastic geometry framework," *Proc. Int. Workshop Bayesian Inference and Maximum Entropy Methods*, 2000.

[27] B. Nazre and R. Chellappa, “Modeling and detection of wrinkles in aging human faces using marked point processes,” *Computer Vision-ECCV*, pp. 178–188, 2012.

[28] W. F. Dengfeng Chai and F. Lafarge, “Recovering linenetworks in images by junction-point processes,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1894–1901, 2013.

[29] O. L. Keresztes, Barna and M. Borda, “Seismic fault detection using marked point processes,” *Image Processing (ICIP), IEEE International Conference on*, pp. 565–568, 2009.

[30] W. Ge and R. T. Collins, “Marked point processes for crowd counting,” *Computer Vision and Pattern Recognition (CVPR) IEEE Conference on*, pp. 2913–2920, 2009.

[31] A. Utasi and C. Benedek, “A 3-d marked point process model for multi-view people detection,” *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 3385–3392, 2011.

[32] A. Utasi and C. Benedek, “A bayesian approach on people localization in multi-camera systems,” *Circuits and Systems for Video Technology, IEEE Transactions on*, pp. 105 – 115, 2013.

[33] W. T. O. Sreetama Basu and D. Racoceanu, “Neurite tracing with object process,” *IEEE Transactions on Medical Imaging*, 2016.

[34] P. W. Soubies, Emmanuel and X. Descombes, “A 3d segmentation algorithm for ellipsoidal shapes. application to nuclei extraction,” *ICPRAM-International Conference on Pattern Recognition Applications and Methods*, 2013.

[35] P. C. M. O. J. Zerubia, “Joint detection and tracking of moving objects using spatio-temporal marked point processes,” *IEEE Winter Conference on Applications of Computer Vision*, pp. 177–184, 2015.

[36] W. K. D. Stoyan and J. Mecke, *Stochastic Geometry and its Applications.* New York: Wiley, 1987.

[37] A. B. S. Descamps, X. Descombes and J. Zerubia, “Automatic flamingo detection using a multiple birth and death process,” *IEEE ICASSP*, pp. 1113–1116, 2008.

[38] J. S. J. MacSleyne, M.D. Uchic and M. Graef, “Three-dimensional analysis of secondary γ' precipitates in rené-88 DT and UMF-20 superalloys,” *Acta Materialia*, pp. 6251–6267, 2009.

[39] P. R. Goudail, François and G. Delyon, “Bhattacharyya distance as a contrast parameter for statistical processing of noisy optical images,” *JOSA A*, pp. 1231–1240, 2004.

[40] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions and the bayesian restoration of images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI*, pp. 721–741, 1984.

[41] G. R. J. A. Srivastava, U. Grenander and M. I. Miller, “Jumpdiffusion markov processes on orthogonal groups for object pose estimation,” *Journal of Statistical Planning and Inference*, pp. 15–37, 2002.

[42] C.-R. Hwang, “Simulated annealing: theory and applications,” *Acta Applicandae Mathematicae*, pp. 108–111, 1988.

[43] P. S. P. Salamon and R. Frost, “Facts, conjectures, and improvements for simulated annealing,” *SIAM Monographs on Mathematical Modeling and Computation*, 2002.

[44] G. C. AG Eldin, X. Descombes and J. Zerubia, “A fast multiple birth and cut algorithm using belief propagation,” *Image Processing (ICIP), IEEE International Conference on*, pp. 2813–2816, 2011.

[45] G. C. AG Eldin, X. Descombes and J. Zerubia, “Multiple birth and cut algorithm for multiple object detection,” *Journal of Multimedia Processing and Technologies*, 2012.

[46] J. Besag, “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society. Series B*, pp. 259–302, 1986.

[47] C. Bouman and M. Shapiro, “A multiscale random field model for bayesian image segmentation,” *Image Processing, IEEE Transactions on*, pp. 162–177, 1994.

[48] M. Comer and E. Delp, “Segmentation of textured images using a multiresolution gaussian autoregressive model,” *Image Processing, IEEE Transactions on*, pp. 408–420, 1999.

[49] S. X. Yu, “Segmentation using multiscale cues,” *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[50] L. G. H. Lombaert, Y. Sun and C. Xu, “A multilevel banded graph cuts method for fast image segmentation,” *International Conference on Computer Vision*, pp. 259–265, 2005.

[51] F. B. Timothée Cour and J. Shi, “Spectral segmentation with multiscale graph decomposition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1124–1131, 2005.

[52] E. P. X. Descombes, J. F. Mangin and M. Sigelle, “Fine structures preserving markov model for image processing,” *Proceedings of the 9th Scandinavian Conference on Image Analysis*, pp. 349–356, 1995.

[53] H. Tjelmeland and J. Besag, “Markov random felds with higher-order interactions,” *Scandinavian Journal of Statistics*, pp. 415–433, 1998.

[54] R. Paget and I. D. Longstaff, “Texture synthesis via a noncausal nonparametric multiscale markov random feld,” *IEEE Transactions on Image Processing*, pp. 925–931, 1998.

[55] P. H. T. Pawan M. Kumar and A. Zisserman, “Objcut: Efficient segmentation using top-down and bottom-up cues,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 530–545, 2010.

[56] M. C. J. S. M. U. J.P. Simmons, P. Chuang and M. D. Graef, “Application and further development of advanced image processing algorithms for automated analysis of serial section image data,” *Modelling and Simulation in Materials Science and Engineering*, 2009.

[57] W. F. Dengfeng Chai and M. Y. Yang, "Combine markov random fields and marked point processes to extract building from remotely sensed images," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 365–370, 2012.

[58] N. C. A. L. B. A. G. M.M Sidi Youssef, C. Mallet, "Combining top-down and bottom-up approaches for building detection in a single very high resolution satellite image," *IEEE International Geoscience and Remtote Sensing Symposium*, pp. 4820–4823, 2014.

[59] G. G. Lafarge, Florent and X. Descombes, "Geometric feature extraction by a multimarked point process," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 1597–1609, 2010.

[60] M. L. C. Huixi Zhao and M. D. Graef, "A unified markov random field/marked point process image model and its application to computational materials," *IEEE Conference on Image Processing*, pp. 6101–6105, 2014.

[61] V. A. Krylov and J. D. Nelson, "Fast road network extraction from remotely sensed images," *Advanced Concepts for Intelligent Vision Systems*, pp. 227–237, 2013.

[62] O. V. Y. Boykov and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pp. 1222–1239, 2001.

[63] M. Comer and E. Delp, "The em/mpm algorithm for segmentation of textured images: analysis and further experimental results," *Image Processing, IEEE Transactions on*, pp. 1731–1744, 2000.

[64] P. F. C. Avenel and D. Béréziat, "Parallel birth and death process for cell nuclei extraction in histopathology images," *International Conference on Parallel Processing (ICPP)*, pp. 429–438, 2013.

[65] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 1999.

[66] S. Li, "Modeling image analysis problems using markov random fields," *Handbook of statistics, Elsevier Scinence*, 2000.

[67] I. Csiszàr and G. Tusnàdy, "Information geometry and alternating minimization procedures," *Statistics and decisions*, 1984.

[68] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International journal of computer vision*, pp. 109–131, 2006.

[69] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," *Computer Vision Eighth IEEE International Conference on*, pp. 105–112, 2001.

[70] P. K. Szummer, Martin and D. Hoiem, "Learning crfs using graph cuts," *European Conference on Computer Vision (ECCV)*, pp. 582–595, 2008.

[71] X. D. S. Ben Hadj, F. Chatelain and J. Zerubia, "Parameter estimation for a marked point process within a framework of multidimensional shape extraction from remote sensing images," *Proc. of ISPRS Conference on Photogrammetry Computer Vision and Image Analysis*, pp. 1–6, 2010.

[72] H. Zhao and M. L. Comer, "A hybrid markov random field/marked point process model for analysis of materials images," *IEEE transaction on computational imaging*, 2016.

VITA

VITA

Education

Degree	Date	School
BSEE	Aug. 2007	Shanghai Jiao Tong University
MSEE	Mar. 2010	Shanghai Jiao Tong University
MSEE	May 2010	Georgia Institute of Technology
PhD	Aug. 2016	Purdue University

Professional Experience

- Research Assistant, Shanghai Jiao Tong University, School of Electronic, Information and Electrical Engineering, Shanghai, China Sep. 2008-Mar. 2010
- Research Engineer (intern), Global Research Center, General Electric, Shanghai, China Mar. 2009-May 2009
- DSP Software Engineer, SRS Labs Inc., Shanghai, China Mar. 2010-Oct. 2010
- Research Assistant, Purdue University, School of Electrical Engineering, West Lafayette, Indiana, USA Aug. 2011-July 2016

Publications

- Huixi Zhao, Hua Yang, Shibao Zheng, “An efficient method for detecting ghosts and left objects in intelligent video surveillance”, *International Conference on Image and Signal Processing*, pp. 1-6, 2009.

- Huixi Zhao, Mary Comer and M. D. Graef, “A unified Markov random field/marked point process image model and its application to computational materials”, *IEEE Conference on Image Processing*, pp. 6101-6105, 2014.
- Huixi Zhao, Mary Comer, “A hybrid Markov random field/marked point process model for analysis of materials images”, *IEEE Transactions on Computational Imaging*, 2016.

Invited Talks

- “Marked point process models for microscope images of materials”, *IS&T International Symposium on Electronic Imaging*, San Francisco, California, USA, February, 2014.