Camera based Primary Separation Vessel Interface Level Detection and Estimation Utilizing Markov Random Field based Image Processing

by

Zheyuan Liu

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

Abstract

The level of the froth middling interface in primary separation vessel plays an important role in overall bitumen recovery in conventional oil sands bitumen extraction process. To maintain the interface within a certain range of level, the accurate measurement is always desired. Online camera detector usually has the best performance. However, it is not always reliable. The objective of this thesis is to develop a new approach to improve the reliability of camera sensor, which is Markov random field based image processing technique.

An experimental setup was designed to simulate the liquid interface. Oil and water were used to form an interface because they are immiscible under normal condition. An online camera was installed to capture the image contain the interface level. Two differential pressure sensors were also installed, and the actual interface height could be determined based on those measurements.

A Markov random field based supervised image segmentation technique was proposed to convert the raw image to a binary image. The interface level could be determined based on the vertical profile of averaged horizontal pixel values of the segmented image. The interface level estimations obtained using image processing technique were validated and compared with the estimations using traditional differential pressure sensors for various operating conditions. All results showed the agreement. An extended iterated conditional modes algorithm was proposed by introducing the k-means clustering as the initial estimation in the aim to reduce the

computational cost. A customized neighborhood system was designed and aimed to reduce interface estimation variance.

An approach of Gaussian mixture model and Markov random field based unsupervised image segmentation was also proposed. Expectation maximization algorithm was used to estimate the parameters in the model. The segmented images were compared with those obtained using Gaussian mixture model based unsupervised segmentation approach, and only the segmentation obtained using proposed Markov random field based approach agreed with those using the supervised technique. Interface level estimations were also compared and showed the agreement.

To predict the segmented image and interface level, the spatial temporal Markov random field based auto-logistic model was proposed to obtain the prediction. A two-step approach was proposed: the observed image at the future time was predicted using a modified random walk model, and then the image segmentation at the future time could be estimated using the proposed spatial temporal Markov random field based auto-logistic model. Both predicted observed image and segmented image at the future time were validated by comparing with the real observation and segmentation. Though a number of pixels were mis-segmented, the interface level prediction was still agreed with real estimation.

This thesis is dedicated to my dear parents
for their love, endless support
and encouragement.

Acknowledgements

Edmonton's fall is beautiful but short. My two-year graduate study was tough but memorable.

I would like to give my greatest and foremost gratitude to my supervisor, Dr. Biao Huang, who gave me the opportunity of exploring academic research in this excellent computer process control (CPC) research group. Without Dr. Huang's patient and explicit guidance, as well as continuous support and encouragement, I am not able to complete the work of this thesis. My spatial thanks also go to my co-supervisor, Artin Afacan, who helped me in operating experiment, reviewing my thesis and providing many valuable suggestions.

It is my honor to getting involved in such great CPC research group. All the people in the group are talent, professional and helpful. Specifically, I would like to thank two post-Doctoral fellows: Dr. Alireza Fatehi from Iran and Dr. Hariprasad Kodamana from India. Dr. Alireza Fatehi has already left our group for a long time, however, he was the one helped me a lot at the beginning of my research journey. He was also the one who provided the main idea of my thesis work. Dr. Hariprasad Kodamana helped me on academic research during the past half year. This thesis cannot be completed without his patient assistance in explaining concepts and academic support. I would also like to give my sincere thanks to all my present and previous colleagues for their help, listed but not limited as the follows: Yanjun Ma, Chao Shang, Wenhan Shen, Rahul Raveendran, Shabnam Sedghi, Xunyuan Yin, Ruomu Tan, Ouyang Wu, Yaojie Lu, Ming Ma, Anahita Sadeghian, Elham Naghoosi, Mengqi Fang, Lei Fan, Fadi Ibrahim and many else. My sincere gratefulness also goes to Sheng Zhang and several close friends here at Edmonton for their accompany and

inspiration: Ran Li, Xiaolong Li, Wenjie Liu, Qipei Mei, Jack Wang, Jiarui Wang, Tianyu Wang, Muqing Yang, and Yue Zhao.

Financial supports from Natural Sciences and Engineering Research Council (NSERC) and Alberta Innovates Technology Futures (AITF) are also acknowledged.

Last but not least, I would like to give my special gratitude to my dear parents for their endless love, support and encouragement. Especially, great thanks to my father, Dr. Hunju Liu, who is a professor of mechanical engineering at Taiyuan University of Technology in P.R. China. He was my motivation of choosing academic research after received my bachelor degree two years ago. He is my idol.

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Acronyms

DP Cell	*Differential pressure cell/sensor*
EM	*Expectation maximization*
Ex-ICM	*Extended iterated conditional modes*
GLR	*Generalized linear regression*
GMM	*Gaussian mixture model*
GRF	*Gibbs random field*
ICM	*Iterated conditional modes*
IF	*Interface*
i.i.d	*independent identically distributed*
KL	*Kullback-Leibler*
LHS	*Left hand side*
MAP	*Maximum a posteriori*
MFA	*Mean field approximation*
MFT	*Mean field theory*
MRF	*Markov random field*
NS	*Neighborhood system*
PSV	*Primary separation vessel*
SAGD	*Steam assisted gravity drainage*
ST-MRF	*Spatial temporal Markov random field*
w.r.t	*with respect to*

Chapter 1

Introduction

1.1 Industrial Background

The oil sands of Alberta are the world's third largest oil reserves. However, the average cost of producing oil from oil sands is high, as compared to other global oil producers. The high cost of oil production can be attributed to currently used water-based thermal heating technology such as hot-water extraction for open-pit mines and in-situ *steam assistant gravity drainage* (SAGD) for extraction of viscous bitumen from sand, clay and water mixture of oil sands. Even though in-situ extraction can provide smaller foot-print both economically and environmentally, the conventional extraction based froth flotation technique is still the dominant process at current time. The *primary separation vessel* (PSV) is the core unit in the entire process of extracting bitumen from open-pit mined ore.

In the conventional bitumen extraction process, PSV is the main unit where bitumen gets extracted from water, clay and sands based on froth flotation. The separation vessel is usually operated by adding certain chemicals and hot water, injecting air and stirring continuously. Adding certain chemicals and hot water into the vessel is aiming to increase temperature and reduce bitumen viscosity, injecting air is aiming to let bitumen attach air bubbles and float on the top, and continuous stir is aiming to let mixing and separation more efficient and effective. As a result, three layers are formed in primary separation vessel due to density difference, they are *bitumen froth layer* on the top, *tailings* at the bottom, and the layer sandwiched in the middle is

named as *middling layer*. Figure 1.1 shows a schematic diagram of three layers in PSV. Froth layer contains the majority amount of bitumen, the most sands sink down and present in tailings, and water and clay remain in the middling layer.

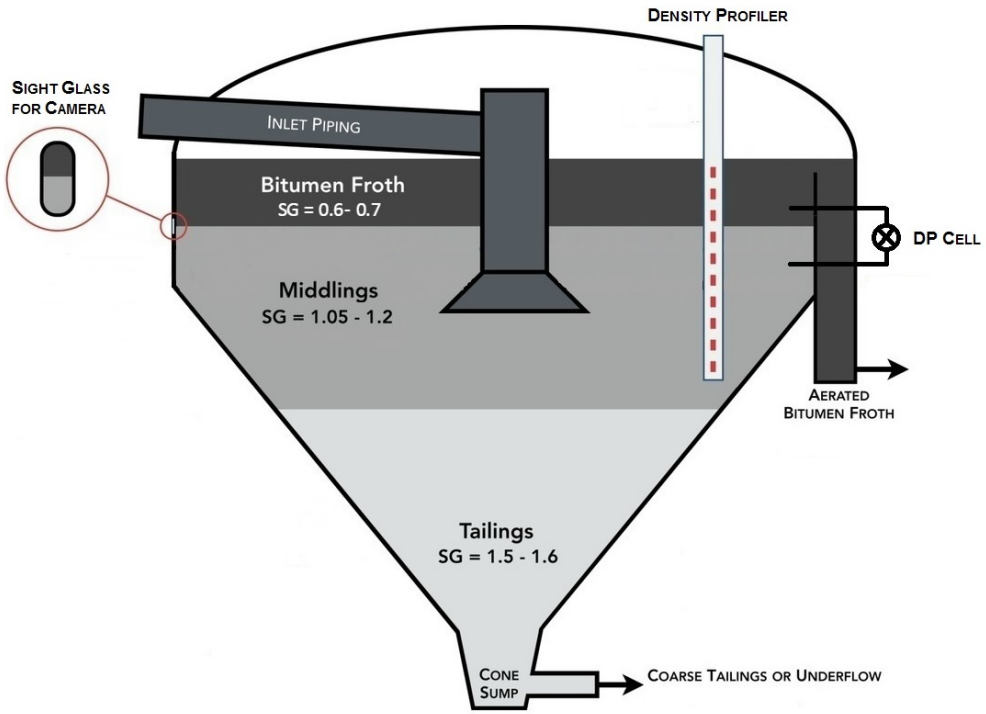


Figure 1.1: *The detailed illustration of primary separation vessel with interface level detectors.*

The maximum of bitumen recovery for the entire process depends on a number of process variables. It has been shown that the interface level between froth and middling layers plays a significant role to overall bitumen recovery [1, 2, 3]. The interface should be maintained at an optimum level [2, 3] to provide both economical and environmental benefits [1]. To control and maintain the froth middling interface level, the accurate detection and estimation are important. Many detectors are installed on PSV; *differential pressure cell* (DP Cell), density profiler, and camera are three types of detectors that are most commonly used in industry currently. They are also indicated on the schematic of PSV in Figure 1.1. In industry, the froth middling interface level is usually measured by a combination of these three types of detectors

because each of them has advantage and disadvantage.

The main advantages of conventional DP cells are: easiness in installation, easiness in data collection, and inexpensiveness in the cost. Whereas, the level detected by DP cells is usually inaccurate because the density of mixture is difficult to estimate. The density profiler is vertically installed inside of the PSV as shown in Figure 1.1. It directly measures the densities at different horizontal levels. The density profiler is selected as level detector since all three layers are formed due to density difference. In theory, density profile should have a sharp change at the interface of two layers by assuming complete mixing in each layer. However, the performance of the density profiler is not always ideal. In addition, the density profiler, which is powered by nuclear, is usually expensive to install and maintain. For camera interface level detectors, the main advantage is that cameras can visually monitor level changes via a sight glass, which is emphasized in the Figure 1.1. Camera is usually the best interface level detector, however, it is not always in good performance [1]. Under some specific conditions, for example, viscous bitumen and blurs are adhered on the sight glass or the interface level is out of sight glass range, camera based level detection usually fails. This is one of main challenges for froth middling interface level detection and estimation in industrial currently.

1.2 Motivation and Objective

As introduced in the industrial background, camera based PSV froth middling interface level detection usually has the best performance but not always. As a result, we are motivated to develop a new approach to improve the accuracy of camera based interface level detection under various conditions, which is the image processing technique. *Markov random field* (MRF) is the most commonly used theory in the area of image analysis [4]. Consequently, the objective of this thesis is to develop a new approach for camera based interface level detection and estimation using Markov random field based image processing technique.

1.3 Experimental Design for Simulating PSV Froth Middling Interface

To experimentally study the PSV froth middling interface based on image processing, we proposed to use the immiscibility between canola oil and distilled water to form an interface. A flow-loop equipment was designed to allow two process liquids, canola oil and distilled water, to form an interface level as a simulation of the interface between froth and middling layers of conventional primary separation vessel. Images of interface could be captured using an on-line camera.

1.3.1 Experimental Setup

Figure 1.2 shows the schematic diagram of the experimental setup. As shown in the figure, there is a main vessel and two holding tanks. The main vessel represents the PSV and two immiscible process liquids, canola oil and distilled water, are filled in the vessel to form an interface. Since canola oil has relative low density, it represents the froth layer in conventional PSV and distilled water represents the middling layer. Two discharge pumps are used to drain oil and water from main vessel into two holding tanks respectively. Other two recycle pumps are used to pump two liquids from holding tanks and recycle them as the input of the main vessel. Two differential pressure sensors (DP Cell) are installed on the main vessel: one at the bottom to measure the total pressure from the bottom of tank up to the liquid surface on the top, and another one installed at 40.64 cm height from the bottom to measure the pressure from the height of 40.64 cm to top liquid surface. Two *liquid level controllers* (LLC) are used to control two discharge pumps and maintain the total liquid height as well as interface height at optimal levels. Two flow meters measure the input flow rates of oil and water respectively before mixing. Two *flow rate controllers* (FRC) are employed to control the recycle pumps in order to obtain images of interface under different operation conditions. The OPTO22 system and LabView program are used to control the process and record the data as a function of time. An online camera is placed beside the main tank at the height around interface level to capture images of

interface. The *Point Grey* software was used to convert and store the numerical data of images. The photographic picture of real experimental setup is shown in Figure 1.3.

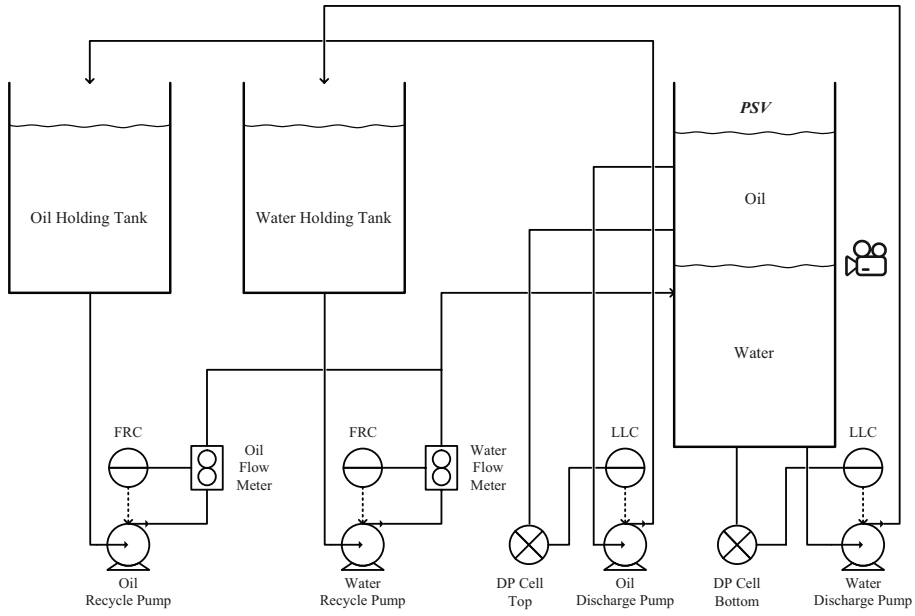


Figure 1.2: *The process flow diagram of experimental setup. FRC represents flow rate controller, and LLC represents liquid level controller.*

1.3.2 First Principle Model of the Experiment

As shown in the process flow diagram in Figure 1.2, the top DP Cell measures the pressure at a level above the interface, and the bottom DP Cell measures the total pressure at the bottom of tank. As a result, the height of interface level can be estimated using first principle based on two pressure measurements. The notations of different liquid heights are needed to be specified beforehand. Table 1.1 summarizes all liquid heights that will be used in computing interface level height.

It is important to note that h_1 is the measurement of total height of liquid in the PSV. Because the measurement of DP Cell is in $in \cdot H_2O$ (inch water) and the liquid in the tank is not only water, the value of h_1 is the equivalent height of same amount

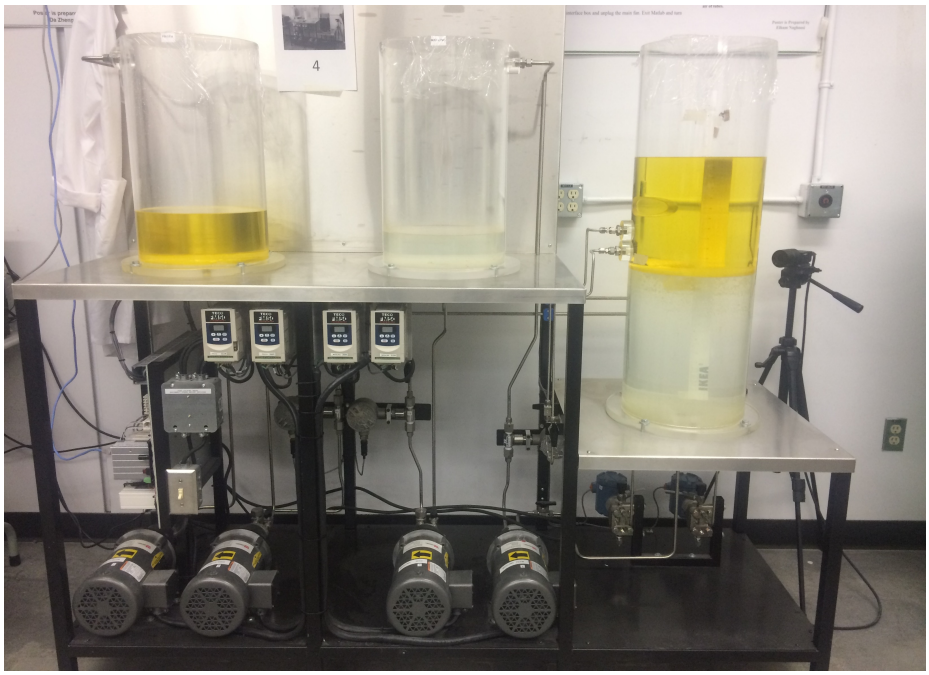


Figure 1.3: *The photographic picture of experimental setup of PSV interface simulation. The yellowish liquid is the canola oil, and the transparent liquid is water. The black camera stands on a tripod and been placed beside the main tank.*

liquid of distilled water. Since the density of oil is smaller than water, h_1 should be less than the real total liquid height. Same as h_2, the value of the measurement also should be less than the real level difference between the top surface and the level where the top DP Cell installed. The constant height, h_{con}, is the distance from the bottom of the tank to the level where the top DP Cell is installed, indeed equal to 40.64 cm.

Based on the pressure difference between the measurements of two DP Cells, we can obtain the following equation:

$$\rho_w g(h_1 - h_2) = \rho_w g h_i + \rho_o g(h_{con} - h_i) \tag{1.1}$$

where ρ_w and ρ_o are the densities of water and oil respectively, and g is the acceleration due to gravity. Rearrange equation (1.1) and the interface level can be directly calculated as:

$$h_i = \frac{\rho_w(h_1 - h_2) - \rho_o h_{con}}{\rho_w - \rho_o} \tag{1.2}$$

Table 1.1: *Summary of liquid heights in first principle model.*

Symbol	Description
h_{con}	constant height, 40.64 cm
h_i	interface height
h_1	bottom DP cell reading
h_2	top DP cell reading

A key assumption that was made for this first principle model is perfect mixing and ideal separation. And to make the above first principle calculation valid, the interface level has to be maintained below the level where the top DP Cell installed. In other words, interface level must be below the height of 40.64 cm.

This first principle estimation will be used as benchmark to compare with the interface level estimation using image processing technique.

1.4 Thesis Contributions

The main contribution of this thesis is the development of an approach for interface level detection and estimation using Markov random field based image processing technique. Besides the method of estimating interface, a number of novelties on MRF based image segmentation are proposed as well. Specifically, the detailed contributions are summarized as follows:

1. Developing an approach for interface level detection and estimation based on image segmentation on raw interface image. The interface level is estimated based on the vertical profile of averaged pixel values for all horizontal levels of segmented image.

2. Proposing an extended *iterated conditional modes* (ICM) algorithm to improve the initial estimation of traditional ICM algorithm using *k-means* clustering. This enhancement will reduce the computational cost compared with the max-

imum likelihood estimation as the initial estimation in traditional ICM algorithm.

3. Designing a customized neighborhood system in order to reduce the variance of interface level estimation.

4. Proposing *mean field approximation* (MFA) in approximating the Gibbs distribution of Markov random field in aim to estimate model parameters using *expectation maximization* (EM) algorithm under a single framework.

5. Developing an approach on dynamic prediction of image segmentation and interface level. This approach will combine *spatial temporal Markov random field* (ST-MRF) and auto-logistic model to achieve the prediction of segmented image and interface level in future time.

1.5 Thesis Outline

The reminder of this thesis is organized as follows:

Chapter 2 introduces Markov random field and optimization framework of MRF based image processing. The extended ICM algorithm and proposed customized neighborhood system are applied by performing supervised image segmentation on experimental images of oil and water interface. The proposed approach of interface level detection and estimation is applied and results are compared with those obtained using first principles.

Chapter 3 presents an unsupervised image segmentation on the same experimental images of interface. Gaussian mixture distribution is assumed on raw pixel values, and mean field approximation is applied on Gibbs distribution of latent Markov random variables. Model parameters are estimated using EM algorithm under a single framework. In this chapter, the estimations of interface level based on unsupervised image segmentation are compared with the results obtained using supervised image segmentation in chapter 2 as well as first principle model.

In chapter 4, a dynamic approach of predicting image segmentation and interface level is developed. First, a predicted image can be obtained using a modified random walk model. Then, the prediction of image segmentation is predicted using a spatial temporal Markov random field based auto-logistic model. Finally, the predictions of image segmentation are compared with the segmented images obtained in chapter 3, and interface heights are compared with the actual heights obtained using first principles.

Chapter 5 draws the conclusions of thesis and provides some future work directions.

Chapter 2

Markov Random Field based Supervised Image Segmentation with Extended ICM Algorithm and Customized Neighborhood System

2.1 Introduction

Image processing is now broadly applied in various areas such as computer vision, medical imaging, and various industrial applications. In general, image analysis includes de-noising, restoration, segmentation, surface reconstruction, edge detection, texture analysis, visual integration and many other applications [4]. Markov random field (MRF) theory provides a feasible way to study and model context dependent entities such as image pixels [4]. According to Bishop [5], a Markov random field, also known as Markov network, is defined to "have a set of nodes each of which corresponds to a variable as well as a set of links each of which connects a pair of nodes". Unlike the Bayesian network, the dependence between each pair of random variables in Markov random field is undirected, therefore, MRF is also called undirected graphical model. In this and the following two chapters, the Markov random field theory will be used as the basis of image analysis.

Image segmentation is one of the applications of image processing using Markov random field theory. Image segmentation can be defined as a process of partitioning

an image into several segments in the aim to simplify the representation of the image [4]. For both industrial operation and laboratory experiment, the liquid interface is a boundary between two different media. Therefore, images which are captured using online camera contain two different media and an interface in between. Image segmentation technique can be applied for those images. Two segments can be partitioned for each image and the interface level can be determined based on the segmented image.

In this chapter, the main novelty is the proposed Markov random field based image segmentation approach for oil and water interface level detection and estimation using the images captured by online camera. The raw image will be segmented to a binary image, and then the interface level can be estimated based on the vertical pixel profile of segmented image. To improve the algorithm for MRF based image segmentation, an extended version of *iterated conditional modes* (ICM) algorithm is proposed by introducing the *k-means* clustering as the initial estimation in the aim to reduce the computational cost. A customized neighborhood system is also proposed to improve the accuracy of interface level estimation and reduce the estimation variance. Since all parameters in the mathematical models in this chapter are assumed known, the proposed approach belongs to supervised image segmentation problem.

The remainder of this chapter is organized as follows: section 2.2 and 2.3 introduce the Markov random field as well as optimization approach in detail. The extended ICM algorithm, customized neighborhood system and the interface level estimation approach are explained in section 2.4. The detailed image segmentation results, the procedure of interface level estimation, as well as the discussion on extended ICM algorithm and customized neighborhood system are shown in section 2.5, and finally conclusions are drawn in section 2.6.

2.2 Introduction of Markov Random Field

2.2.1 Markov Random Field and Neighborhood System

Literally, Markov random field, also known as Markov network or undirected graphical model, is a set of random variables that follow Markov property. In time sequence process, Markov property means that the conditional probability distribution of future state depends only on the present state. The sequence of states having such property is also called Markov chain or Markov process. However, having no time-series information in static image processing, Markov property, in this case, refers to the spatial dependence between each pixel and its neighbors. Therefore, in order to define the Markov random field, a term so called *neighborhood system* needs to be specified in advance.

The neighborhood system defines a way that how image pixels are related. In image processing, each image lattice is called a *site* and denoted as s. The set that contains all sites of an image can be denoted as $S = \{s = (i, j) \mid 1 \leq i \leq H, 1 \leq j \leq W, i, j, H, W \in I\}$, where H and W are the height and width of image in pixels.

According to Li [4], every site in S is related to other sites via neighborhood system. A neighborhood system for S can be defined as:

$$N = \{N_s \mid \forall s \in S\} \tag{2.1}$$

where N_s is the set of sites neighboring the site s. The neighboring relationship has the following two properties:

i A site is not neighboring to itself, $s \notin N_s$

ii The neighboring relationship is mutual, $s \in N_{s'} \Longleftrightarrow s' \in N_s$

In the image processing problem, pixels are regular lattices, the set of neighbors of s is defined as the set of pixels within a circle centered at s and radius of $\sqrt{r}$.

$$N_s = \{s' \in S \mid [\text{dist}(s', s)]^2 \leqslant r, s' \neq s\} \tag{2.2}$$

where $\text{dist}(A, B)$ denotes the Euclidean distance between A and B, and r is the order of the neighborhood system. In the first-order neighborhood system, also called the

4-neighbor system, every interior pixel has 4 neighbors, as shown in Figure 2.1 where pixel i, j is the site considered and 4 sites labeled as 1 are first order neighbors. Similarly, the second-order neighborhood system, also called 8-neighbor system, is shown in Figure 2.1 (b). Other than the neighbors in first order system, there are four additional neighbors labeled as 2 in 8-neighbor system.

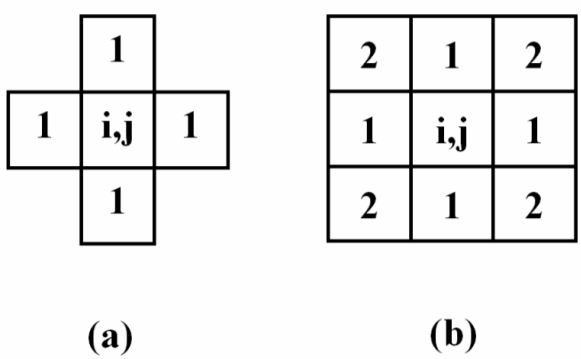


Figure 2.1: *Neighborhood system on a lattice of regular sites. (a) first-order neighborhood system, (b) second-order neighborhood system.*

A *clique* c for (S, N) is defined as a subset of sites in S which consists all possible single and multiple sites. The single-site clique contains one site and the multiple-site clique consists of more than one site. As shown in Figure 2.2, single-site clique and pair-site cliques are shown in (a) and (b) for first-order neighborhood system. The diagonal pair-site cliques in (c), triple-site cliques in (d) as well as quadruple-site cliques in (e) along with (a) and (b) are all possible cliques in second-order neighborhood system.

Having the neighborhood system being specified, the Markov random filed can be formulated in the following ways: F is said to be a Markov random field on S with respect to a neighborhood system N if and only if [4]:

$$P(f) > 0, \forall f \in F \tag{2.3}$$

$$P(f_s \mid f_{S-\{s\}}) = P(f_s \mid f_{N_s}) \tag{2.4}$$

where F is the set of random variables where $F = \{F_s \mid s \in S\}$, and f is the label of each random variable. Pixels are considered as random variables in image analysis.

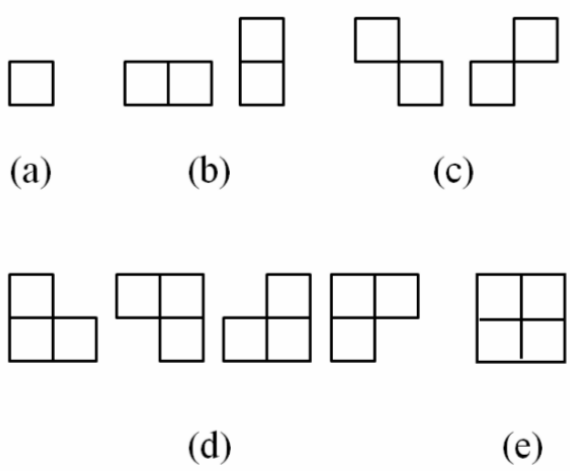


Figure 2.2: *Cliques of a regular site and its 1^{st} and 2^{nd} neighborhood system. (a) and (b) for 1^{st} order system only and (a) $\sim$ (e) for 2^{nd} order system.*

Therefore, in equation (2.4), $f_{S-\{s\}}$ is the set of pixel labels of set difference, f_{N_s} is the set of labels in the neighbors of s and

$$f_{N_s} = \{f_{s'} \mid s' \in N_s\} \tag{2.5}$$

In the above definition, equation (2.3) and (2.4) explain the positivity and the Markovianity of the random field respectively.

The images captured by camera are often corrupted by the noise. The Markov random field is usually assumed as latent or underlying variables in image processing. Each pixel has a corresponding observation, denoted as d_s, which is assumed to be independently observed. The observations of the entire image contain all pixels' observed intensities $d = \{d_s \mid s \in S\}$. In image processing, the labels for both random variable and observation take discrete integers. The observed pixels usually take values between 0 and 255, and the labels of random variable can be different from various situations or problems.

Figure2.3 shows an illustration of a recognition of Markov random field and corresponding observations. All white nodes denoted as f's represent random variables in MRF, and the corresponding observations are those grey nodes denoted as d's. The links between f's represent the dependence of random variables in the field, and each observation is only dependent on its corresponding random variable. To explain the

neighborhood system in this illustration, by considering the variable $f_{i,j}$, its first-order neighbors are $f_{i,j-1}$, $f_{i-1,j}$, $f_{i,j+1}$, and $f_{i+1,j}$. The second-order neighborhood system includes all eight neighbors around it, which are $\{f_{i-1,j-1}$, $f_{i-1,j}$, $f_{i-1,j+1}$, $f_{i,j-1}$, $f_{i,j+1}$, $f_{i+1,j-1}$, $f_{i+1,j}$, $f_{i+1,j+1}\}$.

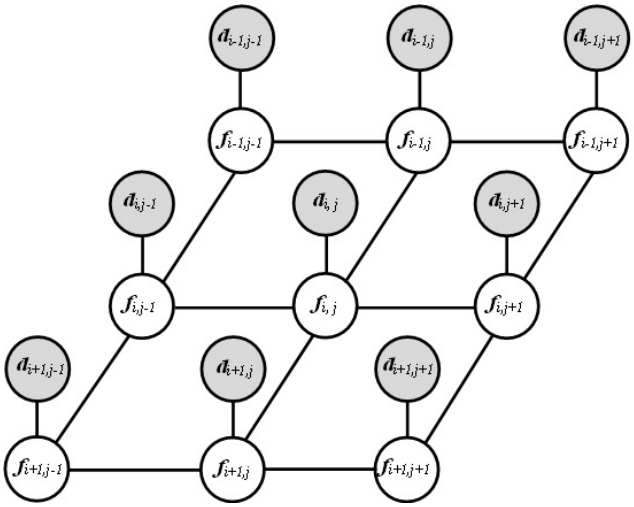


Figure 2.3: *An illustration of Markov random field and corresponding observations. White nodes are random variables, grey nodes are observations.*

2.2.2 Gibbs Distribution and Markov-Gibbs Equivalence

A set of random variables F is said to be a *Gibbs random field* (GRF) on S with respect to N if and only if its configurations obey a *Gibbs distribution* [4]. A Gibbs distribution, which is also called Boltzmann distribution, has its probability density function taking the form

$$P(F = f) = \frac{1}{Z} e^{-\frac{1}{T}U(f)} \tag{2.6}$$

where Z is a normalizing constant called *partition function* that can take form of

$$Z = \sum_{f \in F} e^{-\frac{1}{T}U(f)} \tag{2.7}$$

T is a constant that is usually assumed as 1, $U(f)$ is the *energy function*. The energy is the sum of *clique potentials* over all possible cliques and can be expressed as:

$$U(f) = \sum_{c \in C} V_c(f) \tag{2.8}$$

where $V_c(f)$ is clique potential. C is the set of all cliques.

A Markov random field is featured by the local property, namely the Markovianity. A Gibbs random field is characterized by its global property, namely Gibbs distribution. According to Hammersley-Clifford theorem, the local property of MRF and global property of GRF are equivalent. Literally, the theorem states that F is an MRF on S w.r.t N if and only if F is a GRF on S w.r.t N [4]. There are many proofs for the theorem, one of which can be found in [4].

2.3 Modeling of Images using MRF

In section 2.2, the preliminaries of Markov random field have been introduced. The idea of Gibbs-Markov equivalence is important because it shows the prior knowledge of MRF to obey Gibbs distribution. The next step is to develop a optimization framework to model MRF and to achieve image segmentation. To do that, we need to build an observation model beforehand.

2.3.1 Observation Model

Recall the observation given as $d = \{d_s \mid s \in S\}$ where $S = \{s = (i, j) \mid 1 \leq i \leq H, 1 \leq j \leq W\}$ is usually a rectangular array of pixel values in the size of $H \times W$, and every pixel takes a value in the observation's label set. An observation d can be considered as a transformed or degraded version of an MRF realization f [4]. The transformation may include geometric transformation and blurring, so that the transformed images are often unclear. The degradation is usually because of random factors such as noise [4]. Practically, the observation model of image processing has no blurring, linear transformation as well as independent and identically distributed (i.i.d) Gaussian noise. Simply, the observation model can be expressed as:

$$d = f + e \tag{2.9}$$

where $e \sim \mathcal{N}(0, \sigma^2)$. As a result, the conditional probability distribution of d given f is

$$P(d \mid f) = \frac{1}{\prod_s^m \sqrt{2\pi\sigma^2}} e^{-U(d|f)} \tag{2.10}$$

where the energy function for likelihood distribution is defined as:

$$U(d \mid f) = \sum_{s \in S} (f_s - d_s)^2 / 2\sigma^2 \tag{2.11}$$

and $m = H \times W$ is the number of total pixels [4].

2.3.2 MAP-MRF Framework

Bayesian framework helps to obtain statistical inferences, synthesiszing maximum *a posteriori* (MAP) solution, incorporating the prior informations. It has been found in many applications [4]. In 1984, Geman and Geman [6] first formulated statistical image analysis problems based on Markov random field using maximum *a posteriori* solution. It has been called MAP-MRF framework.

The objective of MAP solution is to maximize the posterior probability, which can be represented as:

$$\arg\max_{f \in F} P(f \mid d) \tag{2.12}$$

According to the Bayes rule, the posterior probability can be computed by using the following equation:

$$P(f \mid d) = \frac{P(f, d)}{P(d)} = \frac{P(d \mid f) P(f)}{P(d)} \tag{2.13}$$

where $P(f, d)$ is the joint probability distribution, $P(d \mid f)$ is the conditional probability density function (p.d.f) of the observation d, also called the likelihood function of f for d, $P(f)$ is the prior probability of f, and finally $P(d)$ is the density of observation d, which is a constant when the observation is given. Therefore, the posterior probability is proportional to the product of likelihood function and the prior probability, which is

$$P(f \mid d) \propto P(d \mid f) P(f) \tag{2.14}$$

As a result, the objective of MAP estimation can be rewritten as:

$$\arg\max_{f \in F} \{P(d \mid f) P(f)\} \tag{2.15}$$

In the image analysis problem based on MRF, $P(f \mid d)$ is indeed the posterior probability distribution of an MRF. According to Markov-Gibbs equivalence, the prior probability follows a Gibbs distribution

$$P(f) = \frac{1}{Z} e^{-U(f)} \tag{2.16}$$

where the parameter T in equation (2.6) is assumed as 1 and $U(f)$ is called *prior energy* function. Based on the observation model mentioned in section 2.3.1, the likelihood probability can be found in equation (2.10). By substituting equations (2.10) and (2.16) into equation (2.14), it can be concluded as:

$$P(f \mid d) \propto e^{-U(d|f)} e^{-U(f)} \tag{2.17}$$

Consequently, the posterior probability can be written as:

$$P(f \mid d) \propto e^{-U(f|d)} \tag{2.18}$$

where

$$U(f \mid d) = U(d \mid f) + U(f) \tag{2.19}$$

is defined as *posterior energy* and $U(d \mid f)$ is called *likelihood energy*. Therefore, to maximize the posterior probability $P(f \mid d)$ is equivalent to minimizing the posterior energy function $U(f \mid d)$, which is

$$\arg\min_{f} U(f \mid d) \tag{2.20}$$

The only parameter that needs to be determined is the variance σ^2 of the noise distribution. Once σ^2 is determined, the minimum of posterior energy $U(f \mid d)$ based on MAP-MRF solution can be completely achieved.

According to Li [4], the procedure of the MAP-MRF approach is summarized as the following steps:

1. Define a neighborhood system N on S and the set of cliques C for N.
2. Define the prior clique potentials $V_c(f)$ and then give the prior energy $U(f)$.
3. Derive the likelihood energy $U(d \mid f)$.
4. Sum $U(d \mid f)$ and $U(f)$ up to yield the posterior energy $U(f \mid d)$.
5. Minimize the posterior energy $U(f \mid d)$ to find the MAP estimation.

2.4 Proposed Approach

There are many existing optimization approaches to obtain the minimum of energy function in MAP-MRF framework. Due to the complexity of interactions between labels, optimal solutions are usually computed by using iterative search techniques [4]. The minimization problem leads to local minimization or global minimization. For a given neighborhood system $N = \{N(f) \mid f \in F\}$, a configuration of MRF reaches *local minimum* of energy U w.r.t. N if

$$U(f') \leqslant U(f) \quad \forall f \in N(f') \tag{2.21}$$

A local minimum is considered as a *global minimum* if the neighborhood is defined to include all other configurations, namely

$$N(f') = \{f \mid \forall f \in F, f \neq f'\} \tag{2.22}$$

Generally, local search is the basis for most minimization algorithms [4]. ICM algorithm, known as *iterated conditional modes* (ICM), is the algorithm that is most commonly used in image analysis based on MRF. However, it suffers from some disadvantages, for instance, more iterations to convergence leading to large computation time. In real time application such as PSV image detection, it is desirable to have an algorithm that has smaller computation time. Further, it is necessary to have the estimation with low variability. Hence, to address these lacunae, in this section, an extended version of ICM algorithm will be proposed by combining with the customized neighborhood system. The approach to detect and estimate interface level will also be proposed in this section.

2.4.1 Extended ICM Algorithm

The basic idea of the ICM algorithm is to maximize local conditional probabilities sequentially since it is difficult to maximize the joint probability of an MRF [6]. Given the observation d and all other labels except site s in k's iteration $f^{(k)}_{S-\{s\}}$, the algorithm iteratively updates each $f^{(k)}_s$ into $f^{(k+1)}_s$ by maximizing $P(f_s \mid d, f^{(k)}_{S-\{s\}})$, which is the conditional posterior probability w.r.t. f_s [4].

There are two assumptions made in computing the conditional posterior probability $P(f_s \mid d, f^{(k)}_{S-\{s\}})$:

1. Observations are conditionally independent given f. Each d_s depends only on f_s and has the same conditional density function $P(d_s \mid f_s)$. As a result,

$$P(d \mid f) = \prod_s P(d_s \mid f_s) \tag{2.23}$$

2. Each f_s depends on the labels in the given neighborhood system N_s.

According to equation (2.14) and based on these two assumptions, the conditional posterior probability w.r.t. f_s in k's iteration is:

$$P(f_s \mid d, f^{(k)}_{S-\{s\}}) \propto P(d_s \mid f_s) P(f_s \mid f^{(k)}_{N_s}) \tag{2.24}$$

Based on MAP-MRF framework in section 2.3.2, maximizing the conditional posterior probability is equivalent to minimizing the corresponding posterior energy function. Therefore, $f_s^{(k+1)}$ can be iteratively updated using the rule:

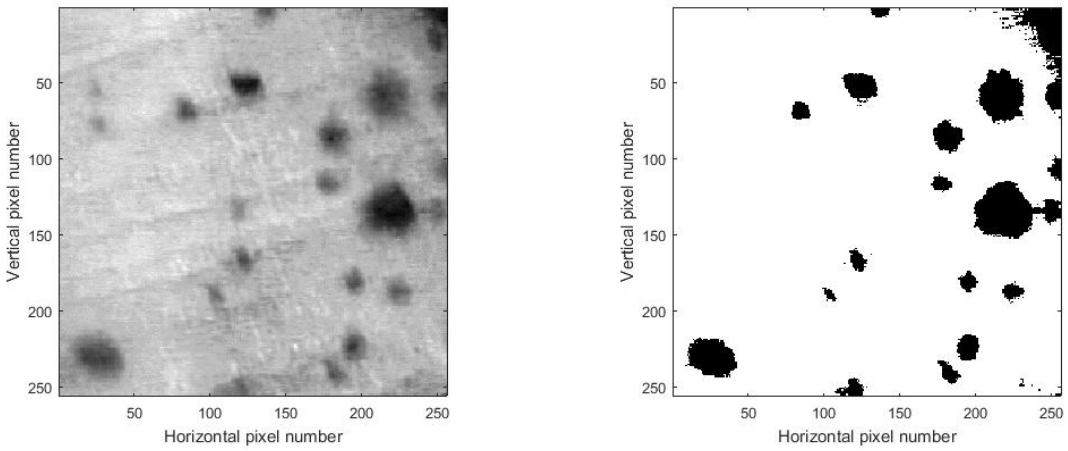
$$f_s^{(k+1)} \longleftarrow \arg\min_{f_s} U(f_s \mid d_s, f^{(k)}_{N_s}) \tag{2.25}$$

where

$$U(f_s \mid d_s, f^{(k)}_{N_s}) = U(d_s \mid f_s) + \sum_{s' \in N_s} U(f_s \mid f^{(k)}_{s'}) \tag{2.26}$$

The likelihood energy of each pixel $U(d_s \mid f_s)$ is also called *potential of observation*, and $\sum_{s' \in N_s} U(f_s \mid f^{(k)}_{s'})$ can be called *potential of neighborhood*.

In image segmentation problem, the objective is to partition the original image into a segmented image that clearly shows the contrast in regions. Therefore, there must exist the difference in pixel intensity in the original image. For example, Figure 2.4a shows an observed image, which has white background with some black areas. The segmentation result is shown to the right, which highlights the dark region and sets the background in pure white. As a result, unlike image restoration problem, where noise distribution is assumed as one i.i.d Gaussian, the noise of image segmentation problem usually has multiple distributions. In the extended ICM algorithm, we assume the



(a) *Original image* **(b)** *Segmented image*

Figure 2.4: *The demonstration of image segmentation based on extended ICM algorithm.*

noise is Gaussian distributed with identical variance but different means. Since the image segmentation in this chapter is supervised, all parameters are known.

According to Li [4], the initial estimator $f^{(0)}$ has the significant impact on either the final outcome or computational cost in the use of ICM algorithm. In the problem of image segmentation, since there are multiple means in the observed image pixel values, a *k-means clustering* method is proposed to obtain the initial estimation as the extension of ICM algorithm. It can be shown that the segmentation result converges rapidly.

The algorithm can be summarized as:

1. Perform k-means clustering to obtain the initial estimation $f^{(0)}$.
2. For each pixel, calculate potentials of observation for different means as well as the potential of neighborhood. Using equation (2.26), the posterior energies can be computed by summing up potentials of observation and potential of neighborhood.
3. Using equation (2.25), we can conclude the MRF recognition by selecting the minimum posterior energy among those calculated in step 2 based on different means.

4. Compute the percentage of pixels that are changed between the current and previous iterations.
5. Repeat steps 2 to 4 until pixel change is less than a given threshold, 0.01%.
6. Exit the iteration and the segmentation result can be obtained.

2.4.2 Customized Neighborhood System

In the MRF based image processing, first and second order neighborhood system, which has been introduced in section 2.2.1, are the most commonly used. However, neighborhood system can be modified based on different situations. For the liquid interface level estimation, a customized neighborhood system was proposed to aim better segmentation results. Figure 2.5 shows the proposed customized neighborhood system. All lattices labeled by ***n*** are considered as neighbors of the center lattice labeled by **i,j**. We can see that this neighborhood system adds two more pixels in horizontal direction by comparing with the traditional first order system. The motivation of adding two more neighbors horizontally is because the interface is ideally leveled and pixels should have more chance to be the same in horizontal direction. By using this new designed neighborhood system instead of first order system, segmentation results are expected to improve by showing a sharper change at interface. The detailed results and discussion will be provided in section 2.5.2.

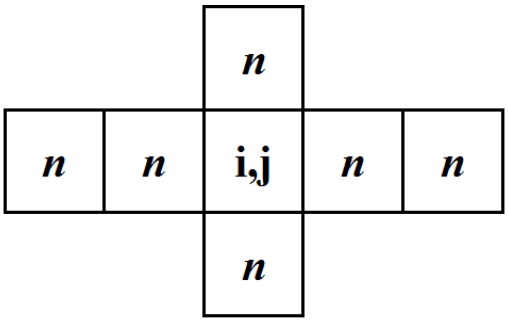


Figure 2.5: *The customized neighborhood system.*

2.4.3 Interface Level Estimation Based on Vertical Profile

By performing image segmentation using extended ICM algorithm with customized neighborhood system, the interface level can be estimated based on the vertical profile of segmented image. The reason why to use vertical profile is same as the idea of developing the customized neighborhood system; that is, interface level between two immiscible media is ideally horizontal. Each vertical profile of the pixels in segmented image should be similar. Therefore, an averaged vertical profile can be calculated. From there, the level of interface can be determined using the averaged vertical profile. The detailed procedure is explained in section 2.5.1.

2.5 Results and Discussion

2.5.1 Segmentation Results and Interface Level Estimation

All image processing results are obtained using MATLAB. Raw images captured using camera have 600 pixels in both horizontal and vertical directions. In MATLAB, the raw image, which is the observation, is stored by a matrix having the same size, namely 600×600. All values in this matrix are integers in the range of $[0, 255]$. Each value represents the pixel intensity of the corresponding pixel. In our experiment, all raw images are in grey scale. In general, the lower pixel intensities represent darker region in the image and higher pixel values represent lighter area.

Figure 2.6 shows the raw image of the experimentally simulated interface level captured using online camera. A clear contrast can be seen such as lighter color is shown on the top and darker color is shown at the bottom. The lighter region represents oil due to the lower density and the darker region represents water. There is a froth region in between, which is the transient region between oil and water and the interface level is a level in this region. Based on this raw image, our goal is to partition this grey scale image into a binary segmented image so that the boundaries of that froth region and interface level can be estimated from there.

By applying the proposed extended ICM algorithm, the black-white binary segmented image can be obtained and shown in Figure 2.7. The white region on the

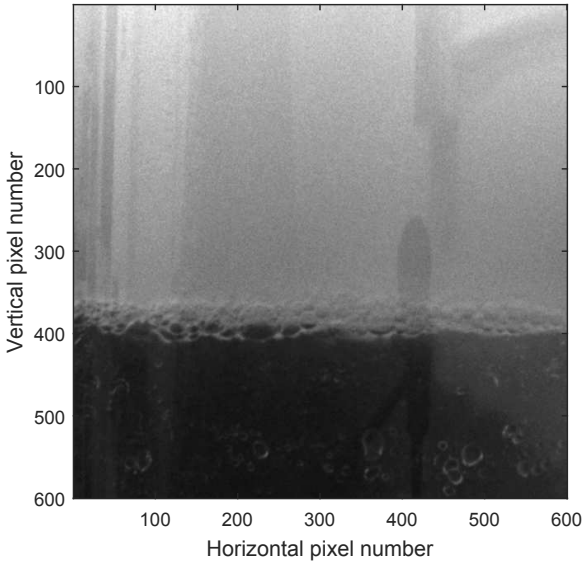


Figure 2.6: *Raw image of interface level captured by camera. Whiter for oil and darker for water. Image was obtained when total flow rate in was* $2L/min$.

top of the figure represents oil and the black region represents water. In other words, image segmentation can also be interpreted as pixel classification. In this work, being segmentation problem, the pixels are classified to binary classes, that is, +1 and -1, corresponding to oil and water respectively, based on the observed pixel intensities and defined neighborhood system. By analyzing this segmentation result, there is a transient zone between pure oil and pure water, which corresponds to the mixing froth region in raw image. This could be due to the following reason: by assuming the ideal separation of oil and water, interface should be a horizontal level just in between two immiscible liquids. The liquid in each region should be homogeneous. However, no matter in real industrial application or in our lab-scale experiment, the ideal separation and homogeneous layered media is practically difficult. Mixture boundaries and interface level can be estimated by analyzing this transient zone.

Due to physical property of liquid mixing, the liquid composition is assumed homogeneous horizontally not vertically. Based on the approach proposed in estimating interface level, the averaged MRF recognition pixel label is calculated for each horizontal level, and a vertical profile of averaged pixel labels over all horizontal levels

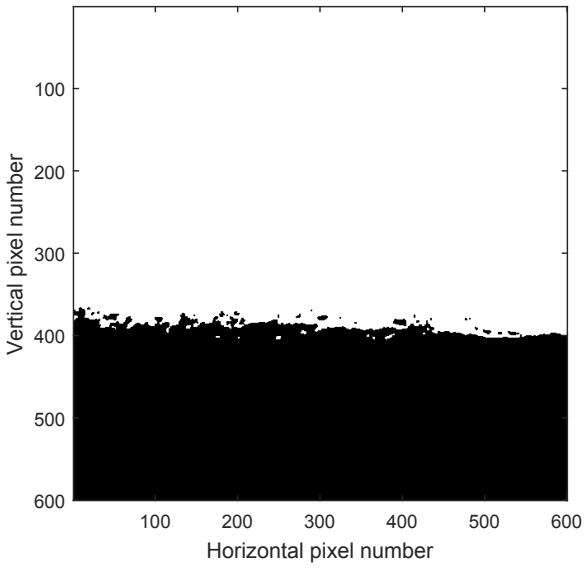
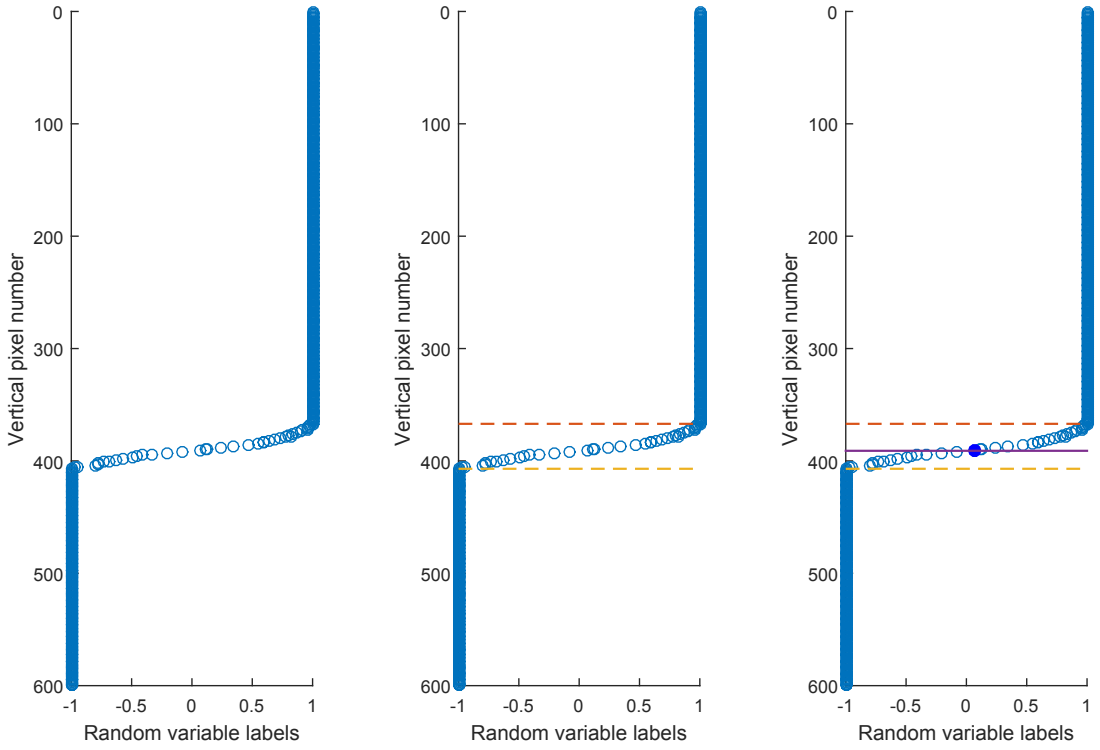


Figure 2.7: *Segmented image by applying MRF and extended ICM algorithm. White for oil and black for water.*

could be plotted as shown in Figure 2.8a. Based on this figure, we can see that the averaged pixel value keeps constant at $+1$ from the top until a level around 350. Then it gradually decreases as level gets lower. Finally, once the averaged pixel value goes down to -1, it remains constant again at -1 until the level reaches the bottom of the image. Based on this vertical profile, three sections can be clarified as $+1$, $(+1, -1)$, and -1. And the entire image can be correspondingly divided into three regions: pure oil, mixture, and pure water respectively. The boundaries of region separation can be estimated as the lowest none $+1$ level and the highest none -1 level. Figure 2.8b shows those two boundaries in the vertical MRF pixel profile plot.

Having determined the boundaries of mixture region, the next step is to identify the interface level between the two boundaries because the exact interface level estimation needs to be compared with the set point level of the interface in the conventional feedback control loop. In theory, when $+1$ is labeled for oil and -1 is labeled for water, the interface level should be at the position where averaged horizontal pixel is in between $+1$ and -1, that is zero. However, the segmented result from raw image does not always have exactly zero in averaged horizontal pixel labels. Therefore, we

consider the interface is at the level that the corresponding averaged horizontal pixel label that is closest to zero, that is, the absolute value reaching the minimum. By following this criterion, the interface level could be easily determined and be shown in Figure 2.8c. Dashed lines indicate the estimation of froth boundaries. The solid line in between indicates the level of interface, and the solid dot on the line is the averaged pixel value at interface level, namely, the one that is closest to zero.

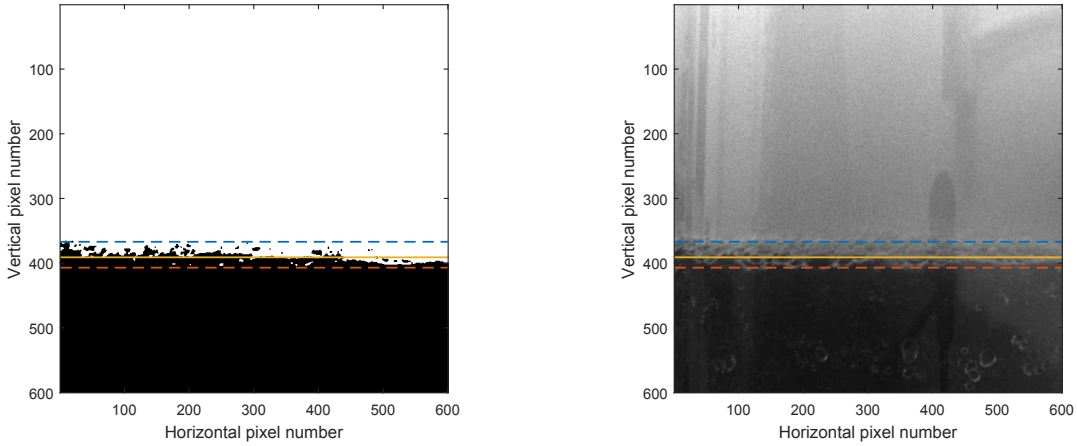


(a) *Pixel profile* **(b)** *Profile with boundaries* **(c)** *Profile with interface*

Figure 2.8: *The vertical profile of averaged MRF recognition pixel for interface level estimation. (a): vertical averaged MRF recognition pixel profile. (b): the vertical profile with the boundary indication. +1 for oil, -1 for water, every label other than* $+1$ *or* -1 *locates in the mixture region. Dashed lines are the upper and lower bound of mixture region. (c) the vertical pixel profile with indication of interface level estimation. Solid line indicates the interface level, blue solid dot is the corresponding averaged pixel label at interface.*

The boundaries of mixture transient region and the estimated interface level could

then be transferred onto both segmented and raw images and are shown in Figures 2.9a and 2.9b respectively. By looking at segmented image with boundaries and interface level, all pixels in froth zone fall in the region bounded by two dash lines. The solid line shows the level where interface is. It is also reasonable because the majority of pixels between upper bound and interface are in white and the majority of pixels between interface and lower bound are in black. By ignoring those shadows caused by reflecting lights, the boundaries and interface line exactly show the levels where changes occur.

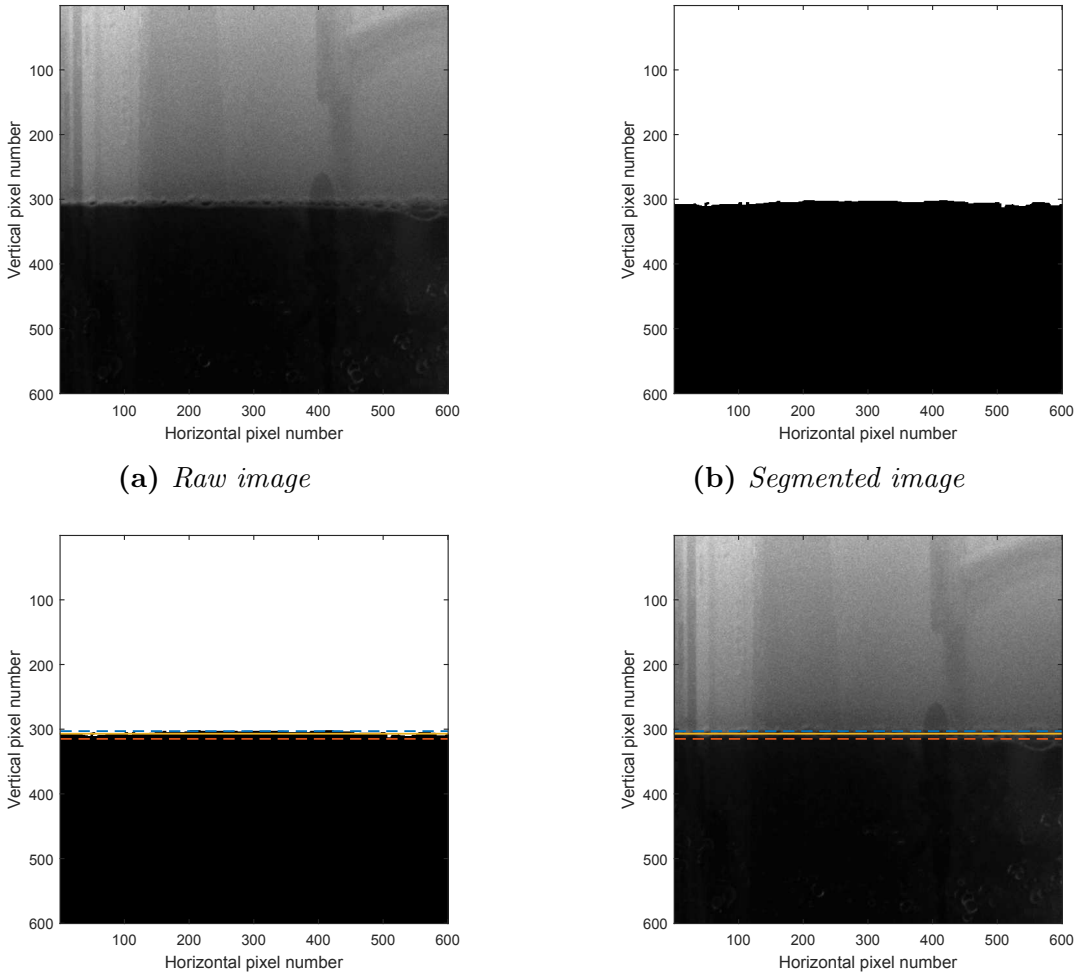


(a) *Segmented image* **(b)** *Raw image*

Figure 2.9: *Mixture boundary and interface level shown on segmented and raw images.*

The raw images captured using online camera are different under different operating conditions. We collected several raw images under the operations of different input flow rates. The raw image shown in Figure 2.6 was captured when the total input flow rate was $2L/min$. Figure 2.10a shows the raw image that was captured when the total input flow rate was $1.5L/min$. The same segmentation algorithm and analyzing approach were applied on this image. Figures 2.10b, 2.10c, and 2.10d show the segmentation result, segmented image with boundaries and interface, and the raw image with boundaries and interface respectively. It can be concluded that the transient region shown in Figure 2.10a is thinner than the one shown in Figure 2.6.

The reason for that is because the lower flow rate results less froth formed. Therefore, the transient zone is thinner and interface level is clearer in this case. The results of boundary and interface level estimation agree with both segmented and raw images as shown in Figures 2.10c, and 2.10d.



(a) *Raw image*

(b) *Segmented image*

(c) *Segmented image with interface*

(d) *Raw image with interface*

Figure 2.10: *Raw and segmented images with interface level indication for another operating condition. The raw image was captured when total input flow rate was* $1.5L/min$.

A total of 10 raw images were captured under various operating conditions. By using the proposed interface level estimation approach based on the vertical profile of averaged pixel values, the interface level can be estimated and represented by a

vertical pixel number. This vertical pixel number is in a range of 1 to 600, where 1 means the highest level on the image and 600 means the lowest level. The real heights of interface levels can be computed using the following calibration equation:

$$\bar{h_i} = -0.01 \cdot IF + 39 \tag{2.27}$$

where $\bar{h_i}$ is the real height estimated using image segmentation technique, and IF is the vertical pixel number of interface level. Table 2.1 tabulates the interface heights of 10 images estimated using first principle model in chapter 1 and the image segmentation technique. All errors are less than 1%, therefore, the interface estimations based on image processing agree with those first principle estimations.

Table 2.1: *The interface heights estimated using supervised image segmentation technique and corresponding errors as compared with the first principle estimation.*

Image Number	First Principle Estimation	Supervised Image Segmentation	
	Interface Height (cm)	Interface Height (cm)	% Error
1	35.04	35.09	0.8%
2	35.94	35.93	0.2%
3	35.98	36.01	0.5%
4	36.01	35.98	0.5%
5	35.99	35.95	0.7%
6	35.82	35.80	0.3%
7	35.80	35.78	0.3%
8	35.75	35.72	0.5%
9	35.71	35.70	0.2%
10	35.67	35.66	0.2%

2.5.2 Performance Improvements Owing to the Extended ICM Algorithm and Customized Neighborhood System

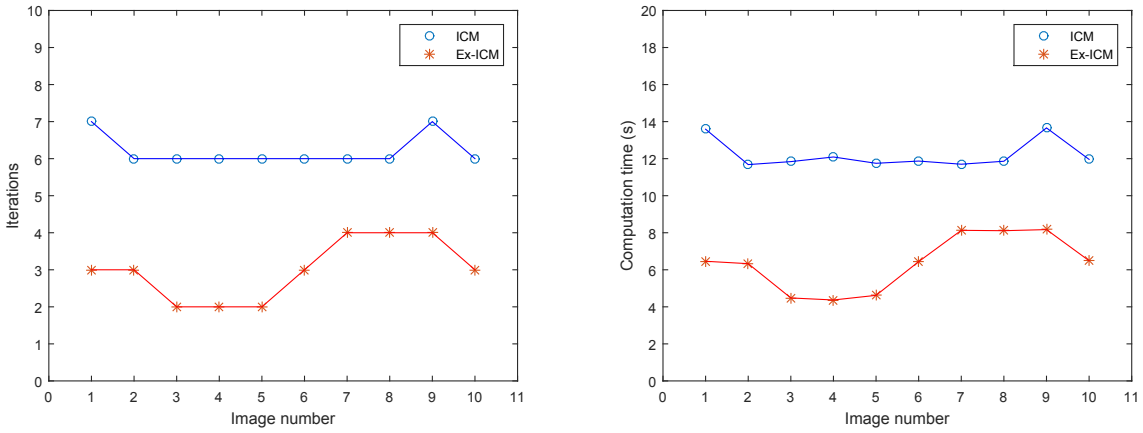
There are two main contributions in the proposed algorithm: one is to extend conventional ICM algorithm by combining k-means clustering as the initial guess; another one is to develop a customized neighborhood system in the case of interface level detection and estimation problem. To verify the efficiency of these two enhancements, the extended version is compared with original ICM algorithm in terms of iterations and computational time; the customized neighborhood system is compared with first order neighborhood system in terms of interface estimation variance.

Figures 2.11a and 2.11b show the comparison between ICM algorithm and our proposed extended version with respect to iterations and computational time respectively. Same 10 images were used to perform both algorithms. The horizontal axes indicate the corresponding image number. Both figures show that both iterations and computational time are significantly reduced by applying the extended ICM algorithm. The reason for this improvement is because of the more accurate initial guess. Therefore,using k-means clustering method based on observed image to obtain the initial estimation is an efficient way to improve the algorithm in terms of computational cost reduction. The numerical values of iterations and computational time that are plotted in Figures 2.11a and 2.11b are tabulated in Table A1 in appendix A.

As for the enhancement of customized neighborhood system, the comparison results are shown in terms of interface level estimation variance. The variance is defined as the summation of the squared difference between the averaged MRF pixel value at estimated interface level and every other averaged MRF pixel value in mixture transient region. It can be formulated as the following:

$$var = \sum_{\bar{f} \in (-1,+1)} (\bar{f} - \bar{f}_{IF})^2 \tag{2.28}$$

where $\bar{f}$ represents the averaged MRF pixel value shown in Figure 2.8, and $\bar{f}_{IF}$ is the averaged value at interface level and corresponds the solid dot shown in Figure 2.8c. According to the definition mentioned above, the lower variance indicates the thinner transient zone. Therefore, the lower variance is always desired.

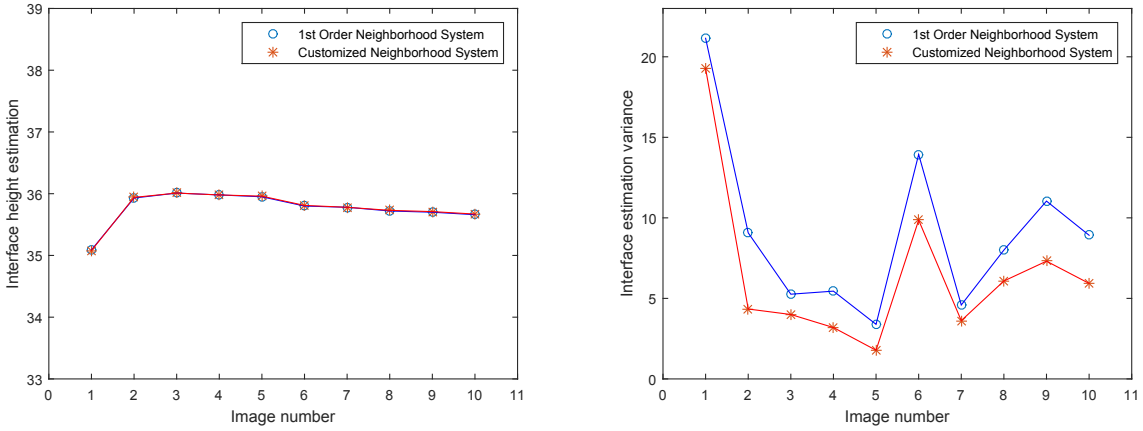


(a) *Iterations* **(b)** *Computational time*

Figure 2.11: *The comparison between ICM algorithm and Extended ICM algorithm with respect to computational cost.*

The same 10 images were used for analyzing the difference of interface height estimation and the variance when using first order neighborhood system and the proposed customized neighborhood system. Figure 2.12a shows the comparison of interface height estimation when using two neighborhood systems. It is obvious that interface estimations for all images can be considered to be the same, the differences are negligible. Figure 2.12b shows the comparison of interface estimation variance when using the first order and the proposed customized neighborhood system. Unlike the comparison of interface height estimation, the estimation variances when using customized neighborhood system for all 10 images are lower than the variances when using standard first order neighborhood system. It can be concluded that using the proposed customized neighborhood system not only has the accurate estimation of interface height, but also can reduce the estimation variance. All numerical values are tabulated in Table A2 in appendix A.

The percentage of improvement for both extended ICM algorithm and customized neighborhood system are shown in Table 2.2. The extended ICM algorithm mainly improve the estimation in terms of computational cost. The computational time was reduced from 30% to 64%, and 47.9% on average. The customized neighborhood system achieves the reduction of variance in interface level estimation. The percentage



(a) *Interface height estimation* **(b)** *Interface estimation variance*

Figure 2.12: *The comparison of interface heights estimation and corresponding variances of image segmentation results by using* 1^{st} *order neighborhood system and customized neighborhood system. The height estimations are similar, but the variances are lower when the customized neighborhood system is used.*

of interface variance varies from 9% up to 52%, and 31.6% on average.

Table 2.2: *The percentage improvement when using new proposed extended ICM algorithm and customized neighborhood system.*

Image Number	1	2	3	4	5	6	7	8	9	10
Ex-ICM	53%	46%	62%	64%	61%	46%	30%	32%	40%	46%
Cust. NS	9%	52%	24%	42%	48%	29%	21%	24%	34%	34%

2.6 Conclusions

In this chapter, a Markov random field based supervised image segmentation technique was proposed for addressing primary separation vessel froth middling interface level detection and estimation. An approach of interface level detection and estimation was proposed based on the vertical profile of averaged pixel values of segmented image. The interface levels estimated using image processing technique were comparable with the estimations using first principle. In addition, an extended ICM

algorithm was also proposed, which was aiming to reduce the computational cost of conventional ICM algorithm. The comparison results showed the reduction of computational cost when using extended ICM algorithm. A customized neighborhood system was proposed in order to reduce the interface estimation variance. It also has been shown that using customized neighborhood system could not only obtain more accurate interface level estimation but also lower interface estimation variance.

Chapter 3

Gaussian Mixture Model and Markov Random Field based Unsupervised Image Segmentation using EM Algorithm and Mean Field Approximation

3.1 Introduction

In the previous chapter, we introduced Markov random field (MRF) based supervised image segmentation. An extended ICM algorithm was proposed by integrating k-means clustering to the ICM algorithm. Furthermore, a customized neighborhood system was also proposed to replace conventional first order neighborhood system to obtain the improvement in reducing estimation variance. Both proposed approaches improved the image segmentation results in terms of computational cost and estimation variance. However, a main disadvantage is that all parameters in MRF mathematical model should be known before applying the algorithm. This is a common shortcoming for all supervised machine learning problems. Therefore, the objective of this chapter is to develop an approach for unsupervised image segmentation in conjunction with parameter estimation algorithm.

The *Expectation Maximization*(EM) algorithm is a feasible approach to find the parameters of maximum-likelihood estimation in an unsupervised way [7]. The EM

algorithm has been proved to be a handy tool for the case when the given data is incomplete due to missing values or hidden variables. In the MRF based image processing problem discussed in chapter 2, all random variables could be considered as the hidden variables. Therefore, the entire data set for the EM algorithm in the MRF based image processing contains both observations and random hidden variables.

As mentioned in chapter 2, the main parameters were assumed to be known in the observation model. Generally, the observation model for unsupervised image segmentation problem is assumed to be the *Gaussian Mixture Model* (GMM) [8, 9, 10, 11]. As the images for segmentation purpose usually have significant differences in pixel intensities among different regions, it makes the mixture model a natural choice for observation model.

However, the image segmentation results obtained using the Gaussian mixture model are usually poor by ignoring the spatial constraints, namely neighborhood relations, among latent variables. Spatial dependence is important in image segmentation as the neighborhood relation is the key assumption in the MRF based image processing. To find a feasible solution by taking into account of spatial constraints, we need to introduce another set of important parameters regarding the neighbors' relationship in the MRF model. These parameters are the ones that explain the penalty of conflict between the considered pixel and its neighbors in energy function of prior Gibbs distribution. In the literature, there are a number of ways to account spatial dependence in the GMM based image segmentation with different assumptions and scenarios [10, 11, 12]. One way is to assume the weight of the GMM for each pixel (random variable) follows the prior Gibbs distribution [10], another way is to use clustering results in calculating spatial dependence [12], and the parameters of spatial constraints can be assigned as specific numerical values in some cases[11]. Whereas, none of existing methods is to estimate the Gaussian mixture model parameters as well as the penalty parameter of MRF spatial constraints in a single framework. In this chapter, we propose a new approach to account MRF spatial constraints and simultaneously estimate parameters of the GMM and MRF spatial dependence using the EM algorithm.

Unlike the standard GMM based parameter estimation using the EM algorithm, a main challenge of GMM and MRF based parameter estimation is the linked hidden variables in MRF. To overcome this challenge, we propose to apply *Mean Field Approximation* (MFA) to solve the dependency problem between random variables. And then the model parameters are estimated using EM algorithm. The final outcome of segmentation is still based on MAP-MRF framework in conventional MRF image analysis. The novelty of the proposed approach is elegantly estimating the penalty parameter of MRF spatial constraints along with the Gaussian mixture model parameters using the EM algorithm under the MAP-MRF framework yielding an unsupervised algorithm for image segmentation.

The organization of this chapter is as follows: a detailed introduction to Gaussian mixture model and mean field approximation will be provided in section 3.2, the formulation of parameter estimation using EM algorithm will be demonstrated in section 3.3, section 3.4 summarizes the entire methodology proposed in this chapter, the experimental validation for interface level detection and estimation will be followed in section 3.5, and finally the conclusions will be drawn in section 3.6.

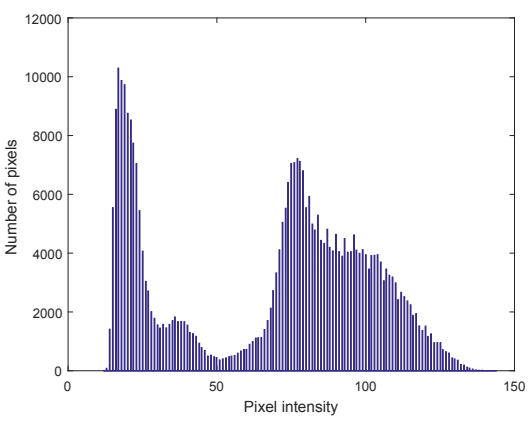
3.2 Gaussian Mixture Model and Mean Field Approximation

3.2.1 Gaussian Mixture Model

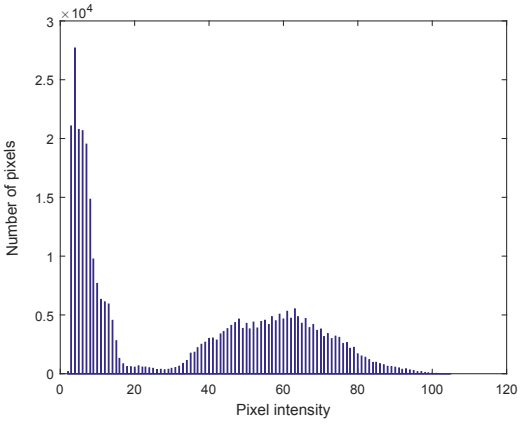
As mentioned in chapter 2, pixel values of an image for segmentation purpose have significant differences among distinct regions, therefore, the mixture model becomes a reasonable choice for modeling the observed image.

Figures 3.1a to 3.1d show the histograms of pixel values of four different observed raw images. All histograms illustrate a similar distribution of mixture model which confirms the idea of modeling observed pixels using a Gaussian mixture model.

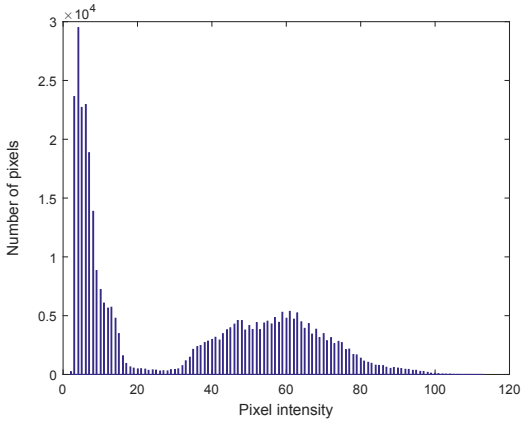
The observed image noise was assumed as Gaussian distribution with unit variance in chapter 2; however, based on the histograms shown in Figures 3.1a to 3.1d, the assumption of unit variance was invalid. Therefore, in this chapter, the raw image



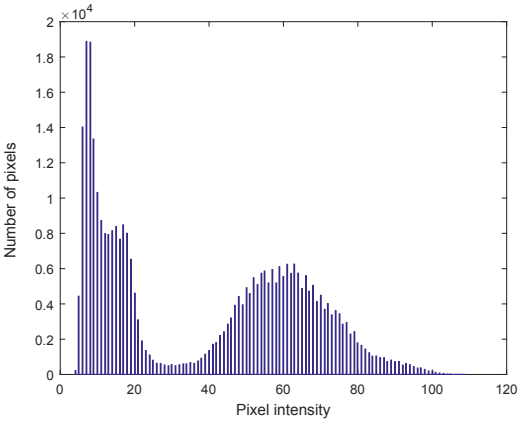
(a) *Histogram of image 1*



(b) *Histogram of image 2*



(c) *Histogram of image 5*



(d) *Histogram of image 8*

Figure 3.1: *Histograms of pixel values of four different observed raw images.*

data will be modeled by a Gaussian mixture model with two different Gaussian components. In general, a Gaussian mixture model can be formulated in the following form [5]:

$$P(d_s \mid \theta_d) = \sum_{k=1}^{K} w_k P_k(d_s \mid \theta_k) \tag{3.1}$$

where d_s represents an observed pixel, θ_d is the observation parameter set, k indicates the Gaussian component in the mixture model, and w_k is the weight of each Gaussian distribution, which satisfies:

$$\sum_{k=1}^{K} w_k = 1 \tag{3.2}$$

Totally, there are K Gaussian components in the mixture model. For each Gaussian component, it takes the standard distribution form as:

$$P_k(d_s \mid \theta_k) = \frac{1}{\sigma_k \sqrt{2\pi}} e^{-\frac{(d_s - \mu_k)^2}{2\sigma_k^2}} \tag{3.3}$$

Consequently, observation parameter set θ_d includes the distribution parameter θ_k, which includes the mean μ_k and the variance σ_k, and the weight for each distribution w_k. Recall the random variables in the Markov random field, for each pixel in the observed image d_s, the corresponding latent variable f_s is called the component identity of d_s in the GMM. This component identity actually indicates the Gaussian distribution from which the observed pixel comes. The probability of this event occurrence is equal to the mixing coefficient of corresponding Gaussian component. An illustration of the graphical explanation is shown in Figure 3.2.

3.2.2 Mean Field Approximation

In chapter 2, Hammersley-Clifford theorem has proved that random variables in the Markov random field obey Gibbs distribution. The penalty parameters, which explain the correlation between every random variable and its neighbors, in the energy function in equation (2.6) are estimated using the EM algorithm. While performing maximum-likelihood estimation using the EM framework, one has to compute the logarithm of the Gibbs distribution. Due to the complexity of partition function, which

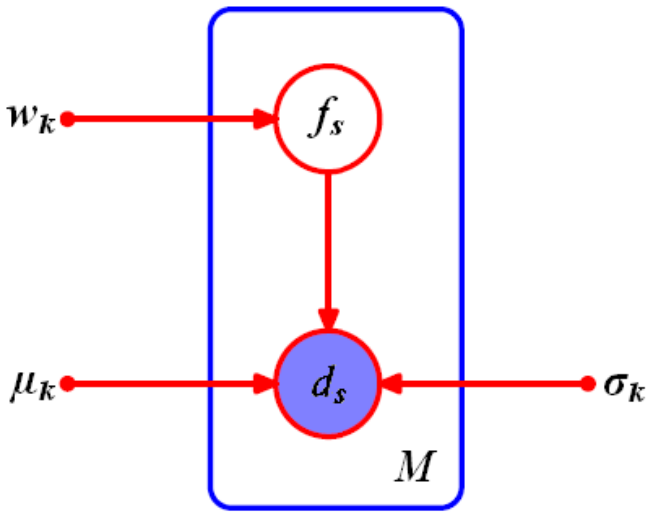


Figure 3.2: *Graphical representation of a Gaussian mixture model for an observed pixel d_s and corresponding latent Markov random variable f_s where $s = 1, ..., M$.*

can be shown in equation (2.7), it is impossible to factorize the Gibbs distribution and then estimate the parameters in the Gibbs distribution using the EM algorithm. As a result, seeking a distribution, which can be factorized to approximate Gibbs distribution, is the key of GMM-MRF based unsupervised image segmentation under the EM framework. *Mean field theory* (MFT) provides a perspective to achieve this objective.

Mean field theory is widely used in physics as well as statistical mechanics and has been applied in many areas such as neural networks and deterministic annealing [13]. Originally, the mean field theory was proposed for dealing with a large number of individual components which interact with each other in physics. It has been extended to deal with uncertainty in the dependency of models in context of graphical models in the area of image processing, neural networks, and artificial intelligence [13]. The objective of applying MFT to our model is to obtain an approximate model such that the inference is easy to compute. As illustrate in Figure 3.3, $P(x)$ represents a dependency distribution among individual components that are interacted. By applying mean field approximation, we obtain independently distributed $R(x)$ to approximate $P(x)$. As a result, instead of calculating the inference of $P(x)$, it is

more straightforward to compute the inference of $R(x)$. In the image segmentation problem, the priori Gibbs distribution is a such dependency model whose inference is hard to compute. To estimate the parameters using the EM algorithm, mean field approximation (MFA) is a good approach to find an approximate distribution and make the inference calculable.

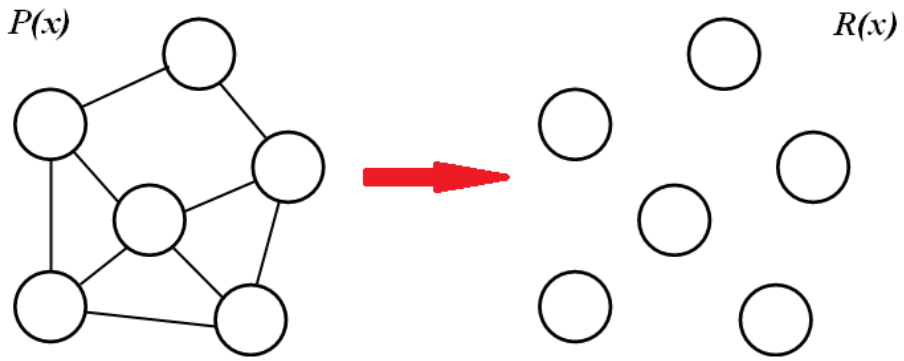


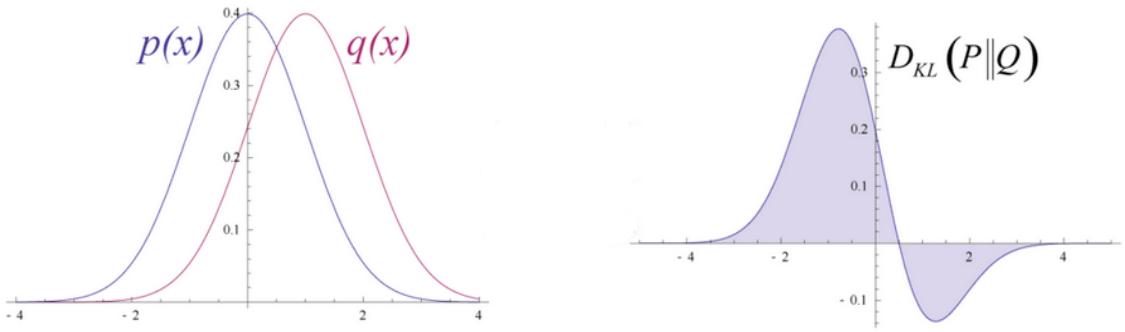
Figure 3.3: *The illustration of applying mean field approximation to a distribution having probabilistic dependence. P(x) is the original distribution, R(x) is the approximate distribution by applying MFA.*

A commonly used approach for obtaining approximate distribution is cross entropy minimization [13]. The cross entropy, also known as *Kullback-Leibler* (KL) *distance* or *divergence*, is a measurement of distance between two distributions. Since Gibbs distribution is a complex distribution that the random variables are interacted, it is difficult to graphically show the KL distance between the original distribution and the approximate distribution. Let us take a simple Gaussian distribution as an example. As shown in Figure 3.4a, the original continuous distribution $q(x)$ will be approximated by a zero-mean Gaussian distribution $p(x)$. The KL distance between the original and approximate distributions can be calculated using the following expression:

$$D_{KL}(p \parallel q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx \tag{3.4}$$

Figure 3.4b shows the graphical representation of KL distance, which is the shaded area. The optimal solution of $q(x)$ is to minimize the KL distance by tuning the

variance of the approximate distribution $p(x)$.



(a) *original $q(x)$, approximate $p(x)$* **(b)** *KL distance*

Figure 3.4: *The illustration of the approximation distribution and the corresponding KL distance to the original distribution.*

In our case, since the random variables in Gibbs distribution are discrete, the KL distance between the original Gibbs distribution and the approximate distribution can be formulated as:

$$D_{KL}(R \parallel P) = \sum_{f \in F} R(f) \log \frac{R(f)}{P(f)} \tag{3.5}$$

where $P(f)$ is the original Gibbs distribution that can be found in equation (2.6) in section 2.2.2, $R(f)$ is the approximate distribution that can be factorized to easily compute the inference

$$R(f) = \prod_{s} R(f_s) \tag{3.6}$$

and $D_{KL}(R \parallel P)$ is the KL distance between $R(f)$ and $P(f)$. Similarly, the optimal solution of $R(f)$ is to minimize the KL distance between two distributions

$$R(f) = \arg\min_{R(f_s)} D_{KL}(R \parallel P) \tag{3.7}$$

and the approximation $R(f)$ can be shown by considering Markov property of random variables, and it can be expressed as:

$$R(f_s) \propto \exp\left(R(f_{N_s}) \log P(f_s \mid f_{N_s})\right) \tag{3.8}$$

And finally, it can be concluded that

$$R(f_s \mid f_{N_s}) \propto \exp\left(U_s(f_s \mid f_{N_s})\right) \tag{3.9}$$

The detailed derivation of equation (3.9) can be found in [4, 13].

3.3 Parameter Estimation using EM Algorithm

3.3.1 Problem Formulation

To formulate Gaussian mixture model and Markov random field based unsupervised image segmentation problem using EM algorithm, it is important to define the data set for the algorithm. The entire data includes observed pixel values, d's and corresponding latent random variables in MRF, f's. A graphical representation is shown in Figure 3.5.

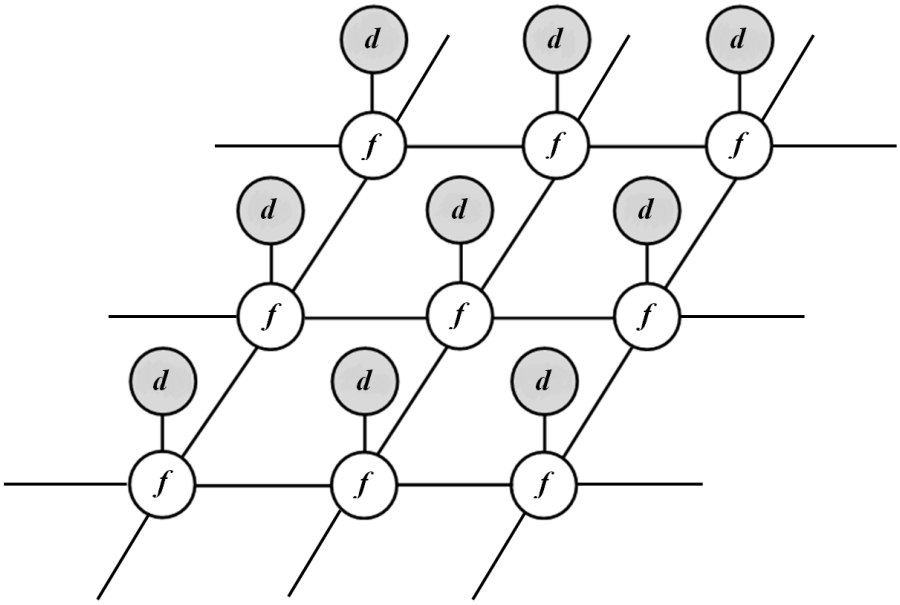


Figure 3.5: *Graphical representation of data set including observations, d's, and latent random variables in MRF, f's.*

The next step is to clarify all parameters that need to be estimated. As introduced in section 3.2.1, parameters from Gaussian mixture model are means, μ_k, variances, σ_k, and the mixing coefficients, w_k. Another set of parameters explaining neighbor's

relationship can be defined as the following. An important assumption for the energy function is that clique potentials take up to two sites [4]. Therefore, the energy function takes the form

$$\begin{aligned} U_s(f_s \mid f_{N_s}) &= V_1(f_s) + \sum_{\{s,s'\} \in C_2} V_2(f_s, f_{s'}) \\ &= \alpha f_s + \sum_{s' \in N_s} \beta f_s f_{s'} \end{aligned} \tag{3.10}$$

Since the single-site parameter, α, does not carry information of neighbor pixels and is usually set to be zero, β is the only parameter that needs to be estimated [4, 10]. As a result, the energy function can be rewritten as:

$$U_s(f_s \mid f_{N_s}, \theta_f) = \sum_{s' \in N_s} \beta f_s f_{s'} \tag{3.11}$$

where β is the penalty parameter of spatial constraints in MRF parameter set, θ_f. Consequently, it can be concluded that the overall parameter set Θ is

$$\Theta = \{\mu_k, \sigma_k, w_k, \beta\}$$

where $k \in \{1, 2, ..., K\}$, and we use the EM algorithm to estimate the parameter set Θ.

The EM algorithm is an iterative approach, it contains two steps in each iteration: E-step and M-step. E-step computes a Q-function by calculating the expectation of likelihood of joint probability of observation and latent variables given the parameters that are estimated in previous iteration. It can be formulated as

$$Q(\Theta \mid \Theta^{(t)}) = E\big[\log P(f, d \mid \Theta) \mid d, \Theta^{(t)}\big] \tag{3.12}$$

M-step is aiming to update parameters by maximizing the Q-function that is derived in E-step in the same iteration, that is

$$\Theta^{(t+1)} = \arg\max_{\Theta} Q(\Theta \mid \Theta^{(t)}) \tag{3.13}$$

Once all parameters are updated, the next iteration starts with E-step again by using updated parameters. The following sections 3.3.2 and 3.3.3 provide the detailed EM algorithm derivation.

3.3.2 Expectation Step

In this section, the Q-function derivation will be demonstrated in detail. Start with the joint probability

$$P(f, d \mid \Theta) = P(d \mid f, \Theta) \cdot P(f \mid \Theta) \tag{3.14}$$

By substituting the equation (3.14) into equation (3.12), the Q-function can be expressed as:

$$\begin{aligned} Q(\Theta \mid \Theta^{(t)}) &= E\big[\log P(f, d \mid \Theta) \mid d, \Theta^{(t)}\big] \\ &= \sum_{f \in F} \big[\log P(f, d \mid \Theta)\big] P(f \mid d, \Theta^{(t)}) \\ &= \sum_{f \in F} \big[\log P(d \mid f, \Theta) + \log P(f \mid \Theta)\big] P(f \mid d, \Theta^{(t)}) \\ &= \sum_{f \in F} \log P(d \mid f, \Theta) P(f \mid d, \Theta^{(t)}) + \log P(f \mid \Theta) P(f \mid d, \Theta^{(t)}) \end{aligned} \tag{3.15}$$

It is obvious that the Q-function can be separated into two parts and one is related to observations and another is related to MRF. Let two parts be denoted as:

$$Q_d(\Theta \mid \Theta^{(t)}) = \sum_{f \in F} \log P(d \mid f, \Theta) P(f \mid d, \Theta^{(t)}) \tag{3.16}$$

$$Q_f(\Theta \mid \Theta^{(t)}) = \sum_{f \in F} \log P(f \mid \Theta) P(f \mid d, \Theta^{(t)}) \tag{3.17}$$

Therefore

$$Q(\Theta \mid \Theta^{(t)}) = Q_d(\Theta \mid \Theta^{(t)}) + Q_f(\Theta \mid \Theta^{(t)}) \tag{3.18}$$

Two sub-Q-functions will be derived separately in the following.

For the first sub-Q-function $Q_d(\Theta \mid \Theta^{(t)})$, since the image pixels are assumed to be independently observed,

$$\begin{aligned} P(d \mid f, \Theta) &= P(d \mid f, \theta_d) \\ &= \prod_{s \in S} P(d_s \mid f_s, \theta_{f_s}) \end{aligned} \tag{3.19}$$

By taking the logarithm,

$$\begin{aligned}
\log P(d \mid f, \Theta) &= \sum_{s=1}^{M} \log P(d_s \mid f_s, \theta_{f_s}) \\
&= \sum_{s=1}^{M} \sum_{k=1}^{K} \delta_{f_s,k} \log P(d_s \mid k, \theta_k) \\
&= \sum_{s=1}^{M} \sum_{k=1}^{K} \delta_{f_s,k} \log \big[w_k P_k(d_s \mid \theta_k) \big] \\
&= \sum_{s=1}^{M} \sum_{k=1}^{K} \delta_{f_s,k} \big[\log w_k + \log P_k(d_s \mid \theta_k) \big] \qquad (3.20)
\end{aligned}$$

where $\delta_{f_s,k}$ is the *dirac delta* function that is defined as:

$$\delta_{a,b} = \begin{cases} 1 & \text{if} \quad a = b \\ 0 & \text{if} \quad a \neq b \end{cases} \qquad (3.21)$$

By substituting equation (3.20) into (3.16)

$$\begin{aligned}
Q_d(\Theta \mid \Theta^{(t)}) &= \sum_{f \in F} \sum_{s=1}^{M} \sum_{k=1}^{K} \delta_{f_s,k} \big[\log w_k + \log P_k(d_s \mid \theta_k) \big] P(f \mid d, \Theta^{(t)}) \\
&= \sum_{f \in F} \sum_{s=1}^{M} \sum_{k=1}^{K} \delta_{f_s,k} \big[\log w_k + \log P_k(d_s \mid \theta_k) \big] \prod_{s=1}^{M} P(f_s \mid d_s, \Theta^{(t)}) \\
&= \sum_{s=1}^{M} \sum_{k=1}^{K} \big[\log w_k + \log P_k(d_s \mid \theta_k) \big] \sum_{f \in F} \delta_{f_s,k} \prod_{s=1}^{M} P(f_s \mid d_s, \Theta^{(t)}) \qquad (3.22)
\end{aligned}$$

To simplify the above equation, it can be shown that [14],

$$\sum_{f \in F} \delta_{f_s,k} \prod_{s=1}^{M} P(f_s \mid d_s, \Theta^{(t)}) = P(k \mid d_s, \Theta^{(t)}) \qquad (3.23)$$

where

$$P(k \mid d_s, \Theta^{(t)}) = \frac{w_k^{(t)} P_k(d_s \mid \theta_k^{(t)})}{\sum_{k=1}^{K} w_k^{(t)} P_k(d_s \mid \theta_k^{(t)})} \qquad (3.24)$$

Finally, we can conclude the derivation of sub-Q-function related with observation by

combining equation (3.22) and (3.23)

$$
\begin{aligned}
Q_d(\Theta \mid \Theta^{(t)}) &= \sum_{s=1}^{M}\sum_{k=1}^{K}\big[\log w_k + \log P_k(d_s \mid \theta_k)\big]P(k \mid d_s, \Theta^{(t)}) \\
&= \sum_{s=1}^{M}\sum_{k=1}^{K}\log(w_k)P(k \mid d_s, \Theta^{(t)}) + \sum_{s=1}^{M}\sum_{k=1}^{K}\log\big(P_k(d_s \mid \theta_k)\big)P(k \mid d_s, \Theta^{(t)})
\end{aligned}
\tag{3.25}
$$

Regarding the derivation of second sub-Q-function, $Q_f(\Theta \mid \Theta^{(t)})$, as shown in equation (3.17), we use the mean field approximation of Gibbs distribution. It has already been shown in equation (3.9), that the exact approximate distribution can be written as:

$$
R(f_s \mid f_{N_s}, \theta_f) = \frac{1}{Z'_s} e^{-U_s(f_s|f_{N_s},\theta_f)} \tag{3.26}
$$

where

$$
Z'_s = \sum_{f_s} e^{-U_s(f_s|f_{N_s},\theta_f)} \tag{3.27}
$$

and

$$
U_s(f_s \mid f_{N_s}, \theta_f) = \sum_{s' \in N_s} \beta f_s f_{s'} \tag{3.28}
$$

In the work of this chapter, the energy function is assumed as *auto-logistic model* which means all random variables take binary values from $\{0, 1\}$. By combining equations (3.26), (3.27), and (3.28), the approximate distribution can be simplified as:

$$
R(f_s \mid f_{N_s}, \theta_f) = \frac{\exp\big(-\sum_{s' \in N_s} \beta f_s f_{s'}\big)}{1 + \exp\big(-\sum_{s' \in N_s} \beta f_{s'}\big)} \tag{3.29}
$$

Therefore, the prior distribution of random variables in the MRF can be approximated as:

$$
\begin{aligned}
P(f \mid \Theta) &= P(f \mid \theta_f) \\
&\approx R(f \mid \theta_f) \\
&= \prod_{s=1}^{M} R(f_s \mid f_{N_s}, \theta_f) \\
&= \prod_{s=1}^{M} \frac{\exp\big(-\sum_{s' \in N_s} \beta f_s f_{s'}\big)}{1 + \exp\big(-\sum_{s' \in N_s} \beta f_{s'}\big)}
\end{aligned}
\tag{3.30}
$$

By taking the logarithm of $P(f \mid \Theta)$, the factorized expression can easily be converted to a summation of logarithm, namely

$$\begin{aligned}
\log P(f \mid \Theta) &= \sum_{s=1}^{M} \log R(f_s \mid f_{N_s}, \theta_f) \\
&= \sum_{s=1}^{M} \log \left[\frac{\exp\big(-\sum_{s' \in N_s} \beta f_s f_{s'}\big)}{1 + \exp\big(-\sum_{s' \in N_s} \beta f_{s'}\big)} \right] \\
&= \sum_{s=1}^{M} \Big(- \sum_{s' \in N_s} \beta f_s f_{s'} \Big) - \log \Big[1 + \exp\big(- \sum_{s' \in N_s} \beta f_{s'} \big) \Big] \qquad (3.31)
\end{aligned}$$

Finally, equation (3.17) can be written as:

$$\begin{aligned}
Q_f(\Theta \mid \Theta^{(t)}) &= \sum_{f \in F} \sum_{s=1}^{M} \log R(f_s \mid f_{N_s}, \theta_f) P(f \mid d, \Theta^{(t)}) \\
&= \sum_{f \in F} \sum_{s=1}^{M} \sum_{k=1}^{K} \delta_{f_s,k} \log R(f_s = k \mid f_{N_s}, \theta_f) \prod_{s=1}^{M} P(f_s \mid d_s, \Theta^{(t)}) \\
&= \sum_{s=1}^{M} \sum_{k=1}^{K} \log R(f_s = k \mid f_{N_s}, \theta_f) \sum_{f \in F} \delta_{f_s,k} \prod_{s=1}^{M} P(f_s \mid d_s, \Theta^{(t)}) \\
&= \sum_{s=1}^{M} \sum_{k=1}^{K} \log R(f_s = k \mid f_{N_s}, \theta_f) P(k \mid d_s, \Theta^{(t)}) \qquad (3.32)
\end{aligned}$$

For deriving equation (3.32), we have assumed two conditions: the first one is

$$R(f_s = k \mid f_{N_s}, \theta_f) = R(f_s \mid f_{N_s}, \theta_f) \qquad (3.33)$$

The second condition is that the neighborhood information in the conditional probability is from previous iteration, thus,

$$Q_f(\Theta \mid \Theta^{(t)}) = \sum_{s=1}^{M} \sum_{k=1}^{K} \log R(f_s = k \mid f_{N_s}^{(t)}, \theta_f) P(k \mid d_s, \Theta^{(t)}) \qquad (3.34)$$

3.3.3 Maximization Step

The next step is to maximize the Q-function obtained in the Expectation step to iteratively update parameters. The overall expression of Q-function can be concluded

by combining equations (3.18), (3.25), and (3.34), that is

$$\begin{aligned} Q(\Theta \mid \Theta^{(t)}) = &\sum_{s=1}^{M}\sum_{k=1}^{K} \log(w_k) P(k \mid d_s, \Theta^{(t)}) \\ &+ \sum_{s=1}^{M}\sum_{k=1}^{K} \log\big(P_k(d_s \mid \theta_k)\big) P(k \mid d_s, \Theta^{(t)}) \\ &+ \sum_{s=1}^{M}\sum_{k=1}^{K} \log R(f_s = k \mid f_{N_s}^{(t)}, \theta_f) P(k \mid d_s, \Theta^{(t)}) \end{aligned} \tag{3.35}$$

Based on equation (3.35), we observe that w_k, θ_k and β can be separately maximized, since they belong to three independent terms. Therefore, our objectives can be expressed as the following form:

$$w_k^{(t+1)} = \arg\max_{w_k} \sum_{s=1}^{M}\sum_{k=1}^{K} \log(w_k) P(k \mid d_s, \Theta^{(t)}) \tag{3.36}$$

$$\theta_k^{(t+1)} = \arg\max_{\theta_k} \sum_{s=1}^{M}\sum_{k=1}^{K} \log\big(P_k(d_s \mid \theta_k)\big) P(k \mid d_s, \Theta^{(t)}) \tag{3.37}$$

$$\beta^{(t+1)} = \arg\max_{\beta} \sum_{s=1}^{M}\sum_{k=1}^{K} \log R(f_s = k \mid f_{N_s}^{(t)}, \theta_f) P(k \mid d_s, \Theta^{(t)}) \tag{3.38}$$

To find the updating expression of $w_k^{(t+1)}$, equation (3.36), we need to introduce the Lagrange multiplier λ with the constraint that $\sum_k w_k = 1$, then solve the equation:

$$\frac{\partial}{\partial w_k}\left[\sum_{s=1}^{M}\sum_{k=1}^{K} \log(w_k) P(k \mid d_s, \Theta^{(t)}) + \lambda\left(\sum_k w_k - 1\right)\right] = 0 \tag{3.39}$$

The detailed derivation of updating weights, $w_k^{(t+1)}$, and Gaussian distribution parameters, $\theta_k^{(t+1)}$ can be found in [14], and results are given as following:

$$w_k^{(t+1)} = \frac{1}{M}\sum_{s=1}^{M} P(k \mid d_s, \Theta^{(t)}) \tag{3.40}$$

$$\mu_k^{(t+1)} = \frac{\sum_{s=1}^{M} d_s P(k \mid d_s, \Theta^{(t)})}{\sum_{s=1}^{M} P(k \mid d_s, \Theta^{(t)})} \tag{3.41}$$

$$(\sigma^2)_k^{(t+1)} = \frac{\sum_{s=1}^{M} P(k \mid d_s, \Theta^{(t)})\big(d_s - \mu_k^{(t+1)}\big)\big(d_s - \mu_k^{(t+1)}\big)^T}{\sum_{s=1}^{M} P(k \mid d_s, \Theta^{(t)})} \tag{3.42}$$

Updating pair-wise potential parameter, also named penalty parameter, β, is one of the main contributions in this work. To find β, we maximize $Q_f(\Theta \mid \Theta^{(t)})$ by taking the derivative of equation (3.34) and set it to zero,

$$\frac{\partial}{\partial \beta}\left[\sum_{s=1}^{M}\sum_{k=1}^{K}\log R(f_s = k \mid f_{N_s}^{(t)}, \theta_f)P(k \mid d_s, \Theta^{(t)})\right] = 0 \tag{3.43}$$

Substitute equation (3.31) into the above equation

$$\frac{\partial}{\partial \beta}\left[\sum_{s=1}^{M}\sum_{k=1}^{K}\left\{\big(-\sum_{s'\in N_s^{(t)}}\beta f_s f_{s'}\big) - \log\Big[1+\exp\big(-\sum_{s'\in N_s^{(t)}}\beta f_{s'}\big)\Big]\right\}P(k \mid d_s, \Theta^{(t)})\right] = 0 \tag{3.44}$$

$$\sum_{s=1}^{M}\sum_{k=1}^{K}\left[\big(-\sum_{s'\in N_s^{(t)}} f_s f_{s'}\big) - \frac{\exp\big(-\sum_{s'\in N_s^{(t)}}\beta f_{s'}\big)\cdot\big(-\sum_{s'\in N_s^{(t)}} f_{s'}\big)}{1+\exp\big(-\sum_{s'\in N_s^{(t)}}\beta f_{s'}\big)}\right]P(k \mid d_s, \Theta^{(t)}) = 0 \tag{3.45}$$

Rearrange equation (3.45)

$$\begin{aligned}&\sum_{s=1}^{M}\sum_{k=1}^{K}\big(-\sum_{s'\in N_s^{(t)}} f_s f_{s'}\big)P(k \mid d_s, \Theta^{(t)}) \\ &\quad= \sum_{s=1}^{M}\sum_{k=1}^{K}\frac{\exp\big(-\sum_{s'\in N_s^{(t)}}\beta f_{s'}\big)}{1+\exp\big(-\sum_{s'\in N_s^{(t)}}\beta f_{s'}\big)}\big(-\sum_{s'\in N_s^{(t)}} f_{s'}\big)P(k \mid d_s, \Theta^{(t)})\end{aligned} \tag{3.46}$$

Denoting $A = \exp\big(-\sum_{s'\in N_s^{(t)}}\beta f_{s'}\big)$, the above equation can be rewritten as:

$$\frac{A}{1+A} = \frac{\sum_{s=1}^{M}\sum_{k=1}^{K}\big(-\sum_{s'\in N_s^{(t)}} f_s f_{s'}\big)P(k \mid d_s, \Theta^{(t)})}{\sum_{s=1}^{M}\sum_{k=1}^{K}\big(-\sum_{s'\in N_s^{(t)}} f_{s'}\big)P(k \mid d_s, \Theta^{(t)})} \tag{3.47}$$

Let the right hand side of equation (3.47) be denoted as B which should be a constant value in each iteration. As a result, the above equation can be simplified as:

$$\frac{A}{1+A} = B \tag{3.48}$$

By solving the above equation, we can obtain

$$A = \frac{B}{1-B} \tag{3.49}$$

Therefore,

$$\exp\big(-\sum_{s'\in N_s^{(t)}}\beta f_{s'}\big)=\frac{B}{1-B} \tag{3.50}$$

and the new β can be updated as:

$$\beta^{(t+1)}=\frac{\log\big(\frac{B}{1-B}\big)}{-\sum_{s'\in N_s^{(t)}} f_{s'}} \tag{3.51}$$

where

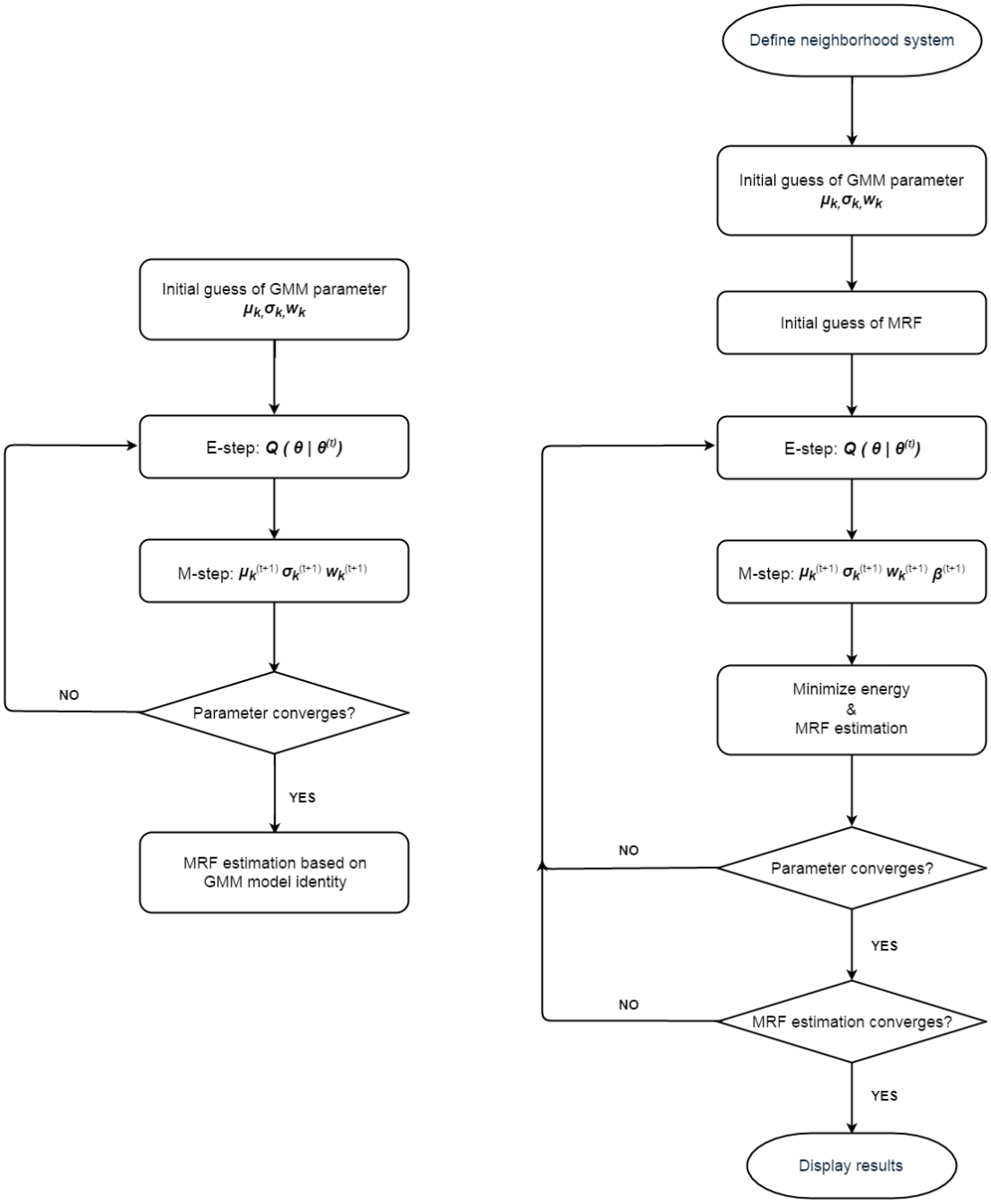
$$B=\frac{\sum_{s=1}^{M}\sum_{k=1}^{K}\big(-\sum_{s'\in N_s^{(t)}} f_s f_{s'}\big)P(k\mid d_s,\Theta^{(t)})}{\sum_{s=1}^{M}\sum_{k=1}^{K}\big(-\sum_{s'\in N_s^{(t)}} f_{s'}\big)P(k\mid d_s,\Theta^{(t)})} \tag{3.52}$$

Next, we summarize the parameter estimation methodology in a concise way.

3.4 Summary of Methodology

The proposed approach encapsulates the EM based parameter estimation for both observation model and MRF spatial dependence as well as MRF based image segmentation in a single framework. The observation model parameter update expressions are dependent on themselves while the update of neighborhood parameter β requires MRF estimation from previous iteration and has been discussed in section 3.3.2. The strategy for MRF estimation is same as the one introduced in chapter 2, which is energy minimization under MAP-MRF framework. The details can be referred to section 2.3.2. Once both the parameters and MRF estimation are converged, the unsupervised image segmentation results are obtained for discussion and further analysis. Before the iterative procedure starts, it is imperative to define a neighborhood system and the initial guess of both GMM parameter and MRF estimation.

By comparing with the conventional GMM based image segmentation algorithm, the main improvement of the proposed algorithm is taking Markov random field spatial constraints into account in every iteration of parameter estimation progress. Figures 3.6a and 3.6b show the complete flow charts of the methodology for EM algorithm based GMM and GMM-MRF unsupervised image segmentation. The added steps in (b) are due to the consideration of neighborhood system.



(a) *GMM* **(b)** *GMM-MRF*

Figure 3.6: *The flow charts of EM algorithm based on (a) GMM and (b) GMM-MRF unsupervised image segmentation.*

3.5 Experimental Validation for Interface Level Detection and Estimation

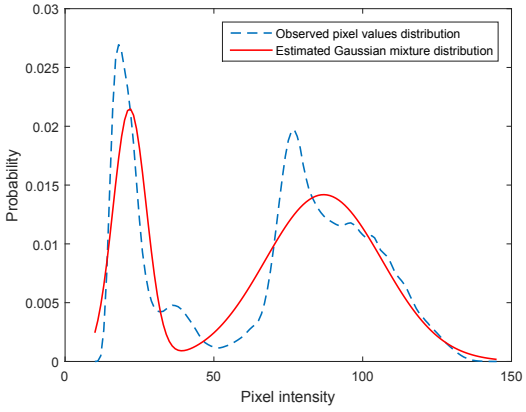
The proposed unsupervised algorithm in this chapter will be validated using the same experimental images that were used in chapter 2. MATLAB was employed to perform image analysis. The proposed algorithm will be validated in two aspects: the validation of Gaussian mixture model for observed images, and the validation of interface level detection and estimation by considering spatial constraints.

3.5.1 Gaussian Mixture Model Validation

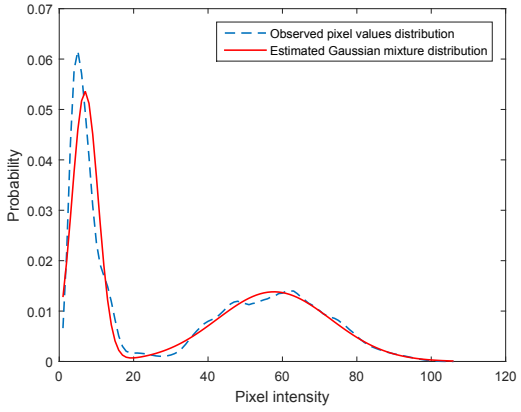
The proposed algorithm for unsupervised image segmentation in this chapter was developed based on the assumption of pixel values in the observed raw image follow a mixture of Gaussian distribution. To validate this assumption, the observed pixel values distribution was estimated first and EM algorithm was used to estimate all parameters of the Gaussian mixture distribution. The comparison between the observed pixel values distribution and the estimated Gaussian mixture distribution for four different images are shown in Figures 3.7a to 3.7d. They correspond to the histograms shown in Figures 3.1a to 3.1d. In Figures 3.7a to 3.7d, all dashed lines represent the observed pixel values distributions, and solid lines represent the estimated Gaussian mixture distributions. All figures show the agreement between the observed pixel values distribution and the estimated Gaussian mixture distribution. Therefore, the assumption of using Gaussian mixture distribution for the modeling is valid.

3.5.2 Validation of Image Segmentation and Interface Level Detection and Estimation

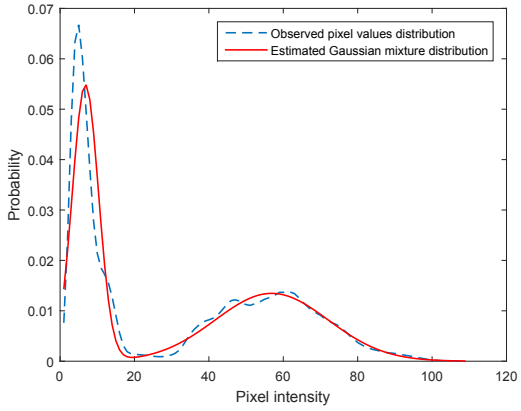
In this section, both image segmentation results and the interface level estimation will be shown by comparing the algorithm based on the GMM of observed image and the proposed algorithm based on both observed image as well as MRF spatial neighbors. As shown in Figures 3.8a and 3.8b, two raw images were selected to



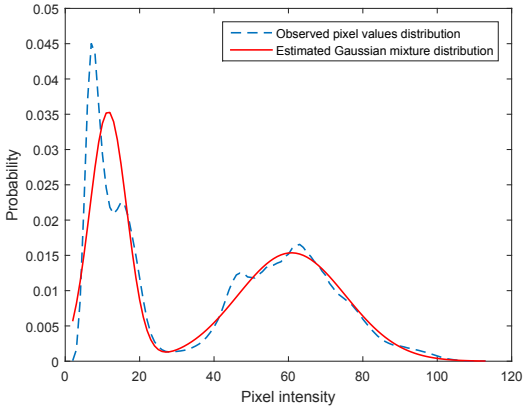
(a) *Distributions of image 1*



(b) *Distributions of image 2*



(c) *Distributions of image 5*



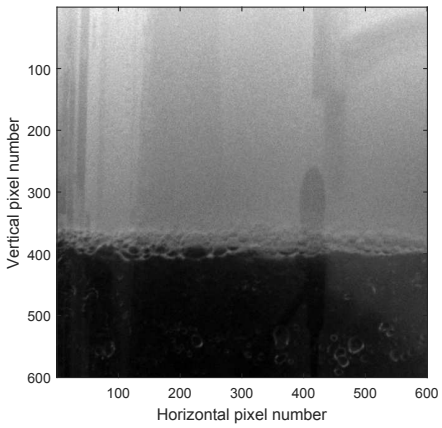
(d) *Distributions of image 8*

Figure 3.7: *The observed pixel values distribution and the estimated Gaussian mixture distribution for four different images.*

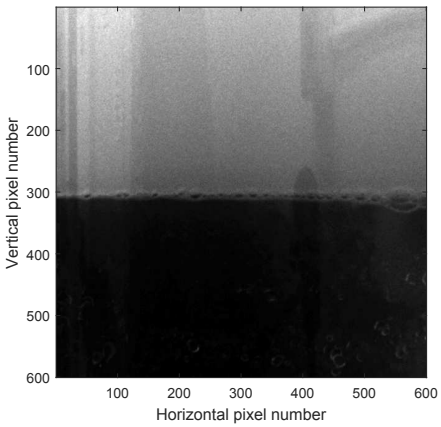
perform both algorithms. Figures 3.8c and 3.8d show the segmentation based on observed pixel values without considering the MRF neighborhood constraints. Figures 3.8e and 3.8f show the segmentation by considering both observed pixel values and the MRF neighborhood constraints. Here, the neighborhood constraints refer to the MRF spatial neighbors' dependence between every single random variable in MRF and its neighbors. Based on these segmented images, it is obvious that observation based segmentation is sensitive to image noise. For instance, both Figures 3.8c and 3.8d show a number of white dots or even large region in the area that should be theoretically in black. By comparing with the original raw images, those errors are caused by some sticky oil droplets attaching on the wall of the tank. In contrast, the two segmented images obtained using proposed algorithm are more robust to those noises caused in real experimental situations as shown in Figures 3.8e and 3.8f. The reason for the robustness of proposed algorithm is because of the consideration of spatial neighbors in the Markov random field.

Having obtained the segmented images, transient zone boundaries and interface level can be estimated based on the approach of using vertical profile of averaged pixels, which was proposed in chapter 2. Figures 3.9a and 3.9b indicate the estimated transient zone boundaries and interface levels in the segmented and observed images. Figure 3.9a shows the levels based on the segmentation result of Gaussian mixture model, the corresponding levels are indicated in the raw image as shown in Figure 3.9b. The levels that are indicated in Figures 3.9c and 3.9d were estimated based on the segmentation result obtained using proposed approach. It can be clearly seen that the GMM based segmentation fails in estimating interface level, especially for estimating the lower boundary. However, the segmentation result obtained by performing the GMM-MRF algorithm can provide the reasonable estimation for both interface and transient zone boundaries as shown in Figure 3.9d.

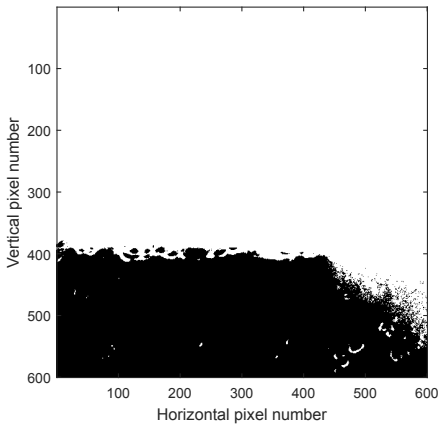
Table 3.1 shows the comparison of the interface heights using different estimating methods, which are first principle estimation, the GMM based image segmentation, and the proposed GMM-MRF based image segmentation. Considering the first principle estimation as the benchmark, the estimation errors in percentage of two algo-



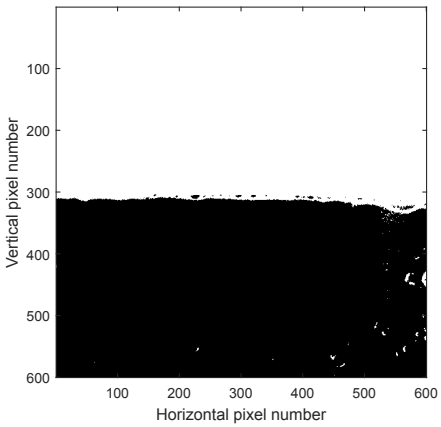
(a) *raw image 1*



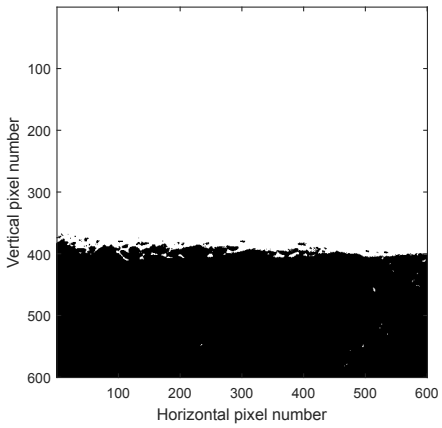
(b) *raw image 2*



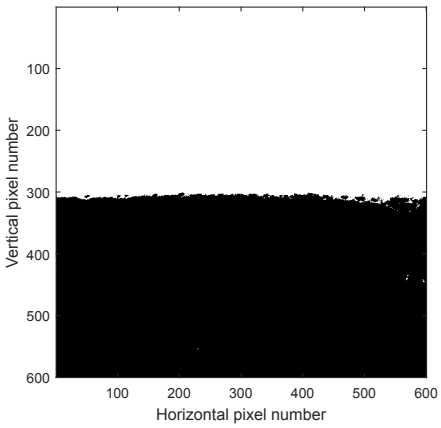
(c) *Observation based segmentation*



(d) *Observation based segmentation*

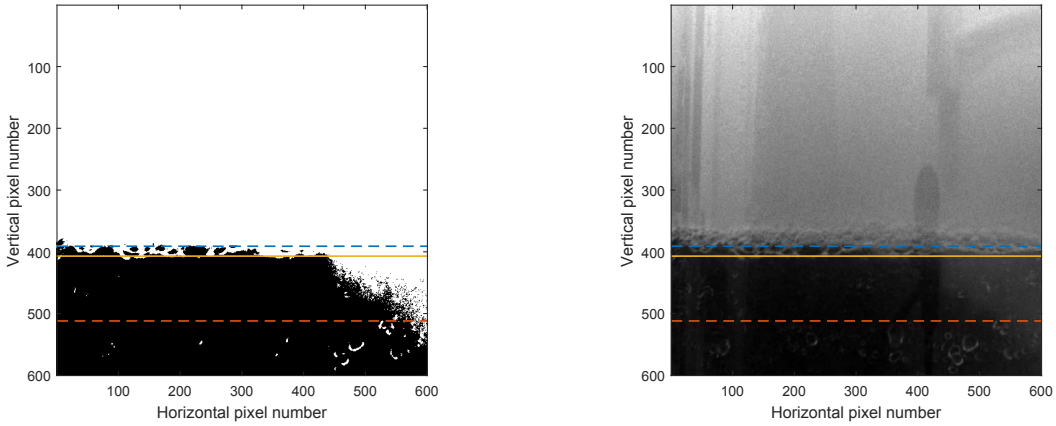


(e) *Neighborhood based segmentation*

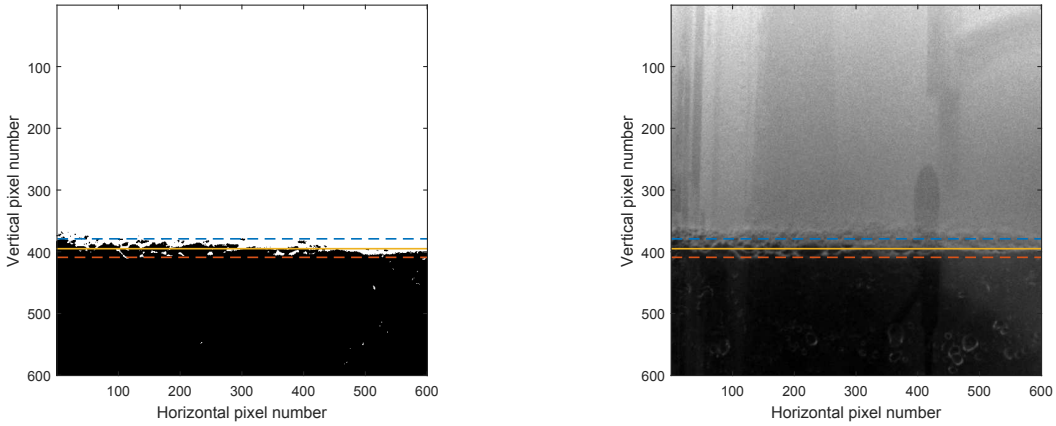


(f) *Neighborhood based segmentation*

Figure 3.8: *The comparison of the segmentation results between two algorithms for two different images. (a) and (b) are raw images, (c) and (d) are the segmented images based on GMM only, (e) and (f) are the segmented images based on GMM-MRF.*



(a) *Observation based* **(b)** *raw image*



(c) *Neighborhood based* **(d)** *raw image*

Figure 3.9: *The estimation of interface levels of segmented images based on two algorithms. (a) and (b) are based on GMM only, (c) and (d) are based on GMM-MRF. Dashed lines are upper and lower boundaries of transient zone, solid line is the estimated interface level.*

rithms based on image segmentation can be obtained and are also shown in Table 3.1. By comparing the percentage errors between two image segmentation algorithms, it can be concluded that the proposed GMM-MRF based image segmentation algorithm has the better performance. The averaged error of proposed algorithm is 0.27%, which is lower than the GMM based algorithm's averaged error 1.25%. Therefore, the proposed algorithm performs well in estimating interface level and the mean field approximation for Gibbs distribution is also valid.

Table 3.1: *The interface heights and corresponding errors based on two algorithms in image segmentation as compared with the first principle estimation.*

Image Number	First Principle Estimation	Image Segmentation			
		GMM		GMM-MRF	
	Interface Height (cm)	Interface Height (cm)	% Error	Interface Height (cm)	% Error
1	35.04	34.93	1.8%	35.05	0.2%
2	35.94	35.86	1.3%	35.92	0.3%
3	35.98	35.91	0.7%	35.99	0.3%
4	36.01	35.94	1.2%	36.03	0.2%
5	35.99	35.93	0.8%	36.00	0.2%
6	35.82	35.75	1.2%	35.84	0.3%
7	35.80	35.70	1.7%	35.82	0.3%
8	35.75	35.67	1.3%	35.76	0.2%
9	35.71	35.63	1.3%	35.74	0.5%
10	35.67	35.60	1.2%	35.68	0.2%

Considering the interface heights estimated using the proposed GMM-MRF based algorithm as the results of unsupervised image segmentation, they can also be com-

pared with those estimations obtained using supervised learning method discussed in chapter 2. Table 3.2 tabulates both supervised and unsupervised estimation results as compared with the first principle estimation. The corresponding errors in percentage are also shown in Table 3.2. None of errors is greater than 1%, which confirms the agreement in estimation of interface level for both supervised and unsupervised approaches.

Table 3.2: *The interface heights and corresponding errors of supervised and unsupervised image segmentation as compared with the first principle estimation.*

Image Number	First Principle Estimation	Supervised Image Segmentation		Unsupervised Image Segmentation	
	Interface Height (cm)	Interface Height (cm)	% Error	Interface Height (cm)	% Error
1	35.04	35.09	0.8%	35.05	0.2%
2	35.94	35.93	0.2%	35.92	0.3%
3	35.98	36.01	0.5%	35.99	0.3%
4	36.01	35.98	0.5%	36.03	0.2%
5	35.99	35.95	0.7%	36.00	0.2%
6	35.82	35.80	0.3%	35.84	0.3%
7	35.80	35.78	0.3%	35.82	0.3%
8	35.75	35.72	0.5%	35.76	0.2%
9	35.71	35.70	0.2%	35.74	0.5%
10	35.67	35.66	0.2%	35.68	0.2%

Furthermore, the interface estimation variance can also be calculated using equation (2.28) for both segmentation results obtained using the GMM based and the proposed GMM-MRF based algorithms discussed in this chapter. Figure 3.10 shows

the comparison of interface estimation variance when using the GMM based and the proposed GMM-MRF based algorithms. The estimation variances for the GMM-MRF algorithm are lower than those using the GMM based algorithm except image 9, where the performance of both algorithms is comparable. Therefore, it can be concluded that the proposed GMM-MRF based algorithm outperforms the GMM based algorithm in terms of segmentation result, interface level estimation and the estimation variance due to the consideration of MRF neighbors.

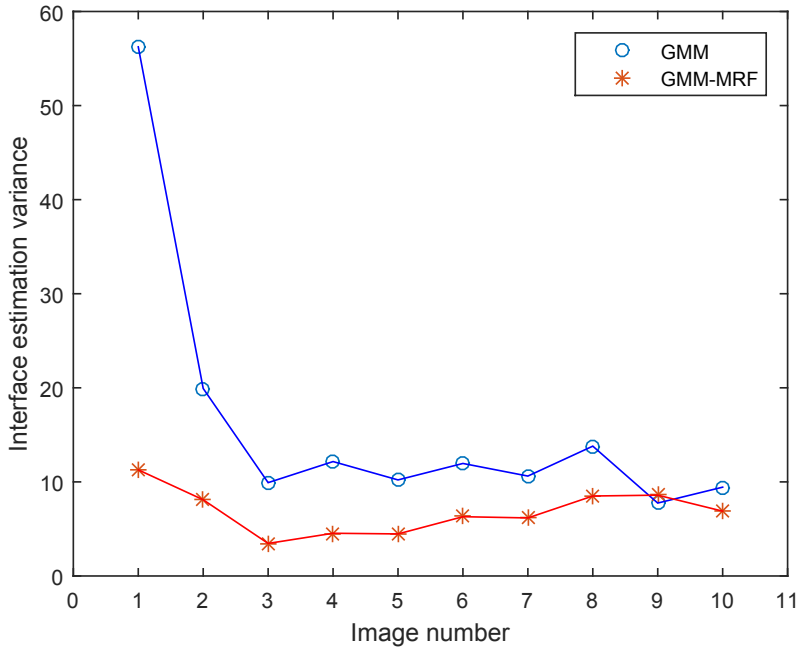


Figure 3.10: *The comparison of interface estimation variances by using GMM and GMM-MRF based algorithms.*

As for the computational cost of the proposed algorithm, it takes almost same number of iterations and has more computational time due to the calculation on penalty parameter of MRF spatial dependence.

3.6 Conclusions

In this chapter, a new algorithm was proposed to obtain unsupervised image segmentation. Gaussian mixture distribution was considered for the modeling of the

observed images. Based on Markov random field framework, the mean field approximation method was used to obtain the approximate distribution of Gibbs distribution in MRF. All model parameters were estimated using the EM algorithm. This proposed approach improved the GMM based segmentation algorithm by considering the spatial neighborhood in MRF. The advantage of the proposed algorithm was providing a new approach with consideration of the spatial constraints such that the penalty parameter of spatial constraints and the parameters in observation model could be estimated simultaneously in a single framework. Furthermore, the proposed algorithm also improved the segmentation result, interface level estimation and the variance of the estimation as compared with those results obtained using the GMM based approach. Nevertheless, a shortcoming of considering neighborhood in the algorithm was increasing the computational cost though it provided more accurate results.

Chapter 4

Dynamic Prediction of Binary Image Segmentation and Interface Level using Spatial Temporal Markov Random Field based Auto-logistic Model

4.1 Introduction

In the previous chapter, we demonstrated that the Markov random field based unsupervised image segmentation can be used to determine interface level detection based on images obtained using a camera. The observed image was modeled by a mixture of Gaussian distribution. The mean field approximation was used to obtain an approximate distribution of Gibbs distribution in the MRF, such that the inference of the approximate distribution is easy to compute. The model parameters were estimated using the EM algorithm. The interface level height estimations were found to agree with those estimates obtained using first principle model. The unsupervised estimations were also comparable with the estimations obtained using supervised learning method proposed in chapter 2. However, the height of interface level in the primary separation vessel (PSV) is a process variable, which changes over time and its prediction could be helpful for economic operation and improved control. As a result, the objective of this chapter is to consider the time series dependence in the modeling of

sequential images for predicting the dynamic interface level.

As illustrated in the previous two chapters, single image is usually modeled by a two-dimensional Markov random field or spatial Markov random field. Therefore, by stacking images along time axis, it forms a three-dimensional data consisting of a sequence of images, and such data can be modeled using the *spatial temporal Markov random field* (ST-MRF) model [4, 15, 16, 17, 18]. Typically, spatial temporal MRF can be applied for various video-based or sequential imaged based applications [4]. For example, spatial temporal MRF has been applied in video-based moving object detection and segmentation for the use of traffic monitoring [15, 16, 17]; the ST-MRF can also be used in contextual image classification for analyzing geographical sequential images [18]. Figure 4.1 illustrates a schematic diagram of spatial temporal Markov random field. The illustration shows three images along time axis at time instants of $t-1, t$, and $t+1$, and each image has the same spatial dimensions with height of h and width of w. Here, f represents the latent random variable in the MRF, and d represents the corresponding observed pixel intensity. Similarly with the definition of neighborhood system of two-dimensional MRF, the first order neighborhood system of the ST-MRF is defined to consist of six nearest neighbors in the three-dimensional volume [4]. For example, by considering the black node labeled as f at time of t, the first order neighborhood system of this node contains spatial neighbors, which are four grey nodes labeled as f at time of t, as well as two temporal neighbors, which are the grey nodes labeled as f at time instants of $t-1$ and $t+1$. The highlighted solid lines between the black node and its spatial neighbors represent the spatial cliques. Similarly, the highlighted dashed lines between the black node and its temporal neighbors represent the temporal cliques. Furthermore, the solid lines between black node and the corresponding observed pixel d in grey represents the observation clique. All three cliques will be considered in the modeling of the ST-MRF.

Logistic model is defined as a regression model where the dependent variables are categorical [19]. According to Besag [20], *auto-logistic model* is similar to the classical logistic model [21] by emphasizing the explanatory variables are surrounding array

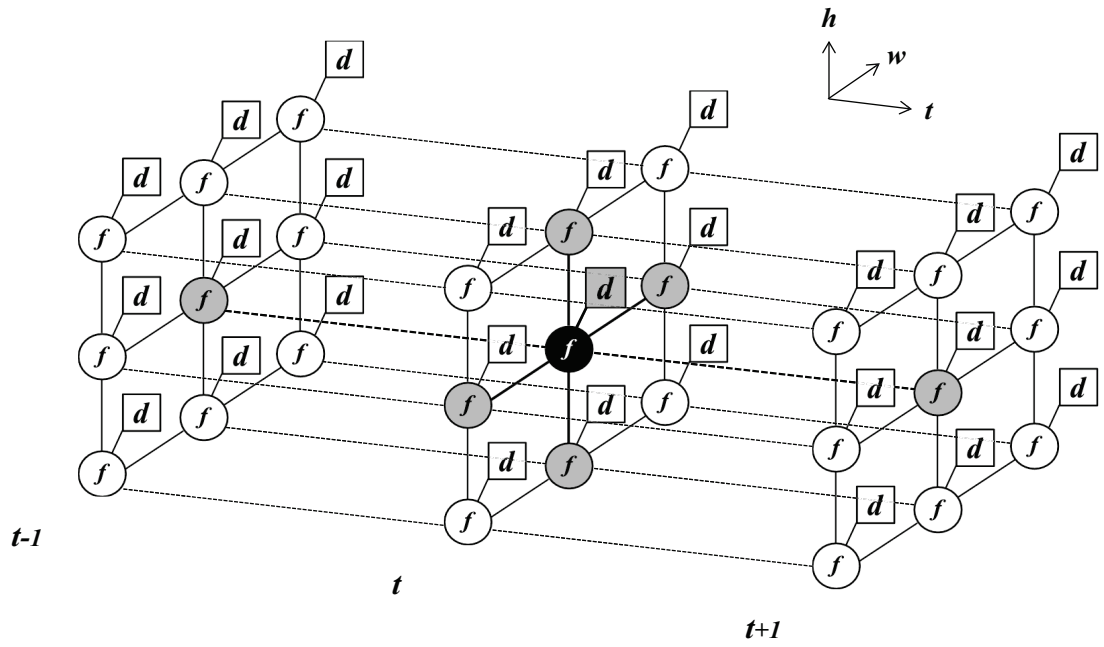


Figure 4.1: *An illustration of spatial temporal Markov random field (ST-MRF).*

variables themselves. The reasons for proposing auto-logistic model in the image segmentation prediction are: first, the image segmentation is binary in the case of interface level detection, and second, spatial temporal MRF based image segmentation considers the dependence of both spatial and temporal neighboring variables, which are equivalent to those surrounding array variables in auto-logistic model. Besag [20] was the first person who proposed to use auto-logistic model in the modeling of binary data with nearest-neighbor system. The auto-logistic model has also been applied to model some typical type of data along with spatial temporal Markov random field, for example, spatial temporal binary data [22, 23]. However, none of the existing work combines the spatial temporal MRF and auto-logistic model in the modeling and processing images. In this chapter, we propose a spatial temporal MRF based auto-logistic model to obtain the prediction of binary image segmentation as well as the interface level based on experimental sequential images. The novelty of this work is to apply ST-MRF and auto-logistic model for image processing by considering typical temporal dependence in process systems engineering.

The remainder of this chapter is organized as the follows: an overview of problem statement and framework formulation are introduced in section 4.2, the detailed mathematical formulation for image segmentation prediction is demonstrated in section 4.3, a summary of the methodology and the algorithm are given in section 4.4, the validation of proposed algorithm as well as numerical results of prediction is shown in section 4.5, and finally section 4.6 concludes this chapter.

4.2 Problem Statement and Formulation

In many conventional chemical engineering processes, the prediction of process variables at future time can be obtained using the model based on present and past process information.Therefore, the temporal neighbor set of ST-MRF model in such situations only contains the one at previous time instant. Figure 4.2 shows the graphical representation of the problem that is dealing with in this chapter. To predict the image segmentation at future time instant of $t+1$ by having the observation, d_t, and MRF inference, f_t, at present time, we propose a two-step approach to obtain the segmentation prediction, f_{t+1}. The first step is to predict the observation at $t+1$ based on the current observation, d_t, using a modified random walk model. The second step is predicting the MRF inference at $t+1$, f_{t+1}. A spatial temporal Markov random field auto-logistic model will be employed in this step by considering spatial and temporal neighborhood dependence as well as the predicted observation obtained in the previous step. For example, as shown in Figure 4.2, consider the black node labeled as f at time instant of $t+1$, the prediction is dependent on four spatial neighbors at time of $t+1$, one temporal neighbor at time of t, as well as the directly linked observation, $\hat{d}$. The observation at the next sample is predicted based on the present one. All related nodes of MRF configuration and observation in predicting black node are labeled in grey as shown in Figure 4.2.

The formulation of the framework is given as the following:

$$\begin{cases} \hat{d}_{t+1} = d_t + e_t & (4.1) \\ f_{t+1} = \mathbf{F}(\hat{d}_{t+1}, f_{N_s,t+1}, f_t) & (4.2) \end{cases}$$

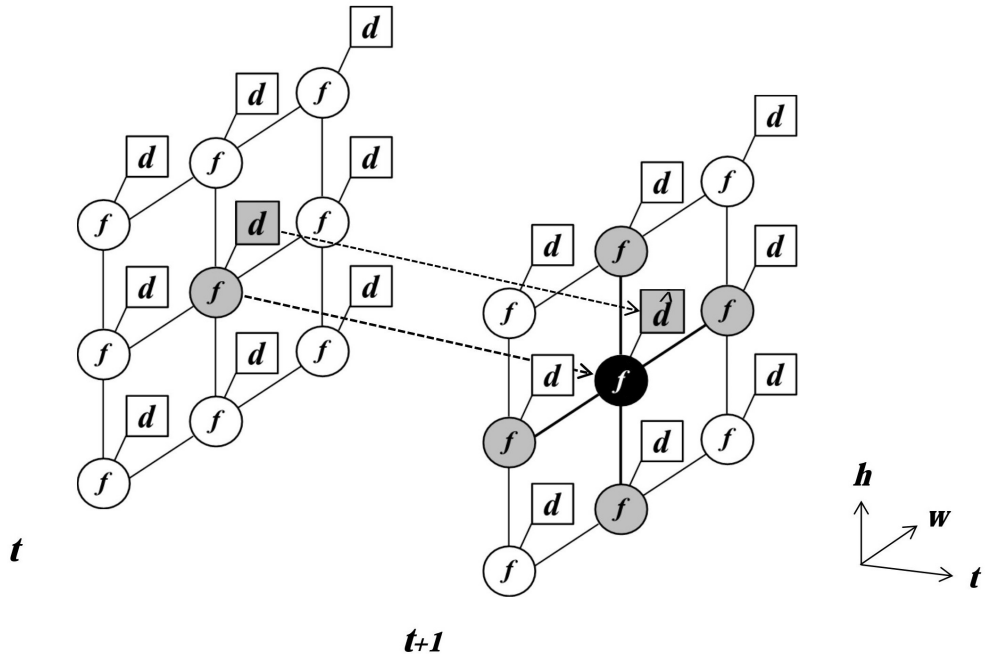


Figure 4.2: *The graphical representation of spatial temporal Markov random field based segmentation prediction.*

where equations (4.1) and (4.2) model the steps 1 and 2 as mentioned above respectively. The random walk model described in equation (4.1) will be discussed in detail in section 4.3.1. Equation (4.2) models the ST-MRF where **F** represents the logistic function and $f_{N_s,t+1}$ denotes the set of spatial neighbors. The detailed derivation of equation (4.2) will be demonstrated in section 4.3.2.

4.3 Mathematical Model for Image Segmentation Prediction

The problem of predicting image segmentation is formulated in equations (4.1) and (4.2). In this section, the detailed model development will be discussed and demonstrated. In section 4.3.1, a modified random walk model will be developed to predict the observation at one step ahead. Other than the completely random noise in conventional random walk process, the noise in this modified model is a weighted summation of three Gaussian noises. This modification is aiming to obtain a better prediction

of the observation in our specific case. The parameter estimation of three Gaussian noises is an off-line process, and then the dynamic prediction of observation can be implemented on-line. Having obtained the predicted observation, the MRF inference prediction at one step ahead can be predicted using the spatial temporal Markov random field based auto-logistic model where the parameters of the model can be estimated using generalized linear regression [22, 23]. For this step, the parameter estimation and MRF inference prediction can be implemented on-line.

4.3.1 Dynamic Observation Model based on Modified Random Walk Process

A model of predicting observation will be developed based on a modified random walk process. We will use a sequence of five images for off-line noise parameter estimation. These images were captured during one operation in every 10 seconds. The time points for five images are denoted as t_0, t_1, t_2, t_3 and , t_4. Assuming that the observed image at present time is obtained by adding a Gaussian noise on each pixel of the image at previous time instant. Therefore, we can obtain four sets of image noise and each set is computed by taking the difference of observed image pixels between time t_i and t_{i-1}, where $i \in \{1, 2, 3, 4\}$. Based on the physical property of liquid mixture, liquid is ideally homogeneous horizontally, and interface level moves in vertical direction. Therefore, the noises are plotted for every horizontal level and are shown in Figures 4.3a to 4.3d. Based on four figures, there are several observations can be made:

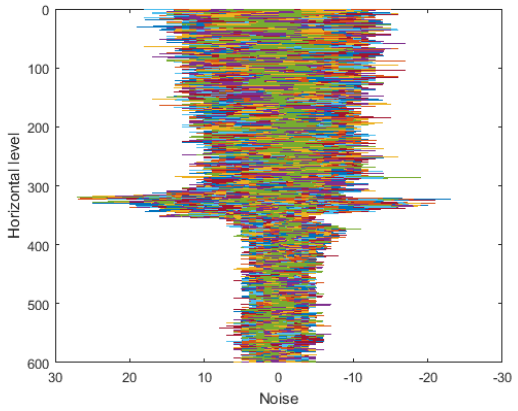
1. four noise figures have the similar profile, which means the dynamic observation can be modeled using one model under same operation condition;

2. the mean value of each level's noises are approximately zero, but variances are different at different levels;

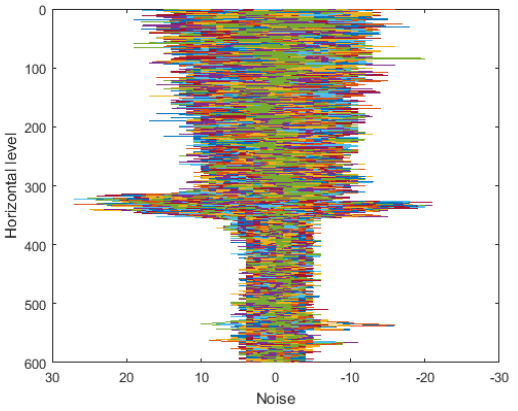
3. noises can be divided into three classes according to different variances.

Consider the noise profile between t_0 and t_1 as an example, Figure 4.4a indicates three Gaussian noises as e_1, e_2 and e_3. By comparing with the raw image at t_0 shown as

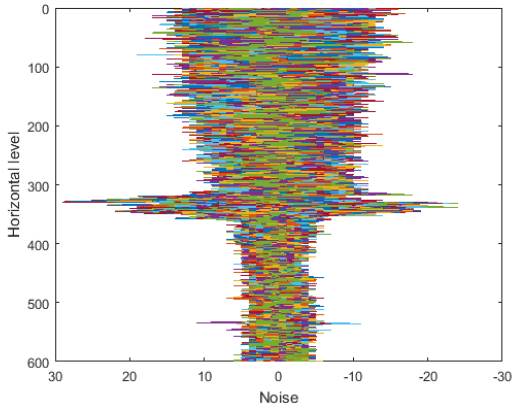
Figure 4.4b, the noise with largest variance, corresponding to the intermediate region, e_2, is sandwiched between two regions having lesser variances, e_1 and e_3, corresponding to top and bottom regions, respectively. As a result, noise at a particular level can be assumed to be composed of these three elements and therefore we are motivated to use a modified random walk process to develop the model.



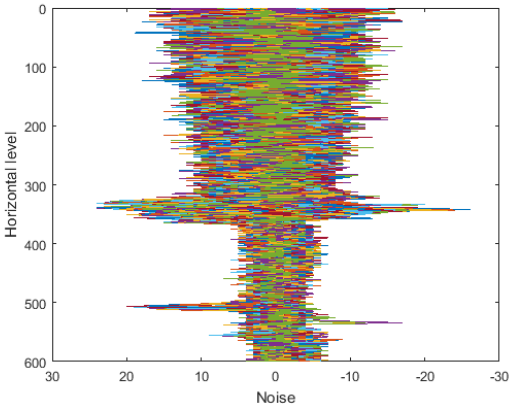
(a) *Noise profile between time step t_0 and t_1*



(b) *Noise profile between time step t_1 and t_2*



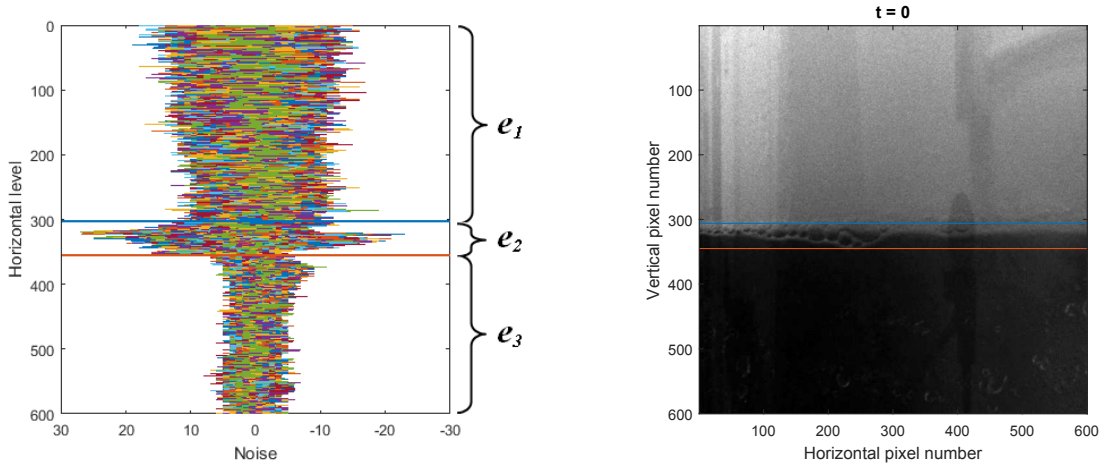
(c) *Noise profile between time step t_2 and t_3*



(d) *Noise profile between time step t_3 and t_4*

Figure 4.3: *The plots of horizontal levels' noises between each pair of contiguous images.*

As shown in equation (4.1), a random walk process is described by adding a random noise to the previous state. However, based on the noise figures in our image prediction, we propose a noise of weighted summation rather than a single noise



(a) *Three Gaussian noises indication* **(b)** *The raw image at* t_0

Figure 4.4: *The indication of three Gaussian noises based on horizontal levels.*

source. The overall noise model can be expressed as:

$$e_t = w_{1,t}e_1 + w_{2,t}e_2 + w_{3,t}e_3 \tag{4.3}$$

where e_1, e_2, e_3 are Gaussian noises and $w_{1,t}$, $w_{2,t}$, $w_{3,t}$ are the corresponding weights. Weights are self defined and aiming to obtain the best noise term at each horizontal level. It has to be noted that weights are a function of time, and they will be different when the time is propagating. Furthermore, weights are a function of levels, which mean we are predicting the observed image level by level. This is reasonable because the noises are distinct at different levels, and the appropriate noise could be computed using the equation (4.3) for each level. Our self defined weights can be computed using the following equation:

$$w_{n,t}^i = \frac{\exp(-(\eta_t^i - m_{n,t})^2)}{\sum_n \exp(-(\eta_t^i - m_{n,t})^2)} \quad n = 1, 2, 3 \quad i \in [1, 600] \tag{4.4}$$

where i is the horizontal level index, η_t^i is the averaged pixel value of i'th level at time t, and $m_{n,t}$ is the mean of observed pixel values corresponding to the three sections where three Gaussian noises are distributed. The values of $m_{n,t}$ can be obtained based on the Gaussian mixture distribution fitting, which has been shown in previous chapter. Taking the image at time instant of t_0 as an example, the observation

distribution and fitted Gaussian mixture distribution are shown in Figure 4.5. The selection of m_1, m_2, and m_3 are also indicated on the graph. The values of m_1 and m_3 are selected as the modes of the Gaussian mixture distribution, and m_2 is defined as the pixel intensity that has the smallest probability between m_1 and m_3. Figure 4.6 shows the vertical profile of averaged pixel value for every horizontal level, η_t. The data at time t_0 is used as an example for the illustration. According to the principle of mean values selection as mentioned above, m_1, m_2, and m_3 are shown in the figure of η_t. It clearly shows that three mean values are individually distributed into each corresponding section.

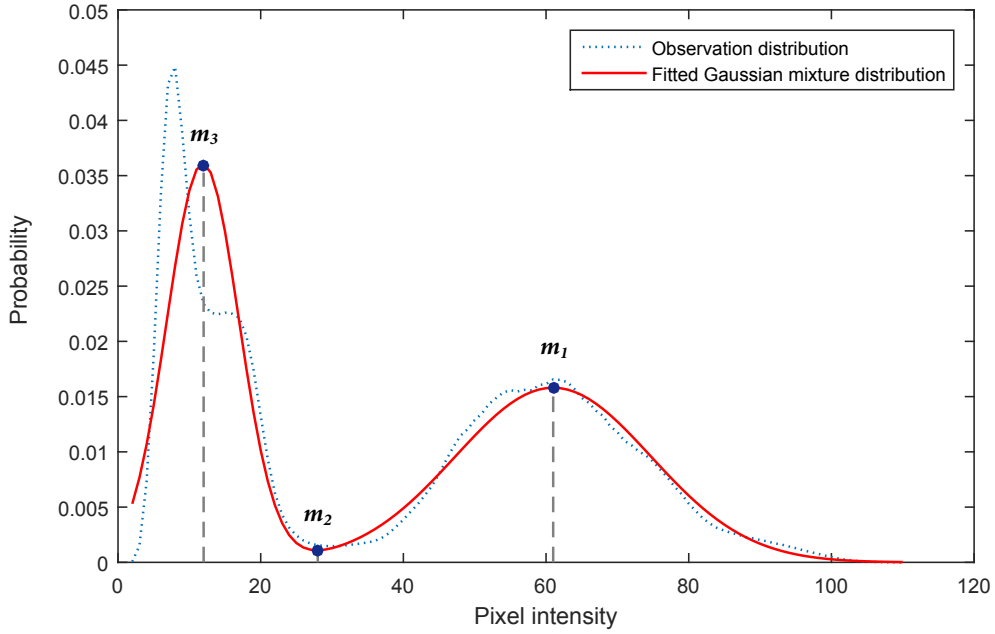


Figure 4.5: *The observation distribution and fitted Gaussian mixture distribution with the indication of three mean values.*

Having determined the weights in equation (4.3), the parameters of three Gaussian noises will be estimated next. All noise data that were shown in Figures 4.3a to 4.3d are used in parameter estimation. A Gaussian mixture model (GMM) is assumed and parameters are estimated using Expectation Maximization (EM) algorithm. Let us denote the overall noise as $\varepsilon = \{e_j, j = 1, 2, ..., M\}$; then the Gaussian mixture

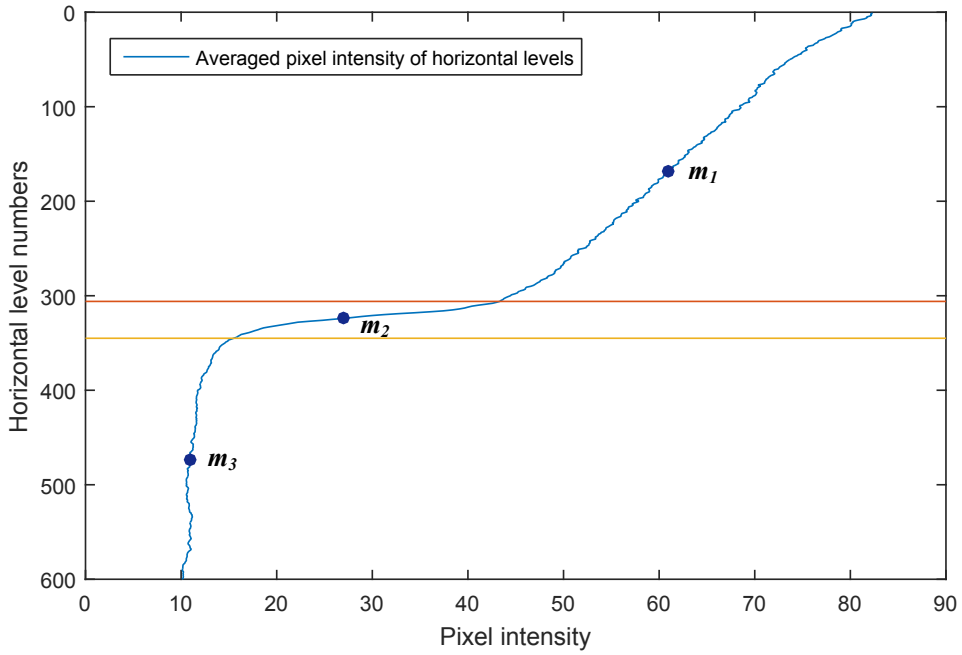


Figure 4.6: *The vertical profile of averaged pixel value for every horizontal level. Determined three mean values, m_1, m_2, m_3 are also indicated.*

model can be expressed as:

$$P(\varepsilon \mid \Theta) = \sum_{k=1}^{3} \alpha_n P_n(\varepsilon \mid \theta_n) \tag{4.5}$$

where α_n are the GMM weights, $P_n(\varepsilon \mid \theta_n)$ are Gaussian noises that can be individually formulated as:

$$P_n(e_j \mid \theta_n) = \frac{1}{\sigma_n \sqrt{2\pi}} \exp\left(- \frac{(e_j - \mu_n)^2}{2\sigma_n^2} \right) \tag{4.6}$$

and $\theta_n = \{\mu_n, \sigma_n\}$ are Gaussian noises parameters. To use EM algorithm for GMM parameter estimation, a latent variable set has to be defined. Denoting y as the latent variable for each noise, the Q-function of the expectation step can be written as:

$$\begin{aligned} Q(\Theta, \Theta^{old}) &= E\big[\log P(\varepsilon, y \mid \Theta) \mid \varepsilon, \Theta^{old} \big] \\ &= \sum_{y \in Y} \log(\mathcal{L}(\Theta \mid \varepsilon, y)) P(y \mid \varepsilon, \Theta^{old}) \end{aligned} \tag{4.7}$$

where the log-likelihood expression can be expressed as:

$$\log(\mathcal{L}(\Theta \mid \varepsilon, y)) = \sum_{j=1}^{M} \log(\alpha_{y_j} P_{y_j}(e_j \mid \theta_{y_j})) \tag{4.8}$$

and the conditional probability of unobserved data is:

$$P(y \mid \varepsilon, \Theta^{old}) = \prod_{j=1}^{M} P(y_j \mid e_j, \Theta^{old}) \tag{4.9}$$

Combining equations (4.7), (4.8) and (4.9), the Q-function can be defined as:

$$Q(\Theta, \Theta^{old}) = \sum_{y \in Y} \sum_{j=1}^{M} \log(\alpha_{y_j} P_{y_j}(e_j \mid \theta_{y_j})) \prod_{j=1}^{M} P(y_j \mid e_j, \Theta^{old}) \tag{4.10}$$

Same as the derivation of EM algorithm in chapter 3, a *dirac delta* function:

$$\delta_{a,b} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases} \tag{4.11}$$

is introduced in the Q-function derivation. Therefore, equation (4.10) can be rewritten as:

$$\begin{aligned} Q(\Theta, \Theta^{old}) &= \sum_{y \in Y} \sum_{j=1}^{M} \sum_{n=1}^{3} \delta_{n,y_j} \log(\alpha_n P_n(e_j \mid \theta_n)) \prod_{j=1}^{M} P(y_j \mid e_j, \Theta^{old}) \\ &= \sum_{j=1}^{M} \sum_{n=1}^{3} \log(\alpha_n P_n(e_j \mid \theta_n)) \sum_{y \in Y} \delta_{n,y_j} \prod_{j=1}^{M} P(y_j \mid e_j, \Theta^{old}) \end{aligned} \tag{4.12}$$

It can be proved that [14]

$$\sum_{y \in Y} \delta_{n,y_j} \prod_{j=1}^{M} P(y_j \mid e_j, \Theta^{old}) = P(n \mid e_j, \Theta^{old}) \tag{4.13}$$

Finally, the Q-function of E-step can be expressed as:

$$\begin{aligned} Q(\Theta, \Theta^{old}) &= \sum_{j=1}^{M} \sum_{n=1}^{3} \log(\alpha_n P_n(e_j \mid \theta_n)) P(n \mid e_j, \Theta^{old}) \\ &= \sum_{j=1}^{M} \sum_{n=1}^{3} \log(\alpha_n) P(n \mid e_j, \Theta^{old}) + \sum_{j=1}^{M} \sum_{n=1}^{3} \log(P_n(e_j \mid \theta_n)) P(n \mid e_j, \Theta^{old}) \end{aligned} \tag{4.14}$$

The detailed derivation of the maximization step can also be found in [14]. The estimation of new parameters in terms of old parameters can be summarized as follows:

$$\mu_n^{new} = \frac{\sum_{j=1}^{M} e_j P(n \mid e_j, \Theta^{old})}{\sum_{j=1}^{M} P(n \mid e_j, \Theta^{old})} \tag{4.15}$$

$$(\sigma^2)_n^{new} = \frac{\sum_{j=1}^{M} P(n \mid e_j, \Theta^{old})\left(e_j - \mu_n^{new}\right)\left(e_j - \mu_n^{new}\right)^T}{\sum_{j=1}^{M} P(n \mid e_j, \Theta^{old})} \tag{4.16}$$

The EM algorithm is an iterative process, as it updates the parameters based on the old estimation in previous iteration until the estimates converge. The eventual results of Gaussian noise parameter estimation are tabulated in Table 4.1. We can see that the means of three distributions are close to zero, and the variances of three distributions agree with the real noises that are shown in Figure 4.3. The overall noise distribution, three estimated Gaussian components and the mixture distribution are plotted in Figure 4.7. This figure shows the agreement between the overall noise distribution and the mixture of three estimated Gaussian noise distributions.

Table 4.1: *Estimated parameters of three Gaussian components in the noise model.*

Parameters	Noise Components		
	e_1	e_2	e_3
mean, μ	0.18	0.96	0.01
variance, σ^2	10.40	24.41	3.06

4.3.2 Spatial Temporal Markov Random Field based Auto-logistic Model

In this section, we propose to use spatial temporal Markov random field in the modeling of image segmentation prediction. To achieve the prediction of image segmentation in time-series, we add the temporal dependence to the model. Since the MRF inference, which is namely the segmentation of image, is binary in our interface estimation problem, we propose the auto-logistic model for such spatial temporal binary

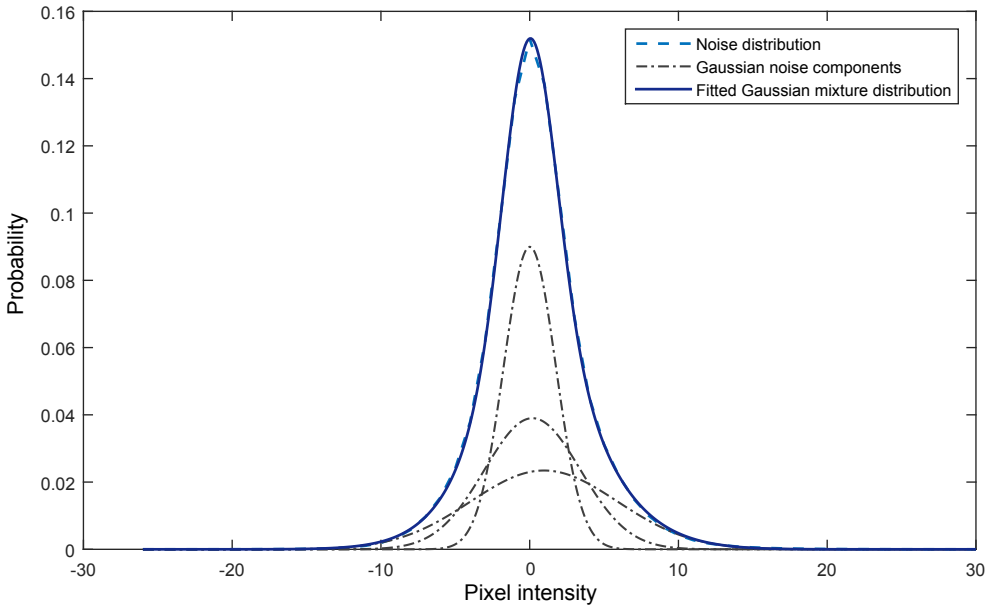


Figure 4.7: *The estimated Gaussian mixture distribution for entire noise distribution as well as three Gaussian components.*

data. A key assumption is that the model we propose follows Markov chain across time for each lattice and the transition probability is auto-logistic [22]. Therefore, the transition probability of an $(h \times w)$-dimensional matrix Markov chain can be defined as:

$$Pr(f_t \mid f_{t-1}) = \frac{1}{C} q(f_t \mid f_{t-1}) \tag{4.17}$$

where C is a normalizing constant and $q(f_t \mid f_{t-1})$ can be defined as:

$$\begin{aligned} q(f_t \mid f_{t-1}) \equiv \exp \Bigg\{ & \theta_{d,t} \sum_{s=1}^{M} \hat{d}_{s,t} f_{s,t} \\ & + \theta_{spa,t} \sum_{s=1}^{M} \sum_{s' \in N_s} \Big[f_{s,t} f_{s',t} + (1 - f_{s,t})(1 - f_{s',t}) \Big] \\ & + \theta_{tem,t} \sum_{s=1}^{M} \Big[f_{s,t} f_{s,t-1} + (1 - f_{s,t})(1 - f_{s,t-1}) \Big] \Bigg\} \end{aligned} \tag{4.18}$$

Here $\hat{d}_{s,t}$ is the predicted observation that was obtained based on the model in previous section, f denotes the random variable in MRF and takes binary value from the set

of $\{0,1\}$. The subscript s represents the number of site, and N_s is the set of spatial neighbors of site s. We propose to use conventional first-order neighborhood system, which means that for a given site at i-th row and j-th column in spatial dimension, $N_s(i,j) = \{f_{i-1,j}, f_{i,j+1}, f_{i+1,j}, f_{i,j-1}\}$. $\theta_{d,t}$ is the parameter of observation, $\theta_{spa,t}$ is the coefficient corresponding to spatial neighborhood, and $\theta_{tem,t}$ corresponds to the temporal dependence.

For a given starting time instant of t_0, the joint distribution of $f_{t_0+1}, ..., f_t$ conditional on f_{t_0} is:

$$\begin{aligned} Pr(f_{t_0+1}, ..., f_t \mid f_{t_0}) = & \prod_{t'=t_0+1}^{t} Pr(f_{t'} \mid f_{t'-1}) \\ = & \frac{1}{C_{t-t_0}} \prod_{t'=t_0+1}^{t} q(f_{t'} \mid f_{t'-1}) \\ = & \frac{1}{C_{t-t_0}} \exp\left\{ \sum_{t'=t_0+1}^{t} \left(\theta_{d,t'} \sum_{s=1}^{M} \hat{d}_{s,t'} f_{s,t'} \right.\right. \\ & + \theta_{spa,t'} \sum_{s=1}^{M} \sum_{s' \in N_s} \left[f_{s,t'} f_{s',t'} + (1 - f_{s,t'})(1 - f_{s',t'}) \right] \\ & \left.\left. + \theta_{tem,t'} \sum_{s=1}^{M} \left[f_{s,t'} f_{s,t'-1} + (1 - f_{s,t'})(1 - f_{s,t'-1}) \right] \right) \right\} \end{aligned} \tag{4.19}$$

Next, we will derive the full conditional probability for each random variable. For the lattice at s-th site and t-th time point, $f_{s,t}$, according to Bayes rule, the full conditional probability in terms of the joint distribution in equation (4.19) can be written as:

$$\begin{aligned} Pr(f_{s,t} \mid \{f_{t_0}, f_{t_0+1}, ..., f_t\} \setminus f_{s,t}) &= \frac{Pr(f_{s,t}, \{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0})}{Pr(\{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0})} \\ &= \frac{Pr(f_{t_0+1}, ..., f_t \mid f_{t_0})}{Pr(\{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0})} \end{aligned} \tag{4.20}$$

where the denominator in equation (4.20) is the integration of joint distribution over

$f_{s,t}$,

$$
\begin{aligned}
&Pr(\{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0}) \\
&\quad = \sum_{f_{s,t}} Pr(f_{t_0+1}, ..., f_t \mid f_{t_0}) \\
&\quad = \sum_{f_{s,t}} Pr(f_{s,t}, \{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0}) \\
&\quad = Pr(f_{s,t} = 0, \{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0}) + Pr(f_{s,t} = 1, \{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0})
\end{aligned} \tag{4.21}
$$

Therefore, the full conditional probability of $f_{s,t}$ can be expressed as:

$$
\begin{aligned}
&Pr(f_{s,t} \mid \{f_{t_0}, f_{t_0+1}, ..., f_t\} \setminus f_{s,t}) \\
&\quad = \frac{Pr(f_{t_0+1}, ..., f_t \mid f_{t_0})}{Pr(f_{s,t} = 0, \{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0}) + Pr(f_{s,t} = 1, \{f_{t_0+1}, ..., f_t\} \setminus f_{s,t} \mid f_{t_0})}
\end{aligned} \tag{4.22}
$$

Let us express the right-hand-side of equation (4.22) in the form of

$$
\frac{\exp\big(A + B(f_{s,t})\big)}{\exp\big(A + B(f_{s,t} = 0)\big) + \exp\big(A + B(f_{s,t} = 1)\big)} \tag{4.23}
$$

where

$$
\begin{aligned}
A \equiv \sum_{t'=t_0+1}^{t} \Bigg(& \theta_{d,t'} \sum_{\tilde{s} \neq s} \hat{d}_{\tilde{s},t'} f_{\tilde{s},t'} + \theta_{spa,t'} \sum_{\tilde{s} \neq s} \sum_{s' \in N_{\tilde{s}}} \Big[f_{\tilde{s},t'} f_{s',t'} + (1 - f_{\tilde{s},t'})(1 - f_{s',t'}) \Big] \\
& + \theta_{tem,t'} \sum_{\tilde{s} \neq s} \Big[f_{\tilde{s},t'} f_{\tilde{s},t'-1} + (1 - f_{\tilde{s},t'})(1 - f_{\tilde{s},t'-1}) \Big] \Bigg)
\end{aligned} \tag{4.24}
$$

consists of the terms not involving $f_{s,t}$, and

$$
\begin{aligned}
B(f_{s,t}) \equiv \quad & \theta_{d,t} \hat{d}_{s,t} f_{s,t} + \theta_{spa,t} \sum_{s' \in N_s} \Big[f_{s,t} f_{s',t} + (1 - f_{s,t})(1 - f_{s',t}) \Big] && (4.25) \\
& + \theta_{tem,t} \Big[f_{s,t} f_{s,t-1} + (1 - f_{s,t})(1 - f_{s,t-1}) \Big] && (4.26)
\end{aligned}
$$

consists of the terms involving $f_{s,t}$. Consequently,

$$
B(f_{s,t} = 0) = \theta_{spa,t} \sum_{s' \in N_s} (1 - f_{s',t}) + \theta_{tem,t}(1 - f_{s,t-1}) \tag{4.27}
$$

and

$$B(f_{s,t}=1)=\theta_{d,t}\hat{d}_{s,t}+\theta_{spa,t}\sum_{s'\in N_s}f_{s',t}+\theta_{tem,t}f_{s,t-1} \tag{4.28}$$

Due to equations (4.27) and (4.28), the expression in equation (4.23) can be rearranged as:

$$\begin{aligned} LHS=&\frac{\frac{\exp\big(A+B(f_{s,t})\big)}{\exp\big(A+B(f_{s,t}=0)\big)}}{\frac{\exp\big(A+B(f_{s,t}=0)\big)}{\exp\big(A+B(f_{s,t}=0)\big)}+\frac{\exp\big(A+B(f_{s,t}=1)\big)}{\exp\big(A+B(f_{s,t}=0)\big)}}\\ =&\frac{\exp\big(B(f_{s,t})-B(f_{s,t}=0)\big)}{1+\exp\big(B(f_{s,t}=1)-B(f_{s,t}=0)\big)} \end{aligned} \tag{4.29}$$

Therefore, the full conditional probability can be finally expressed as:

$$\begin{aligned} &Pr(f_{s,t}\mid\{f_{t_0},...,f_t\}\setminus f_{s,t})\\ &\quad=\frac{\exp\big\{\theta_{d,t}\hat{d}_{s,t}f_{s,t}+\theta_{spa,t}\sum_{s'\in N_s}f_{s,t}(2f_{s',t}-1)+\theta_{tem,t}f_{s,t}(2f_{s,t-1}-1)\big\}}{1+\exp\big\{\theta_{d,t}\hat{d}_{s,t}+\theta_{spa,t}\sum_{s'\in N_s}(2f_{s',t}-1)+\theta_{tem,t}(2f_{s,t-1}-1)\big\}} \end{aligned} \tag{4.30}$$

The full conditional probability can be simplified as the following expression due to the assumption of Markov property in MRF:

$$Pr(f_{s,t}\mid\{f_{t_0},...,f_t\}\setminus f_{s,t})=Pr(f_{s,t}\mid f_{s',t}:(s',t)\in N_{s,t}) \tag{4.31}$$

where $N_{s,t}$ is the set of neighboring sites in both spatial and temporal dimension. Finally, the Markov random field based auto-logistic model can be developed as:

$$\begin{aligned} &Pr(f_{s,t}\mid f_{s',t}:(s',t)\in N_{s,t})\\ &\quad=\frac{\exp\big\{\theta_{d,t}\hat{d}_{s,t}f_{s,t}+\theta_{spa,t}\sum_{s'\in N_s}f_{s,t}(2f_{s',t}-1)+\theta_{tem,t}f_{s,t}(2f_{s,t-1}-1)\big\}}{1+\exp\big\{\theta_{d,t}\hat{d}_{s,t}+\theta_{spa,t}\sum_{s'\in N_s}(2f_{s',t}-1)+\theta_{tem,t}(2f_{s,t-1}-1)\big\}} \end{aligned} \tag{4.32}$$

Having specified the model structure, the next step is to estimate model parameters. Preliminarily, a key assumption made in parameter estimation is that for a MRF inference at a specific time point, model parameters are same for every site in the random field. Generaly, *generalized linear regression* (GLR) is commonly used in parameter estimation of auto-logistic model [22, 23]. Generalized linear regression

usually transfers non-linear regression problem into linear regression through a *link function* with the corresponding distribution of random variables [24]. Due to the binary variable in logistic model, binomial distribution and *logit* link function will be used in the general linear regression. The *logit* link function is defined as:

$$X \cdot \beta = \log\left(\frac{\kappa}{1-\kappa}\right) \tag{4.33}$$

where

$$\kappa = \frac{\exp(X \cdot \beta)}{1 + \exp(X \cdot \beta)} \tag{4.34}$$

By comparing equation (4.34) with equation (4.32), we can obviously conclude that κ is the probability when random variable, $f_{s,t}$, is equal to 1,

$$\kappa = Pr(f_{s,t} = 1 \mid f_{s',t} : (s',t) \in N_{s,t}) \tag{4.35}$$

and then,

$$1 - \kappa = Pr(f_{s,t} = 0 \mid f_{s',t} : (s',t) \in N_{s,t}) \tag{4.36}$$

Therefore, the *logit* link function can also be expressed as:

$$X \cdot \beta = \log\left(\frac{Pr(f_{s,t} = 1 \mid f_{s',t} : (s',t) \in N_{s,t})}{Pr(f_{s,t} = 0 \mid f_{s',t} : (s',t) \in N_{s,t})}\right) \tag{4.37}$$

The parameter set contains all parameters that were mentioned in ST-MRF model as well as an intercept parameter θ_0; therefore, the entire set can be expressed as $\beta = \{\theta_0, \theta_{d,t}, \theta_{spa,t}, \theta_{tem,t}\}$. Correspondingly, the regressor X is the combination of observation, spatial neighbors, the temporal neighbor, as well as a constant, $X = \{1, \hat{d}_{s,t}, (2f_{N_s,t} - 1), (2f_{s,t-1} - 1)\}$. An important note needs to be emphasized is that the model of the image at the start point t_0 is an exception of the ST-MRF modeling because no temporal dependence can be considered for initial segmentation. However, we still use the MRF based auto-logistic model by eliminating the temporal term in both parameter set and regressor.

Finally, since MRF estimation is dependent on spatial neighbors, an initial guess of MRF is always required to start the estimation of MRF at every time point. A method to provide improvised initial guess which is dependent on previous MRF

estimation will be mentioned and discussed in section 4.5.2. The parameter estimation is iteratively updated until the random field inference converges. For every iteration, once the parameters are estimated, κ^{new} can be updated by using equation (4.34) and the updated MRF estimation can be obtained by using the following strategy:

$$f_{s,t}^{new} = \begin{cases} 1 & \text{if } \kappa^{new} \approx 1 \\ 0 & \text{if } \kappa^{new} \approx 0 \end{cases} \tag{4.38}$$

4.4 Summary of Methodology

To summarize the mathematical model that was developed in previous two sections, we start from the framework we proposed in section 4.2. We proposed a two-step algorithm to estimate the prediction of MRF inference. A graphical representation of proposed framework in estimating Markov random field across time is shown in Figure 4.8. The dash arrows represent the first step, which is predicting current observation based on the true observation at previous time instance. All other solid arrows indicate the second step of the estimation. It can be summarized that, as time propagates, the MRF estimation is dependent on the predicted observation and the initial guess of MRF at current time point, as well as the MRF inference at previous time step. Both predicted observation and the initial guess are obtained based on the true observation and estimated MRF at previous time instant, respectively.

Regarding as the first step, the observation prediction, the detailed algorithm was provided in section 4.3.1 and the procedure of prediction is shown in Figure 4.9. From the flow chart, we can summarize that:

1. once the true observation d_t was obtained, a Gaussian mixture distribution was fitted and three mean values m_1, m_2, m_3 were determined based on the fitted Gaussian mixture distribution;

2. then the self-defined weights of three Gaussian noises were calculated for each horizontal level using equation (4.4);

3. the overall noise, e_t, which is the weighted summation of three Gaussian noises,

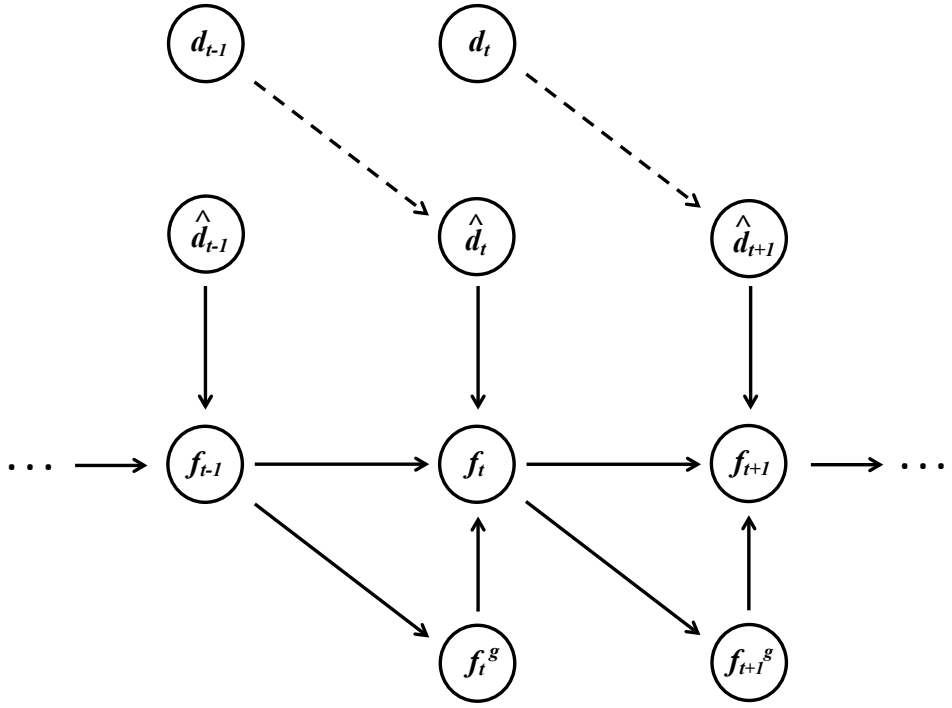


Figure 4.8: *The graphical representation of proposed framework in estimating Markov random field across time. Dashed arrows represent the first observation prediction step as expressed in equation (4.1), solid arrows represent the second MRF estimation step as expressed in equation (4.2). As time propagates, the MRF estimation is dependent on the predicted observation and the initial guess of MRF at current time point, as well as the MRF inference at previous time point. Both predicted observation and the initial guess are obtained based on the true observation and estimated MRF at previous time point respectively.*

was computed using equation (4.3), and parameters of three Gaussian noises e_1, e_2, e_3 were estimated off-line;

4. finally, the predicted observation for future time was obtained by using random walk model in equation (4.1), which adds the computed noise on current true observation.

For the second step, Markov random field estimation based on ST-MRF autologistic model, the detailed derivation of the model and parameter estimation were demonstrated in section 4.3.2. A flow chart of the procedure on how to develop the

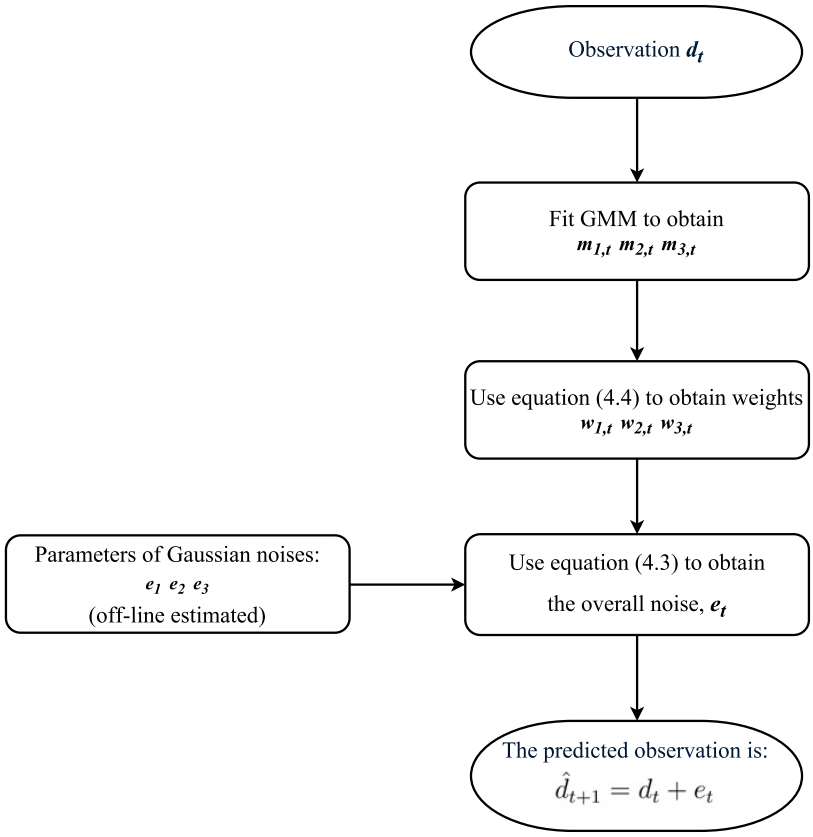


Figure 4.9: *The detailed procedure of observation prediction.*

ST-MRF auto-logistic model as well as how to estimate model parameters is illustrated in Figure 4.10. The spatial temporal Markov random field auto-logistic model of predicting MRF inference at future time was initially built based on predicted observation, $\hat{d}_{t+1}$, the present MRF estimation, f_t, and the initial guess, f^g_{t+1}. Then, generalized linear regression was performed to estimate model parameters, β^{new}. The MRF inference was updated based on new estimated parameters. If the MRF estimation did not converge, the estimation of random field from previous iteration, f^{old}_{t+1}, was used in developing model and estimating new parameters. Once the MRF estimation converges, the final estimation of future time,f_{t+1}, was obtained. For the MRF estimation at initial time instant of t_0, the procedure of estimating f_{t_0} was similar as the procedure described above, but eliminated the consideration of temporal dependence due to the lack of temporal dependence for the starting image.

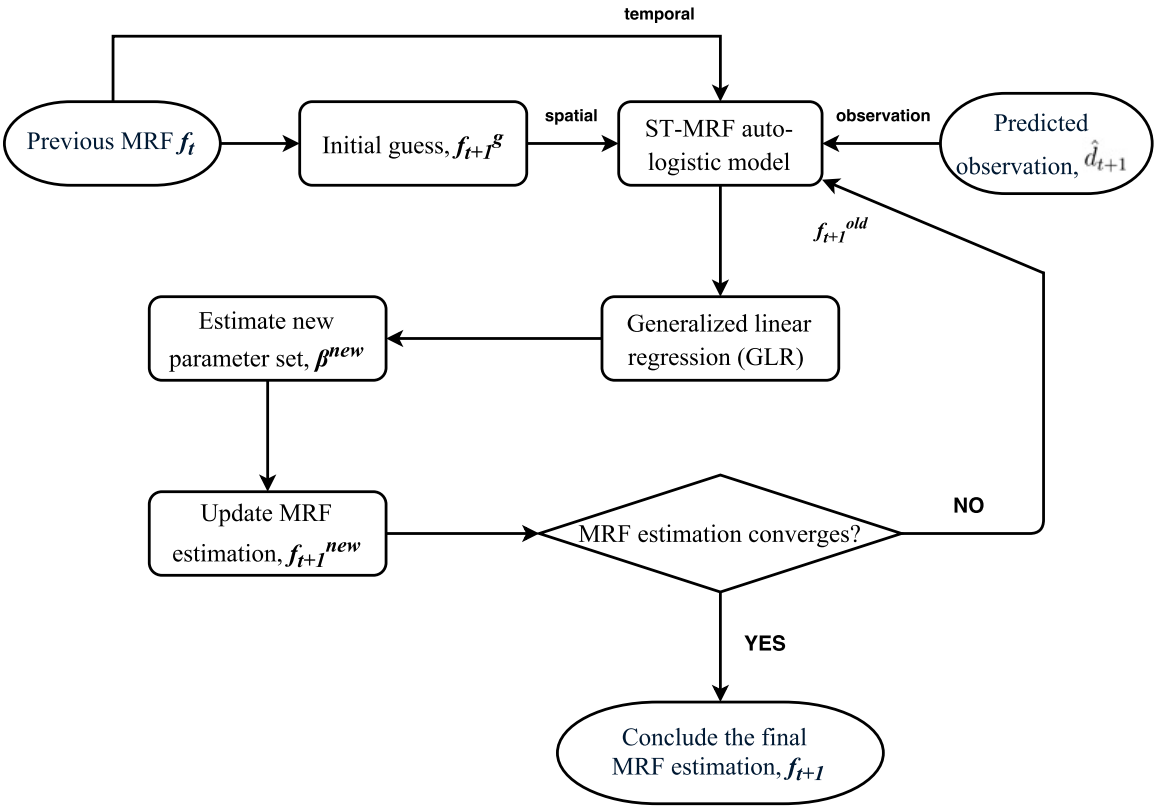


Figure 4.10: *The detailed schematic diagram of Markov random field estimation based on ST-MRF auto-logistic model.*

4.5 The Prediction of Image Segmentation for Interface Level Estimation

In this section, two sets of sequential images were obtained using the same experimental setup, and are used to validate the proposed methodology of predicting image segmentation for interface level estimation in this chapter. The experimental design and setup are shown in chapter 1. Two sets of sequential images were obtained under different operating conditions: one for total flow rate $1.25L/min$ and the other for total flow rate $1.75L/min$, respectively. For convenience of notation, we denote the total flow rate of $1.25L/min$ as *low flow rate*, and the total flow rate of $1.75L/min$ as *high flow rate*. Each set of images has five images in total, and the time instants of five images are denoted as t_0, t_1, t_2, t_3, and t_4. The images at t_0 are starting images, segmentation prediction starts from the time point of t_1. MATLAB was employed to

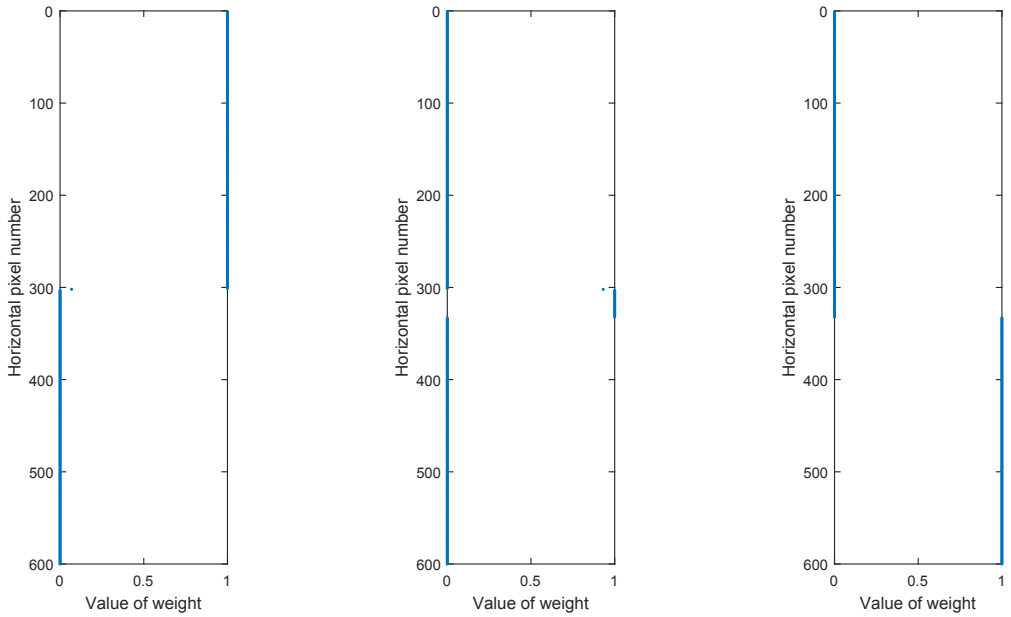
perform image processing and plot figures.

According to the algorithm, observation prediction will be validated first in section 4.5.1. Then section 4.5.2 will show the prediction of image segmentation for estimating interface level by comparing with the results that were obtained using unsupervised algorithm in chapter 3. The improved initial guess for this oil-water interface level estimation will also be discussed in section 4.5.2.

4.5.1 Validation of Observation Prediction

In section 4.3.1, we introduced the methodology for predicting observation. To obtain a better prediction, we proposed a random walk model with modified noise term. Based on the off-line data training, the noise was determined as a form of weighted summation of Gaussian distributions which is expressed in equation (4.3). The weights were self-defined as a function of level, and were also changed as time propagates. The goal of defining such weights was to explicitly assign the corresponding noise to each level of image. Taking the data at time t_1 as an example, to confirm the validation of our self-defined weights, the vertical profiles of three weights for every horizontal level are shown in Figures 4.11a, 4.11b, and 4.11c, respectively. It is shown that nearly all the values for three weights are either 0 or 1, which means among three noises: e_1, e_2, and e_3, only the one that has weight of 1 is added for each level. The weight of e_1 has the value of 1 from the top up to a level of 300-ish, and 0 for the rest; w_2 has the value of 1 for only about 30 levels in the middle; and w_3 equal to 1 for those levels at which w_1 and w_2 are equal to 0. These results match the noise observation that is shown in Figure 4.4a in section 4.3.1. Therefore, the self-defined weights are valid for predicting observed images.

As the next step, we validate whether the predicted observation agree with true observation or not. Two figures are plotted to show the comparison between predicted observation and true observation in pixel intensities for entire image as well as the overall distribution of pixel intensities. Figures 4.12a and 4.12b show the pixel intensities of entire image in a 3-dimensional view. Figure 4.12a shows the true observation at time instant of t_1, and Figure 4.12b shows the predicted observation for



(a) w_1, *weight of* e_1 **(b)** w_2, *weight of* e_2 **(c)** w_3, *weight of* e_3

Figure 4.11: *The plots of weights of three Gaussian noises for all levels.*

t_1 using the developed model based on the true observation at time instant of t_0. We can conclude that the prediction is acceptable because the overall pixel intensities distribution is matched. The predicted image slightly appears noisy due to the added random noise. Figure 4.13 shows the overall distributions of pixel intensities for both true and predicted observation at a same time point. These figures are based on the data at t_2. Same as the comparison shown in Figure 4.12, the overall distributions agreed with each other. However, the distribution of predicted observation is slightly smoother. This might be caused by the off-line estimated noises that are based on entire data set rather than the noise at present time.

4.5.2 Image Segmentation Prediction

The main objective of this chapter was to predict the segmentation of future time point based on observed image and segmentation at present time. To validate the algorithm which was introduced in section 4.3.2, we demonstrate the comparison

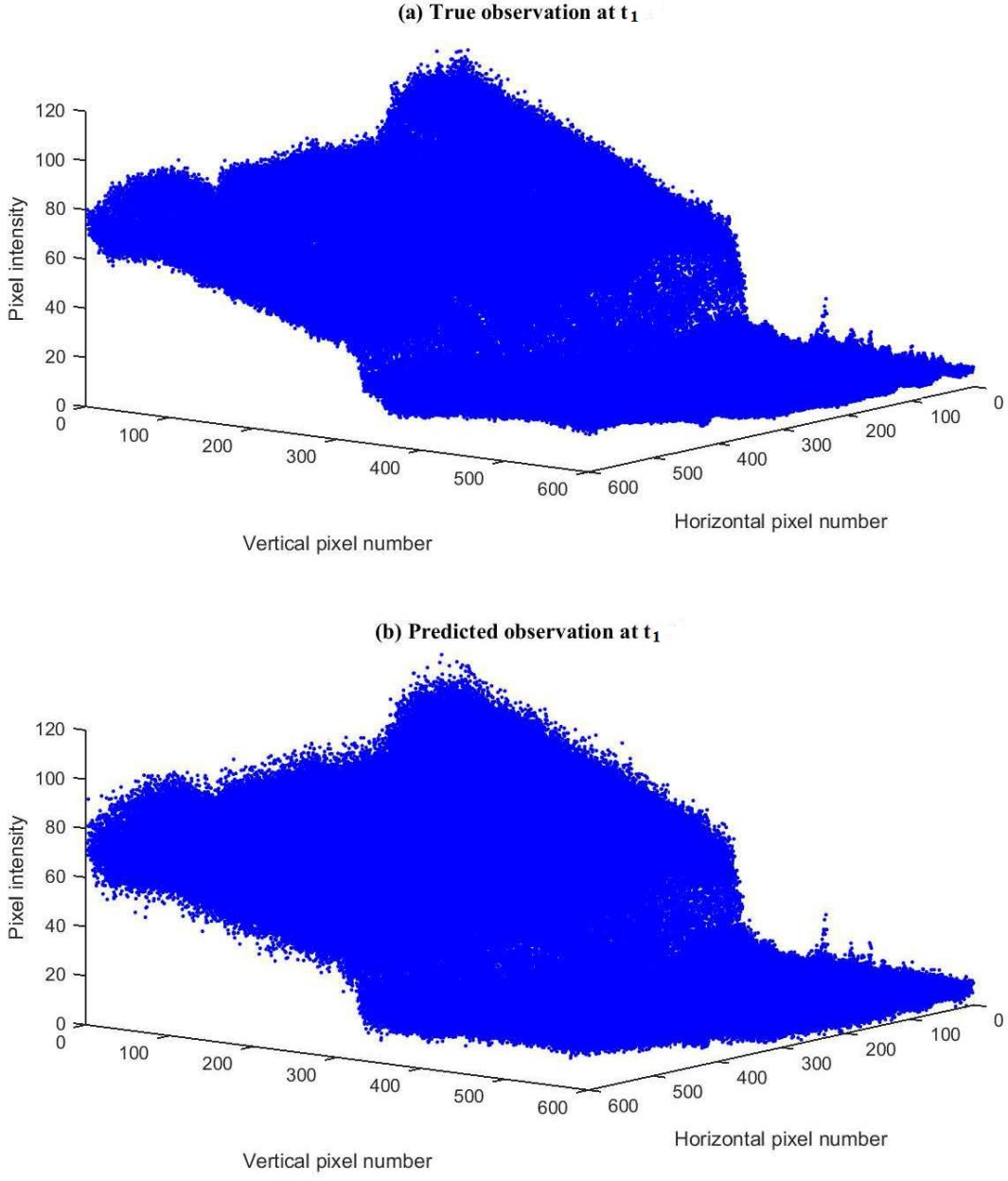


Figure 4.12: *The compare between (a) true observation and (b) predicted observation on pixel intensities of entire image. Figures are plotted using the data at* t_1 *as an example.*

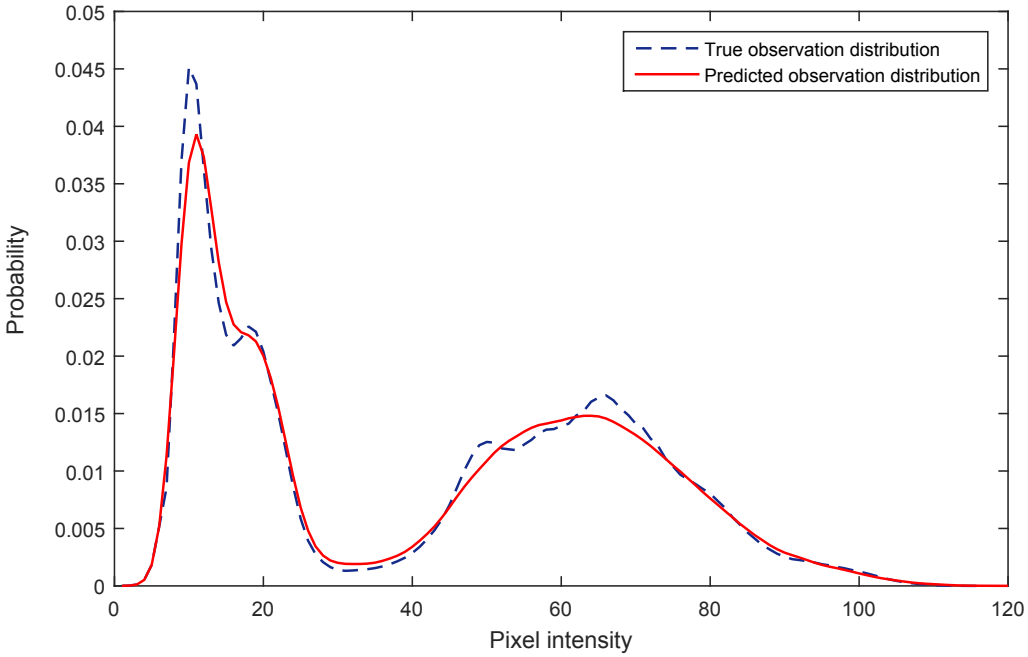


Figure 4.13: *Pixel intensities' distribution of true observation and predicted observation. Figures are plotted using the data at t_2 as an example.*

between the predicted segmentation as well as *true* segmentation at the same time point. The *true* segmentation in this section refers to the segmentation result obtained by using the unsupervised algorithm in chapter 3. Initially, a way to provide a good initial guess of the segmentation will be discussed and demonstrated.

Based on the off-line data training experience, we noticed that the initial guess of the segmentation is fairly important for producing a good prediction outcome. Generally, there are two options for initial guess: the first option is to use the exact random field estimation at previous time point, and another option is to generate a totally randomized initialization. For option one, the main problem of using such initial guess is the high correlation between initialization and the temporal neighbors. As a result, the outcome of MRF prediction is almost same as the previous MRF estimation, which means the prediction is poor based on such initial guess. For the second option, randomized initialization could occasionally results in a good prediction of segmentation; however, contingency is the main disadvantage of using such

initialization.

By combining the advantages of two initialization options mentioned above, we propose an improved initial guess for the case of such image segmentation prediction in a process; that is named as *conditional randomized initialization*. This initial guess is randomized but conditioned on the segmentation at previous time point. The main idea of generating such conditional randomized initialization is based on the method proposed to estimate the interface level, which is plotting a vertical profile of averaged random variables' labels for every level. Here, we are trying to generate a similar vertical profile as the initialization. The following summarizes the procedure of how to create an initial guess:

1. a vertical profile of mean values for all levels is generated, and the value are gradually decreasing from 1 to 0 in the range from 1 (top) to 600 (bottom). The value of $\frac{1}{2}$ is fixed at the level where the interface level of previous segmentation locates;

2. a random field, f^0, in the size of (600×600) is generated by repeating the vertical profile 600 times in horizontal direction;

3. then, f^0 is contaminated by a Gaussian noise with zero mean and variance of $\frac{1}{9}$, $\mathcal{N}(0, \frac{1}{9})$;

4. finally, the initial guess of MRF, f^g_{t+1}, can be finalized with the following strategy:

$$f^g_{s,t+1} = \begin{cases} 1 & \text{if} \quad f^0_{s,t+1} \geq \frac{1}{2} \\ 0 & \text{if} \quad f^0_{s,t+1} < \frac{1}{2} \end{cases} \tag{4.39}$$

   where $\{s = (i, j) \mid 1 \leq i \leq 600, 1 \leq j \leq 600\}$.

Following the procedure as shown above, an example of the improved initialization is shown in Figure 4.14 based on the assumption that the interface level at previous segmentation is at 400-th level. Figure 4.15a shows a vertical profile of initial mean values. Figure 4.15b shows the vertical profile of averaged random variable labels for the initial MRF guess, which is shown in Figure 4.14. It can be seen that the

initial guess of MRF, as shown in Figure 4.14, has the similar overall inference as the true segmentation; however, the region around interface level is totally randomized. Therefore, by using the method of generating conditional randomized initial guess, the initialization is reducing both the correlation with temporal neighbors and the contingency of randomization.

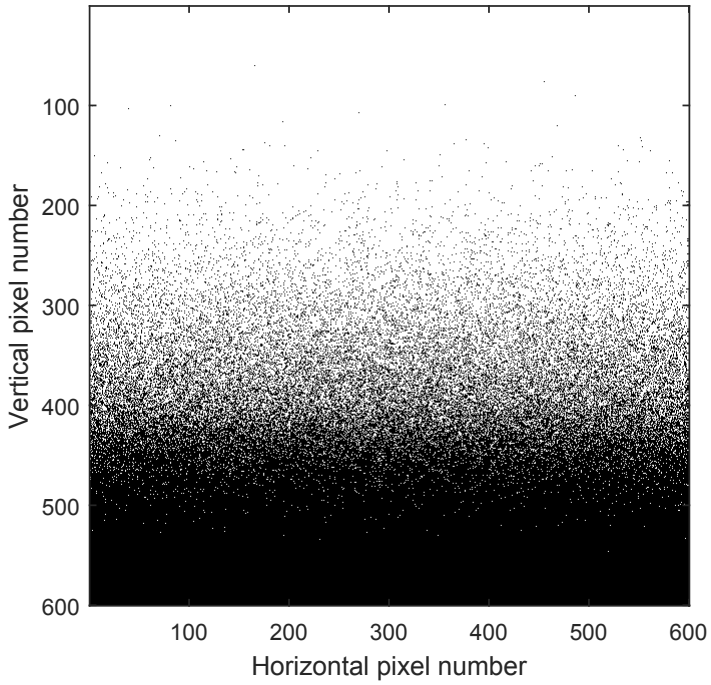
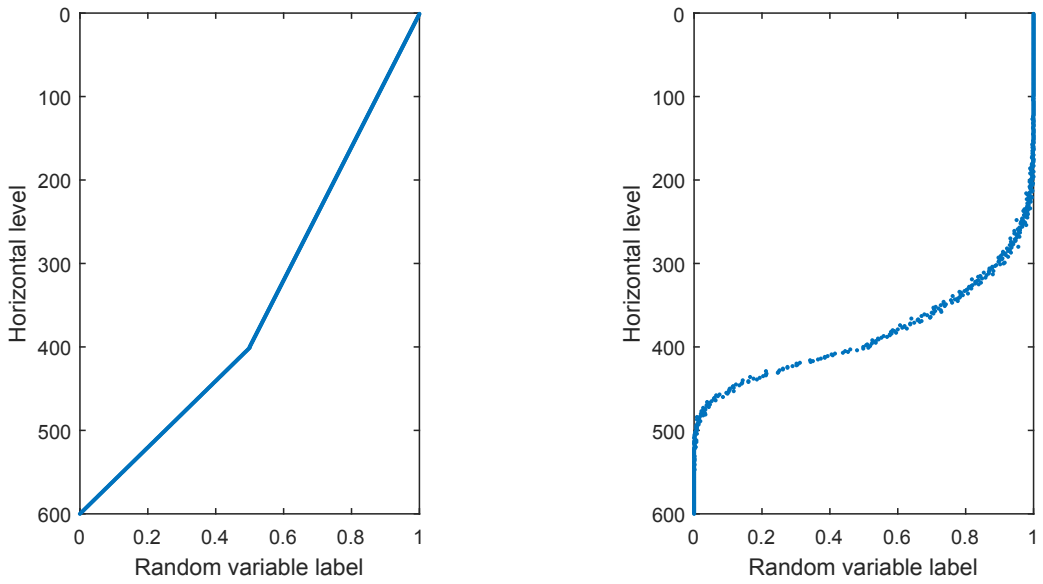


Figure 4.14: *The smart initial guess of Markov random field prediction for interface level estimation.*

Having specified the conditional randomized initial guess, the segmentation prediction can be obtained by following the steps in the flow chart, which was shown in Figure 4.10. Taking the third time point t_2 as an example, the predicted image segmentation for low flow rate and high flow rate are shown in Figures 4.16a and 4.16c, respectively. Correspondingly, the true segmentation for both operations are shown in Figures 4.16b and 4.16d. The comparisons of Figures 4.16a versus 4.16b as well as 4.16c versus 4.16d show the agreement. Only a number of *noisy* pixels above the interface are considered as *mis-segmentation*. The statistics of those mis-segmented pixels

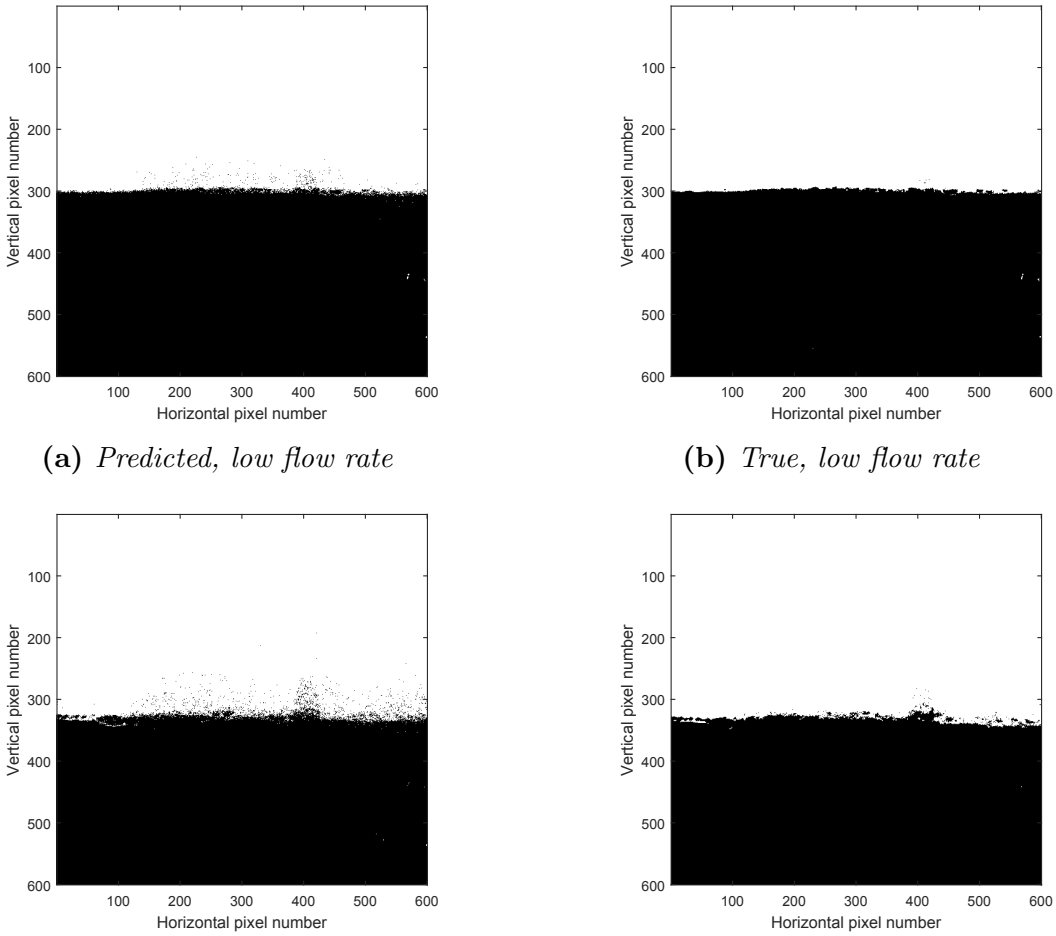


(a) *initial means* **(b)** *averaged random variable labels*

Figure 4.15: *The vertical profile of (a) initial means and (b) averaged random variable labels of the initial guess.*

are tabulated in Table 4.2 for all predictions. The number of mis-segmented pixels varies from 1469 up to 5131. The percentage of mis-segmentation was calculated by using the number of mis-segmented pixels divided by the total number of image pixels, $360,000$. As shown in Table 4.2, the highest mis-segmentation percentage is as much as 1.43%, which is a relatively small amount.

The validation of segmentation prediction can also be verified by comparing the interface level estimation. Although there are a number of mis-segmented pixels appearing in the prediction of image segmentation, the accuracy of interface estimation is not affected. First of all, for those four segmented images shown in Figures 4.16a to 4.16d, the interface levels can be estimated based on the approach of interface level estimation, which was proposed in chapter 2 and be visually shown on the images. Figures4.17a to 4.17d show the same segmentation comparison with interface indicated. It is difficult to specify the difference of interface estimation between Figures 4.17a and 4.17b as well as Figures 4.17c and 4.17d. To numerically compare the estimation, the interface heights of predicted segmentation were first calculated using



(a) *Predicted, low flow rate*

(b) *True, low flow rate*

(c) *Predicted, high flow rate*

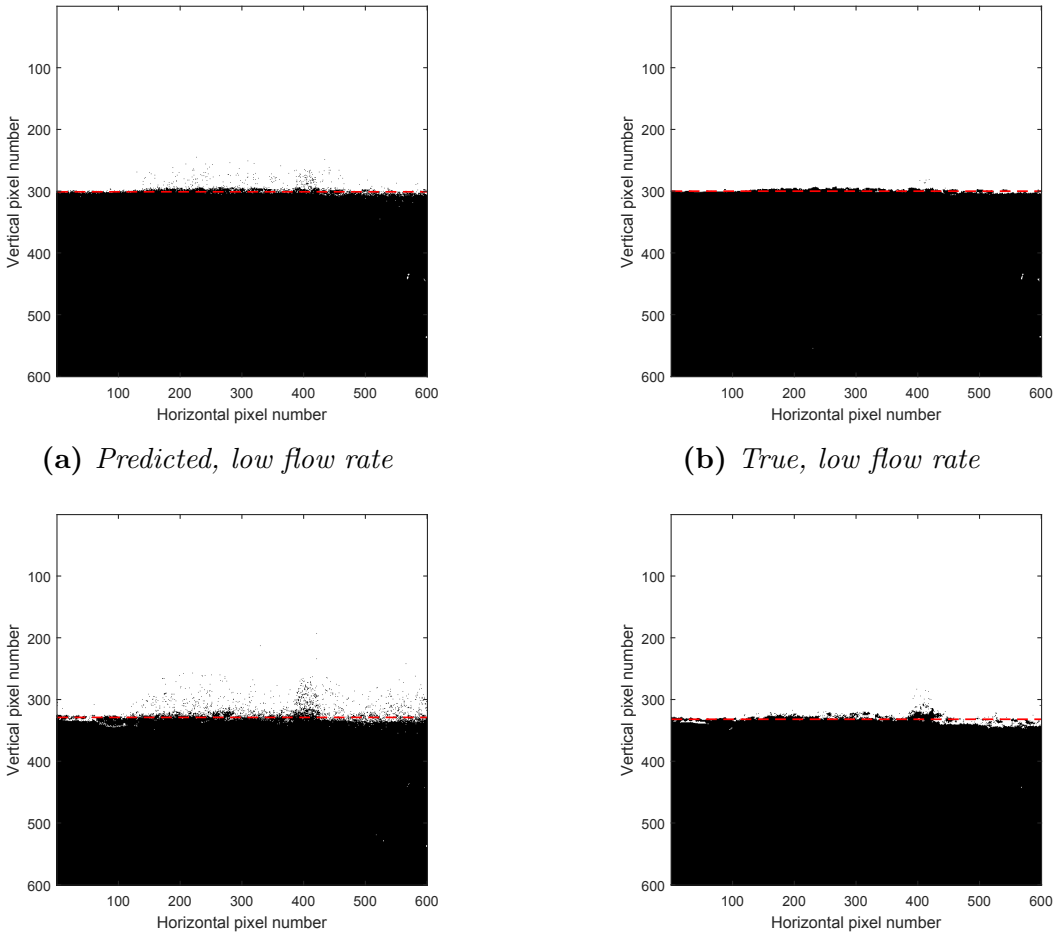
(d) *True, high flow rate*

Figure 4.16: *The comparison between predicted segmentation and true segmentation for both low and high flow rate operations.*

Table 4.2: *The mis-segmentation between predicted and true segmentation in number of pixels and percentage for both low and high flow rate operation.*

Time Point	Difference between Predicted & True Segmentation			
	Low Flow Rate Operation		High Flow Rate Operation	
	Number of Pixels	Percentage %	Number of Pixels	Percentage %
t_1	2078	0.58%	4245	1.18%
t_2	6418	1.78%	4069	1.13%
t_3	1469	0.41%	4186	1.16%
t_4	3027	0.84%	5131	1.43%

the calibration equation in chapter 2 and then compared with those heights obtained based on first principle method, denoted as *actual* interface height. All values along with the percentage errors for both low and high flow rate operations are tabulated in Table 4.3. Percentage errors are all less than 1%, thereby validating the proposed approach.



(a) *Predicted, low flow rate*

(b) *True, low flow rate*

(c) *Predicted, high flow rate*

(d) *True, high flow rate*

Figure 4.17: *The comparison between predicted segmentation and true segmentation with interface indication for both low and high flow rate operations.*

Additionally, considering the correlation along time series can not only achieve the prediction of both the image segmentation and the interface level but also improve the accuracy of the image segmentation and the interface level. Table 4.4 shows the

Table 4.3: *Estimation of interface heights based on predicted segmentation is compared with the actual heights for two operating conditions.*

Operating Condition	Time Point	Actual Interface Height (cm)	Predicted Interface Height (cm)	% Error
Low Flow Rate	t_1	35.94	35.94	0%
	t_2	35.98	35.93	0.8%
	t_3	36.01	36.00	0.2%
	t_4	35.99	36.00	0.2%
High Flow Rate	t_1	35.80	35.82	0.3%
	t_2	35.75	35.78	0.5%
	t_3	35.71	35.74	0.5%
	t_4	35.67	35.71	0.7%

comparison of the interface level estimation based on image segmentation between two MRF based models by considering with and without time sequence correlation. MRF is the model without considering correlation, and ST-MRF is the model considering correlation. A clear improvement is shown by comparing the percentage error between the estimations based on two models, which shows that the interface level estimation is more accurate by considering the time sequence correlation.

4.6 Conclusions

In conventional chemical process, process variables are usually varied and recorded as a function of time. The interface level of primary vessel is also a time varying variable. Based on camera sensor and proposed image processing technique, we motivated to consider temporal dependence in the modeling and achieve image segmentation prediction. We proposed a two-step algorithm: first, a predicted observation model was

Table 4.4: *The comparison of the interface level estimation based on image segmentation between two models by considering with (ST-MRF-based) and without (MRF-based) time sequence correlation.*

Operating Condition	Time Point	MRF-based Interface Height (cm)	% Error	ST-MRF-based Interface Height (cm)	% Error
Low Flow Rate	t_1	35.89	0.8%	35.92	0.3%
	t_2	35.92	1.0%	35.95	0.5%
	t_3	35.95	1.0%	36.01	0%
	t_4	35.94	0.8%	36.00	0.2%
High Flow Rate	t_1	35.77	0.5%	35.82	0.3%
	t_2	35.71	0.7%	35.78	0.5%
	t_3	35.65	1.0%	35.74	0.5%
	t_4	35.61	1.0%	35.69	0.3%

developed based on random walk process with the modification on noise term, and then, a spatial temporal Markov random field (ST-MRF) auto-logistic model was formulated to predict the segmentation at future time. The noise in the observation prediction model was replaced by a weighted summation noise which was aiming to obtain a better prediction on observed images. The ST-MRF auto-logistic model was developed based on the assumption of Markov chain along time-series. Model parameters were estimated using the generalized linear regression. The first step of the algorithm was validated by comparing the predicted observation with the real observation in overall distribution and the pixel intensities. The predicted segmentation was validated by showing the percentage differences from the true segmentation in the number of pixels that were mis-segmented as well as the interface level estimation difference. The developed model and algorithm were confirmed by showing the agreement with accepted inaccuracy.

Chapter 5

Conclusions

5.1 Summary of This Thesis

This thesis is focused on addressing the problem on primary separation vessel (PSV) froth middling interface level detection and estimation utilizing image processing technique based on Markov random field (MRF) theory. Specifically, Markov random field based image segmentation was proposed and applied throughout the thesis.

The industrial background of PSV froth middling interface level detection and estimation, the motivation and objective of solving this problem using MRF based image processing technique have been introduced in chapter 1. An experimental setup was designed in order to simulate a liquid interface. An online camera was able to capture interface images for the use of image processing. The first principle estimation of experimental interface level was also derived as a benchmark to compare with the estimation results obtained using image processing technique.

Chapter 2 introduced the basis of Markov random field based image segmentation: neighborhood system, Gibbs distribution, Markov-Gibbs equivalence and MAP-MRF optimization framework. An approach for interface level estimation based on segmented image was proposed. The extended iterated conditional modes (ICM) algorithm and the customized neighborhood system were also proposed in supervised image segmentation in order to obtain good segmentation results. Results have shown that the interface level estimations based on segmented images were comparable with first principle estimations. In addition, the proposed extended ICM algorithm could

significantly reduce the computational cost and the proposed customized neighborhood system could decrease the estimation variance.

In chapter 3, the work was mainly focused on improving MRF based image segmentation technique from supervised learning to unsupervised learning. Gaussian mixture model was proposed to model the observed images. Meanwhile, mean filed approximation was applied to obtain an approximate distribution of Gibbs distribution. Expectation maximization (EM) algorithm was used to iteratively estimate model parameters of both observation model and hidden MRF model under a single framework. The proposed approach in this chapter, on one hand, could achieve unsupervised image segmentation; on the other hand, it improved the image segmentation results by considering the spatial constraints in the latent Markov random field. The interface level estimation algorithm based on segmented image was also applied in this chapter and results were comparable with first principle estimations as well.

In chapter 4, a model for dynamic prediction of image segmentation and the interface level was developed based on spatial temporal Markov random field (ST-MRF) and auto-logistic model. A two-step approach was proposed: first of all, the observed image was predicted based on a modified random walk process; in the second step, the image segmentation could be predicted using spatial temporal Markov random field based auto-logistic model. For the first step, based on the recorded image data, the noise in the random walk model was modified as a weighted summation of three Gaussian noises. For the second step, the auto-logistic model was developed based on ST-MRF and model parameters were iteratively estimated by generalized linear regression. The image segmentation prediction errors were acceptable, and interface level estimations based on segmented images were also comparable with first principle estimations.

5.2 Directions for Future Work

In this thesis, we have proposed a new approach for camera based PSV froth middling interface level detection and estimation, that is Markov random field based

image processing. The original intention of proposing image processing technique in such problem was trying to address the interface level detection under some complex situations such as when the conventional camera based detection approach fails. To further investigate image processing technique in detecting interface level, some directions for future work could be summarized as the following:

1. One common situation that conventional camera based interface level detection fails is that the interface level does not present in the range of sight glass. To experimentally simulate such situation, one can paste a black cover on the transparent tank and make an artificial sight glass. To estimate interface level, one possible solution is to develop a vertical profile of pixel intensities, then match the portion that is shown in sight glass, and the interface level could be estimated based on the relativity between the portion in sight glass and the overall profile.

2. According to experienced site engineers, another common situation that the conventional camera detection usually fails is when viscous bitumen or blurs are adhered on sight glass. Markov random field based image processing could denoise those adhered blurs and bitumen on the image based on typical neighboring dependence. And then the interface level could be more accurately detected and estimated.

3. Regarding the unsupervised image segmentation, the proposed Gaussian mixture model could only estimate the rough overall distribution. To obtain a more accurate estimated distribution of observed image, a mixture of beta distribution could be proposed, and the corresponding terms of energy function might also be modified.

Bibliography

[1] B. Li, F. Xu, Z. Ren, and A. Espejo. Primary separation vessel interface control. In *Advanced Control of Industrial Processes (ADCONIP), 2011 International Symposium on*, pages 262–264. IEEE, 2011.

[2] P. Jampana, S.L. Shah, and R. Kadali. Computer vision based interface level control in separation cells. *Control Engineering Practice*, 18(4):349–357, 2010.

[3] A. Narang, S.L. Shah, T. Chen, E. Shukeir, and R. Kadali. Design of a model predictive controller for interface level regulation in oil sands separation cells. In *2012 American Control Conference (ACC)*, pages 2812–2817. IEEE, 2012.

[4] S.Z. Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.

[5] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[6] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

[7] A.P. Demrsrsa, N.M. Lamb, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal ofthe Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[8] K.A. Tran, N.Q. Vo, T.T. Nguyen, and G. Lee. Gaussian mixture model based

on hidden markov random field for color image segmentation. In *Ubiquitous Information Technologies and Applications*, pages 189–197. Springer, 2014.

[9] Q. Wang. Gmm-based hidden markov random field for color image and 3d volume segmentation. *arXiv preprint arXiv:1212.4527*, 2012.

[10] T. Nguyen. *Gaussian mixture model based spatial information concept for image segmentation*. PhD thesis, University of Windsor, 2011.

[11] H. He, K. Lu, and B. Lv. Gaussian mixture model with markov random field for mr image segmentation. In *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, pages 1166–1170. IEEE, 2006.

[12] W. Cai, L. Lei, and M. Yang. A gaussian mixture model-based clustering algorithm for image segmentation using dependable spatial constraints. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, volume 3, pages 1268–1272. IEEE, 2010.

[13] M. Haft, R. Hofmann, and V. Tresp. Model-independent mean-field theory as a local method for approximate propagation of information. *Network: Computation in Neural Systems*, 2009.

[14] J.A. Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[15] S. Kamijo, K. Ikeuchi, and M. Sakauchi. Segmentations of spatio-temporal images by spatio-temporal markov random field model. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 298–313. Springer, 2001.

[16] Z. Yin and R. Collins. Belief propagation in a 3d spatio-temporal mrf for moving object detection. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[17] R. Huang, V. Pavlovic, and D. Metaxas. A new spatio-temporal mrf framework for video-based object segmentation. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis-MLVMA'08*, 2008.

[18] F. Melgani and S.B. Serpico. A markov random field approach to spatio-temporal contextual image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 41(11):2478–2487, 2003.

[19] D.A. Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.

[20] J.E. Besag. Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 75–83, 1972.

[21] D.R. Cox and E.J. Snell. *Analysis of binary data*, volume 32. CRC Press, 1989.

[22] J. Zhu, H.C. Huang, and J. Wu. Modeling spatial-temporal binary data using markov random fields. *Journal of Agricultural, Biological, and Environmental Statistics*, 10(2):212–225, 2005.

[23] Z. Wei and H. Li. A hidden spatial-temporal markov random field model for network-based analysis of time course gene expression data. *The Annals of Applied Statistics*, pages 408–429, 2008.

[24] J.M. Hilbe. Generalized linear models. *The American Statistician*, 48(3):255–265, 1994.

Appendix A

Appendix to Chapter 2

Table A1: *The summary of comparison between ICM and extended ICM algorithm in terms of computational cost.*

Image Number	Iterations		Computation Time (s)	
	ICM	Ex-ICM	ICM	Ex-ICM
1	7	3	13.59	6.46
2	6	3	11.68	6.33
3	6	2	11.84	4.48
4	6	2	12.09	4.37
5	6	2	11.75	4.62
6	6	3	11.87	6.44
7	6	4	11.70	8.13
8	6	4	11.86	8.11
9	7	4	13.66	8.17
10	6	3	11.96	6.49

Table A2: *The summary of comparison between 1^{st} order and customized neighborhood system in terms of interface estimation and variances.*

Image Number	Interface height Estimation (cm)		Interface Estimation Variance	
	1st	Cust.	1st	Cust.
1	35.08	35.09	21.15	19.27
2	35.94	35.93	9.07	4.34
3	36.01	36.01	5.26	4.00
4	35.98	35.98	5.45	3.19
5	35.96	35.95	3.39	1.78
6	35.81	35.80	13.94	9.86
7	35.78	35.78	4.59	3.62
8	35.73	35.72	8.00	6.08
9	35.71	35.70	11.04	7.31
10	35.67	35.66	8.92	5.93