

# SAR Sea Ice Recognition Using Texture Methods

by

Bing Yue

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Applied Science

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2002

©Bing Yue 2002

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

*No one can cheat you out of ultimate success but yourselves.*

*—Emerson*

## Abstract

With the development of remote sensing techniques, a vast amount of SAR sea ice imagery is being provided by satellite platforms. As an important aspect of measurement, monitoring, and understanding of sea ice evolution during the seasons, the generation of ice type maps is a fundamental step in the interpretation of these data. The abundant texture information in SAR imagery is useful for segmentation of the pertinent ice types. Many texture analysis approaches have been proposed in the literature which can be identified into three categories: geometrical-based, statistical-based and model-based. Among these methods, to explore a coherent theoretical framework to support robust and powerful algorithms for sea ice segmentation is necessary.

This thesis will focus on two methods: gray level co-occurrence probability (*GLCP*) method and Markov random field (*MRF*) method. *GLCP* method can extract texture features with different frequencies along different directions in the image space. Using *MRF* method, texture is analyzed as having preferred relations and interactions that can be articulated mathematically, and then a Bayesian framework can be employed to make inferences. Quantizations, displacement, orientation, window size and texture statistics are important parameters in *GLCP* method. Accurate estimation of the texture models is an important concern for a successful segmentation using *MRF* method. In this thesis, the potential of these two methods for texture analysis and unsupervised segmentation are investigated and tested using synthetic, Brodatz and SAR sea ice images, and their texture distinguishing abilities are also compared.

## Acknowledgements

I would like to express my gratitude to the following individuals, groups and organizations:

First of all, special thanks are due to my supervisor, Professor David A. Clausi, for his guidance in defining this research, encouraging independent thinking and offering valuable advice during the whole course of my Master program.

I would also like to thank my readers, Professor Ed Jernigan and Professor Paul Fieguth from Systems Design Engineering, for their helpful comments on my thesis.

I give my thanks for the support and criticism from VIP members, for their part in making the MASc. experience both pleasurable and memorable. Thanks go to Zohreh Azimifar, Fakhry Khellah, Xinyu Chen, Xiaoli Tang, John Morton, Simon Booth, and Michael Jamieson.

I acknowledge with thanks the financial support I received over the course of my graduate program from the Faculty of Engineering of the University of Waterloo and CRYSYS (CRYosphere SYStem in Canada) research project.

Finally I thank my family for always being there with love when I needed them.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Statement . . . . .	3
1.2	Thesis Objectives . . . . .	5
1.3	Thesis Organization . . . . .	6
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	Introduction of SAR System . . . . .	7
2.2	Sea Ice Types and Its Physical Properties . . . . .	13
2.3	Texture Analysis . . . . .	16
2.3.1	Texture Definitions . . . . .	17
2.3.2	Texture Methods . . . . .	18
2.4	Challenge of Sea Ice Segmentation . . . . .	21
<b>3</b>	<b>Statistical Texture Analysis</b>	<b>22</b>
3.1	First Order Methods . . . . .	22
3.2	Second Order Methods . . . . .	24

3.3	The Comparison of Second Order Methods . . . . .	26
3.4	<i>GLCP</i> Texture Features . . . . .	28
3.4.1	<i>GLCM</i> . . . . .	28
3.4.2	Texture Feature Extraction from <i>GLCM</i> . . . . .	33
3.5	Statistical Second Order Texture Analysis for Sea Ice Segmentation	37
<b>4</b>	<b>Model-based Texture Analysis</b>	<b>40</b>
4.1	Markov Random Fields . . . . .	42
4.1.1	Neighbourhood System and Clique . . . . .	42
4.1.2	Markov Random Field Definition . . . . .	44
4.2	Markov-Gibbs Equivalence . . . . .	47
4.2.1	Gibbs Random Field Definition . . . . .	47
4.2.2	Hammersley-Clifford Theorem . . . . .	49
4.3	Image Models . . . . .	50
4.3.1	Auto-Model and Related Texture Synthesis . . . . .	52
4.3.2	Gauss-model . . . . .	60
4.3.3	Gauss-Model Texture Synthesis . . . . .	65
4.4	Model-based Texture Segmentation Algorithm Design . . . . .	69
4.4.1	Prior Image Models . . . . .	69
4.4.2	Model Size and Parameter Estimation . . . . .	73
4.4.3	<i>MRF</i> Segmentation in Bayesian Framework . . . . .	78
4.4.4	<i>MRF</i> Segmentation Algorithm . . . . .	83

<b>5 Experiments and Results Analysis</b>	<b>88</b>
5.1 Research Questions . . . . .	89
5.2 Research Question One . . . . .	92
5.2.1 Method . . . . .	92
5.2.2 Experiment Results . . . . .	93
5.3 Research Question Two . . . . .	95
5.3.1 Method . . . . .	97
5.3.2 Experiment Results . . . . .	99
5.4 Research Question Three . . . . .	109
5.4.1 Method . . . . .	109
5.4.2 Experiment Results . . . . .	111
5.5 Image segmentation Results . . . . .	124
<b>6 Conclusions and Future Work</b>	<b>145</b>
6.1 Conclusions . . . . .	145
6.2 Future Work . . . . .	148
<b>A Radarsat Image and Sea Ice Types</b>	<b>150</b>
A.1 Image Modes of Radarsat . . . . .	150
A.2 Processing Levels of Radarsat Image . . . . .	150
A.3 Ice Types . . . . .	151
<b>Bibliography</b>	<b>153</b>

# List of Tables

3.1	<i>GLCM</i> parameters of Brodatz and SAR sea ice images used in this thesis. . . . .	39
4.1	Auto-model parameters corresponding to the textures in Fig. 4.6 . .	60
4.2	GMRF model parameters corresponding to the textures in Fig. 4.9	67
4.3	GMRF texture models and $C_k$ values corresponding to the synthesized textures in Fig. 4.12. . . . .	76
4.4	The minimum $C_k$ value distribution of Radarsat SAR sea ice image with different <i>GMRF</i> texture model orders . . . . .	77
5.1	The <i>GMRF</i> models used to generate the synthetic textures used in the experiments in Chapter 5. . . . .	93
5.2	Average deterioration of the standard deviation of the estimated <i>GLCP</i> texture features and the <i>GMRF</i> model parameters from window sizes 96 to 64, 32, 16, 8. . . . .	94
5.3	Bhattacharyya error bounds and Fisher criteria of the texture pairs in Fig. 5.2 and 5.6 . . . . .	106
5.4	Segmentation error of the test images in section 5.5. . . . .	132

A.1 Sea ice types . . . . . 152

# List of Figures

2.1	Geometric characteristics of radar image . . . . .	9
2.2	Radarsat operational modes (adapted from [48]). . . . .	12
2.3	Sea ice samples . . . . .	14
2.4	Illustration of the functionality of textures in pattern recognition of human vision system . . . . .	18
3.1	A summary of the relative merits of the <i>GLRL</i> , the <i>GLD</i> , and the <i>GLCP</i> (adapted from [Connors 80]). . . . .	27
3.2	Eight directions for the creation of <i>GLCM</i> of pixel s. . . . .	29
3.3	Sub-image window from the texture image . . . . .	30
3.4	Calculation of the <i>GLCM</i> . . . . .	31
3.5	The <i>GLCMs</i> of different textures . . . . .	32
3.6	Original texture image . . . . .	34
3.7	The <i>GLCP</i> texture features extracted from a texture image . . . . .	35
4.1	Neighbourhood systems and cliques . . . . .	44
4.2	The Markovianity property . . . . .	46

4.3	Texture samples . . . . .	53
4.4	Pair-cliques involved in the third order <i>MRF</i> texture model . . . . .	54
4.5	Auto-model neighbourhood system. . . . .	57
4.6	Synthetic textures generated using the auto-models in Table 4.1 . . . . .	58
4.7	A image (size: $4 \times 4$ ) with a toroidal boundary condition . . . . .	62
4.8	The parameters of the fifth order <i>GMRF</i> model. . . . .	63
4.9	Synthetic textures generated using the <i>GMRF</i> models in Table 4.2 . . . . .	68
4.10	Image reconstructions based on different prior knowledge . . . . .	70
4.11	Image components defined for model-based segmentation algorithm . . . . .	71
4.12	Synthesized textures corresponding to the models in Table 4.3 . . . . .	75
4.13	Neighbourhood systems of intensity and lable images. . . . .	82
4.14	Model-based <i>MRF</i> texture image segmentation algorithm flowchart. . . . .	84
4.15	Parameter estimation in Fig. 4.14 . . . . .	85
4.16	Variances of Brodatz texture energy $U_1$ vs different energy window sizes. . . . .	86
4.17	Variances of Radarsat sea ice texture energy $U_1$ vs different energy window sizes . . . . .	87
5.1	Change of standard deviation of the estimated <i>GLCP</i> texture features and the <i>GMRF</i> model parameters with different window size . . . . .	96
5.2	Test image pairs for the second research question . . . . .	100
5.3	Distributions of the two sets of the <i>GMRF</i> model parameters and the <i>GLCP</i> texture features estimated from Fig. 5.2 using $n = 8$ on their Fish discriminant vector $w$ . . . . .	102

5.4	Distributions of the two sets of the <i>GMRF</i> model parameters and the <i>GLCP</i> texture features estimated from Fig. 5.2 using $n = 16$ on their Fish discriminant vector $w$ . . . . .	103
5.5	Distributions of the two sets of the <i>GMRF</i> model parameters and the <i>GLCP</i> texture features estimated from Fig. 5.2 using $n = 32$ on their Fish discriminant vector $w$ . . . . .	104
5.6	Another test image pair for the second research question . . . . .	107
5.7	Distributions of the two sets of the <i>GMRF</i> texture model and the <i>GLCP</i> texture features estimated from Fig. 5.6 on their Fish discriminant vector $w$ ( $n=8,16,32$ ) . . . . .	108
5.8	Three texture images separated by straight boundary . . . . .	110
5.9	The averaged <i>GLCP</i> texture features over 50 arbitrarily selected rows from Fig. 5.8 a . . . . .	113
5.10	The averaged <i>GMRF</i> model parameters over 50 arbitrarily selected rows from Fig. 5.8 a . . . . .	114
5.11	The averaged <i>GLCP</i> texture features over 50 arbitrarily selected rows from Fig. 5.8 b . . . . .	115
5.12	The averaged <i>GMRF</i> model parameters over 50 arbitrarily selected rows from Fig. 5.8 b . . . . .	116
5.13	The averaged <i>GLCP</i> texture features over 50 arbitrarily selected rows from Fig. 5.8 c . . . . .	117
5.14	The averaged <i>GMRF</i> model parameters over 50 arbitrarily selected rows from Fig. 5.8 c . . . . .	118

5.15	The estimated texture parameters of pixel $s$ using $5 \times 5$ window could be more similar to the parameters of texture B than A. . . . .	119
5.16	Segmentation of Brodatz texture image . . . . .	121
5.17	Segmentation of Brodatz texture image . . . . .	122
5.18	Segmentation of Brodatz texture image . . . . .	123
5.19	Segmentation of Synthetic texture image . . . . .	125
5.20	Segmentation of Brodatz texture image . . . . .	127
5.21	Segmentation of Brodatz texture image . . . . .	129
5.22	Segmentation of Brodatz texture image . . . . .	130
5.23	<i>MRF</i> Mid-segmentation results of Fig. 5.22 . . . . .	131
5.24	Aerial SAR image (sub-image of Figure 2(b) in [16]) . . . . .	135
5.25	<i>GLCP</i> segmentation result of Fig. 5.24 . . . . .	136
5.26	<i>GLCP</i> segmentation result edge map of Fig. 5.25. . . . .	137
5.27	<i>MRF</i> segmentation result of Fig. 5.24 . . . . .	138
5.28	<i>MRF</i> segmentation result edge map of Fig. 5.27. . . . .	139
5.29	Radarsat SAR sea ice image obtained from Canadian Ice Service . .	140
5.30	<i>GLCP</i> segmentation result of Fig. 5.29 . . . . .	141
5.31	<i>GLCP</i> segmentation result edge map of Fig. 5.30. . . . .	142
5.32	<i>MRF</i> segmentation result of Fig. 5.29 . . . . .	143
5.33	<i>MRF</i> segmentation result edge map of Fig. 5.32. . . . .	144
A.1	Radarsat-1 (adapted from [48]). . . . .	151

# Chapter 1

## Introduction

The remotely sensed image in its digital form, whether the image be a photomicrograph, an aerial photograph, or a satellite image, is stored in the computer as a 2-D or  $n$ -D array if the image is multi-spectral. These images contain a vast amount of information about objects within a scene, e.g. relative brightness, spatial proximity, size etc. The computer vision branch of the field of artificial intelligence, such as image coding, restoration, enhancement, and segmentation, is then employed to develop algorithms for analyzing the content of the image.

Human beings are adept at visually interpreting images produced by remote sensing devices. When visually interpreting a remotely sensed image, a person synergistically takes into account the characteristics of the data such as context, edges, texture and tonal variation. However, it is practically impossible for the computer-assisted segmentation algorithm to incorporate all these characteristics simultaneously. One could ask: why try to mimic and improve on the capability of human using a computer? Jensen summarizes several reasons [32]:

1. There are certain thresholds beyond which the human interpreter cannot de-

tect just-noticeable-differences in the image. It is commonly known that the human vision system can discriminate only 8 to 16 shades of grey levels when interpreting continuous-tone black-and-white remote sensing images. If the data were originally recorded with 256 grey levels, which is usually the case of most remotely sensed data, there may be more subtle information present in the image than the interpreter can extract visually.

2. The results obtained by computer are almost always repeatable, whereas, those of interpreters are subject to variability, making the interpretations generally unrepeatable. The interpretations also vary from person to person.
3. When tracking of a great amount of detailed quantitative information, the computer is much more adept and efficient at storing and manipulating such tedious information and possibly making a more unbiased conclusion.

Early image segmentation algorithms are based only on the use of tonal information, i.e. the intensity value of the image [32]. Thus, it is not surprising that there has been considerable activity in trying to incorporate other characteristics into the digital segmentation procedure. One of the successful explorations in this area is the work of Haralick and Galloway *et al* beginning from the 1970's [44, 25]. They demonstrated the power of texture information in image segmentation. From then on, using texture in image segmentation became of interest in the literature.

Image segmentation can be defined as a process that partitions a digital image into disjoint regions [8]. For our purpose, a region is a connected set of pixels. The segmentation can be performed using either of two methods: supervised or unsupervised. In a supervised segmentation, the identity and location of some of the object types in the image are known *a priori* through a combination of field work, analysis of maps, and personal experience. The analyst attempts to locate

specific sites in the remotely sensed image that represent homogeneous examples of the known land-cover types. These areas are commonly referred to as training sites because the spectral characteristics of these known areas are used to train segmentation algorithms for eventual land cover mapping of the remainder of the image. In an unsupervised segmentation, the identities of land cover types to be specified as classes within a scene are not generally known *a priori* because ground truth is lacking or surface features within the scene are not well defined. The computer is required to group the pixels into different classes according to some statistically determined criteria. It is then the responsibility of the analyst to label these clusters. The unsupervised segmentation obtains statistical features from the given image instead of from the training sites, this makes it more difficult to compute statistical features of a class accurately.

## 1.1 Problem Statement

Satellites have flown for about four decades, and the polar regions have been the subject of their routine surveillance for more than half that time [22]. Our observations of polar regions have evolved from spot explorations and icing records to routine global records obtained by various satellites. Thanks to such abundant data, scientists now know a great deal about the ice-covered seas, which constitute about ten percent of the Earth's surface. Just as Carsey *et al* state in [6]: "We are now already at a point of transition in sea ice studies: we are concerned less about ice itself and more about its role in the climate system." In this system, the sea ice cover has been placed in an unique position. Its degeneration is regarded as the key predictor of global warming.

Canada is considered to be a polar nation, with vast tracts of arctic land and

ocean areas. More than 170,000 *km* of shoreline are above 60°N and a further 6800 *km* of maritime shoreline are affected by ice for some part of the year, which results in more than ninety percent of Canada's maritime shoreline being affected by ice [20]. Realizing the importance of sea ice study, the CRYSYS (a Canadian led interdisciplinary science investigation in the NASA earth system enterprise and earth observing system program [15]) research project has clearly identified its first scientific goals as to develop capabilities for improved satellite-based measurement, monitoring and understanding of cryospheric variables over a range of spatial and temporal scales for the coming decade [21].

Using synthetic aperture radar (SAR) images for sea ice measurement, monitoring and interpretation is now an important research area. Historically, sea ice reconnaissance, in support of Arctic resource exploration and shipping operations, was one of the primary factors that drove the development of SAR technology in Canada [22]. As early as the mid-1970s, SAR was recognized as a potentially suitable sensor class for all weather sea ice mapping, but the understanding of sea ice properties that could be observed by radar systems was not consistent. Because the SARs of that period were not calibrated instruments, quantitative measurements, made on the ice surface or acquired by calibrated, non-imaging sensors, were treated as auxiliary data by the sea ice image interpreter.

Ice reconnaissance continues to this day and is one of the major commercial uses of SAR technology. As research has progressed, SARs have become calibrated instruments and quantitative measurements made by these systems are becoming increasingly important in sea ice research. With the more powerful SARs becoming operational, they will provide an unprecedented volume of imagery. This wealth of SAR data could cause a serious problem if the method for interpretation of these data is not improved. Computer-assisted, semi-automated or automated methods

for evaluations of sea ice data are necessary research directions. As one research area included in CRYSYS scientific objectives, this thesis will focus on exploring the computer-assisted automated texture segmentation algorithm for identifying different sea ice types in SAR imagery.

## 1.2 Thesis Objectives

The following are the objectives of this thesis:

- Present background knowledge on SAR systems and sea ice properties related to SAR sea ice interpretation. The topic will focus on the relationship of the physical properties of sea ice to the corresponding backscatter characteristics on the SAR sea ice image.
- Review the texture analysis techniques related to image segmentation in the literature.
- Present the statistical-based grey level co-occurrence probability (*GLCP*) approach for texture features extraction, investigate the feasibility of using the extracted features for texture image and SAR sea ice image segmentation.
- Explore the potential of using Markov random field (*MRF*) models for texture analysis. Present methods used to synthesize *MRF* textures and model parameter estimation.
- Present a texture segmentation algorithm using Markov random field model coupled with a Bayesian framework.

- Investigate the texture distinguishing ability of the *GLCP* texture features and the *GMRF* models. Compare the segmentation performance of the *GLCP* and *MRF* methods for synthetic, Brodatz, and SAR sea ice images.

### 1.3 Thesis Organization

Chapter 2 will present the background knowledge about SAR systems, sea ice properties as well as the texture definitions assumed in the following chapters. In Chapter 3, the popular statistical-based texture analysis methods are reviewed first, followed by the detailed demonstration of *GLCP* method used for texture image segmentation. Chapter 4 will cover the topics on *MRF* theory, *MRF* model estimation and texture synthesis. A methodology using *MRF* technique coupled with the Bayesian framework for texture segmentation will also be developed. Several research questions about the separability of *GLCP* texture features and *GMRF* models will be analyzed in Chapter 5. And the segmentation results of synthetic texture image, Brodatz texture images and SAR sea ice images are also demonstrated. Chapter 6 provides research conclusions and gives some suggestions for future work.

# Chapter 2

## Background

This chapter will first present some background knowledge on SAR systems. This will be followed by a discussion on sea ice properties related to sea ice interpretation. Texture definition, from the image processing perspective, follows. The challenges involved in sea ice segmentation are discussed in the final section.

### 2.1 Introduction of SAR System

“Remote sensing is broadly defined as collecting and interpreting information about a target without being in physical contact with the object [50].” Different from passive remote sensing system such as thermal IR, which detects the available energy reflected or radiated from the earth surface, a radar system sends and receives its own energy. It operates in the radio and microwave bands of electromagnetic spectrum ranging from one meter to a few millimeters in wavelength. The system “illuminates” the terrain with electromagnetic energy, detects the energy returning from the object, called radar return, and records it as an image. Without depending

on outside illuminating energy, a radar system operates independently of lighting and weather conditions. In addition, the system can control the “look” direction so as to “illuminate” the terrain in the optimum direction and enhance features of interested . Fig. 2.1 illustrates the geometric characteristics of radar imagery acquired by a side-looking radar. The range or look direction is the direction of the radar illumination that is at right angles to the direction the aircraft is traveling and parallel to the horizontal plane of the Earth.

The two basic radar systems are real aperture radar and synthetic aperture radar, i.e. SAR, which differ primarily in the method each uses to achieve resolution in the azimuth flight direction. In [50], Sabins provides a detailed description about the properties of radar systems and terrain that can determine the intensity of the radar return. The following paragraphs are a brief summary from [50].

The relative elements that determine the intensity of the radar return can be categorized into two aspects: radar system properties and terrain properties. The radar system properties include: wavelength, depression angle, spatial resolution and polarization. Terrain properties include: dielectric property, surface roughness and feature orientation.

Radar systems typically operate at one of four different wavelengths: Ka-band ( $0.86\text{ cm} \sim 1.1\text{ cm}$ ), X-band ( $2.4\text{ cm} \sim 3.8\text{ cm}$ ), C-band ( $3.8\text{ cm} \sim 7.5\text{ cm}$ ) and L-band ( $15\text{ cm} \sim 30\text{ cm}$ ). As the wavelengths increase, the penetrability of the radar signal through the objects on the Earth increases, but the resolution decreases. Using filters, the radar antennas selectively send and receive polarized energy. The pulse of energy sent out by the antenna may be vertically or horizontally polarized, and the backscattered energy can also be recorded in the vertical and horizontal directions. Considering the sending and receiving directions together, there are four kinds of radar images: HH, VV, HV and VH. The ability of a radar system to record

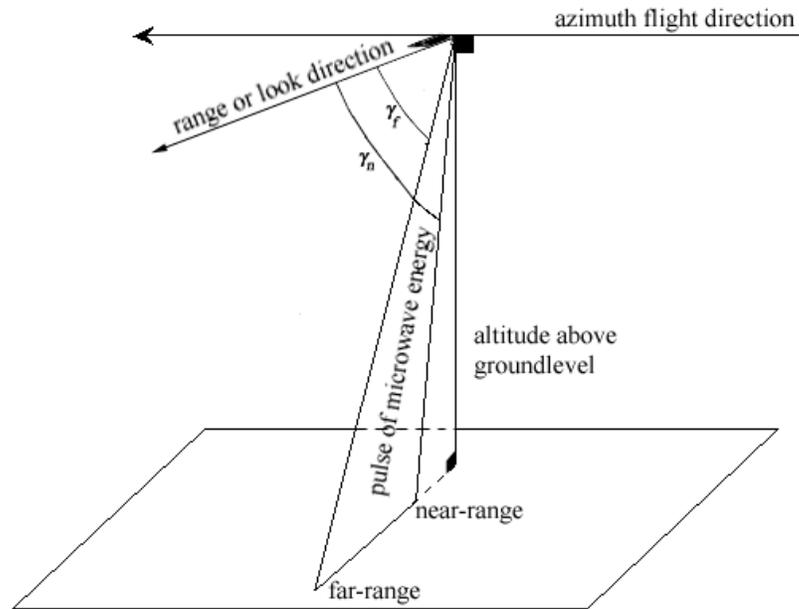


Figure 2.1: Geometric characteristics of radar image acquired by side-looking radar. different types of polarized energy results in valuable earth resource information for certain applications [50].

Resolutions in the range and azimuth direction are determined by the engineering characteristics of the radar system. The combination of range resolution and azimuth resolution determines the dimension of the ground resolution cell, which in turn determines the spatial resolution of a radar image. An important characteristic that can affect range resolution of the SAR image is the depression angle ( $\gamma$ ), defined as the angle between a horizontal plane and a beam from the antenna to a target on the ground. The near-range depression angle ( $\gamma_n$ ) and far-range depression angle ( $\gamma_f$ ) are shown in Fig. 2.1. The equation for range resolution of

SAR is

$$SAR_r = \frac{c \tau}{2 \cos \gamma} \quad (2.1)$$

where  $\tau$  is the pulse length,  $c$  is the speed of light ( $c = 3 \times 10^8 m/s$ ). According to Eqn. 2.1, one method for improving range resolution is to shorten the pulse length. But one cannot infinitely do so because of two reasons: (1) shortening the pulse length results in the reduction of the total amount of energy in each transmitted pulse, and the energy cannot be reduced below the level required to produce a sufficiently strong return from the terrain; (2) the physical limitation of the radar antenna. The equation for the azimuth resolution of SAR is

$$SAR_a = \frac{D}{2} \quad (2.2)$$

where  $D$  is the antenna length. Because the coherent nature of the SAR signal produces speckle in the image, to remove the speckle, the image is usually processed using several looks, i.e. an average takes place. In this case, the azimuth resolution is adjusted as following:

$$SAR_a = N \frac{D}{2} \quad (2.3)$$

where  $N$  is the number of looks. It is interesting to notice that the azimuth resolution of a SAR system is independent of range distance, i.e., the distance from antenna to the target.

The backscatter of the terrain is a quantitative measure of the intensity of radar energy returned to the antenna. Physical properties of terrain that influence the backscatter include dielectric property, surface roughness and feature orientation. Each of them will be discussed below.

The dielectric constant is a physical characteristic of matter, which is used to measure the matter's interaction with electromagnetic energy. With increas-

ing moisture, the dielectric constant of the object increases, and causes increased backscatter, which in turn indicates an increasing brightness in image tone. For sea ice, different ice types each with different dielectric constant lead to various brightnesses on SAR sea ice image, which could be utilized in sea ice type's identification.

Radar backscatter is also strongly influenced by the surface roughness. Surface roughness is measured using distance units and is determined by relative roughness of the ground to the radar wavelength. The average surface roughness within a SAR ground resolution cell determines the intensity of the radar return from that cell. In practice, the Rayleigh criterion is used to indicate whether a surface is considered smooth or rough. A surface is said to be smooth if

$$h < \frac{\lambda}{8 \sin \gamma} \quad (2.4)$$

where  $h$  is the vertical relief of the surface,  $\lambda$  is the signal wavelength, and  $\gamma$  is the depression angle. The smooth surface reflects all energy away from the radar system. The surface with intermediate roughness reflects part of the energy and scatters the remainder. The rough surface diffusely scatters all energy, causing a relatively strong backscattered component that produces a bright signature on the image. For a side-looking radar system, the slant distance, i.e. the distance from target on the terrain to antenna, can also affect the smooth and rough criteria. For example, a surface with vertical relief  $h$  in the far range has a smaller depression angle than that in the near range area with the same vertical relief  $h$ , according to 2.4, the surface in the far range will appear smoother to the radar signal.

The geometric relationship between the preferred orientation and the look direction also influences the radar signature. Usually, features trending normal to the look direction are enhanced by highlights and shadow. Feature trending parallel with the look direction produces no highlights or shadows and are suppressed in an

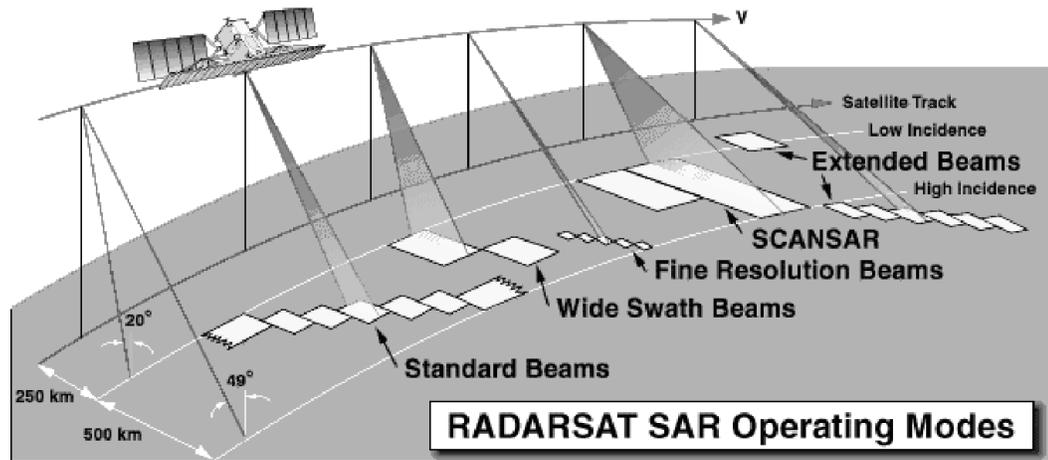


Figure 2.2: Radarsat operational modes (adapted from [48]).

image.

Some of SAR sea ice images used in this thesis are captured by Radarsat ScanSAR wide mode beam. Launched in November 1995, Radarsat is a sophisticated Earth observation satellite developed by Canada to monitor environmental change and the planet's natural resources. With 25 operating modes, it can provide useful information to scientific users in the fields of agriculture, cartography, hydrology, forestry, ice studies and coastal monitoring.

Fig. 2.2 shows the 25 image modes that Radarsat can apply [48]. Each image mode corresponds to one beam position. From the figure, it is clear that Radarsat is a side-looking imaging system, and the captured images are on the right side of the track. For each Radarsat beam mode, its image products are processed to different levels of geometric accuracy and radiometric calibration. The ScanSAR wide mode data has already been converted to ground range image, and the direction of the image is the same as the direction of the satellite track. For reference, Radarsat Parameters and Operational Modes are described in Appendix A.

## 2.2 Sea Ice Types and Its Physical Properties

According to the World Meteorological Organization, ice can be categorized into four main types: new ice, young ice, first year ice and old ice [13]. Lewis *et al* and Weeks *et al* give a very detailed description of various sea ice types as well as their physical properties in [51] [20]. This section just provides a brief introduction on these topics based on their description.

Initial ice formation begins at the water surface at the beginning of winter, where the heat loss is greatest. The sea ice growing at this very beginning stage is called new ice, which usually is less than 10 *cm* thick. With continued freezing, the new ice begins to form a solid cover from 10 *cm* to 30 *cm* thick called young ice. At the end of the winter, the sea ice after one winter's growth becomes the first year ice, usually more than 30 *cm* thick. When summer season comes, sea ice undergoes dramatic changes. The most obvious change is the large decrease in salinity. Some first year ice survives the summers, and becomes old year or multi-year ice. The detailed descriptions for various ice types are tabulated in Appendix A.3.

SAR systems record the backscattering of the radar signal reflected by the objects on the Earth. When illuminating the objects, the microwave signal can be reflected, scattered, transmitted. The transmitted signal can again be reflected, scattered and transmitted by the lower surfaces of the target. For sea ice, the presence of moisture and salinity on the ice surface and inside the ice can significantly increase radar reflectivity and decrease its transmission. Scattering depends upon the surface roughness and inhomogeneities within ice. Spatial variations in the dielectric properties also contribute to the magnitude of the scattering.

The electromagnetic properties of the ice sheet are heavily dependent on the distribution of liquid brine and gas within the ice, and they are extremely important

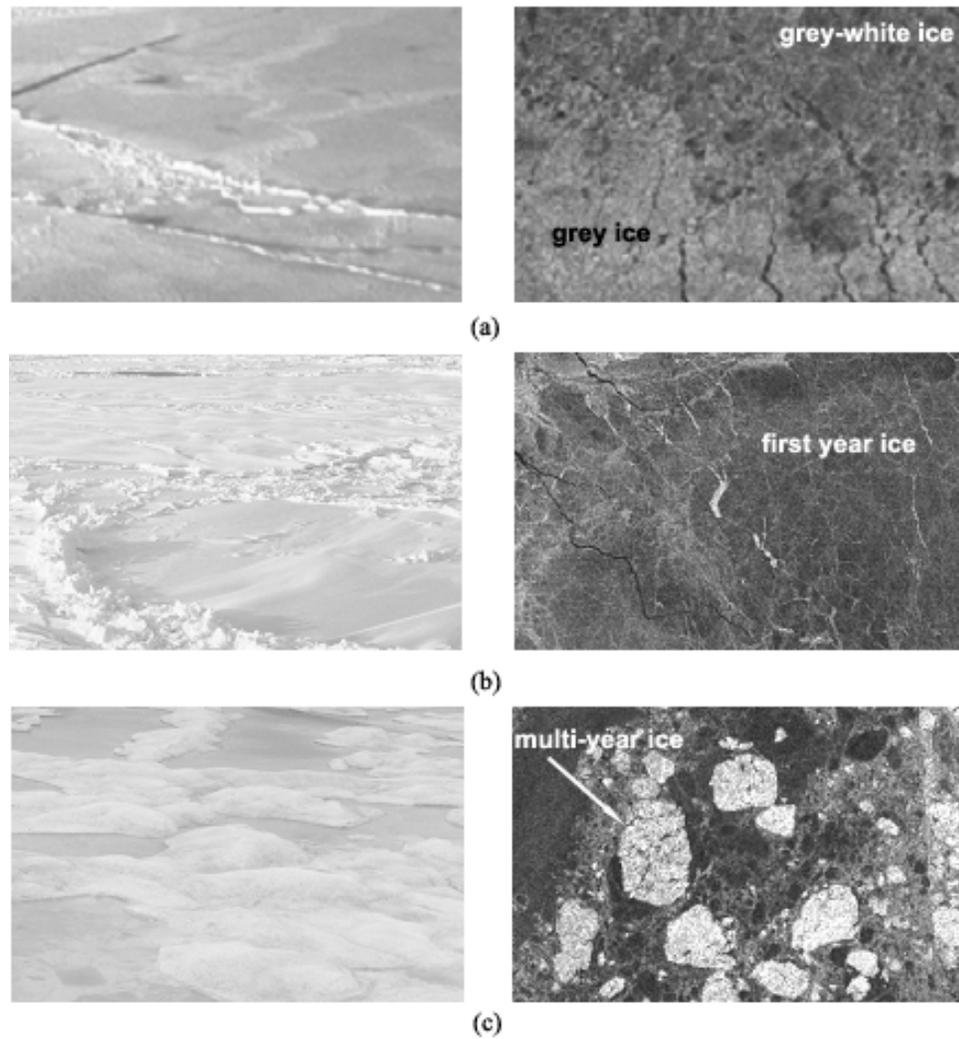


Figure 2.3: Sea ice samples (L: sea ice pictures, R: Radarsat ScanSar wide mode images, adapted from [1]). (a) Young ice. (b) First year ice. (c) Multi-year ice.

with regard to radar return. Usually, young ice and first year sea ice are relatively saline with a loose structure containing many air bubbles which make it difficult to penetrate with microwave energy. As a result, most of the energy is reflected. Old ice is a low dielectric substance because of little brine content which permits significant penetration by microwave energy.

Surface appearances of different ice types are quite different. Fig. 2.3 shows some sea ice samples of young ice, first year ice and multi-year ice (the pictures and images are provided by Dr. Roger De Abreu of Canadian Ice Service [1]). The left column shows the ice pictures. The grey ice are fractures with ragged edges, usually without floe structure or ridging, and the grey-white ice are stronger, thicker ice sheet than grey ice. Surface topography of first-year ice is generally sharp and angular with moderate relief. Un-deformed first-year ice is flat with little freeboard. When it has been deformed, the ridges consist of very distinct collections of angular blocks. The surface of multi-year ice is undulating with smooth and rounded features, often with significant vertical relief. Multi-year ice typically has a rolling hummocky appearance. Un-deformed multi-year floes have a surface relief of 10 *cm* to 20 *cm* as a result of differential melting. The undulating surface of old ice is easily identified from an aircraft if there is little snow cover. Multi-year ridges are also distinct from first-year ridges because they are weathered and rounded, with little or no sign of the original block structure evident. The densities of multi-year ice are lower than those of first-year ice in the upper layers. This reflects the increase in porosity of multi-year ice due to the evacuation of brine inclusions and the enlargement of brine cavities and drainage channels during the melt season.

The Radarsat ScanSAR wide mode sea ice image samples are shown on the right column in Fig. 2.3. The grey ice usually has light grey tone and spongy

texture in the absence of floes and ridges. The grey-white ice appears dark grey with straight-edged floes and angular fractures. The first year ice usually has grey or dark tone with some bright ridge lines, floes are often angular, floe size may be large. For the multi-year ice, it has bright tone in winter sense, and much less tone in melt scenes, floes are often rounded with mottled texture.

Sea ice is a dynamic, complicated target in SAR image, its appearance changes with weather, season and location. For example, although usually there is big difference in the backscattering character of first year and old ice, this difference can be concealed by various facts which can either change the surface smoothness or the dielectric constant of the ice, such as periodic warming, snow cover, free water intrusion [13]. The general introduction of this section, however, is not necessarily representative of all practical cases, and care should be exercised in their interpretation.

## 2.3 Texture Analysis

From 1970's, the importance of texture information for object recognition and classification has been explored by many researchers and their studies show the fact that incorporation of texture information is crucial [44, 25, 30, 26, 7]. In his precursory paper about textural features for image classification, Haralick defined fourteen textural features based on grey level spatial dependencies, and illustrated their application in category identification tasks of several different kinds of image data. More than eighty percent test set identification accuracy was obtained. The results successfully indicated that textural features had a general applicability for a wide variety of image classification applications [44].

### 2.3.1 Texture Definitions

The word texture originally referred to the appearance of woven fabric. In the image processing field, there is still no universally accepted definition for texture. Part of the difficulty is the extremely large number of texture attributes that people would like to subsume under a definition. Various definitions are given which represent the same basic ideas. “Texture is the local change in intensity of the colour within some defined spatial region [52].” “Texture is concerned with the spatial (statistical) distribution of grey tones [44].” “Texture is the frequency of change and arrangement of tones on an image. Fine, medium, and coarse are some terms used to describe texture [50].” “Texture is defined as the spatial variation of digital numbers within an image [53].”

According to the above definitions, if the grey level is constant everywhere in the object, or nearly so, one can say that the object has no texture. If the grey level varies significantly within the object, then the object has texture. When there is no spatial pattern and the tone variation between features is wide, a surface with fine texture will dominate the image, and as the spatial pattern becomes more definite and involves more and more resolution cells, a coarser texture results.

Texture contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment. An example given in Fig. 2.4 can show the importance of texture in object recognition. If one perceives a object in an isolated visual field (i.e. with no surrounding clues), one would be hard pressed to determine if this object was the texture of a fire or the leaf of a beautiful tree during autumn based on tone alone. However, if texture was considered, then the distinction would become a trivial one since the leaf image would appear very fine in its tone while the surface of the fire would be much more

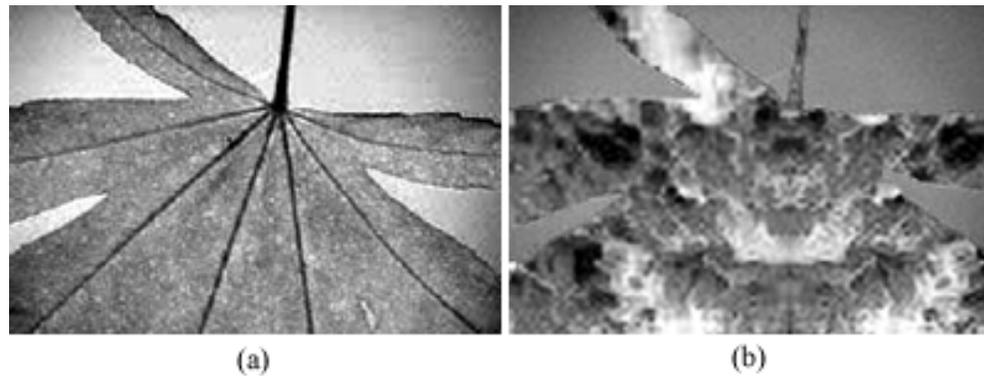


Figure 2.4: Illustration of the functionality of textures in pattern recognition of human vision system. (a)Texture of leaf. (b)Texture of fire.

coarse. In this case, although both objects in Fig. 2.4 have the same shape, one can still tell their difference based on their different style of grey level distribution. This variation in grey tone that constitutes the monochromatic image derived features are called textures.

### 2.3.2 Texture Methods

Tone and texture information are used simultaneously by the human observer for visual interpretation of the scene although at times one can dominate the other. When the ranges of tones in an area of interest are comparable to the ranges of tones in the entire image, human interpretation draws heavily on textural appearance. When seeking to measure texture, one attempts to quantify the nature of the variation in grey level within an object as texture features. Normally, a texture feature is independent of the object's position, size and shape. There are many approaches to extract the texture features in the literature, they can be identified into

three major categories: geometrical, statistical, and model-based texture analysis [14] [10].

Geometrical texture analysis assumes that the texture pattern is a spatial arrangement of texture primitives, which may be of varying or deterministic shapes, such as circles, hexagons, or dot patterns. The texture image is formed from the primitives by placement rules which specify how the primitives are oriented with respect to each other. Texture feature extraction then becomes the task of locating the primitives and quantifying their spatial arrangement. Examples of such textures include tiling of the plane, cellular structures such as tissue samples, and a picture of brick wall [14]. Because sea ice does not have this kind of primitive structures, this method is not applicable to sea ice image interpretation.

Statistical based texture analysis can be broken down into two branches: spectral and spatial based. In spectral based approach, the texture features are extracted in frequency domain by applying the Fourier transform or the multi-channel filters to the original image. The power spectrum, Gabor filter and wavelet transformation are among the commonly used methods in this area. The power spectrum of the image is the Fourier transform of the autocorrelation function, and the multi-channel filters are essentially a set of band-pass filters in frequency domain. Theoretically, the Gabor filter can be regarded as a special wavelet function, and both of them belong to the latter method. The spatial or pixel based approach extracts texture features based on the pixel's grey level of the original image [8] [32]. The texture features of each pixel are the measurements such as smooth, coarse, complex, regular for a local region in the original image. The texture images from the above measurements, also called texture feature maps, can then be used as other "features" in the segmentation process.

Model-based texture analysis is a mathematical process which can synthesize

or describe the texture image. The numerical parameters of the model can be used to establish the distribution law of a texture pattern in a variety of relaxation segmentation processes, or simply used as features to classify the textures.

There is a fundamental difference between model-based and statistical texture analysis. In a model-based study, the texture model has the capability to generate the texture which matches the observed texture; using a statistical method, the texture features are measured without an ideal or representative texture in mind, and the texture features extracted from the image cannot be used in general to synthesize a texture image.

In [13] [12], Clausi explored deeply the power spectrum, Gabor filter and grey level co-occurrence probability method (referred as *GLCP*, which is a spatial based statistical method). Several guiding conclusions obtained by the author. First, power spectrum texture features has poorer classification ability compared to the co-occurrence texture features. Second, the co-occurrence and Gabor features, each can generate a local estimate of the directional frequency, are strongly correlated. Third, the segmentation capabilities of Gabor filters and *GLCP* are different. The Gabor filters are able to capture information in a multi-resolution manner, whereas the co-occurrence features are unable to capture the multi-resolution features if an image contains patterns with different texture resolution. But in the texture classification point of view, the Gabor filters are not ideal candidates since their inherent multi-resolution ability is not utilized for fixed sized samples. According to the author, the Gabor filters are a better approach for image segmentation, and the *GLCP* is better suited for supervised classification than Gabor filters.

The *GLCP* is currently the most widely used method for the SAR sea ice type identification in the literature. Compared to the spectral based method, there are fewer publications exploring systematically the relative classification ability between

it and the model-based method in the literature by now. So, the topic of this thesis will focus on the spatial based statistical texture analysis and model-based texture analysis approaches.

## 2.4 Challenge of Sea Ice Segmentation

Sea ice is a complex and dynamic material. Its representation on SAR image can be determined by many variables as mentioned in section 2.1 and 2.2 as well as weather and geographical conditions. The latter include temperature, wind condition, rate of freezing and salinity of sea water etc. All these variables make the sea ice in SAR image show a very complicated appearance. It is difficult even for human to make an interpretation sometimes. The difficulties for auto-distinguishing of sea ice types using computer can be summarized as following:

1. The surfaces of same ice type might have different degree of smoothness because of their different distances to SAR sensor, which results in their different tone representations on SAR image.
2. Tracing the boundary between different ice types is an unsolved problem. Different types of sea ice mix up with each other along the boundary areas which makes the segmentation problem complicated.
3. The scale of each ice type is different, one ice type may stretch from several meters to several kilometers. It is difficult to decide the neighbourhood size for algorithms based on neighbourhood analysis (i.e. analyzing one region's characteristics based on the characteristics of its neighbourhood regions).

# Chapter 3

## Statistical Texture Analysis

A review of the statistical texture analysis techniques, intended to provide a general framework of this field, as well as the algorithm of *GLCP* approach for texture segmentation are represented in this chapter.

### 3.1 First Order Methods

For some images, image segmentation could be easily implemented using only the intensity value of each pixel. In such cases, no high order texture features are needed. But most of the time, the identification of class types in the image can not be done so easily based only on the grey levels of the pixels. Then the texture features become an helpful information in the segmentation process. The simplest approach of texture analysis method in statistical category is known as first order methods [32]. The statistics are based on individual pixel values, not on the relationship between pixels. Texture features such as mean, variance, standard deviation, gradient, and skewness are usually extracted from the image [32].

Before the 1990s, the first order statistical texture analysis was widely used in SAR image processing. Shuchman *et al* analyzed the X-band (HH) SAR data of sea ice (resolution 15 *m*, collected in March and April 1987) with respect to discriminating open water, new ice, first year ice, and multi-year ice [43]. The authors stated that mean versus variance could segment out open water and first year ice, and skewness versus modified skewness could segment the new ice and multi-year categories. As the conclusion, it was stated that *GLCP* method generated very similar result to the mean and standard deviation analysis. However, [43] is short of further explanation on the similarity criteria and what *GLCP* texture statistics were used. Burns and Kasischke [7] investigated the application of a simple texture extraction technique based on the calculation of local grey level variance statistics. In order to indicate the performance of their technique, second order statistical texture analysis method was also used for comparison purpose (discussed in Section 3.2). They found that the latter one performed slightly better on the data set than their variance technique for classifying first year and multi-year ice types. They concluded that the simpler variance technique would be preferable in an operational ice mapping system given the considerations of accuracy and timeliness. Barber *et al* used the average grey tone signatures from X-band HH polarized SAR sea ice image in a supervised classification scheme [16]. As the authors expected, the classification results were poor. Classification accuracies were 51 and 33 percent for the training and test data sets respectively. The authors analyzed that the reasons for high classification error lie in the large variability of grey levels of the first year rough class.

## 3.2 Second Order Methods

If an image shows regions with equal first order statistics, then second order statistics should be used. As with the first order approach, the second order texture analysis also makes use of the grey level of each image pixel. The main difference between these two is that in the second order approach the texture features are extracted from an intermediate relationship matrix which is calculated using the pixels within a pixel's neighborhood and used to save some statistical relationships between each pixel and its neighbours. Whereas in the first order method, the texture features are extracted directly from the gray levels of the image. There are several commonly used methods in this category [47]: grey level difference (*GLD*), grey level run length (*GLRL*), and grey level co-occurrence probability (*GLCP*).

*GLD* method tends to indicate the presence and direction of "edges" within an image [46, 47]. The location and presence of edge information is a clue to the visual system in the recognition process and has been shown to contribute to increase classification accuracy. Let  $f(m, n)$  be the image, for any given displacement value  $\eta(\Delta m, \Delta n)$ , let

$$f_\eta(m, n) = |f(m, n) - f(m + \Delta m, n + \Delta n)|$$

where  $\Delta m, \Delta n$  are integers. Let  $p_\eta$  be the probability density of  $f_\eta(m, n)$ , that is

$$p_\eta(i|\eta) = p(f_\eta(m, n) = i).$$

If there are  $k$  grey levels,  $p_\eta$  will be a  $k$ -dimensional vector whose  $i^{th}$  element is the probability that  $f_\eta(m, n)$  will have value  $i$ . Given the displacement value  $\eta$ , a texture is said to be coarse when  $\eta$  is smaller comparing to the texture element size. The  $p_\eta$  value of coarse textures tend to concentrate near  $i = 0$ . Whereas, for a fine texture, the grey levels with separation  $\eta(\Delta m, \Delta n)$  are usually different, then  $p_\eta$  will

be more spread. In this way, the values in  $p_\eta$  can tell the coarseness information about the texture. For various magnitudes of  $\eta$ , several texture features, called contrast, entropy, angular second moment, inverse difference moment etc, can be calculated from  $p_\eta$ .

*GLRL* statistics represents contiguous occurrences of identical grey level values in an image [47]. In this method, the *GLRL* matrix  $R(i, j|\theta)$  whose element  $r(i, j|\theta)$  denotes the number of occurrences of a particular grey level  $i$  of a specified run length  $j$  along orientation  $\theta$  is constructed. For a coarse texture, for each grey level  $i$ , longer run lengths are expected in  $R(\theta)$ , and for fine texture, grey levels change with higher frequency, the shorter run lengths are obtained. So, for texture with different coarseness, the distribution of their  $R(\theta)$  is different. Based on  $r(i, j|\theta)$ , several texture features, called short run emphasis, long run emphasis, grey level distribution, run length distribution, and run percentages, etc., can be calculated.

The *GLCP* method can measure textural characteristics such as homogeneity, grey level linear structure, contrast, entropy, and image complexity [13]. They are estimated based on the two dimensional joint probability matrix  $p(i, j|\delta, \theta)$  called grey level co-occurrence matrix (*GLCM*). Each element in the *GLCM* is the number of occurrences of a pairwise combination of grey levels  $i$  and  $j$ , given that the displacement  $\delta$  and the orientation  $\theta$ . A different *GLCM* is required for each  $(\delta, \theta)$  pair. The *GLCM* can be regarded as a data structure which is used to store the relationship information between pixels, and then the *GLCP* texture features can be extracted from it. The detailed implementation of this method will be described in section 3.4.

### 3.3 The Comparison of Second Order Methods

Among the three approaches introduced in the last section, *GLCP* is the one that is most commonly used. The reason is detailed in this section.

A comparison between *GLD*, *GLRL* and *GLCP* is needed since the first choice a researcher faces is the selection of the method to be used. Among many publications, Weszka *et al* [31] and Connors [47] explored this problem more directly and clearly using different approaches. The metric of comparison employed by Weszka is a classification comparison. Such a procedure has three drawback:

1. A large data base is needed to obtain a high confidence level in the result.
2. The results are data dependent, i.e. the comparison results may not be indicative to relative power of each algorithm for some texture types other than the textures used in the experiment.
3. This method cannot be used to determine whether the set of texture statistics defined in each method is inadequate or the method itself does not contain all the important texture information needed to do the discrimination.

Connors provided a more systematical and theoretical comparison. The comparison procedure was designed to compare the relative loss of important texture context information experienced in going from the digital image to the intermediate relationship matrices of each of the algorithms considered, where the matrices are the grey level difference density  $p_\eta(i|\eta)$ , the grey level run length matrix  $R(i, j|\theta)$ , and the grey level co-occurrence matrix  $p(i, j|d, \theta)$ . A texture algorithm is said to experience a loss of important texture context information if there exist two visually distinct textures which cannot be discriminated by this algorithm. To

determine whether an algorithm experiences a loss of important texture context information, they found out whether the intermediate relationship matrices contain information which will allow the discrimination of these two textures. If the corresponding matrices extracted from the two visually distinguishable textures are equal, an information loss is experienced. The detailed demonstration can be found in [47].

The textures used by Connors are similar to the ones used by perceptual psychologist B. Julesz in his investigations of human texture perception [33]. In order to show how well each algorithm can sense pattern changes, the textures used in his study also have the same first order distribution, i.e., the same mean, variance, and skewness. The conclusions of the research can be summarized by Fig. ??.

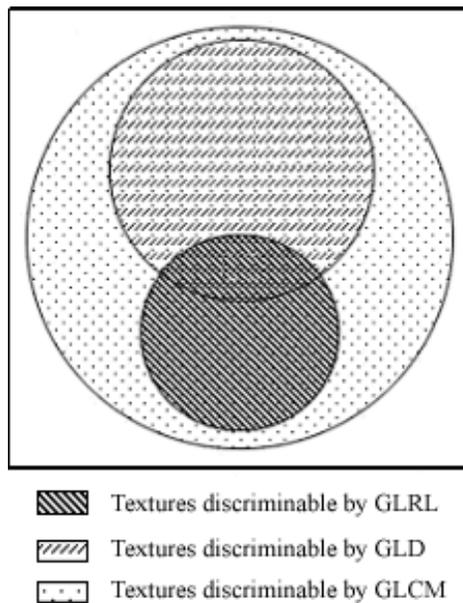


Figure 3.1: A summary of the relative merits of the *GLRL*, the *GLD*, and the *GLCP* (adapted from [Connors 80]).

The area without texture pattern stands for the set of texture pairs that can not be distinguished by all methods. One important conclusion can be drawn from this figure. Over the classes of textures *GLCP* is more powerful than *GLRL* and *GLD*, i.e., the texture pairs which can be distinguished by *GLRL* and *GLD* methods are a proper subset of the set of texture pairs which can be discriminated by *GLCP* method. These theoretically derived results agree very well with the experimentally performed comparison done by Weszka *et al* [31].

### 3.4 *GLCP* Texture Features

*GLCP* texture features extract discriminating information from *GLCM* with different frequencies along different directions in the image space. Commonly used orientations are shown in Fig. 3.2:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$ . The orientations of  $180^\circ$ ,  $225^\circ$ ,  $270^\circ$  and  $315^\circ$  are redundant to the previous four directions.

#### 3.4.1 *GLCM*

Figs. 3.3 and 3.4 show an example about the generation of *GLCM* from a five by five sub-image window with  $\theta = 90^\circ$  and  $\delta = 1$ . Suppose there are six intensity levels (4, 5, 6, 7, 8, 9) in the image. To determine the *GLCM*, the program will scan the sub-image window and count the occurrences of each pair of pixel. For example, the first co-occurring pixel pair is (4, 5). The second is (6, 5), and so on. The un-normalized *GLCM* can be obtained by accumulating the number of occurrences of each relationship defined by  $(\delta, \theta)$ . For example, the (5, 7) occurs twice, and the (8, 9) occurs once. Therefore the (5, 7) element in the un-normalized co-occurrence matrix is set to two while the (8, 9) element is set to one. Note that such a *GLCM* is

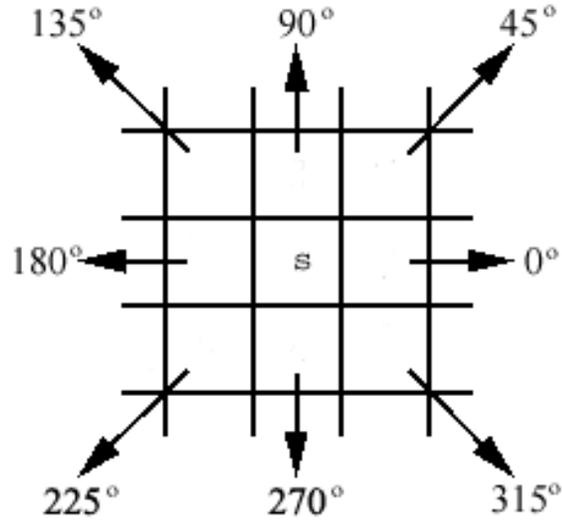


Figure 3.2: Eight directions for the creation of *GLCM* of pixel  $s$ .

asymmetric. In order to consider  $90^\circ$  and  $270^\circ$  orientation together, the asymmetric co-occurrence matrix above is added by its transpose. To obtain the normalized *GLCM* in Fig. 3.4, it is just needed to divide each element in the un-normalized co-occurrence matrix by the number of total pairs used to build it, which is also the sum of all the elements in the un-normalized co-occurrence matrix as follows:

$$P(i, j|\delta, \theta) = \frac{C(i, j|\delta, \theta)}{\sum_{i,j=4}^9 C(i, j|\delta, \theta)} \quad (3.1)$$

where  $C(i, j|\delta, \theta)$  is the element in the un-normalized co-occurrence matrix.

The notation *GLCM* mentioned in this thesis refers to the normalized *GLCM*. Let  $P$  is the probability element in *GLCM* of Fig. 3.4, it is clear that

$$\sum_{i,j=4}^9 P(i, j|\delta, \theta) = 1. \quad (3.2)$$

If the full dynamic range of an typical image is used, each *GLCM* is a  $256 \times 256$

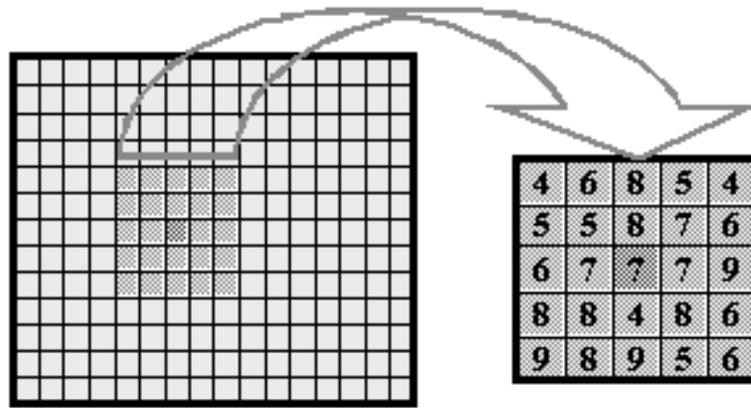


Figure 3.3: Sub-image window from the texture image

matrix (65536 entries). If the sub-image window size is ten by ten, then there are only  $2 \times (10 \times 9) = 180$  entries in the matrix. In this case, a *GLCM* can be quite sparse. In order to have a *GLCM* that adequately reflects the joint probability distribution, it must contain a reasonably large average occupancy level. To solve this problem, the quantization method should be used. The decision that one has to make is how many intensity levels are needed to represent a set of texture successfully. In [36], Soh and Tsatsoulis extracted sample sites from test images with different textural contexts. They devised a test using six texture statistics and six different uniform quantization levels: 8, 32, 64, 128 and 256. The displacement is set to one and orientations to  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$ . The different values of quantization level are evaluated using five measurements based on Euclidean distance along each textural feature between each pair of sample sites. The objective of the Euclidean distance measurements is to provide a visual presentation of the trend between the differences among each successive pair of quantization schemes. Based on their results, they found that the 8-level quantization scheme should not be used because of its poorer distinguishing ability compared to other four schemes.

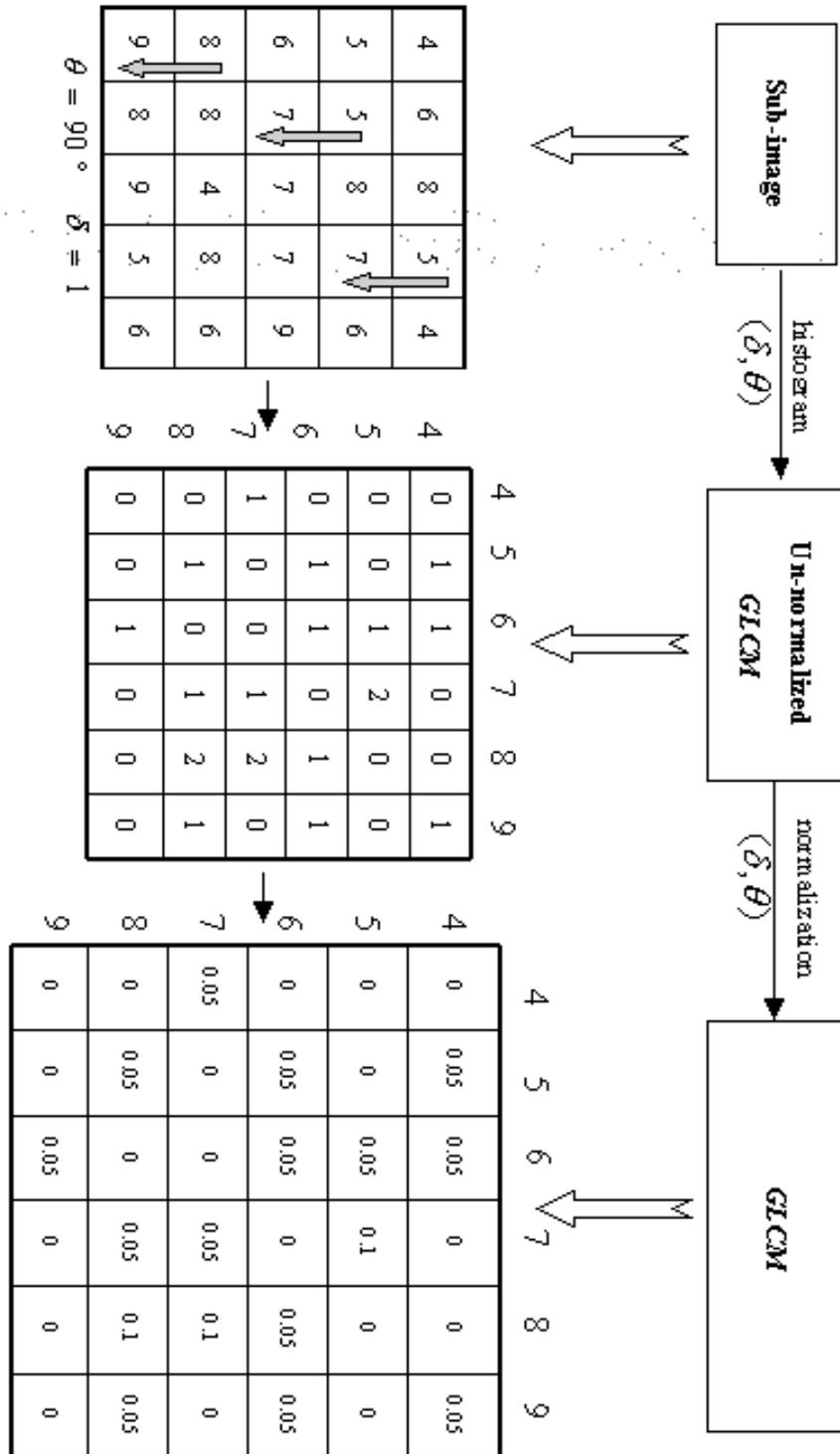


Figure 3.4: Calculation of the GLCM

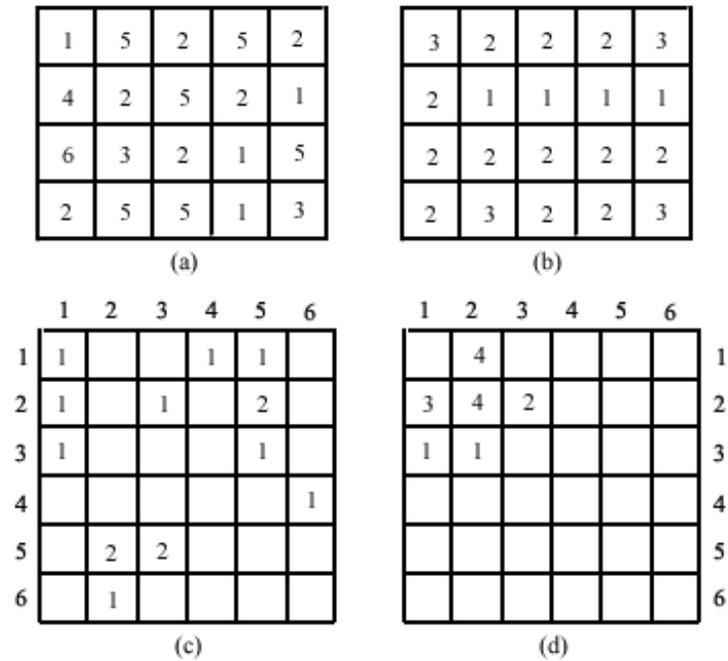


Figure 3.5: The *GLCMs* of different textures. (a) Intensity values of high texture. (b) Intensity values of low texture. (c) *GLCM* of a. (d) *GLCM* of b.

The distribution of the non-zero elements in *GLCM* is expected to characterize the texture of the image. For example, the degree of dispersion that the elements in *GLCM* have about the diagonal characterizes the texture of the sub-image. A small dispersion along the diagonal means that the texture is coarse compared to the size of sub-image window (the texture elements are larger than the window, and within the area of window, the texture is smooth). The *GLCM* of high texture region Fig. 3.5 a and low texture region Fig. 3.5 b are shown in Fig. 3.5 c and Fig. 3.5 d. The different element distributions of these two *GLCMs* illustrate the above explanation.

### 3.4.2 Texture Feature Extraction from *GLCM*

Based on the *GLCM*, a number of texture statistics can be applied. Haralick suggested 14 statistics [44]. Connors and Harlow then showed that the information contained within the *GLCM* is not adequately captured by any single statistic [47]. Barber and LeDrew further demonstrated that some of the statistics are highly correlated and others are not. Based on the separability analysis using univariate (signal *GLCP* texture statistics for discriminating class types) and multivariate (several *GLCP* texture statistics together for discriminating class types), they concluded that the best discrimination could be achieved from a set of three texture features considered simultaneously [2]. Later on, based on the correlation analysis and experiment results using Brodatz, Limex and SAR sea ice image, Clausi suggested the following three co-occurrence texture statistics [13]:

*Dissimilarity:*

$$\sum_{i,j=0}^{N-1} P(i, j|\delta, \theta)|i - j| \quad (3.3)$$

*Correlation:*

$$\sum_{i,j=0}^{N-1} \frac{P(i, j|\delta, \theta)(i - \mu_i)(j - \mu_j)}{\sigma_i \sigma_j} \quad (3.4)$$

*Entropy:*

$$- \sum_{i,j=0}^{N-1} P(i, j|\delta, \theta) \log P(i, j|\delta, \theta) \quad (3.5)$$



Figure 3.6: Original texture image

where  $N$  is the grey level of the image,  $(\mu_i, \mu_j)$  and  $(\sigma_i, \sigma_j)$  is the standard deviation of row  $i$  and column  $j$  of  $P(i, j|\delta, \theta)$  defined as follows:

$$\mu_i = \sum_{i,j=0}^{N-1} P(i, j|\delta, \theta) i \quad (3.6)$$

$$\sigma_i = \sum_{i,j=0}^{N-1} P(i, j|\delta, \theta) (i - \mu_i)^2 \quad (3.7)$$

The above feature extraction methods are also called grey level shift invariant statistics [13]. The segmentation based on these statistics will not be a function of pixel intensity values. Applying the texture statistics to *GLCM*, the texture feature maps can be obtained for each pixel of the image. The texture feature maps of a texture image shown in Fig. 3.6 are given in Fig. 3.7. The window size is five by five (left column) and fifteen by fifteen (right column).

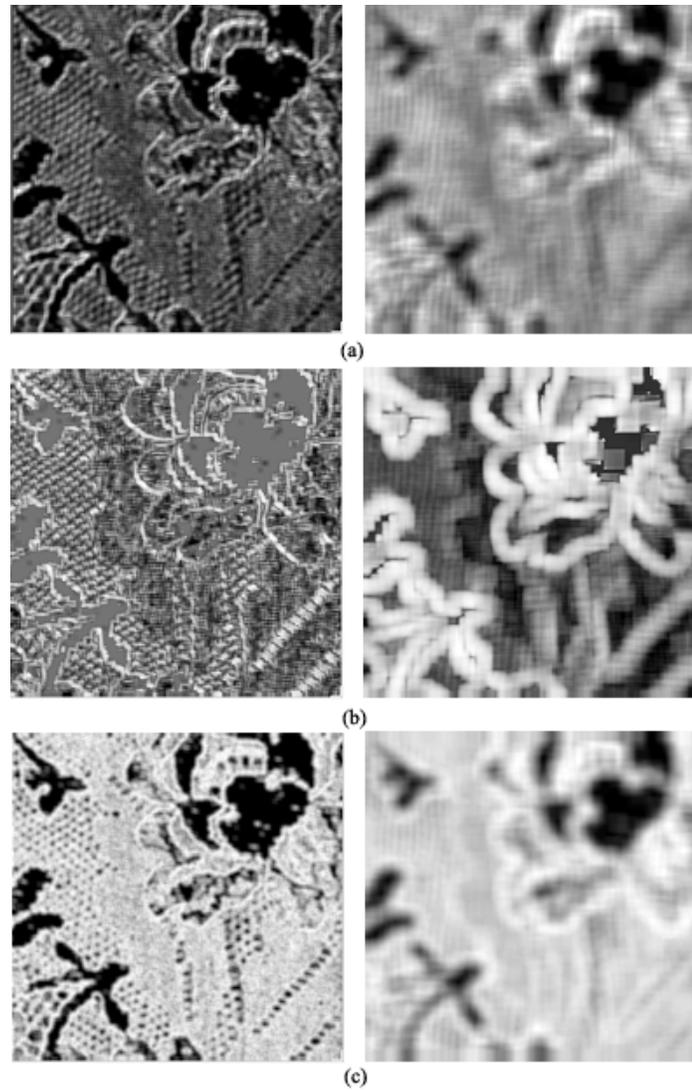


Figure 3.7: The *GLCP* texture features extracted from the texture image shown in Fig. 3.6 ( $\delta = 1$ ,  $\theta = 45$ , window size: Left column:  $5 \times 5$ , Right column:  $15 \times 15$ ). (a) Dissimilarity. (b) Correlation. (c) Entropy.

Observing the texture feature maps in Fig. 3.7, and analyzing the above equations at the same time, several conclusions can be drawn:

1. Dissimilarity measures the amount of local variation in the image along a certain orientation and displacement. It is high when the region within the sample window has a high contrast. In image (a), the boundary areas between the bright flower and the dark background have higher values.
2. Correlation measure the linear dependency of grey levels of neighbourhood pixels along a certain orientation and displacement. When the scale of local texture is much large than the sub-image window, correlation is typically high, and when the local texture has a scale similar or smaller than the window size, there will be low correlation between pairs of pixels.
3. Entropy measures the degree of disorganization in the local window along a certain orientation and displacement. It is high when the elements in *GLCM* have relatively equal values, which means that no pair of intensity values dominates within the sample window.

Besides the texture statistics, there are a number of other parameters that the user should set when using the *GLCP* method. Since the dimension of the *GLCM* is determined by the grey levels of the image, the first parameter needed to set is the quantization level. If the quantization level is large, the *GLCM* will be large and the computation is burdensome. Conversely, if the quantization level is too small, some texture information will be lost. The window size of the sub-image is another important parameter. It will determine how many neighbourhood pixels are involved to build the *GLCM* for the pixel concerned. If the window size is too small, no sufficient statistical information from the pixels in the window could

be obtained, if the window size is too large, some unrelated neighbourhood pixels could interfere the statistics. For the segmentation problem, generally speaking, coarse textures require relatively smaller windows, and smooth texture can have larger window size for a consistent measurement. Comparing the texture feature maps on the left column and right column in Fig. 3.7, one can see the clear boundary obtained from five by an five by five window size is blurred when using an fifteen by fifteen window size. The selection of  $\theta$  is data dependent. For the texture images that do not have dominate orientation, several orientation can be averaged together [36]. As with the window size selection, the selection of  $\delta$  should be considered together with the scale of the texture in the image: small  $\delta$  for fine textures and larger  $\delta$  for coarse textures.

### 3.5 Statistical Second Order Texture Analysis for Sea Ice Segmentation

Reviewing the literature on second order statistical texture analysis methods applied in SAR sea ice type interpretation, *GLCP* method is the most preferred algorithm currently used [16] [42] [43]. The popularity of this method are just consistent with what Connors and Harlow proved in [47].

Holmes *et al* used X-band SAR sea ice image with three by three meter resolution and HV polarization for sea ice classification [42]. The texture measurements are entropy and contrast which are obtained by considering the average of orientations. According to the authors, the intensity range of different ice type has a better separation in entropy and contrast texture feature maps than in the original image. The classification is based on the intensity thresholds of each ice type which are

setup from the texture feature maps of training data. There are several uncertainties in this paper. First, as the authors said the reason for using the average entropy and average contrast was to obtain texture measures which are insensitive to the orientation of the sensor. By doing this, they had already made an assumption that the orientation of the sensor is not important for the textures in sea ice image. Second, the separation of ice types by intensity thresholds is quite image dependent and not robust, different image types (e.g. different polarization) may have different thresholds.

Barber and LeDrew [2] explored five co-occurrence texture statistics of *GLCP* method using an X-band HH SAR image. The five statistics are uniformity, correlation, entropy, dissimilarity and contrast. Classification accuracy based on different orientations, texture statistics and displacements are compared. Significantly better results were obtained at  $\theta = 0^\circ$  than other orientations. The improved accuracies were obtained from the several texture statistics rather than signal one.

In 1999, L. K. Soh and C. Tsatsoulis [36] explored the potential of *GLCP* texture features in SAR sea ice classification. The authors evaluated parameters of *GLCM* ( $\delta$ ,  $\theta$ , quantization level) quantitatively to determine which texture statistics (including energy, contrast, correlation, homogeneity, entropy, autocorrelation, dissimilarity, cluster shade, cluster prominence, maximum probability) were best for mapping sea ice texture. They tested different choices for  $\delta = 1, 2, \dots, 32$ ,  $\theta = 0, 45, 90, 135$ , and quantization levels = 8, 16, 32, 64, 128, 256 by examining the effects of applying the texture maps to a Bayes classifier. They showed that a complete grey level representation of the images was not necessary for texture segmentation, and the displacement in texture measurements was more important than orientation. The experiments concluded that the best *GLCP* implementation in representing sea ice texture was one that utilizes a range of displacement val-

	Orientation ( $\theta$ )	Displacement ( $\delta$ )	Texture statistics	Quantization level
Brodatz	( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ )	1, 3	dissimilarity, correlation, entropy	16
Sea Ice	average of ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ )	1, 2	dissimilarity correlation, entropy	16

Table 3.1: *GLCM* parameters of Brodatz and SAR sea ice images used in this thesis.

ues such that both micro-textures and macro-textures of sea ice can be adequately captured.

In *GLCP* method, one of the most important parameters, which most researchers chose arbitrarily is the window size for calculating the *GLCM* for each pixel. [36] used  $64 \times 64$ , [2] used  $25 \times 25$ , and [17] used  $32 \times 32$  and  $64 \times 64$ . By now, no one provides definite criteria to decide the window size. The selection has to be a compromise between being large enough to provide stable estimates yet small enough not to severely distort the values computed for the small isolated patches of some ice types. Some general considerations need to be taken for choosing the window size: the resolution of the SAR image, the scale of the texture and the displacement value used. The finer the texture, the smaller the window size should be.

Based on the suggestions given in [36] and [13], the parameters used to build *GLCM* for Brodatz texture images and SAR sea ice images in this thesis are listed in Table 3.1. A k-means classifier is used for segmentation. The parameters used for the synthetic texture images is the same as the Brodatz texture images. Because the textures of sea ice do not seem to have obvious directional tendency, the texture features along four directions are averaged.

# Chapter 4

## Model-based Texture Analysis

Different images have different natures. An image processing or analysis technique only applies well for certain kinds of images. Thus it is important for the designer of such techniques to be able to characterize, i.e., to model, the image that is to be processed. Once the model of the image is obtained, it will serve to explain the dominant statistical characteristics of the given data, and the subsequent processing of the images can be efficiently done using the model. In this regard, any analytical expression that explains the nature and extent of dependency of a pixel intensity on intensities of its neighbours can be said to be a model [9].

Image models were used by some researchers even in the early years, but the awareness of their importance was enhanced at the end of the 1970s by a workshop on image modelling held in Chicago in August, 1979 [49]. Since then, the research activities in the image modelling area have increased considerably. Two-dimensional time series and random field models probably are the most common classes of image models which focus on the attempt to characterize the relationship among the neighbouring pixels. Owing to the cross-fertilization of ideas from

image processing, spatial statistics, and statistical physics, the last two decades have witnessed a significant amount of research activity in image modelling with special emphasis on Markov random fields (*MRFs*) [49]. The philosophy of modelling images through the local interaction of pixels as specified by *MRF* models is intuitively appealing due to the following reasons:

1. One can systematically develop algorithms for a variety of problems based on sound principles rather than ad hoc heuristics [28].
2. It is easy to derive quantitative performance measures for characterizing how well the image analysis algorithms work [39].
3. *MRF* models can be used to incorporate prior contextual information or constraints in a quantitative way [27].
4. The *MRF*-based algorithms tend to be local which permits parallel updating of the pixel sites, and tend to parallel hardware implementation in a natural way (the full parallel potential is realized by assigning one processor to each site of the intensity process)[27].

The practical use of *MRF* models is largely ascribed to the equivalence between *MRFs* and Gibbs distributions established by Hamersley and Clifford in 1971 and further developed by Besag in 1974 for the joint distribution of *MRFs* [5]. This enables us to model vision problems by a mathematically sound yet tractable means for the image analysis in the Bayesian framework. From the early 1980s, a plethora of papers has appeared on various aspects of *MRF* models including application in texture synthesis [14], image classification and segmentation [19] [35] [41], image restoration and artificial neural networks [27] [39]. Issues such as parameter estimation and hypothesis testing in *MRF* models also have been addressed [5] [57].

The following two sections will provide a detailed description of *MRF* theory as well as Markov-Gibbs Equivalence theorem.

## 4.1 Markov Random Fields

### 4.1.1 Neighbourhood System and Clique

Let  $X$  represent an image. Then  $X(i, j)$  represents a random variable at a site  $(i, j)$  on the  $M \times M$  lattice system  $S$ . For convenience,  $X(i, j)$  can be indexed as  $X_s$ ,  $s = 1, 2, \dots, M^2$ . A label is an event that may happen to a site in the image defined on  $S$ . Let  $L$  be a set of labels, in the discrete case, a label  $L_s$  assumes a discrete value in a set of  $G$  labels  $\{l_1, l_2, \dots, l_G\}$ . The labeling problem is to assign a label from the label set  $L$  to  $X_s$ . For example,  $L$  can be a discrete set  $\{0, 1, \dots, 255\}$  which represents the quantized intensity value of a pixel site  $s \in S$ , where elements in  $S$  index the image pixels. In segmentation problems,  $X_s$  is assigned a label from the set  $L = \{class1, class2, \dots, classL\}$ . The set  $\{X_1, X_2, \dots, X_{M^2}\}$  is called a configuration of the sites on  $S$  in terms of labels in  $L$ .

The inter-relationship between the label of one pixel and the labels of its surrounding pixels are maintained by a so-called neighbourhood system. A neighbourhood system for  $S$  is defined as

$$N = \{N_s | s \in S\} \quad (4.1)$$

where  $N_s$  is the set of sites neighbouring  $s$ . The neighbouring relationship has the following properties:

1. A site is not neighbouring to itself:  $s \notin N_s$ ;

2. The neighbouring relationship is mutual:  $s' \in N_s \Leftrightarrow s \in N_{s'}$ .

In the first order neighbourhood system, also called the four-neighbourhood system, every (interior) site has four neighbours, as shown in Fig. 4.1 (a) where  $s$  denotes the considered site. In the second order neighbourhood system, also called the eight-neighbourhood system, there are eight neighbours for every (interior) site, as shown in (b). The numbers  $a = \{1, \dots, 5\}$  shown in (c) indicate the outermost neighbouring sites in the  $a^{\text{th}}$  order neighbourhood system. The order of a neighbourhood reflects its extent. The shape of neighbourhood set may be described as a hull enclosing each site in the set, and the hull can be thought of as a type of interface which can separate the site from the outside world, i.e. the existence of the site only relies on its neighbourhood comprised in the hull. The first order and second order system denoted here are different from the same terminologies used in statistical texture analysis of the previous chapter. In *MRF* texture analysis, they are the terminologies used to describe the neighbourhood size of a pixel. In statistical texture analysis, they are used to differentiate the complexity of the statistical methods applied to an image.

The pair  $(S, N)$  constitutes a graph in the usual sense where  $S$  contains the pixels and  $N$  is the neighbourhood system of  $S$ . A clique  $c$  for  $(S, N)$  is defined as a subset of sites in  $S$  in which every pair of distinct sites are neighbours. It consists either of a single site  $c = \{s\}$ , or of a pair of neighbouring sites  $c = \{s, s'\}$ , or of a triple of neighbouring sites  $c = \{s, s', s''\}$ , and so on. Let  $C_1$ ,  $C_2$  and  $C_3$  denote the collections of single-site, pair-site and triple-site cliques respectively. So the collection of all cliques for  $(S, N)$  is  $C = C_1 \cup C_2 \cup C_3 \dots$ , where “ $\dots$ ” denotes all possible sets of large cliques.

The type of a clique is determined by its size, shape and orientation. (d)-(h)

in Fig. 4.1 show clique types for the first and second order neighbourhood systems for a lattice. The cliques of first order neighbourhood system are shown in (d) and (e). The clique types for the second order neighbourhood system include (d) and (e) as well as (f), (g) and (h). Obviously, the number of cliques grows rapidly with the order and so do the involved computational requirements.

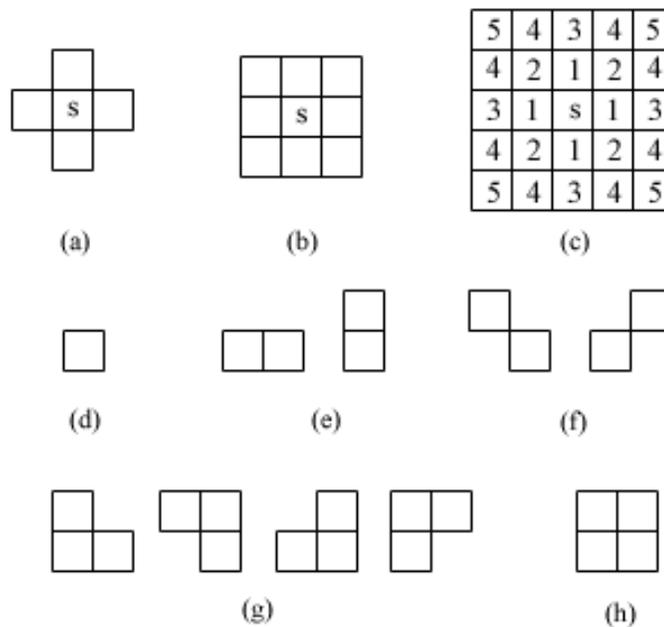


Figure 4.1: Neighbourhood systems and cliques

### 4.1.2 Markov Random Field Definition

After defining the neighbourhood system, configuration problem, and clique, the definition of *MRF* can be presented. Observing an image, if it is not a random noise image, one can notice that neighbouring pixels typically have similar brightness.

The intensity value of a pixel is highly dependent on the intensity values of its neighbourhood pixels, and the textures distributed within a local region tend to be the same. This phenomenon can be called local similarity. In fact, in image processing, the notion of near neighbour dependence is pervasive [14] [28]. *MRF* is a type of mathematical tool to model this dependence.

Let  $X = \{X_1, \dots, X_{M^2}\}$  be a family of random variables defined on the lattice system  $S$ . The family  $X$  is called a random field. Lowercase letters denote the values assumed by these random variables, i.e. the notation  $X_s = x_s$  denotes the event that  $X_s$  takes the value  $x_s$ . The joint event  $X = x$  where  $x = \{x_1, \dots, x_{M^2}\}$  can be regarded as a configuration of  $X$ , or in other words, a realization of the field  $X$  with significant interaction between neighbourhood sites.

For a discrete label set  $L$ , the probability that random variable  $X_s$  takes the value  $x_s$  is denoted  $P(X_s = x_s)$ , abbreviated  $P(x_s)$ , and the joint probability is denoted  $P(X = x) = P(X_1 = x_1, \dots, X_{M^2} = x_{M^2})$  and abbreviated  $P(x)$ .  $X$  is said to be a *MRF* on  $S$  with respect to a neighbourhood system  $N$  if and only if the following two conditions are satisfied:

1. *Positivity:*

$$P(x) > 0 \quad \forall x \quad (4.2)$$

2. *Markovianity:*

$$P(x_s | x_{S-s}) = P(x_s | x_{N_s}) \quad (4.3)$$

where  $S - s$  is the set difference,  $x_{S-s}$  denotes the set of labels at the sites in  $S - s$  and  $x_{N_s}$  stands for the set of labels at the sites neighbouring  $s$ . The left hand of equation (4.3) is called the local characteristics of *MRF* and it turns out that the joint probability of any process satisfying equation (4.2) is uniquely determined by these local characteristics [5].

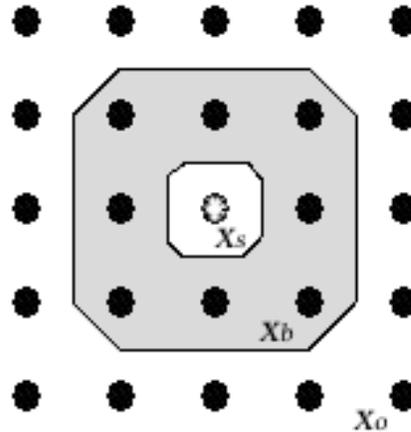


Figure 4.2: The Markovianity property: the boundary set  $\{x_b\}$  separates the inside pixel  $x_s$  and outside pixels  $\{x_o\}$ .

The Markovianity depicts the local characteristics of image  $X$ : a site interacts with only the neighbouring sites. In other words, only neighbouring sites have direct interactions with each other. Fig. 4.2 shows a second order *MRF* neighbourhood. The pixel set  $\{x_b\}$  located in the boundary decouples the inside pixel  $x_s$  and outside pixel set  $x_o$ . The pixels located in boundary set  $\{x_b\}$  are said to be the neighbourhood of  $x_s$ . Notice that if the neighborhoods are large enough to encompass the dependencies, any process which satisfies equation (4.2) and (4.3) will be a *MRF*. In practice, based on prior knowledge, effort is made to select the size of neighbourhood. It should be large enough to contain the necessary information to model different classes of images, and at the same time small enough to ensure feasible computational loads. A *MRF* is said to be homogeneous if  $P(x_s|x_{N_s})$  is independent of the relative position of site  $s$  in  $S$ .

In image processing, it may be necessary to define a few coupled *MRFs* for some problems, each defined on one of the spatially interwoven set of sites. For example,

in the related tasks of image segmentation, two *MRFs*, one for pixel values ( $x_s$ ) and the other for the underlying label values ( $l_s$ ), can be defined on the image lattice and its label set respectively. They are coupled to each other via the conditional probability  $P(x_s|l_s)$ .

## 4.2 Markov-Gibbs Equivalence

Theoretically, A *MRF* can be specified in two ways: the conditional probabilities approach  $P(x_s|x_{N_s})$  and the joint probability approach  $P(x)$ . Besag argues for the joint probability approach in view of the disadvantages of the conditional probability approach [5]. Firstly, no obvious method is available for deducing the joint probability from the associated conditional probabilities. Secondly, the conditional probabilities themselves are subject to some non-obvious and highly restrictive consistency conditions. Thirdly, the natural specification of equilibrium of statistical process is in terms of the joint probability rather than the conditional distribution of the variables. Fortunately, a theoretical result about the equivalence between *MRFs* and Gibbs distribution provides a mathematically tractable means of specifying the joint probability of an *MRF*.

### 4.2.1 Gibbs Random Field Definition

Gibbs models originated from statistical physics and were introduced into image modelling by Hassner and Sklansky in 1983 [27]. A Gibbs distribution  $P(x)$  relative to  $(S, N)$  measures the probability of the occurrence of a particular configuration  $x$  in the image domain. It can be represented as follows:

$$P(x) = Z^{-1}e^{-\frac{1}{T}U(x)} \quad (4.4)$$

where

$$Z = \sum_{x \in X} e^{-\frac{1}{T}U(x)} \quad (4.5)$$

is a normalizing constant called the partition function,  $T$  is a constant called the temperature which controls the degree of peaking in the density  $P(x)$ , and  $U(x)$  is the energy function taking the form

$$U(x) = \sum_{c \in C} V_c(x). \quad (4.6)$$

The energy is a sum of clique potentials  $V_c(x)$  over all possible cliques  $c$ . The value of  $V_c(x)$  depends on the local configuration on the clique  $c$ . By choosing a particular function  $V_c(x)$ , the *a priori* contextual information can be incorporated in the model.

A set of random variables  $x$  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. A *GRF* is said to be homogeneous if  $V_c(x)$  is independent of the relative position of the clique  $c$  in  $S$ . It is said to be isotropic if  $V_c$  is independent of the orientation of  $c$ . To specify a *GRF* distribution if it is homogeneous or isotropic is simpler than specifying one without such properties [27]. The homogeneity is assumed in most *MRF* vision models for mathematical and computational convenience. The isotropy is a property of direction-independent regions.

Sometimes, it may be convenient to express the energy of a Gibbs distribution as the sum of several terms, each ascribed to cliques of a certain size, that is:

$$U(x) = \sum_{s \in C_1} V_1(x_s) + \sum_{\{s, s'\} \in C_2} V_2(x_s, x_{s'}) + \sum_{\{s, s', s''\} \in C_3} V_3(x_s, x_{s'}, x_{s''}) + \dots \quad (4.7)$$

This implies a homogeneous Gibbs distribution because  $V_1$ ,  $V_2$  and  $V_3$  are independent of the locations of  $s$ ,  $s'$  and  $s''$ . For non-homogeneous Gibbs distributions, the clique functions should be written as  $V_1(s, x_s)$ ,  $V_2(s, s', x_s, x_{s'})$ , and so on.

$P(x)$  measures the probability of the occurrence of a particular configuration  $x$ . The more probable configurations are those with lower energies. The temperature  $T$  controls the sharpness of the distribution. When the temperature is high, all configurations tend to be equally distributed. Near zero temperature, the distribution concentrates around the global energy minimum. Usually,  $T$  is assumed to be 1 unless otherwise stated.

## 4.2.2 Hammersley-Clifford Theorem

An *MRF* is characterized by its local property (Markovianity) whereas a *GRF* is characterized by its global property (the Gibbs distribution). The Hammersley-Clifford theorem establishes the equivalence of these two types of properties. The theorem states:

*$X$  is an MRF on  $S$  with respect to  $N$  if and only if  $X$  is a GRF on  $S$  with respect to  $N$ .*

Many proofs of the theorem exist (e.g. in [5]). According to this theorem, if  $X$  is a Gibbs distribution, then for a configuration  $x$  of  $X$ , one has

$$P(x_s | x_{S-s}) = \frac{e^{-\sum_{c \in A} V_c(x)}}{\sum_{x'_s} e^{-\sum_{c \in A} V_c(x'_s)}} \quad (4.8)$$

where  $A$  consists of cliques containing  $s$  and  $x'_s$  denotes any possible values on position  $s$ . Eqn. (4.8) only depends on  $x_s$  and its neighbours.

The practical value of this theorem is that it provides a simple way of specifying the joint probability by providing an explicit formula for the joint probability distribution  $P(X = x)$  in terms of the energy function. One can specify the joint probability  $P(X = x)$  by specifying the potential functions together with

the neighbourhood system. In this way, the *a priori* knowledge or preference about interactions between labels is also easily encoded in the image processing procedure.

How to choose the forms and parameters of the potential functions for a proper encoding of constraints is a major topic in *MRF* modelling [29] [37]. The forms of the potential functions determine the form of the Gibbs distribution. When all the parameters involved in the potential functions are specified, the Gibbs distribution is completely defined. To calculate the joint probability of an *MRF*, which is a Gibbs distribution, it is also necessary to evaluate the partition function  $Z$ . Because it is the sum over a combinatorial number of configurations of  $X$ , the computation is usually intractable. The explicit evaluation can be avoided in maximum probability based *MRF* vision models when  $U(x)$  contains no unknown parameters. However, this is not true when the parameter estimation is also a part of the problem. In the latter case, the energy function  $U(x) = U(x|\theta)$  is also a function of parameters  $\theta$  and so is the partition function  $Z = Z(\theta)$ . The evaluation of  $Z(\theta)$  is required. To circumvent the formidable difficulty therein, the joint probability is often approximated in practice.

### 4.3 Image Models

Most texture research can be characterized by the underlying assumption made about the texture formation process. There are two major assumptions, and the choice of the assumption depends primarily on the type of textures to be considered [14]. The first assumption, which is called the placement rule, considers a texture to be composed of primitives. Macro-textures have large primitives, whereas micro-textures are composed of small primitives. The texture is generated by some placement rules which specify how the primitives are oriented, both on the image

field and with respect to each other. The tissue samples are a typical example of this kind of textures. But the textures of sand, grass, and water are not appropriately described by the placement rule. The key feature of these texture images is that the primitives are random in shape and cannot be easily described. The second viewpoint regarding texture generation processes involves the stochastic assumption. The value at each pixel site is considered a random variable, and the texture is regarded as a sample from a probability distribution on the image space. This assumption of texture formation can be applied to the sand, water and grass textures. Fig. 4.3 shows the texture samples in these two categories.

The interest of this thesis is in texture models with stochastic process approach. Ideally, people would like to find a stochastic process that is physically meaningful and related to the texture which is being modeled. Texture models serve a dual role in that they can describe images and also can serve to generate synthetic images from the model parameters. In the stochastic process approach, the grey level  $x_s$  at a pixel site  $s$  is not independent of the grey levels at other sites in the image, and the principal concern is about the interaction and relationship between the pixels.

*MRF* image models belong to the stochastic category. The study of it has had a long history beginning with Ising's 1925 thesis on ferromagnetism [53]. Although the Ising model used in his thesis was not proved to be a realistic model for magnetic domains, it is approximately correct for phase-separated alloys, idealized grass, and some crystals [24]. In 1974, Besag allowed a natural extension of the model to the case of variables that have integer ranges, either bounded or unbounded [5]. A particular *MRF* model favours its own class of patterns by associating them with larger probabilities than other pattern classes. In the segmentation process, they are employed to model the *a priori* probability of contextual dependent pattern of the images. For examples: a colour Gauss Markov random field (*GMRF*) model

is used by Panjwani and Healey for modeling a colour image [40]. Barker and Rayner [3] used an isotropic *MRF* model and *GMRF* model for an unsupervised segmentation algorithm. The isotropic *MRF* model is used to model an image consisting of regions of constant but different grey scales, corrupted with an i.i.d. noise process, and *GMRF* model for individual textures. Chellappa and Chatterjee [10] used the *GMRF* model in a supervised segmentation algorithm. The *GMRF* model parameter and some derived features calculated using the model parameters are used as the texture features for classification. Cross and Jain [14] used an auto-binomial model to model and synthesize Brodatz textures. Derin and Cole [18] used a specific Gibbs distribution model considering up to 4-pixel clique size for texture segmentation. Other studies that made use of *MRF* and *GRF* models for texture or image modelling were conducted by Geman and Geman [27].

This section will focus on most common *MRF* models in image processing: auto-model and Gauss-model. And in fact, Gauss-model can also be regarded as a special case of auto-models.

### 4.3.1 Auto-Model and Related Texture Synthesis

Contextual constraints on two pixel sites, which are encoded in the Gibbs energy as pair-clique potentials, are the lowest order constraint for conveying contextual information in *MRF* texture models. Fig. 4.4 shows the pair-cliques involved in the first to third order models.

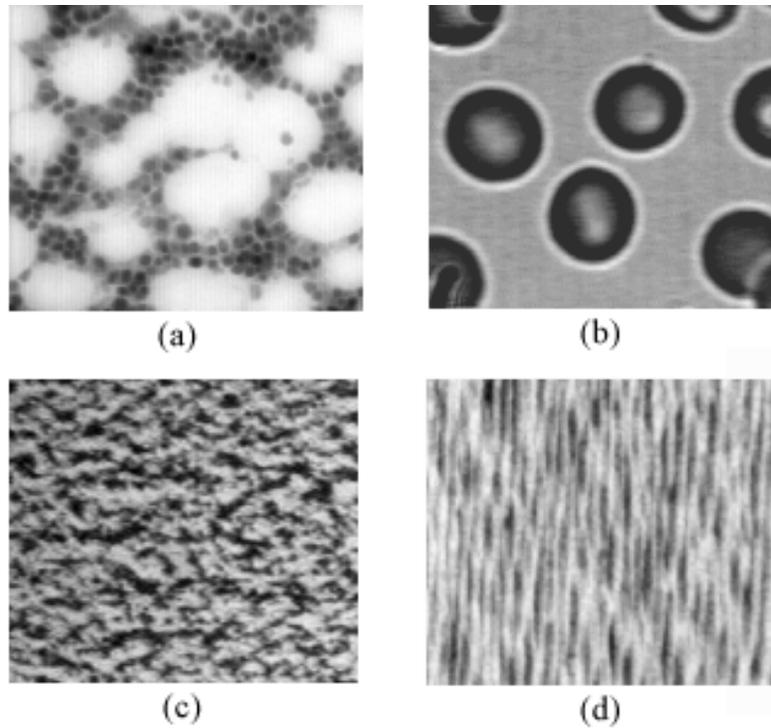


Figure 4.3: Texture samples: (a,b) Texture samples with circle primitives that can be synthesized by placement rule models (adapted from Matlab demo image database). (c,d) Texture samples that can be synthesized by stochastic processes (Brodatz textures).

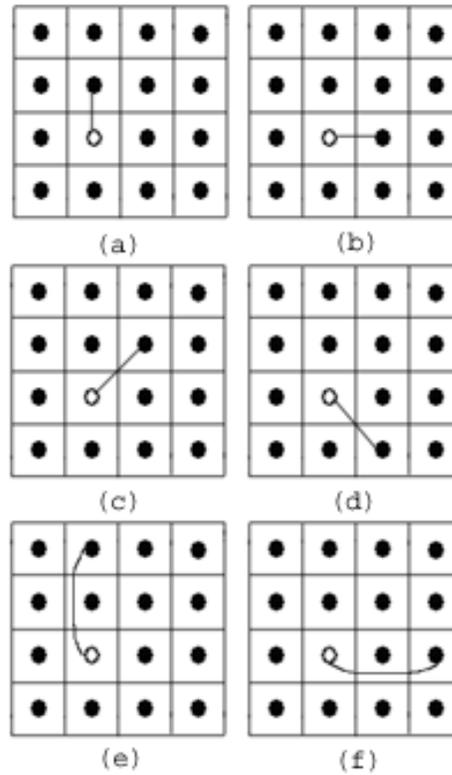


Figure 4.4: Pair-cliques involved up to the third order *MRF* texture model. (a-b) Pair-cliques involved in the first order model. (a-d) Pair-cliques involved in the second order model. (a-f) Pair-cliques involved in the third order model.

From the previous section, it can be observed that with clique potentials of up to two sites, the energy takes the form

$$U(x) = \sum_{s \in S} V_1(x_s) + \sum_{\{s, s'\} \in C_2} V_2(x_s, x_{s'}) \quad (4.9)$$

The above equation is widely used because of its simple form and low computational cost, and at the same time it still can convey contextual information [37]. Some specific *MRF* or *GRF* models can be specified by proper selection of  $V_1$  and  $V_2$ .

When  $V_1(x_s) = x_s G_s(x_s)$  and  $V_2(x_s, x_{s'}) = \beta_{s,s'} x_s x_{s'}$  where  $G_s(\cdot)$  is an arbitrary function and  $\beta_{s,s'}$  is a constant reflecting the pair-clique interaction between  $s$  and  $s'$ , the energy equation has the form

$$U(x) = \sum_{\{i\} \in C_1} x_s G_s(x_s) + \sum_{\{s,s'\} \in C_2} \beta_{s,s'} x_s x_{s'} \quad (4.10)$$

The above energy expression is called auto-models by Besag [5]. It can be further classified according the assumptions made about individual  $x_s$ . An auto-model is said to be an auto-logistic model if the  $x_s$  takes on values in the discrete label set  $\{0, 1\}$  or  $\{-1, 1\}$ . The corresponding energy is of the following form

$$U(x) = \sum_{\{s\} \in C_1} \alpha_s x_s + \sum_{\{s,s'\} \in C_2} \beta_{s,s'} x_s x_{s'} \quad (4.11)$$

where  $\beta_{s,s'}$  can be viewed as the interaction coefficients. The conditional probability for the auto-logistic model with  $\{0, 1\}$  is

$$P(x_s | x_{N_x}) = \frac{e^{\alpha_s x_s + \sum_{s' \in N_s} \beta_{s,s'} x_s x_{s'}}}{\sum_{x_s \in \{0,1\}} e^{\alpha_s x_s + \sum_{s' \in N_s} \beta_{s,s'} x_s x_{s'}}} = \frac{e^{\alpha_s x_s + \sum_{s' \in N_s} \beta_{s,s'} x_s x_{s'}}}{1 + e^{\alpha_s + \sum_{s' \in N_s} \beta_{s,s'} x_{s'}}} \quad (4.12)$$

When  $N_s$  represents the nearest neighbourhood on a lattice system (four nearest neighbours on a 2D lattice system), the auto-logistic model is reduced to the Ising model. An auto-model is said to be an auto-binomial model if  $x_s$  is taken on values in  $\{0, 1, \dots, R-1\}$  and every  $x_s$  has a conditionally binomial distribution of  $M$  trials and probability of success  $q$ . An auto-model is said to be an auto-normal model, also called Gauss-model, if the label set  $L$ 's joint distribution is multivariate normal. This model will be described in the next section.

One of the primary goals of *MRF* models is the description of real textures. The parameters of *MRF* models completely describe the texture structure. Given the model parameters, one can synthesize textures. This method is widely used

in stochastic image compression, random texture synthesis and lattice-physics simulations [23]. To illustrate the ability of describing the texture structure of *MRF* texture models, several *MRF* textures will be generated using auto-logistical models in this section and Gauss-models in the next section.

To generate auto-model *MRF* textures that are a visual representation of a *MRF*, a procedure that yields a sample from a *MRF* with given parameters is required, i.e., generate texture samples  $x^i$  statistically sampled from the random field  $X$  using the distribution law  $p(\cdot)$  of Eqn. (4.4). Gibbs sampler is a widely used technique for such a synthesis problem. The general steps involved in the Gibbs sampler for the discrete binary valued case are [23]:

**Step 1** Start with a sample from an i.i.d binary random field with  $p(x_s) = 0.5$ .

**Step 2** Visit each site in the image. At each site, sample  $x_s$  from  $p(x_s|x_r, r \in N_s)$

**Step 3** Repeat Step 2 many times, until the stability is reached.

For the auto-logistic model case, the conditional probability given by Eqn. (4.12) can also be written as [14]:

$$P(x_s|x_{N_s}) = \frac{\exp(x_s T)}{1 + \exp(T)} \quad (4.13)$$

where for the first order model T is:

$$T = \alpha + \beta(1, 1)(x_{s+r1} + x_{s-r1}) + \beta(1, 2)(x_{s+r2} + x_{s-r2})$$

and  $\tilde{N}_s = \{r1, r2\} = \{(0, -1), (-1, 0)\}$ . For the second-order model T is:

$$T = \alpha + \beta(1, 1)(x_{s+r1} + x_{s-r1}) + \beta(1, 2)(x_{s+r2} + x_{s-r2}) + \\ \beta(2, 1)(x_{s+r3} + x_{s-r3}) + \beta(2, 2)(x_{s+r4} + x_{s-r4})$$

and  $\tilde{N}_s = r1, r2, r3, r4 = (0, -1), (-1, 0), (-1, -1), (1, -1)$ .

The distribution of neighbourhood set  $\tilde{N}_s$  around pixel  $s$  up to fourth order is shown in Fig. 4.5.

		r8	r6	r10	
r7	r3	r2	r4	r9	
r5	r1	s	-r1	-r5	
-r9	-r4	-r2	-r3	-r7	
		-r10	-r6	-r8	

Figure 4.5: Auto-model neighbourhood system.

Some *MRF* texture examples of size  $128 \times 128$  generated from auto-logistic models are shown in Fig. 4.6. The model parameters are given in Table 4.1 (the model parameters are adapted from [14]). These simple texture samples are representative results but not direct attempts to imitate real textures. However, by artificially setting the model parameters, an intuitive understanding of the functionality of the model parameters in controlling the underlying structure of the texture can be obtained.

Textures I-a, I-b and I-c are generated using first order models (Ising model). Because the parameters along vertical and horizontal direction have the same values, these textures demonstrate an isotropic structure. All the parameters of texture I-a are set to zero, i.e.  $\alpha = \beta(1, \cdot) = 0$ , which makes the intensity value of each pixel independent of all the other pixels and the probability of a pixel being black or white is fifty percent. The texture generated using this models is a typical noise texture. As the values of  $\beta(1, \cdot)$  change from 0 in texture I-a to 1.52 in

texture I-c, the texture clustering increases clearly. Many of the applications of the Ising model involve studying the checkerboard-like patterns obtained with negative clustering parameters [24]. This is illustrated by texture I-d, where the most likely configuration is a black pixel surrounding by four white pixels or vice versa.

Figure 4.6: Synthetic textures generated using auto-models in Table 4.1. (I) Ising textures. (II) Anisotropy textures. (III) Attraction-repulsion textures.

Textures II-a, II-b and II-c show extreme anisotropy of first order and second order models. The value  $\beta(1, 1)$  controls the horizontal clustering and  $\beta(1, 2)$  con-

trols vertical clustering. In texture II-a, a large positive value of  $\beta(1, 2)$  and a large negative value of  $\beta(1, 1)$  result in clean vertical lines. On the contrary, texture II-b has short thick and noisy horizontal lines because of the small positive value of the vertical clustering parameters. The clustering along the NW-SE direction of texture II-c is pronounced since the parameter in this direction is 1.9 and the parameters in all the other directions are relatively smaller.

An attraction-repulsion process involves having low-order parameters positive (resulting in clustering), but high-order parameters negative (to inhibit the growth of clusters) [14]. If high-order parameters are also positive, large clusters would result, whereas negative high-order parameters yield small clusters. Texture III-a and III-b shows the effect of anisotropic clustering with inhibition. Texture III-b contains longer horizontal and vertical lines than texture III-a because of the large values of the first order clustering parameters  $\beta(1, 1)$  and  $\beta(1, 2)$ . Texture III-c and III-d shows two isotropic attraction-repulsion textures. Cluster sizes are small here because of the high order inhibition.

The synthetic texture samples shown in Fig. 4.6 demonstrate that the *MRF* parameters control the strength and direction of the clustering of the image. The model parameters themselves are sufficient to generate images. As Besag explains in the discussion of his lattice model paper [5], the pattern formation process, although specified locally, implies a global pattern. The consistency conditions enforced by the *MRF* model cause a pattern over the entire lattice. The patterns generated by varying the model parameters can be studied and analyzed. Directionality, coarseness, grey level distribution, and sharpness can all be controlled by choosing of the model parameters.

Model Number	Model parameters						
	$\alpha$	$\beta(1, 1)$	$\beta(1, 2)$				
I-a	0	0	0				
I-b	-2.2	1.1	1.1				
I-c	-3.04	1.52	1.52				
I-d	5.09	-2.25	-2.16				
	$\alpha$	$\beta(1, 1)$	$\beta(1, 2)$	$\beta(2, 1)$	$\beta(2, 2)$		
II-a	-0.26	-1	2.1	0.13	0.015		
II-b	-2.04	1.93	0.16	0.07	-0.075		
II-c	-1.9	-0.1	0.1	1.9	0.02		
	$\alpha$	$\beta(1, 1)$	$\beta(1, 2)$	$\beta(2, 1)$	$\beta(2, 2)$	$\beta(3, \cdot)$	$\beta(4, \cdot)$
III-a	2.19	-0.088	-0.009	-1	-1		
III-b	0.16	2.06	2.05	-2.03	-2.10		
III-c	-0.97	0.94	0.94	0.94	0.94	-0.42	-0.49
III-d	-4.6	2.62	2.62	2.17	2.17	-0.87	-0.85

Table 4.1: Auto-model parameters corresponding to the textures in Fig. 4.6

### 4.3.2 Gauss-model

One of the important characteristics of image data is the special nature of the statistical dependence of the grey level at a lattice point on those of its neighbours. Gauss-model provides a way of modelling this relationship by representing  $x_s$  as a linear weighted combination of the neighbourhood pixels and additive noise. The random field modeled by a Gauss-model is called the Gauss Markov random field (*GMRF*). Comparing with auto-logistical and auto-binomial models, the Gauss-model introduces much more diverse texture types and has been successfully used to simulate, segment, and classify different image textures with a limited spatial range of pixel interactions.

Let  $X$  be an observation from an image defined on a two dimensional  $M \times M$  lattice system  $S$  where  $S = \{(i, j); 0 \leq i, j \leq M - 1\}$ . The two dimensional *GMRF* models characterize the statistical dependency among pixels by satisfying Eqn. 4.2 and 4.3. Since *GMRF* models are defined only for symmetric neighbourhood sites, often  $N$  is equivalently characterized using an asymmetrical neighbourhood set  $\tilde{N}$ , i.e. if  $r \in \tilde{N}$  then  $-r \notin \tilde{N}$  and  $N = (r : r \in \tilde{N}) \cup (-r : r \in \tilde{N})$ . A hierarchy of *GMRF* models can be defined. For example, when  $\tilde{N} = \{r1, r2\} = \{(0, -1), (-1, 0)\}$ , a first order *GMRF* model is obtained, and when  $\tilde{N} = \{r1, r2, r3, r4\} = \{(0, -1), (-1, 0), (-1, -1), (-1, -1)\}$  a second order model is obtained and so on.

Given a finite image, it can be analyzed as a finite slice of an underlying infinite lattice image. This approach leads to the class of *GMRF* models known as infinite lattice *GMRF* models. In general the infinite lattice models do not give computationally attractive algorithms for image processing. Another class of models, known as finite lattice models, obtained by assuming special boundary conditions, yields computationally efficient processing algorithms using fast transforms like discrete Fourier, discrete sine and cosine [9]. Because of this, the finite models will be used in this thesis, and the boundary conditions are considered as follows: the left and right edges as well as the top and bottom edges are considered adjacent. In other words, the field is toroidal. Fig. 4.7 shows a four by four small image under this boundary conditions. For example, in the first order model case,  $x_{(1,1)}$  has four neighbourhood pixels:  $x_{(1,2)}$ ,  $x_{(2,1)}$ ,  $x_{(4,1)}$ , and  $x_{(1,4)}$ .

The finite lattice *GMRF* model under this boundary assumption can be represented as follows [9]:

$$x_s = \sum_{r \in N_s} \theta_r x_{s+r} + e_s \quad (4.14)$$

$x$  is a real number and  $e_s$  is a zero mean, Gaussian noise, with autocorrelation given

Figure 4.7: A image (size:  $4 \times 4$ ) with a toroidal boundary condition

by

$$E[e_s e_{s+r}] = \begin{cases} \sigma^2 & r = (0, 0) \\ -\theta_r \sigma^2 & r \in N_s \\ 0 & \text{otherwise} \end{cases} \quad (4.15)$$

Let the vector  $\underline{\theta}$  contain the parameter  $\theta_r, r \in N_s$  in the above equation. The  $\underline{\theta}$  for the fifth order *GMRF* model is shown in Fig. 4.8, where  $\theta_{(0,-1)} = \theta_{(0,1)}$  and  $\theta_{(-1,0)} = \theta_{(1,0)}$  etc.  $[\underline{\theta}, \sigma^2]$  are the parameters of *GMRF*.

By defining

$$\underline{\mathbf{x}} = [x_{(0,0)}, x_{(0,1)}, \dots, x_{(0,M-1)}, \dots, x_{(M-1,M-1)}]^T$$

and

$$\underline{\mathbf{e}} = [e_{(0,0)}, e_{(0,1)}, \dots, e_{(0,M-1)}, \dots, e_{(M-1,M-1)}]^T,$$

Eqn. 4.14 can be written as:

$$B(\underline{\theta})\underline{\mathbf{x}} = \underline{\mathbf{e}} \quad (4.16)$$

Figure 4.8: The parameters of the fifth order *GMRF* model.

where  $B(\underline{\theta})$  is a block-circulate symmetric matrix:

$$B(\underline{\theta}) = \begin{bmatrix} B_{0,0} & B_{0,1} \dots & B_{0,M-1} \\ B_{0,M-1} & B_{0,0} \dots & B_{0,M-2} \\ \dots & & \\ B_{0,1} & \dots & B_{0,0} \end{bmatrix} \quad (4.17)$$

where each  $B_{0,i}$  is an  $M \times M$  matrix and  $B_{0,i} = B_{0,M-i}$ .

For example, when using the simplest first order model,  $\underline{\theta} = \{\theta_{(0,-1)}, \theta_{(0,1)}, \theta_{(-1,0)}, \theta_{(1,0)}\}$ , let  $\theta_1 = \theta_{(0,-1)} = \theta_{(0,1)}$  and  $\theta_2 = \theta_{(-1,0)} = \theta_{(1,0)}$ . Applying Eqn. 4.14 to each pixel



where  $\mu$  is an  $M \times 1$  vector of the conditional means. Substituting  $B(\theta)$  into Eqn. 4.19, it can be found that the single-site and pair-site clique potential functions for the Gauss-model are:

$$V_1(x_s) = (x_s - \mu_s)^2 / 2\sigma^2 \quad (4.20)$$

and

$$V_2(x_s, x_{s'}) = \theta_r (x_s - \mu_s)(x_{s+r} - \mu_{s+r}) / 2\sigma^2 \quad (4.21)$$

The conditional distribution at pixel  $s$  given the neighbourhood set  $N_s$  is:

$$P(x_s | x_r, r \in N_s) = \frac{e^{-U(x_s | x_r, r \in N_s)}}{Z} \quad (4.22)$$

where  $Z$  is the partition function of the conditional Gibbs distribution. Based on Eqn. 4.20 and 4.21

$$U(x_s | x_r, r \in N_s) = \frac{1}{2\sigma^2} ((x_s - \mu_s)^2 - 2\sum_{r \in N_s} \theta_r (x_s - \mu_s)(x_{s+r} - \mu_{s+r})) \quad (4.23)$$

### 4.3.3 Gauss-Model Texture Synthesis

The synthesis procedure of *GMRF* textures begins with the representation given by Eqn. 4.16. Assuming that  $\theta$  takes values so that  $B^{-1}(\theta)$  exists, the image vector  $\underline{x}$  can be written as:

$$\underline{x} = B^{-1}(\theta)e \quad (4.24)$$

It is a heavy computational burden to implement Eqn. 4.24 by directly inverting an  $M^2 \times M^2$  matrix. But under the assumption of a toroidal boundary of image  $x$ ,  $B^{-1}(\theta)$  will also be a block-circulate matrix. Let  $\Lambda$  be the covariance matrix of image  $x$ , then the *GMRF* texture sample  $x$  can be computed as [23]:

$$x = \text{ifft2} \{ \text{sqrt}(\text{fft2}(\Lambda)) \cdot \text{fft2}(e) \} \quad (4.25)$$

where  $fft2$  and  $ifft2$  denote two-dimensional discrete Fourier and inverse Fourier transform.  $sqrt()$  is the square root operator and “.” denotes element-by-element operations, and  $e$  is an i.i.d. Gaussian noise image sample.

Fig. 4.9 shows sixteen  $64 \times 64$  *GMRF* texture samples synthesized using the models given in Table 4.2 (the parameters of the models are adapted from [9]). Inspecting the samples, one can see that they are quite varied in textural patterns. First, the orientations of the texture are controlled by the setting of parameters in the model, just like the auto-model synthesized textures in Fig. 4.6. By setting the vertical and horizontal parameters to zero, textures (b), (c), and (g) show strong diagonal directions. Texture (o) is obtained from texture (n) by considering an interaction of a pair of WN-ES direction neighbourhood pixels. The clear horizontal oriented line structure in texture (n) is diffused in texture (o). And by adding the EN-WS pair of pixels, a more macro-structure pattern in texture (e) than texture (o) is generated. The stretch of the line structure in the horizontal direction almost disappears. Texture (a) and texture (p) each have a vertical stretching tendency because the parameter along the vertical direction is bigger than the parameter along the horizontal direction. But the negative value of texture (p) makes the scale of the texture along the horizontal more narrow and sharp than texture (a). Visually, texture (a) appears like a blurred version of texture (p). Texture (a) and texture (l) have the same set of parameters. The different is the model orders. Texture (a) has a first order model, and texture (l) has a second order model. The interactions between neighbourhood pixels in texture (a) happens within a smaller neighbourhood than texture (l). This makes the change of grey levels in texture (a) look more continuous than texture (l).

Model Number	Model parameters															
	$\theta_{(1,0)}$	$\theta_{(0,1)}$	$\theta_{(1,-1)}$	$\theta_{(1,1)}$	$\theta_{(2,0)}$	$\theta_{(0,2)}$	$\theta_{(2,-2)}$	$\theta_{(2,2)}$	$\theta_{(3,0)}$	$\theta_{(0,3)}$	$\theta_{(4,0)}$	$\theta_{(0,4)}$				
(a)	.2794	.1825														
(b)			.28	.14												
(c)			.14	.28												
(d)	.3357	.3246	-.25	-.2126												
(e)	-.2061	-.2061	.0536	.0536	-.0123	-.0580										
(f)			-.2341	.4682	.0655	.0655	-.0655	-.0655								
(g)			.28	-.22												
(h)	.12	-.10			.08	-.09			-.11	.11	-.07	.09				
(i)	.16	.10							.12	-.14						
(j)	.10	.16							-.14	.12						
(k)	.12	-.10	-.09	.08					-.11	.11						
(l)					.1825	.2794										
(m)	.12	-.24			.16	-.18										
(n)	-.24	.12			-.18	.16										
(o)	-.24	.12		.11	-.18	.16										
(p)	-.12	.18														

Table 4.2: GMRF model parameters corresponding to the textures in Fig. 4.9

Figure 4.9: Synthetic textures generated using the *GMRF* models in Table 4.2

## 4.4 Model-based Texture Segmentation Algorithm Design

The approach of model-based image segmentation is to articulate the regularities between neighbouring pixels mathematically, and then to exploit them in a Bayesian framework to make inference that which label is most likely to happen. During the segmentation process, the prior distribution models are needed to capture the tendencies and constraints that characterize the scenes of interest. In general, the following problems should be addressed in the segmentation approach:

1. Design of prior models;
2. Statistical inference to specify the model parameters;
3. Specification of the posterior distribution given the observed image;
4. Estimation of the true image based on the posterior distribution coupled with a Bayesian decision rule.

The rest of this section will present in detail a segmentation algorithm motivated by the above considerations.

### 4.4.1 Prior Image Models

The importance of prior information in image analysis is illustrated by Winkler [54] through an interesting experiment on image restoration. Based on different prior models, the recovered images from the same observed image looked totally different. Image (b) in Fig. 4.10 is restored based on smooth and isotropic Ising

Figure 4.10: (a) A degraded image. (b) Reconstructed image based on smooth and isotropic Ising model. (c) Reconstructed image based on vertical-stripe auto-model. model assumption, whereas image (c) is restored based on vertical stripes auto-model.

To define the prior image models for the segmentation algorithm, the meaning of the image  $X$  mentioned afterwards should be explained first. As in the previous section, let  $X$  be a two dimensional  $M \times M$  image defined on the lattice system  $S$ . Associating with each pixel  $s$  is a random variable  $X_s$  which contain all of the relevant information for the image processing task. In segmentation,  $X$  is the pixel intensity array and a corresponding texture label array, i.e.,  $X = \{X^I, X^L\}$ . Each label gives the texture type of the associated pixel. Fig. 4.11 illustrates the

Figure 4.11: Image components defined for model-based segmentation algorithm. (a) Intensity image (observable). (b) Label image (unobservable).

intensity component  $X^I$  and label component  $X^L$  of image  $X$ .

Designing the prior models about the image  $X$  means selecting the appropriate *MRF* models which can describe the underlying statistical structure of the intensity image and the label image. The selected models should correctly describe the relations and constraints among the pixels in image. Because the kind of knowledge that the prior distribution can represent is articulated in terms of small local collections of variables, this leads to a distribution on  $X$  with a more or less “local neighbourhood structure” which can just be modelled using *MRF*.

Viewing the literature, several *MRF* models are presented for various image processing problems. Krishmanachari and Chellappa have demonstrated the applicability of using *GMRF* models to model the intensity image [35]. The conditional distribution of the intensity image  $X^I$  at site  $s$  which carries a class label  $l_s$  can be written as[39]:

$$P(X_s^I = x_s | X_r^I = x_r, r \in N_s, X_s^L = l_s) = \frac{e^{-U(X_s^I = x_s | X_r^I = x_r, r \in N_s, X_s^L = l_s)}}{Z} \quad (4.26)$$

where  $U(\cdot)$  is the energy function which takes the form:

$$\begin{aligned} U(X_s^I = x_s | X_r^I = x_r, r \in N_s, X_s^L = l_s) \\ = \frac{1}{2\sigma_{l_s}^2} ((x_s - \mu_s)^2 - 2\sum_{r \in N_s} \theta_r^{l_s} (x_s - \mu_s)(x_{s+r} - \mu_{s+r})) \end{aligned} \quad (4.27)$$

Where  $\theta_r^{l_s}$  represents the *GMRF* model parameter of class  $l_s$ . When the intensity image is zero mean, Eqn. 4.27 can take a simpler form:

$$U(X_s^I = x_s | X_r^I = x_r, r \in N_s, X_s^L = l_s) = \frac{1}{2\sigma_{l_s}^2} (x_s^2 - 2\sum_{r \in N_s} \theta_r^{l_s} x_s x_{s+r}) \quad (4.28)$$

Usually, the label image does not have as many intensity levels as the intensity image. Its intensity levels are the number of texture types in the image. Therefore, the prior model for label image is not as complicated as that for intensity image. The main consideration for the prior model of label image is the smoothness constraint. In fact, a generic contextual constraint in our daily life is the smoothness. It assumes that physical properties in a neighbourhood of space or in an interval of time present some coherence and generally do not change abruptly. This phenomenon also happens in an image. The local conditional probability of the label image can be modelled by a so called pairwise interaction model [39]:

$$P(X_s^L = l_s | X_r^L = l_r, r \in \psi_s) = \frac{\exp\{-U(l_s | l_r)\}}{\sum_{l'_s=(1,2,\dots,L)} \exp\{U(l'_s)\}} \quad (4.29)$$

where  $\psi_s$  is the neighbourhood set of a pixel  $s$  in the label image, and each pixel carries a class label  $l_s, l_s \in \{1, 2, \dots, L\}$ . The energy function  $U(l_s)$  calculates the number of neighbours in  $\psi_s$  that belong to same class as  $l_s$ :

$$U(X_s^L = l_s | X_r^L = l_r, r \in \psi_s) = \beta \sum_{r \in \psi_s} \delta(l_s - l_r) \quad (4.30)$$

where  $\delta(\cdot)$  is the Kronecker delta.

According to Eqn. 4.29, the likely pictures for the label image are therefore the ones that respect the prior expectations: they segment into regions of constant intensities. Whenever all labels in the neighbourhood  $\psi_s$  take the same value as  $l_s$ , they incur a high positive potential, which results in a higher probability; otherwise, if they are not all the same, they incur a small potential, which results in a lower probability. In other words, Eqn. 4.29 and 4.30 provide a method for constructing smooth label images by giving a higher probability to smoothness distribution.

#### 4.4.2 Model Size and Parameter Estimation

To estimate the *GMRF* model, two major problems have to be tackled: the choice of appropriate neighbourhood size (i.e., the order of the *GMRF* model) and the estimation of parameters in the models.

Although using *GMRF* models to describe the structure of the texture is theoretically appropriate, accurate estimation of the model parameters using the observed data is still an open issue. In the case of this thesis, the parameters for the two image models mentioned in the previous section should be estimated. The least squares estimate is selected in this thesis, which can provide appropriate estimates for segmentation purposes and computational efficiency [38].

Given a neighbourhood set  $N_s$  for each pixel site  $s$  in  $S$ , the finite lattice system  $S$  can be partitioned into mutually exclusive and totally inclusive subsets  $S_I$ , the interior set and  $S_B$ , the boundary set, such that:  $S_B = \{s = (i, j) : s \in S, (s+r) \notin S \text{ for at least one } r \in N_s\}$ , and  $S_I = S - S_B$ . Then the least square estimation of the parameters are [34]:

$$\hat{\underline{\theta}} = \left[ \sum_{S_I} Q_s Q_s^T \right]^{-1} \left[ \sum_{S_I} Q_s x_s \right] \quad (4.31)$$

and

$$\hat{\sigma}^2 = \frac{1}{M^2} \sum_{S_I} [x_s - \hat{\theta}^T Q_s]^2 \quad (4.32)$$

Where

$$Q_s = [x_{s+r_1} + x_{s-r_1}, \dots, x_{s+r_n} + x_{s-r_n}]^T \quad (4.33)$$

$r_1, \dots, r_n$  is the asymmetrical neighbourhood set  $\tilde{N}_s$  defined in the beginning of section 4.3.2. The neighbourhood set of Gauss-model is the same as the auto-model neighbourhood set shown in Fig. 4.5.  $M^2$  is the number of pixels in  $S_I$ . The parameters up to fifth order model are shown in Fig. 4.8.

Model order is one of the important concerns for a successful segmentation. Usually, the more complicated the texture, the higher order model should be used. But it is difficult to explicitly adapt the interaction structure to a particular texture by choosing the most characteristic model size and also excluding probably less significant interactions from a model. Kashyap and Chellappa presented a method generated from Bayes procedure for estimation and choice of the model order for the *GMRF* texture [34] [9].

Let  $N^1, N^2, \dots, N^k, \dots$  be the first, second,  $\dots$ ,  $k^{th}$ ,  $\dots$  order neighbourhood. The decision rule for the choice of appropriate neighbourhood size  $k$  is: choose the neighbourhood  $N_k$  if

$$k = \arg \min_k \{C_k\} \quad (4.34)$$

where

$$C_k = -2 \sum_{s \in S} \ln(1 - \theta_k^T \phi_{s,k}) + N^2 \ln \sigma_k + m_k \ln N^2 \quad (4.35)$$

$$\theta_k = \text{col} [\theta_r, r \in \tilde{N}_s^k] \quad (4.36)$$

$$\phi_{s,k} = \text{col} \left[ \cos \frac{2\pi}{N} (s^T r), r \in \tilde{N}_s^k \right] \quad (4.37)$$

Figure 4.12: Synthesized textures corresponding to the models in Table 4.3. (a) Model A. (b) Model B. (c) Model C. (d) Model D. (e) Model E.

Model	Model order	Model parameters	$\sigma^2$	$C_k$
A	2	$\theta_{0,-1} = 0.2707$ $\theta_{-1,0} = 0.5001$ $\theta_{-1,-1} = -0.1471$ $\theta_{-1,1} = -0.1224$	1.000	
B	1	$\theta_{0,-1} = 0.0135$ $\theta_{-1,0} = 0.4985$	1.1571	944.0
C	2	$\theta_{0,-1} = 0.2786$ $\theta_{-1,0} = 0.5000$ $\theta_{-1,-1} = -0.1496$ $\theta_{-1,1} = -0.1283$	0.9788	474.2
D	3	$\theta_{0,-1} = 0.2790$ $\theta_{-1,0} = 0.5004$ $\theta_{-1,-1} = -0.1494$ $\theta_{-1,1} = -0.1287$ $\theta_{0,-2} = -0.0001$ $\theta_{-2,0} = -0.0004$	0.9797	562.8
E	4	$\theta_{0,-1} = 0.2787$ $\theta_{-1,0} = 0.5011$ $\theta_{-1,-1} = -0.1479$ $\theta_{-1,1} = -0.1328$ $\theta_{0,-2} = 0.0006$ $\theta_{-2,0} = -0.0012$ $\theta_{-1,2} = 0.0010$ $\theta_{-2,1} = 0.0035$ $\theta_{-1,-1} = -0.0008$ $\theta_{-1,-2} = -0.0016$	0.9795	603.5

Table 4.3: GMRF texture models and  $C_k$  values corresponding to the synthesized textures in Fig. 4.12.

where  $\tilde{N}_s^k$  is the asymmetrical half of  $N_s^k$ , and  $m_k$  is the number of parameters in neighbourhood  $\tilde{N}_s^k$ .

To illustrate the usefulness of the decision rule, texture (a) in Fig. 4.12 is synthesized using the GMRF model (A) shown in Table 4.3. Take texture (a) as the test original image, the estimated models of texture (a) using second order to fifth order model are given in (B) to (E) in Table 4.3. Each estimated model's statistics  $C_k$  is computed and shown in Table 4.3 too. Comparing the  $C_k$  values, one can see that the decision rule correctly picked up model (C), which has the same order as model (A). Textures (b) to (e) in Fig. 4.12 are synthesized based on the estimated models (B) to (E) in Table 4.3. It is clear that the textures corresponding to the inappropriate models (B), (D) and (E) are not quite similar to texture (a). Texture (c) which is synthesized using the same order model as texture (a) is very

similar to texture (a). In general, the textures corresponding to a specific model  $\xi$  and another model, which includes all the neighbours in model  $\xi$  and some extra neighbours may appear similar, making visual judgement subjective. However, the quantitative decision rule may correctly reject these over-parameterized models.

The data in Table 4.4 is calculated from 144 Radarsat sea ice image samples. For each sample,  $C_k$  was calculated using second to fifth order estimated *GMRF* models. The numbers associated with each order represent the number of times an model had the minimum  $C_k$ . Observing Table 4.4, fourth order model should be a preferred choice for Radarsat SAR sea ice texture. The fourth order model is used in the SAR sea ice image segmentation in this thesis.

Model Order	Times of Minimum $C_k$
2	19
3	6
4	74
5	45

Table 4.4: The minimum  $C_k$  value distribution of Radarsat SAR sea ice image with different *GMRF* texture model orders

There are many natural and artificial image textures that can be modeled adequately by *GMRF* models with multiple pair-clique interactions. But as an image model, the Gauss-model has some draw backs. First, it assumes the infinite continuous range of the signal values and, in the strict sense, does not represent the digital images that have a finite set of pixel values. Second, this model assumes that all the neighbourhood pixels interact with each other within a square neighbourhood system, and the model only considers the interaction between pairwise pixels. Such

simplification leaves aside quite a number of natural textures that usually have larger and more diverse texture structure and greater geometric differences.

For a pairwise interaction model, there is only one parameter  $\beta$  in Eqn. 4.30, which characterizes the binding between textures of the same class, to be estimated. However, it is difficult to estimate the exact value of  $\beta$ . Based on the segmentation experiments,  $\beta$  could be set between 1 and 3. The nature of the segmentation result also depends on the order of the pairwise interaction model. The choice of model order depends on the knowledge of intensity image, and it is quite flexible. For example, it is preferable to choose the first-order model if one knows *a priori* that the boundaries are either horizontal or vertical. It is also not necessary that the neighbourhood pixels are connected with each other. For example, in the segmentation approach for coastline detection [56], 8 nearest neighbourhood pixels and 8 neighbourhood pixels located at a certain distance from the site  $s$  are selected. In this thesis, without any specific assumption to the distribution of the texture types in the image, the second order label model neighbourhood is used.

### 4.4.3 MRF Segmentation in Bayesian Framework

Some image analysis can be implemented using ad hoc techniques, but for robust and powerful algorithm design, there is a need for theory. When analysis is based on precisely formulated mathematical models, one can study the performance of algorithms analytically and design optimal methods. The Bayesian approach is one of the promising attempts to provide a such base[54]. The Bayesian paradigm allows one to consistently combine the prior model with the data, and an optimization process can be designed to obtain a MAP solution.

The basic theory for model-based image segmentation using Markove random

field coupled with Bayesian framework in the literature is introduced in this section. The segmentation algorithm developed in this thesis based on the theory will be explained in detail in the next section.

An optimization-based segmentation algorithm involves three tasks: problem representation, objective function and optimization process. The representation deals with how to provide the solution. For an image labeling problem, one may use a chain of boundary locations to represent the solution [41], or one may alternatively use a region map to perform the same task [28, 55, 35]. The label image  $X^L$  defined in section 4.4.1 is, in fact, a region map representation. The second task is how to formulate the objective function for the optimization. The formulation determines how the constraints and prior knowledge about image are encoded into the function. The encoded information may be pixel intensity, context relations between pixels, or some statistical features of pixel values. The third task deals with optimizing the objective, i.e., how to search for the optimal solution in the solution space. The main concern in this stage is the tradeoff between the appropriate solution and computation efficiency. These three issues are related to one another. First of all, the scheme of representation influences the formulation of the objective function and the design of the search algorithm. The formulation of the objective function then affects the search.

In the segmentation algorithm presented in this thesis, based on the employed image models, the Bayesian theory will involve in the formulation setup, and the objective function in the form of an energy function will be minimized to obtain an optimized solution.

Under the definition for the image  $X = \{X_I, X_L\}$ , the observation on the image is not complete: one observes the intensity image, but not the label image. The segmentation purpose is then to estimate the label image based on the observed image

and the prior information about the observed and label image. The segmentation can be formulated as an optimization process involving maximizing a posteriori:

$$(X^L)^* = \underset{X^L}{\operatorname{arg\,max}} [P(X^L|X^I) = \frac{P(X^I|X^L)P(X^L)}{P(X^I)}] \quad (4.38)$$

This approach can be called a *MAP* estimate: given the observed intensity image  $X^I$ , choose the most likely label image  $(X^L)^*$  which maximizes the posteriori distribution of Eqn.4.38. The posterior distribution contains the information relevant to the image segmentation process. It reveals the likely and unlikely states of the label image. The denominator  $P(X^I)$  is difficult to evaluate. But because  $X^I$  is the observed intensity image,  $P(X^I)$  is then a constant which can be ignored. The analysis of the posterior distribution will require only ratios, not the absolute probabilities.

Searching all the possible configurations of label image  $\{X^L\}$  for a global optimum is computationally infeasible. For example, even for the very simple case of segmentation of a  $128 \times 128$  image into two classes, there are  $2^{2^{14}}$  possible label configurations. To overcome the computational difficulty, the following paragraph will show how the optimal solution can be obtained by using a local optimization approach: the iterated conditional mode method (*ICM*), which was first suggested by Besag [4].

The intensity image is considered to be composed of a set of  $k \times k$  sub-images  $E_s$  centered at each pixel  $s$ .  $E_s$  can be called the energy window. As shown in Fig. 4.13, pixel  $s$  in the intensity image is centered by a  $9 \times 9$  energy window  $E_s$ . In each of these windows the texture labels are considered to be homogeneous, i.e., all the pixels in this window carry the same texture label. Let  $X_{s*}^I$  denote the intensity array in the energy window  $E_s$ . Using the Gibbs formulation, the joint probability in the window  $E_s$  can be written as [39]:

$$P(X_{s*}^I | X_s^L = l^s) = \frac{\exp(-U_1(X_{s*}^I | X_s^L = l^s))}{Z_1(l_s)} \quad (4.39)$$

where  $Z_1(l_s)$  is the partition function of texture  $l_s$  and according to Eqn. 4.20 and Eqn. 4.21, the energy in  $E_s$  with a free boundary is:

$$U_1(X_{s*}^I | X_s^L = l^s) = \frac{1}{2(\sigma^{l_s})^2} \sum_{s' \in E_s} \{x_{s'}^2 - \sum_{r \in \hat{N}_{s'} | r \pm s' \in E_s} \theta_r^{l_s} x_{s'}(x_{s'+r} + x_{s'-r})\} \quad (4.40)$$

For convenience, re-write Eqn. 4.29 as follows, it is the distribution function for the texture label at site  $s$  conditioned on the labels of the neighbouring sites  $\psi_s$  shown in Fig. 4.13:

$$P(L_s | L_r, r \in \psi_s) = \frac{\exp(-U_2(L_s | L_r, r \in \psi_s))}{Z_2} \quad (4.41)$$

where  $Z_2$  is partition function of label image. According to Bayesian theory, the following equation can be written:

$$P(X_s^L = l_s | X_{s*}^I, X_r^L = l_r, r \in \psi_s) = \frac{P(X_s^I | l_s) P(l_s | l_r, r \in \psi_s)}{P(X_{s*}^I)} \quad (4.42)$$

Since  $X_{s*}$  is already observed,  $P(X_{s*}^I)$  in Eqn. 4.42 is a constant, the numerator is a production of two exponential functions and Eqn 4.42 becomes:

$$\begin{aligned} & P(X_s^L = l_s | X_{s*}^I, X_r^L = l_r, r \in \psi_s) \\ &= \frac{1}{Z_p} \exp(-U_P(X_s^L = l_s | X_{s*}^I, X_r^L = l_r, r \in \psi_s)) \end{aligned} \quad (4.43)$$

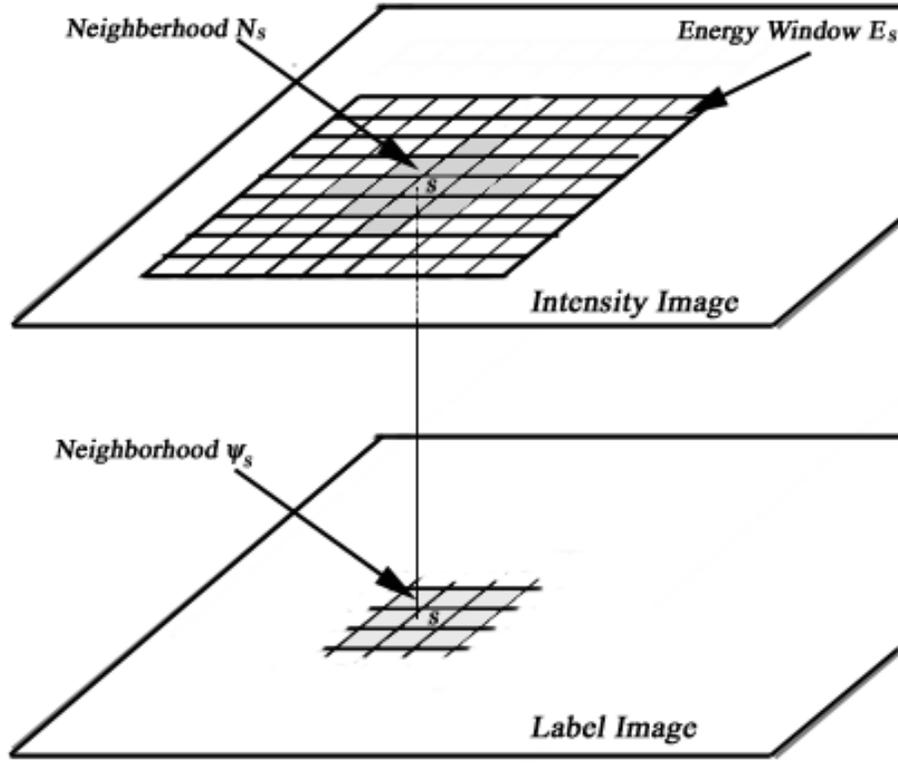


Figure 4.13: Neighbourhood systems of intensity and lable images.

From Eqn. 4.39 and 4.41 , the energy  $U_P(\cdot)$  in Eqn. 4.43 can be written as:

$$\begin{aligned} U_P(X_s^L = l_s | X_{s^*}^I, X_r^L = l_r, r \in \psi_s) \\ = \log(Z_1(l_s)) + U_1(X_{s^*}^I | l_s) + U_2(l_s | l_r, r \in \psi_s) \end{aligned} \quad (4.44)$$

Compared with Eqn. 4.38, Eqn. 4.43 and 4.44 are the local expression of probability and energy. Avoiding the computational burden of solving Eqn. 4.38 for a global  $X^L$ , Eqn. 4.43 can be used to find a local optimum  $X_s^L$  [35] [39] [4]. In this case, the solution is an approximation to the MAP estimation. The

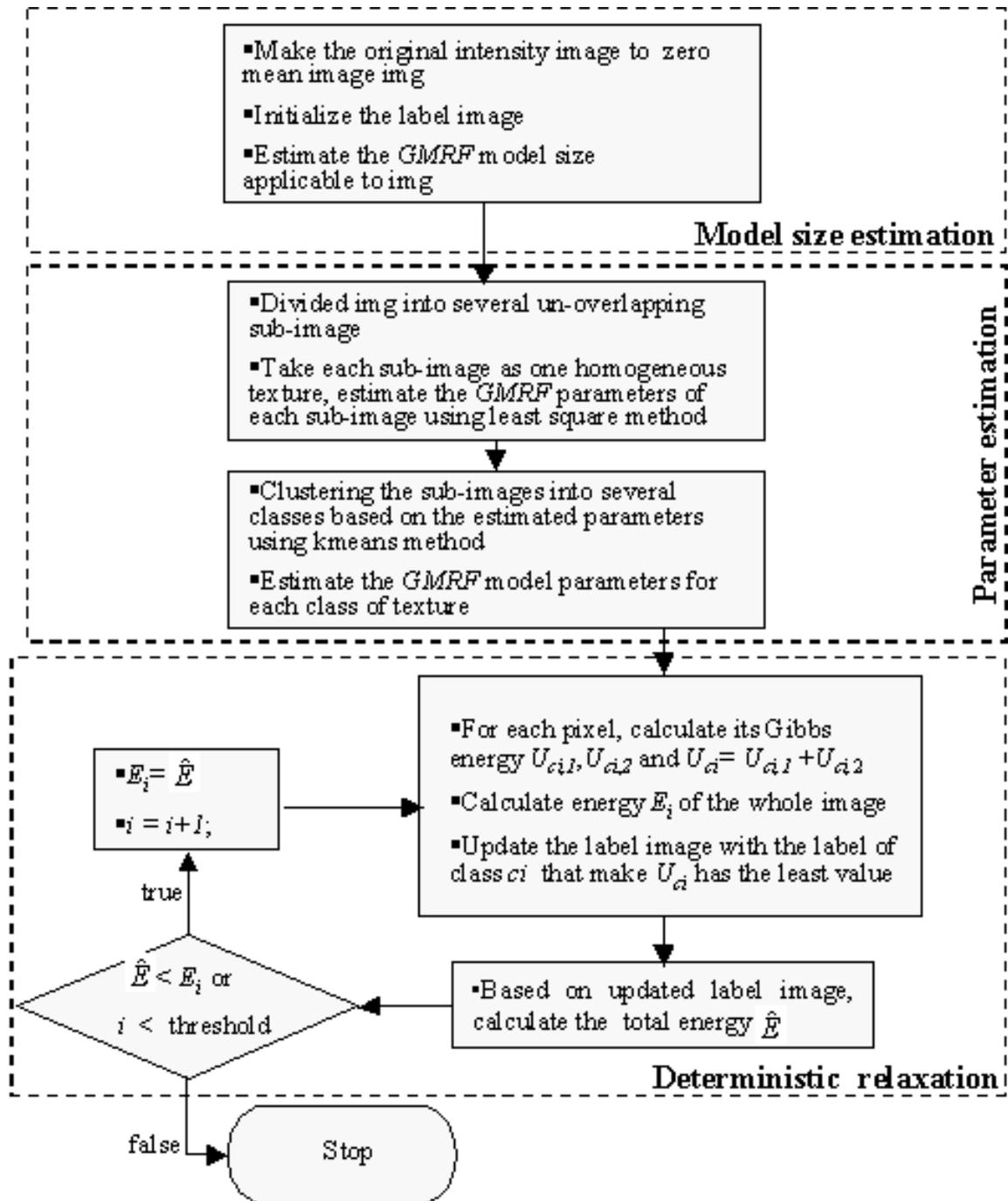
term  $\log(Z_1(l_s))$  is the summation of energy of each intensity array configuration in the energy window  $E_s$ , and it is dependent on the texture class. It can be evaluated explicitly given the model parameters of each class, but the computation is very cumbersome. When the  $\sigma^2$  of each texture model do not have a significant difference, the energy summation of all possible configurations of each class do not have a significant difference as well. In this case, this term can be ignored.

#### 4.4.4 MRF Segmentation Algorithm

Based on the theory described in the above section, the flow chart of the *MRF* segmentation algorithm used in this thesis is illustrated in Fig. 4.14. This algorithm will be used in Chapter 5 to generate all the *MRF* segmentation results.

Three steps are involved in the algorithm: model size estimation, model parameter estimation and the deterministic relaxation. Before explaining the segmentation algorithm, one point should be made clear. As shown in model size estimation part of Fig. 4.14, all the references to intensity image in the following description correspond to the zero mean image. A zero mean image can be obtained as follows: for each pixel  $(i, j)$  in the intensity image, the local mean of pixel  $(i, j)$  is obtained by computing the mean of a square window around pixel  $(i, j)$  in the original intensity image. Then the intensity value of pixel  $(i, j)$  of the zero mean image is obtained by subtracting the local mean from the original intensity value of pixel  $(i, j)$ . The purpose of making image zero mean is to utilize Eqn. 4.28 which is obtained under the zero mean assumption.

The method introduced in section 4.4.2 is used for model size estimation. The steps involved in parameter estimation part can be illustrated using Fig. 4.15. First, the original image is divided into several un-overlapping sub-images, in this

Figure 4.14: Model-based  $MRF$  texture image segmentation algorithm flowchart.

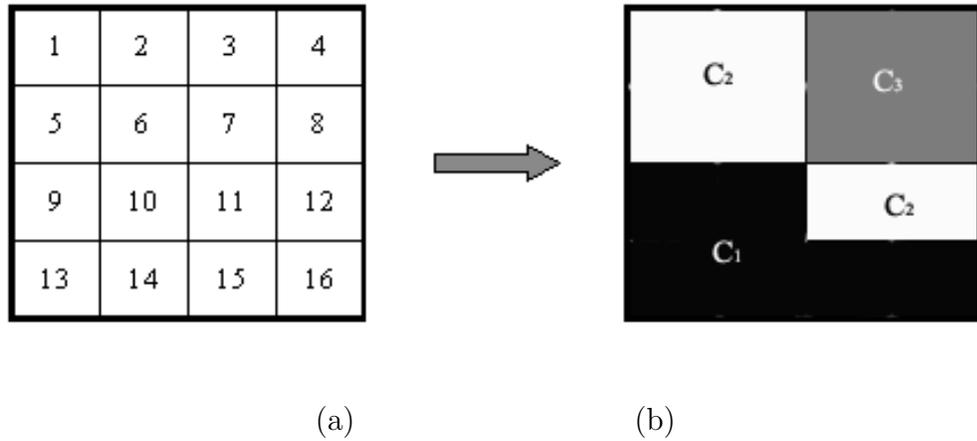


Figure 4.15: Parameter estimation part in Fig. 4.14. (a) Original image. (b) Coarse segmentation.

example, the original image is divided into 16 sub-images. For each sub-image, its *GMRF* model is estimated. so 16 models are estimated from the 16 sub-images. The kmeans method is then used to group these 16 models into 3 classes as shown in figure b. Finally, the *GMRF* models  $\underline{\theta}^{C_1}$ ,  $\underline{\theta}^{C_2}$  and  $\underline{\theta}^{C_3}$  for the three classes  $C_1$ ,  $C_2$  and  $C_3$  are estimated.

The first step in the deterministic relaxation segmentation process involves computing the Gibbs energy  $U_{C_i} = U_{C_i,1} + U_{C_i,2}$  of each pixel for each class  $C_i$ .  $U_{C_i,1}$  is the energy calculated from the intensity image and  $U_{C_i,2}$  is the energy calculated from the label image.  $E_i$  is the total energy of the whole image which is the summation of the energy of each pixel. As shown in Fig. 4.14, during each visit to site  $s$ , its energy  $U_{C_i}$  can be calculated, and the class corresponding to the lowest energy  $U_{C_i}$  is selected. The label image is then updated with the class label of the lowest energy. The process is repeated until there is no further change in the total energy of the image, or the the number of iterations exceeds a threshold. Since the

energy is not increasing at each pixel visiting, the segmentation process is bound to converge.

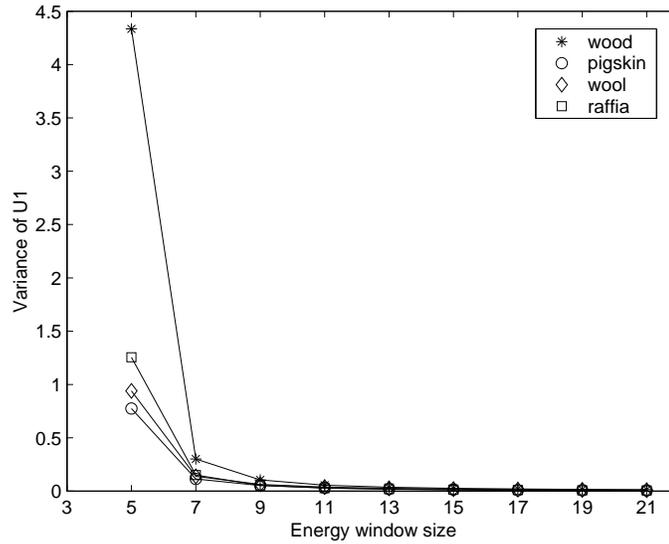


Figure 4.16: Variances of Brodatz texture energy  $U_1$  vs different energy window sizes.

In order to be compatible with  $U_{C_i,2}$ , which is only the energy of one neighbourhood window,  $U_{C_i,1}$  values are normalized by dividing the size of the energy window  $k \times k$ . The normalized  $U_{C_i,1}$  is in fact the average energy of each pixel in  $E_s$ . The energy window size  $k$  is an important parameter that needs to be selected carefully. Like the common condition when applying an average filter, if the size of the filter is too big, too much blurring results. The edge information will be heavily damaged. The same consideration applies to the value of  $k$ . It should be as small as possible to keep the edges between textures neat, and at the same time, for a uniform texture, the average energy of a pixel calculated within the energy window should be stable.

Fig. 4.16 plots the variance of  $U_1$  with different energy window size of four

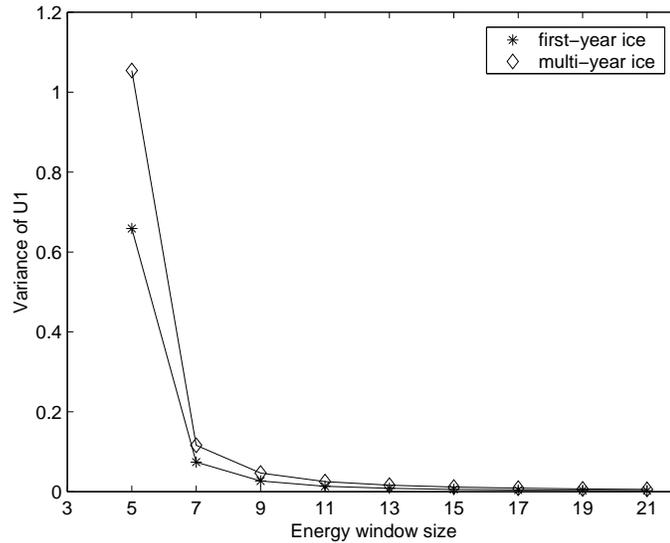


Figure 4.17: Variances of Radarsat sea ice texture energy  $U_1$  vs different energy window sizes

Brodatz textures. The x axis represents window sizes from 5 to 21. From the plot, it can be seen that when the window size is larger than 13, there is not much change in variance values. In the following experiments with Brodatz textures, the energy window size is set to 13. For sea ice texture image, 10 first year ice image samples and 10 multi-year ice image samples are selected from the Radarsat sea ice images. Fig. 4.17 shows the typical variances of  $U_1$  of first year and multi-year ice image with different energy window sizes. Based on the plot, the energy window size of sea ice image is set to 15.

## Chapter 5

# Experiments and Results Analysis

The *GLCP* method has been used to classify sea ice types in SAR imagery for years. Reviewing the image processing literature, there is still strong supportive evidence that this method is the currently preferred texture analysis algorithm for sea ice type identification [42, 2, 36]. However, this method suffers from having to select quantization, displacement, window size, orientation as well as the texture statistics. All suggested choices of these values are based on experiments using selected images of some areas. When model-based *MRF* methods are used for texture analysis, they are demonstrated to have different abilities compared to the *GLCP* method [12]. Encouraging results have been obtained for unsupervised texture segmentation using Brodatz textures [38, 18]. Unfortunately, the texture appearance of a consistent ice type is not as regular as a Brodatz texture, and most of the time, different ice types as well as open water are interwoven with each other. Trained human operators often need ancillary information to properly segment a SAR sea ice image. Can *MRF* methods be used in SAR sea ice image segmentation? Comparing the *GLCP* and *MRF* methods, which one can produce

a better segmentation result? Or under certain circumstances, which one is a better candidate? By now, there exists very limited published research investigating the potential of *MRF* methods in SAR sea ice image segmentation, since most *MRF* testing is usually represented using Brodatz textures. As a precursory and preliminary research project, this thesis will explore these concerns focusing on the following three research questions.

## 5.1 Research Questions

**Research Question One** *How does window size influence the estimated individual GLCP texture features and GMRF model parameters?*

The same window size ( $n$ ) can be used to determine the *GLCP* texture features as well as the *GMRF* model parameters. From the procedure described in Fig. 3.3 and Fig. 3.4, the window size determines the pixels used to calculate the *GLCM*. Similarly, the *GMRF* model parameters of the intensity image are also estimated based on the pixels involved in the window  $n$ . So, for both methods, the size of  $n$  is a very important parameter, and different number of pixels in  $n$  can result different estimation results. It is necessary to evaluate the effect of window size  $n$  on the stability of each estimated *GLCP* texture feature and *GMRF* model parameter. Conclusions on this research question can help to select the suitable window size for both methods.

**Research Question Two** *How does the window size  $n$  influence the separability of the clusters of the estimated GLCP texture features and GMRF model parameters?*

This research question explores the texture distinguishing ability of *GLCP* texture features vs *GMRF* model parameters from the classification point of view. As

described in Chapters 3 and 4, the segmentation approach of the *GLCP* method is based on the texture features of each pixel. First, the texture features are extracted from the image, then they are used as the input to a supervised or unsupervised classifiers. Similarly, using the *ICM* method, the *GMRF* model is estimated from the image first, then the segmentation is an optimization process for selecting a texture label for each pixel which makes the pixel have the smallest energy. In this process, the texture model is used to calculate the energy. Given a pair of textures, the separation of the *GLCP* texture features between the two textures versus the separation of the *GMRF* models provides a means to evaluate the texture distinguishing abilities of these two methods. Evaluating the texture distinguishing ability will also help to deduce the source of the problem from a poor segmentation result. It may be caused by the poorly estimated texture features or models from the very beginning, or the subsequent segmentation approach causes the texture information reserved in the texture features or models lose.

From the second research question, another issue can also be addressed, i.e., given a sufficiently large window size, which method has a better texture distinguishing ability? This can be regarded as a complement to the second research question. The second research question explores the effect of different window sizes on the texture distinguish ability of the *GLCP* texture features and *GMRF* model parameters. But what will happen if sufficiently large window sizes are provided for each method? The question is inspired by the fundamental difference of statistical-based and model-based texture analysis techniques. *MRF* texture models are associated with the natural formation process of the analyzed textures. In theory, the model-based texture analysis technique can not only describe the texture structure, but also has the ability to re-construct the textures. But, there is no known method to synthesize textures based on the *GLCP* texture features. From this point, it is

a logical inference that the *MRF* texture model contains more texture structure information for distinguishing texture types than the *GLCP* texture features.

**Research Question Three** *Given a window, what is the effect on the estimated *GLCP* texture features and *GMRF* model parameters if the window contains multiple textures with the possibility of irregular boundaries?*

The first and second research questions deal with homogeneous textures, i.e., the *GLCP* texture features or *GMRF* model parameters are estimated from the “pure” texture samples. But usually, given a natural image, some of the selected texture samples from the image may contain multiple textures. For example, sea ice may be crashed and broken into pieces because of the weather conditions. Different ice types are interwoven with each other in the image. Given a window, the bigger it is, the greater the probability that various ice types will be included in it. In this case, the estimation of *GLCP* texture features or *GMRF* model parameters may be damaged. What is the relationship of the estimated texture features from the multi-texture window with the texture features of each texture in the window?

The *GLCP* texture features are measurements that can describe a qualitative aspect of the texture. In contrast, the *GMRF* texture models describe the relationship of one pixel with its neighborhood pixels quantitatively. The texture can be uniquely generated once its texture model is known. If more than one texture class is found in a window, the estimated *GMRF* model from this window could not represent any class. In the segmentation process, the poorly estimated *GMRF* texture models will be applied to the whole image when calculating each pixel’s energy. In the *GLCP* method, one poor estimation from a multi-texture window will only affect the segmentation result of that window, and could not be spread to the whole image.

In the following three sections, the three research questions stated above will be explored based on several experiments.

## 5.2 Research Question One

**Research Question One** *How does window size  $n$  influences the estimated individual GLCP texture features and GMRF model parameters.*

### 5.2.1 Method

For the first research question, given a texture, the relative increase of standard deviation of each estimated *GLCP* texture feature and each *GMRF* model is calculated.

Given a window  $n$ , the *GLCP* texture features and *GMRF* model parameters can be estimated based on pixel values in this window. In each case, a multi-dimensional feature space is created. For example, given a texture sample, if two *GLCP* texture statistics are used with  $\delta = 1$  and  $\theta = 0^\circ/180^\circ, 90^\circ/270^\circ$ , then the *GLCP* texture feature space is four. If a third order *GMRF* texture model is estimated from this window, then six pairs of model parameters are involved, the *GMRF* feature space is dimensioned to six. The experiment is designed to check the effect of different window sizes on the stability of the estimation in each dimension for both methods.

One synthetic texture ( $1024 \times 1024$ ) generated using the models (A) in Table 5.1, one Brodatz texture ( $1024 \times 1024$ ), and one SAR sea ice image ( $768 \times 768$ ) are used for the experiments. From each texture image, for each window size ( $n = 8, 16, 32, 64, 96$ ), 60 samples of a certain size are randomly se-

Model	Model parameters					
	(-1,0)	(0,-1)	(-1,-1)	(-1,1)	(-2,0)	(0,-2)
A	0.520252	0.0934154	0.0303413	0.0180476	-0.148331	-0.0216434
B	0.468389	0.308257	-0.0755398	-0.0755797	-0.100678	-0.0407557
C	0.423393	0.406875	-0.178478	-0.1887020	-0.121439	-0.0649544

Table 5.1: The *GMRF* models used to generate the synthetic textures used in the experiments in Chapter 5.

lected. The window sizes used for selecting the samples are 8, 16, 32, 64, 96. The three texture statistics dissimilarity, entropy and correlation with  $\delta = 1$  and  $\theta = 0^\circ/180^\circ, 45^\circ/225^\circ, 90^\circ/270^\circ, 135^\circ/315^\circ$  are used in the *GLCP* method. The third order *GMRF* texture model for synthetic texture, and fourth order *GMRF* model for Brodatz and sea ice textures. The test experiment is designed to keep watching on the relative change of standard deviation in each data space with different window sizes. The variance  $VAR_{feature}^{sample\ size}$  of each estimated feature from the 60 samples with a certain sample size is calculated first. For example, the *GLCP* entropy texture feature with  $\delta = 1$  and  $\theta = 0^\circ/180^\circ$  has five variance values:  $VAR_{E(1,0/180)}^8, VAR_{E(1,0/180)}^{16}, VAR_{E(1,0/180)}^{32}, VAR_{E(1,0/180)}^{64}$  and  $VAR_{E(1,0/180)}^{96}$ . For a comparison purpose, the change of standard deviation  $V_{feature}^{sample\ size}$  is obtained by normalizing each  $VAR_{feature}^{sample\ size}$  by the variance of the 96 sample size as follows:

$$V_{feature}^{sample\ size} = \frac{VAR_{feature}^{sample\ size}}{VAR_{feature}^{96}}$$

## 5.2.2 Experiment Results

A typical plot of this experiment is given in Fig. 5.1 using pigskin texture of Brodatz image. It shows the changes of standard deviations  $V_{feature}^{sample\ size}$  of each

<i>GLCP</i> texture features				
	96 to 8	96 to 16	96 to 32	96 to 64
Synthetic	9.3270	5.4479	2.9316	1.5454
Brodatz	6.8539	4.2216	2.3303	1.2397
Sea ice	2.5119	2.0420	1.3403	0.9901
<i>GMRF</i> model parameters				
	96 to 8	96 to 16	96 to 32	96 to 64
Synthetic	28.3209	8.5046	3.9449	1.6601
Brodatz	16.4391	4.8464	2.2325	1.2402
Sea ice	21.3758	7.3539	3.1928	1.7490

Table 5.2: Average deterioration of the standard deviation of the estimated *GLCP* texture features and the *GMRF* model parameters from window sizes 96 to 64, 32, 16, 8.

estimated *GLCP* texture feature and each *GMRF* model parameter from window sizes 96 to window size 8. Fig 5.1 a plots the six *GLCP* texture features ( $\theta = 45^\circ/225^\circ, 135^\circ/315^\circ$  show the similar changes as  $\theta = 0^\circ/180^\circ, 90^\circ/270^\circ$ ), and figure b plots the six *GMRF* model parameters (the last four parameters show similar changes as the first six). Several observations can be obtained from Fig. 5.1. First, larger window size leads to more stable estimates than smaller window size. Second, with the window size decreasing, the standard deviation of each estimated *GLCP* texture feature and the *GMRF* model parameter increase exponentially. Third, with the window size decreasing, the standard deviation of each *GMRF* model parameter increases faster than the *GLCP* texture features, especially from window size 32 to window size 8.

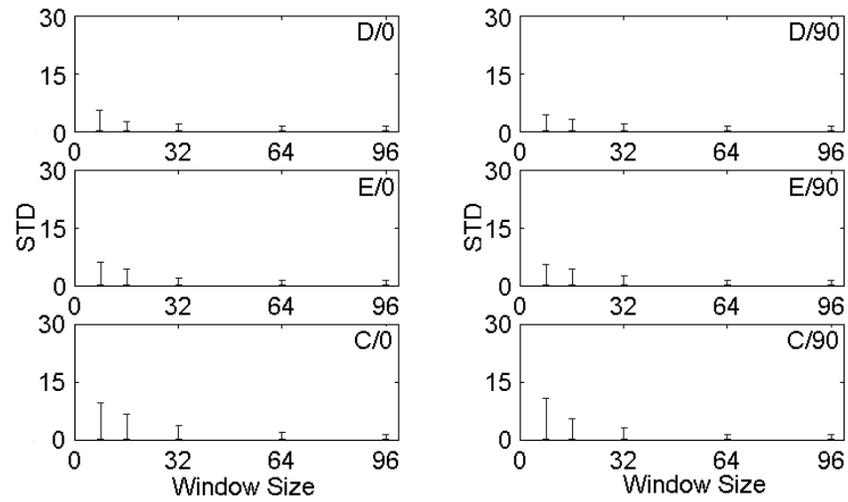
These observations are supported by the numbers in Table 5.2, which summa-

izes the average increase of the standard deviation of each estimation from the three textures used in this experiment. For example, using the *GLCP* method, the biggest standard deviation increase is a factor of 9.3270, which happened in synthetic texture from window size 96 to window size 8. Whereas, the biggest increase using the *GMRF* method is 28.3209, which also happened in synthetic texture from window size 96 to window size 8. Comparing the average increases of standard deviations from larger window size (96) to smaller window size (64 32 16 8), the *GMRF* model always exceeds the *GLCP* texture features. For the estimated *GMRF* model parameter, bigger standard deviations could make the same parameter positive in one estimation window and negative in another estimation window. As demonstrated in Section 4.3.3, the positive and negative value of *MRF* model parameter behave quite differently in the texture formation process. In this case, the big standard deviations will heavily damage the estimation of the texture model.

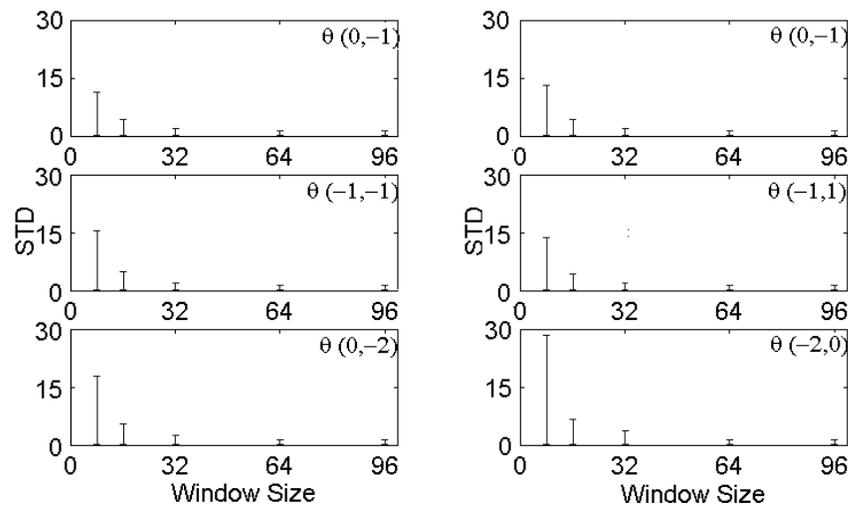
Based on the experiment, one can see that to obtain a stable estimation, the *GMRF* needs more data than *GLCP*. A big window size for a robust estimation should be more necessary and efficient for the *GMRF* than *GLCP*.

### 5.3 Research Question Two

**Research Question Two** *How does the window size  $n$  influence the separability of the clusters of estimated *GLCP* texture features and *GMRF* model parameters.*



(a)



(b)

Figure 5.1: Change of standard deviation of the estimated *GLCP* texture features and the *GMRF* model parameters with different window size. (a) *GLCP* texture features (D: dissimilarity, E: entropy, C: correlation,  $\delta = 1$ ). (b) *GMRF* model parameters.

### 5.3.1 Method

This research question explores the effect of different window size on the overall feature space for each method. The experiments are designed to consider the multi-dimensional feature space of each method together by applying a space transformation technique. In fact, evaluating the separability of the whole feature space is more consistent for the texture classification approach, which considers the feature space together instead of each one individually.

Using the Fisher linear discriminant (*FLD*) technique [45], the clusters of *GLCP* features and *GMRF* model parameters estimated from a pair of textures will be projected into one dimensional feature space. The separability of each method can be evaluated by comparing the Fisher inter-class separation of the projected clusters and the Bhattacharyya error bound [45] between the clusters in the multi-dimensional feature space.

#### Fisher Linear Discriminant

The distance between clusters is a common measurement for their separability. But it is difficult to visualize the distance between clusters where the feature space dimension is larger than three. Using the *FLD* technique, one can project  $d$  dimensional data samples onto a line  $\omega$  and calculate a weighted distance between the projected clusters [45].

The discriminant vector  $\omega$  for classes  $c_1$  and  $c_2$  can be found by optimizing the following Fisher criteria:

$$J(\omega) = \frac{\omega^T S_B \omega}{\omega^T S_w \omega} \quad (5.1)$$

where  $S_B$  is the between-class matrix which is the squared distance between the class means  $m_1$  and  $m_2$ .

$$S_B = (m_1 - m_2)(m_1 - m_2)^T \quad (5.2)$$

and  $S_w$  is the within-class scatter matrix which measures the variance of classes.

$$S_w = S_1 + S_2 \quad (5.3)$$

where  $S_1$  and  $S_2$  is the scatter matrices of each class.

$$S_i = \sum_{x \in C_i} (x - m_i)(x - m_i)^T \quad (5.4)$$

By taking the derivative of  $J(w)$ , the optimal discriminant vector  $w$  is obtained:

$$w = S_w^{-1}(m_1 - m_2) \quad (5.5)$$

Let  $x$  be an  $m \times d$  matrix, where  $m$  is the number of samples in the class and  $d$  is the dimension of samples. The following transformation

$$y = w^t x \quad (5.6)$$

converts a  $d$ -dimensional problem to a more manageable one-dimensional one. The Fisher inter-class distance is the value of Fisher criterion (Eqn. 5.1) obtained by using the optimal discriminant vector  $w$  in Eqn. 5.5.

To plot the distributions of two sets of multi-dimensional *GLCP* texture features or *GMRP* model parameters on their Fish discriminant vector  $w$  estimated from two textures, Eqn. 5.6 is used in this research question. Observing the distributions on  $w$ , one can visually evaluate the separation between two data sets.

### **Bhattacharyya Error Bound**

A further insight of the separability of *GLCP* texture features vs *GMRP* models can be obtained by calculating the upper bound of classification error between the feature clusters of the texture pair in the original multi-dimensional feature space.

Let  $\alpha_1$  and  $\alpha_2$  represent two clusters, then the upper error bound of classification error of them is [45]:

$$P(\text{error}) \leq P^\beta(\alpha_1)P^{1-\beta}(\alpha_2) \int p^\beta(y|\alpha_1)p^{1-\beta}(y|\alpha_2)dy \quad \text{for } 0 \leq \beta \leq 1 \quad (5.7)$$

where  $P(\alpha_1)$  and  $P(\alpha_2)$  represent the probability of class  $\alpha_1$  and  $\alpha_2$ , and  $p(y|\alpha_1)$  and  $p(y|\alpha_2)$  represent the class *pdfs*. When the two classes  $\alpha_1$  and  $\alpha_2$  have the same probability of occurrence, i.e.,  $\beta = 1/2$ , Eqn. 5.7 is referred to as the Bhattacharyya bound expressed as:

$$P(\text{error}) \leq \sqrt{P(\omega_1)P(\omega_2)}e^{-k(1/2)} \quad (5.8)$$

where the Gaussian case for  $k(1/2)$  is:

$$k(1/2) = 1/8(\mu_1 - \mu_2)^t \left[ \frac{\Sigma_1 + \Sigma_2}{2} \right]^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{|\frac{\Sigma_1 + \Sigma_2}{2}|}{\sqrt{|\Sigma_1||\Sigma_2|}} \quad (5.9)$$

### 5.3.2 Experiment Results

Three texture image pairs are used in the experiments: synthetic ( $1024 \times 1024$ ), Brodatz ( $1024 \times 1024$ ) and SAR sea ice texture images ( $768 \times 768$ ) Fig. 5.2. The synthetic texture images are generated using the *GMRF* texture model (A) and (B) in Table 5.1. For each pair of texture images, 60 sample images with size 8, 16, 32 and 64 are randomly selected from each texture. The selected *GLCP* parameters are dissimilarity and entropy along  $0^\circ/180^\circ$ ,  $45^\circ/225^\circ$ ,  $90^\circ/270^\circ$ ,  $135^\circ/315^\circ$  directions with  $\delta = 1$ . So for each texture sample, eight co-occurrence texture features are extracted. The third order *GMRF* model is used for synthetic textures (six model parameters). The fourth order *GMRF* model is applied to Brodatz and SAR sea ice textures (ten model parameters). The correlation texture statistics is not used in the experiment for two reasons. First, among the three texture statistics, correlation contains the least texture distinguishing information compared with dissimilarity

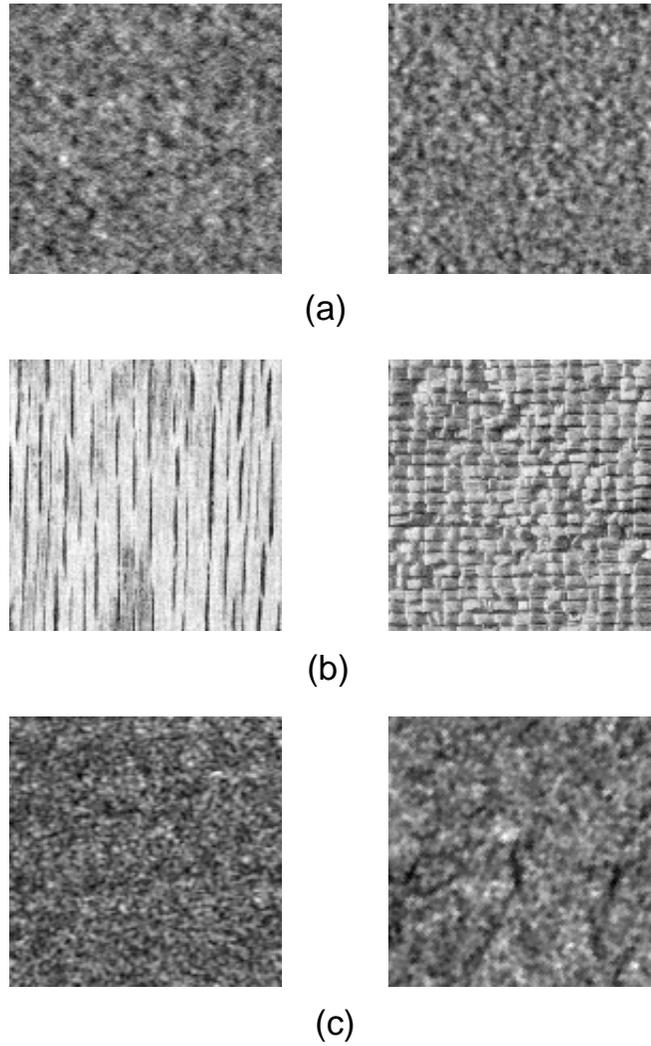
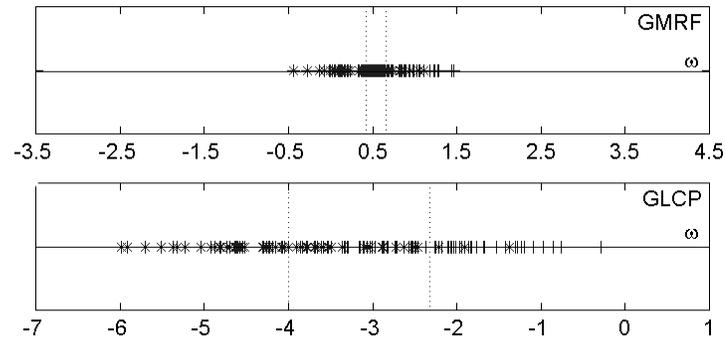


Figure 5.2: Test image pairs for the second research question. (a) Synthetic *GMRF* textures (model A and B in Table 5.2.1). (b) Brodatz textures. (c) SAR sea ice textures (L: first year ice R: multi-year ice).

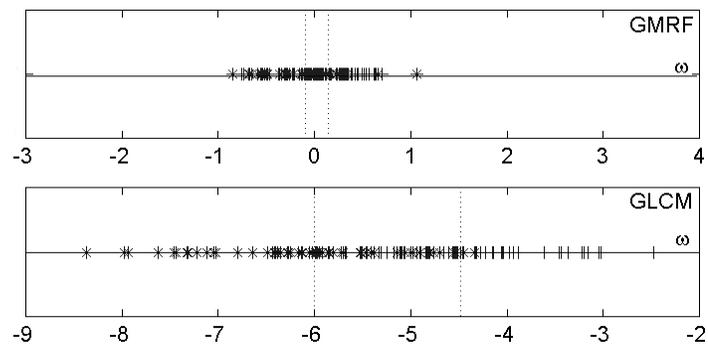
and entropy. Second, the *GLCP* and *GMRF* should have the similar feature space dimensions for comparison. The projected *GMRF* model parameters on  $\omega$  of each texture pair are plotted in Fig. 5.3 ( $n = 8$ ), Fig. 5.4 ( $n = 16$ ) and Fig. 5.5 ( $n = 32$ ). For each texture pair, its projected *GLCP* texture feature clusters on  $\omega$  are also plotted under *GMRF* plot with same  $x$  axis unit. The vertical dash lines in each figure show the mean position of each projected cluster.

Table 5.3 reports the Bhattacharyya error bounds (BEB) of each texture pair and the Fisher criteria ( $J$  Fisher inter-class distance) of each projected texture pair plotted in Figs. 5.3, 5.4 and 5.5 (the first three rows). Each data cluster is assumed to have a normal distribution.

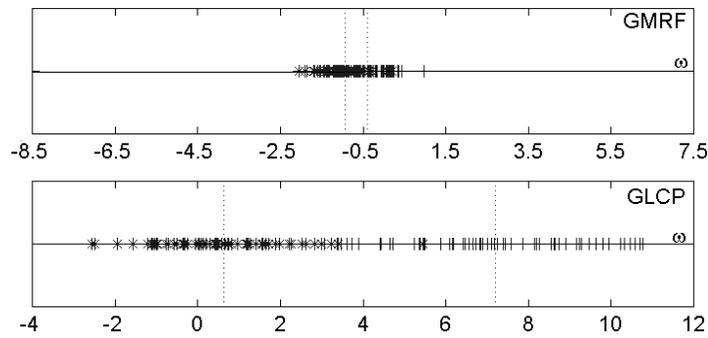
Observing the distribution of the two data clusters projected on the line  $w$  from Fig. 5.3 to Fig. 5.5 as well as the corresponding values of Fisher inter-class distance listed in Table 5.3, there exists a very clear tendency. First, in Fig. 5.5 where  $n = 32$ , all the *GMRF* models have better separations than *GLCP* texture features, especially for the sea ice texture pair. As the window size decreases to  $16 \times 16$ , the cluster center distance of *GMRF* between the projected synthetic and Brodatz texture pairs becomes less than cluster center distance of the *GLCP*. In Table 5.3, the observation from Fig. 5.4 is supported by the BEB and  $J$  values, i.e., compared with *GLCP*, *GMRF* has bigger BEB values and smaller  $J$  values for synthetic and Brodatz texture pair. These observations mean that as the window size decreases to  $16 \times 16$ , the separability of *GMRF* model for synthetic and Brodatz texture pair becomes inferior to *GLCP* texture features. Finally as the window size decreases to  $8 \times 8$ , the superiority of *GLCP* texture features over *GMRF* model for separating all the three texture pairs are very obvious from both the plots and the corresponding BEB and  $J$  values. The same experiment based on a  $64 \times 64$  window size generates a similar result as the  $32 \times 32$  window size, i.e., the separability



(a)

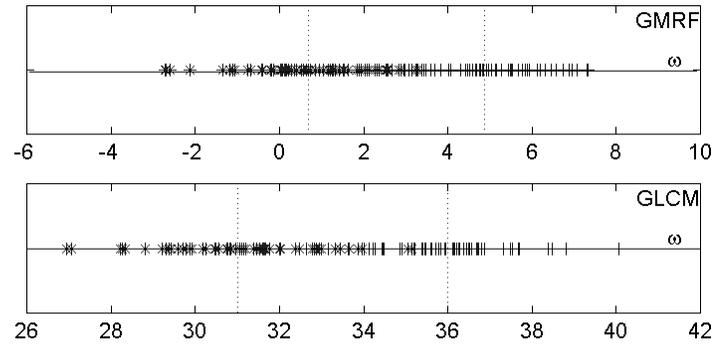


(b)

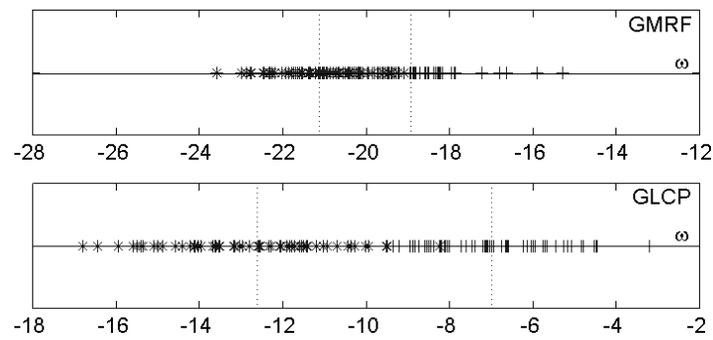


(c)

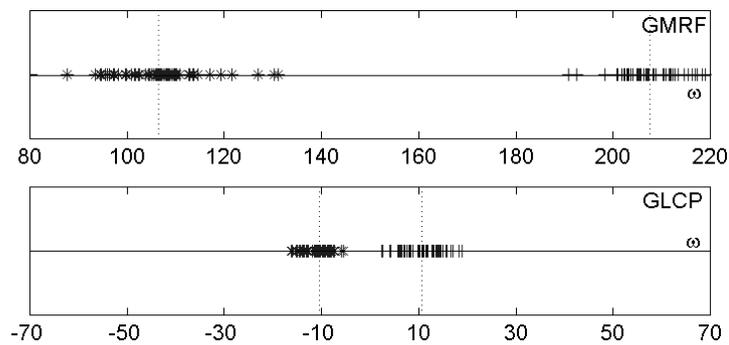
Figure 5.3: Distributions of the two sets of the *GMRF* model parameters and the *GLCP* texture features estimated from Fig. 5.2 using  $n = 8$  on their Fish discriminant vector  $w$ . (a) Synthetic textures. (b) Brodatz textures (c) SAR sea ice textures.



(a)

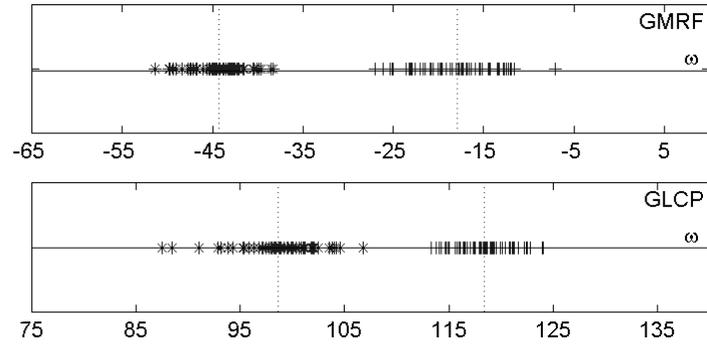


(b)

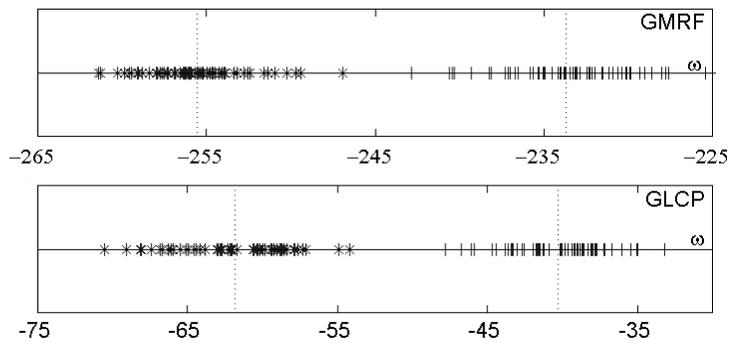


(c)

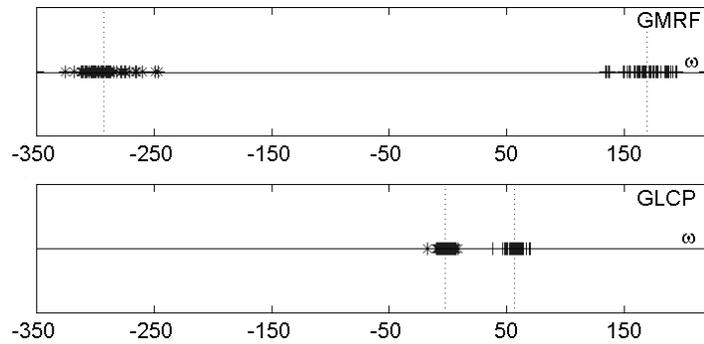
Figure 5.4: Distributions of the two sets of the *GMRF* model parameters and the *GLCP* texture features estimated from Fig. 5.2 using  $n = 16$  on their Fish discriminant vector  $w$ . (a) Synthetic texture. (b) Brodatz texture. (c) SAR sea ice texture.



(a)



(b)



(d)

Figure 5.5: Distributions of the two sets of the *GMRF* model parameters and the *GLCP* texture features estimated from Fig. 5.2 using  $n = 32$  on their Fish discriminant vector  $w$ . (a) Synthetic texture. (b) Brodatz texture. (c) SAR sea ice texture.

of the *GMRF* texture model is better than the *GCLP* texture features. Another inference can also be drawn from the above observations. With the window size decreasing, the separability of *GMRF* models for textures deteriorates faster than *GLCP* texture features. This observation is consistent with the conclusion of the first research question.

Based on experiments, two conclusions can be made. First, both methods prefer the larger window size. Second, as the window size decreases, the separability of *GMRF* models become inferior to the *GLCP* texture features.

As mentioned in section 5.1, one issue related to the second research question needs to be analyzed. Given a sufficiently large window size, which method has a better texture distinguishing ability?

To answer this question, the column of  $32 \times 32$  window size in Table 5.3 need to be analyzed. One can see that the BEB values of all the three texture pairs have a magnitude equal or less than  $10^{-3}$ . This means the Bhattacharyya error bounds of *GMRF* clusters are improved not bigger than  $10^{-3}$  comparing with *GLCP* texture features. In this case, one could say the *GMRF* texture model and *GLCP* texture features have the similar separability for these three texture pairs. Observing the parameter set of model (A) and (B) in Table 5.1, they are not as similar to each other as (B) and (C). To avoid one-sidedness, Fig. 5.6 shows the texture samples synthesized from model (C) and (B). Fig. 5.7 illustrates their projected feature cluster distributions on  $w$  (a:  $n = 8$ , b:  $n = 16$ , c:  $n = 32$ ). Their Bhattacharyya error bounds and Fisher inter-class distance are also reported in Table 5.3 in the row of synthetic\*. When the sample size is  $32 \times 32$ , different from all the first three texture pairs in Fig. 5.5, the *GLCP* texture features of Fig. 5.6 has a better separability than *GMRF* texture models. But again, if checking Table 5.3, the difference of error bounds between *GMRF* models and *GLCP* texture features are

Bhattacharyya error bounds (BEB) and Fisher criterion ( $J$ )				
8 × 8 window size				
	GMRF		GLCP	
	BEB	$J$	BEB	$J$
synthetic	$3.5 \times 10^{-1}$	0.24	$2.9 \times 10^{-1}$	1.68
Brodatz	$3.7 \times 10^{-1}$	0.24	$2.0 \times 10^{-1}$	1.52
sea ice	$2.5 \times 10^{-1}$	0.55	$4.8 \times 10^{-2}$	6.58
synthetic*	$2.2 \times 10^{-1}$	1.15	$1.7 \times 10^{-2}$	12.26
16 × 16 window size				
	GMRF		GLCP	
	BEB	$J$	BEB	$J$
synthetic	$1.7 \times 10^{-1}$	4.19	$1.3 \times 10^{-1}$	4.98
Brodatz	$1.3 \times 10^{-1}$	2.21	$4.4 \times 10^{-2}$	5.64
sea ice	$7.9 \times 10^{-13}$	101.03	$1.1 \times 10^{-3}$	21.07
synthetic*	$2.8 \times 10^{-2}$	8.06	$9.7 \times 10^{-7}$	51.21
32 × 32 window size				
	GMRF		GLCP	
	BEB	$J$	BEB	$J$
synthetic	$4.0 \times 10^{-4}$	26.33	$3.0 \times 10^{-3}$	19.75
Brodatz	$3.2 \times 10^{-3}$	21.85	$3.5 \times 10^{-3}$	21.54
sea ice	$1.5 \times 10^{-51}$	462.21	$6.7 \times 10^{-8}$	58.74
synthetic*	$1.5 \times 10^{-7}$	55.72	$6.9 \times 10^{-25}$	219.06

Table 5.3: Bhattacharyya error bounds and Fisher criteria of the texture pairs in Fig. 5.2 and 5.6

negligible.  $64 \times 64$  window size generates the similar result as  $32 \times 32$  window size.

Based on the supplementary experiment above, another conclusion in the second research question can be obtained. With a large window size, the *GMRF* model and *GLCP* texture features have similar separability.

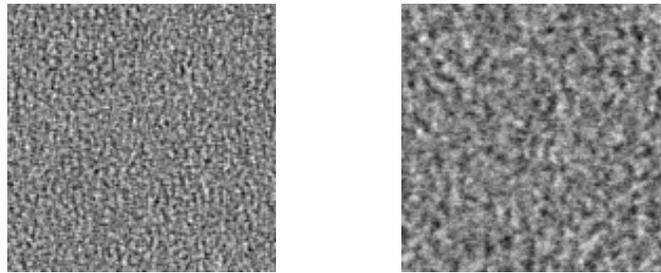
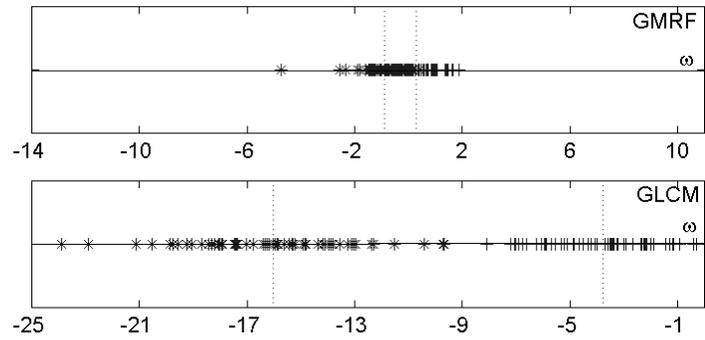
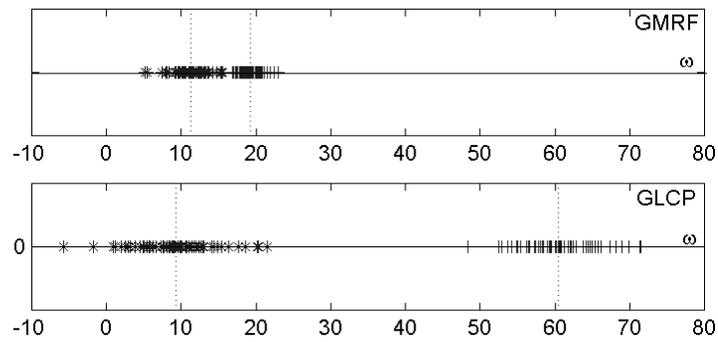


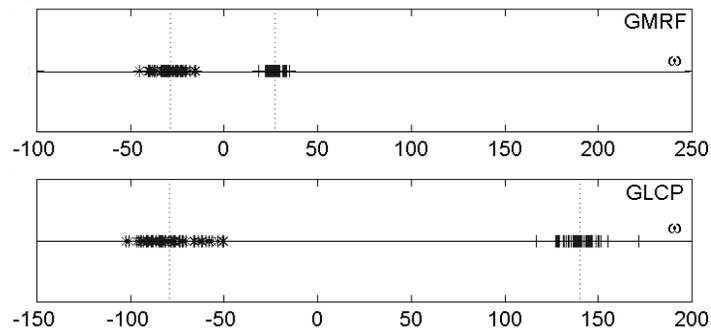
Figure 5.6: Another test image pair for the second research question (Left: model C, Right: model B in Table 5.1).



(a)



(b)



(c)

Figure 5.7: Distributions of the *GMRF* texture model and the *GLCP* texture features estimated from Fig. 5.6 on their Fish discriminant vector  $w$ . (a)  $n = 8 \times 8$ . (b)  $n = 16 \times 16$ . (c)  $n = 32 \times 32$ .

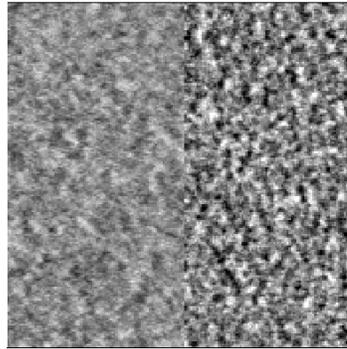
## 5.4 Research Question Three

**Research Question Three** *Given a window, what is the effect on the estimated  $GLCP$  texture features and  $GMRF$  model parameters if the window contains multiple textures with the possibility of irregular boundary*

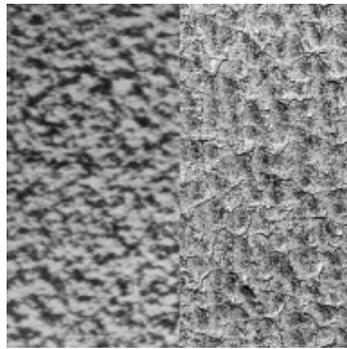
### 5.4.1 Method

Given an estimation window with more than one texture, the extracted  $GLCP$  texture features or the estimated  $GMRF$  texture model from this window can not represent correctly any texture type in this window because of the “un-pure” data. How does each texture type in a given window affect the estimation result? What is the relationship of the extracted  $GLCP$  texture features or the estimated  $GMRF$  texture model from a multi-texture window with the  $GLCP$  texture features or the  $GMRF$  model of each texture in this window?

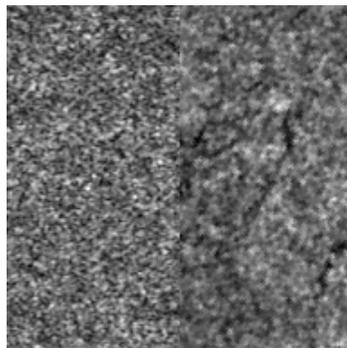
Based on the calculation of  $GLCP$  texture features and  $GMRF$  model parameters, a reasonable inference is that the extracted  $GLCP$  texture features or estimated  $GMRF$  model from a multi-texture window should have a linear relationship with the  $GLCP$  texture features or  $GMRF$  texture model of each texture in this window. For example, if two texture types A and B in window  $n$ , then  $F = a \times F^A + b \times F^B$ , where  $F$  is any co-occurrence texture feature of window  $n$ ,  $a$  is the ratio of texture A in window  $n$ ,  $b$  is the ratio of texture B in window  $n$ , and  $a + b = 1$ . The same relationship for the  $GMRF$  model parameters:  $\theta = a \times \theta^A + b \times \theta^B$ , where  $\theta$  is a parameter of the  $MRF$  model. Although not proven mathematically, the above relationship can be demonstrated by the following experiment using the three texture images in Fig. 5.8.



(a)



(b)



(c)

Figure 5.8: Three texture images separated by straight boundary. (a) Synthetic texture. (b) Brodatz texture. (c) SAR sea ice texture.

### 5.4.2 Experiment Results

Fig. 5.8 shows three texture images (a: synthetic, b: Brodatz, c: SAR sea ice) which are separated by straight vertical boundary from the middle. For each pixel in the image, its *GLCP* texture statistics dissimilarity, entropy ( $\delta = 1, \theta = 0^\circ/180^\circ, 45^\circ/225^\circ, 90^\circ/270^\circ, 135^\circ/315^\circ$ ) and *GMRF* model parameters (third order for synthetic texture, fourth order for Brodatz and SAR sea ice textures) are estimated based on a window ( $n = 16, n=32$ ) centering on this pixel.

Fig. 5.9 shows the *GLCP* texture features averaged over 50 arbitrarily selected rows in Fig. 5.8 a (a:  $n = 16$ , b:  $n = 32$ ). The area between the two vertical lines is the boundary area, i.e., when the window moves into the boundary area, it will contain two textures from the left and right sides. The *GLCP* texture features along  $\theta = 45^\circ/225^\circ, 135^\circ/315^\circ$  directions have the similar results as the  $\theta = 0^\circ/180^\circ, 90^\circ/270^\circ$  directions and they are not plotted out. Fig. 5.10 shows the first four *GMRF* model parameters averaged over the same 50 rows. The rest of the model parameters have the similar results as the first four and are not plotted out. Observing these plots, one can see that within the boundary areas, the values of the *GLCP* texture features or the *GMRF* model parameters change approximately linearly from the value of the left texture to the value of the right texture. This phenomena is more clear in (b) of both figures because the larger window size makes the estimations more stable. The similar situation in the boundary areas happened for Brodatz texture (Fig. 5.11, Fig. 5.12) and SAR sea ice texture (Fig. 5.13, Fig. 5.14).

The above experiment shows that the estimations in both methods can be damaged by the multi-texture windows along the boundary areas. Given a window with multiple textures, each estimated *GLCP* texture feature seems to be a weighted lin-

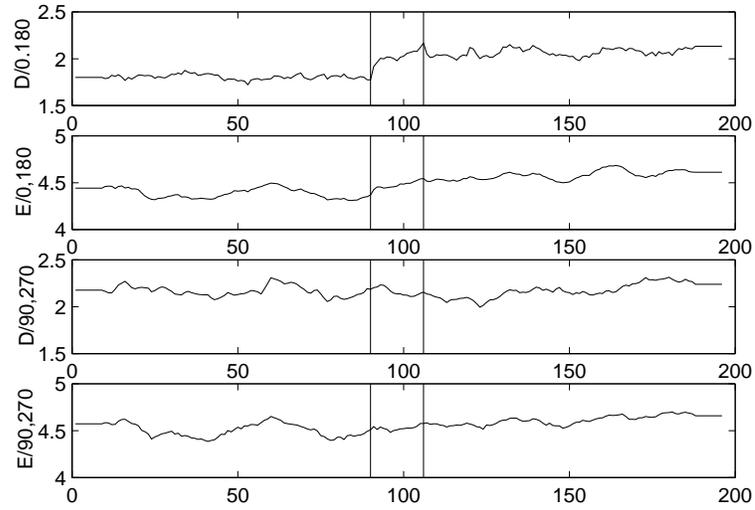
ear combination of the corresponding *GLCP* texture feature of each texture. The same conclusion applies to each estimated *GMRF* model parameter.

Based on the above conclusions, one possible mis-classification in the boundary areas can be illustrated using Fig. 5.15. Texture A and B has a right-angle boundary. When an  $5 \times 5$  window is located on pixel  $s$ , 9 pixels in this window belong to texture A, and 16 pixels belong to texture B. Based on the above experiment, for both methods, the estimated parameters should be more similar to texture B rather than A. The pixel  $s$  could be mis-classified into texture B. In this case, the right-angle boundary between the two textures could be corroded and moved into texture A in the segmentation result.

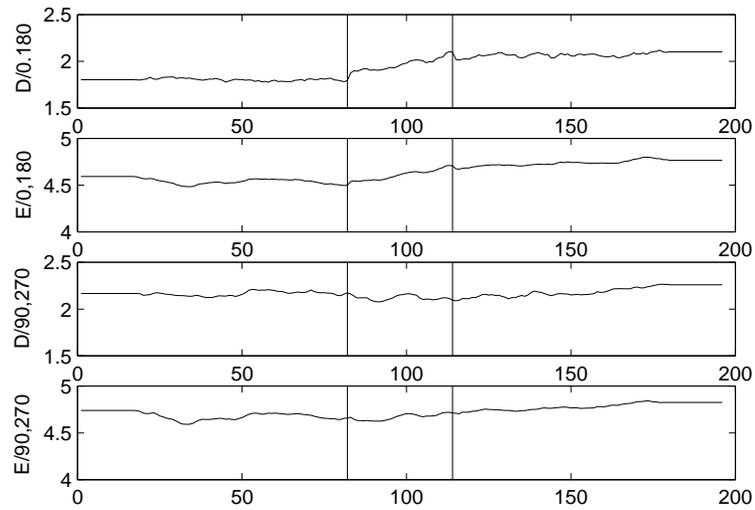
In the practical segmentation problem, the poor estimation caused by the multi-texture windows can not be avoided since the irregular boundaries between textures exist anyway. How does the damaged estimations affect the segmentation results for each method? Fig 5.16 to Fig. 5.18 shows the segmentation results of three Brodatz texture images with different boundary situation. To concentrate on the evaluation of boundary effect on the segmentation results, all the original images ( $192 \times 192$ ) contain the same two Brodatz textures paper and pigskin.

Fig. 5.16 a shows the first original image with a straight vertical boundary, Fig. 5.16 b represents the true segmentation. Using  $16 \times 16$  window size, both the *GLCP* and *MRF* are easily able to segment the original image, and produce accurate segmentation results. For comparison, the boundary from the *MRF* is better than the *GLCP*.

The next original image Fig. 5.17 a has a sinusoidal boundary with three periods, which makes it more difficult to distinguish than the first original image. The  $16 \times 16$  window size is used, and the segmentation result is again about the same.

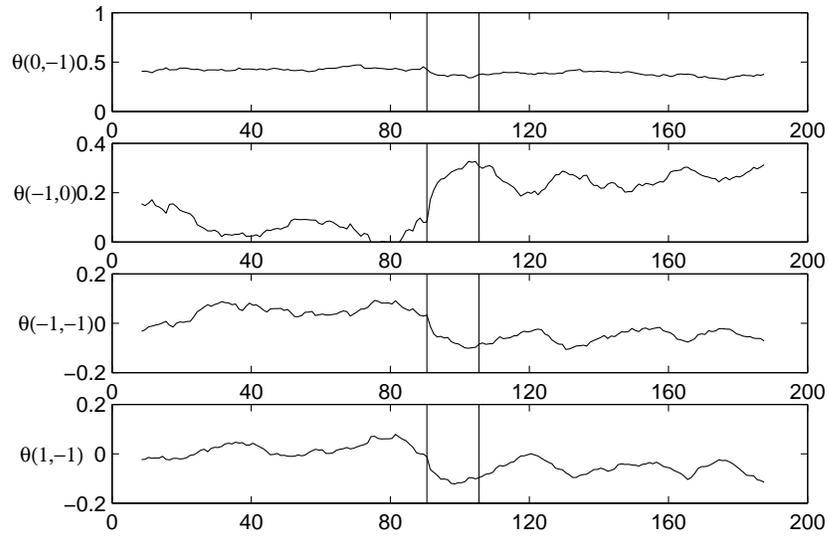


(a)

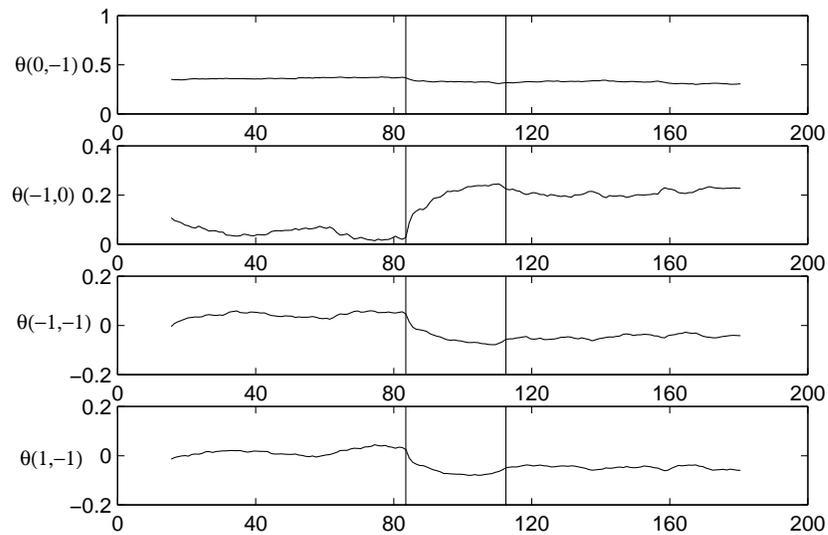


(b)

Figure 5.9: The averaged *GLCP* texture features over 50 arbitrarily selected rows from Fig. 5.8 a (the area between the two straight lines is the boundary area). (a)  $n = 16$ . (b)  $n = 32$ .



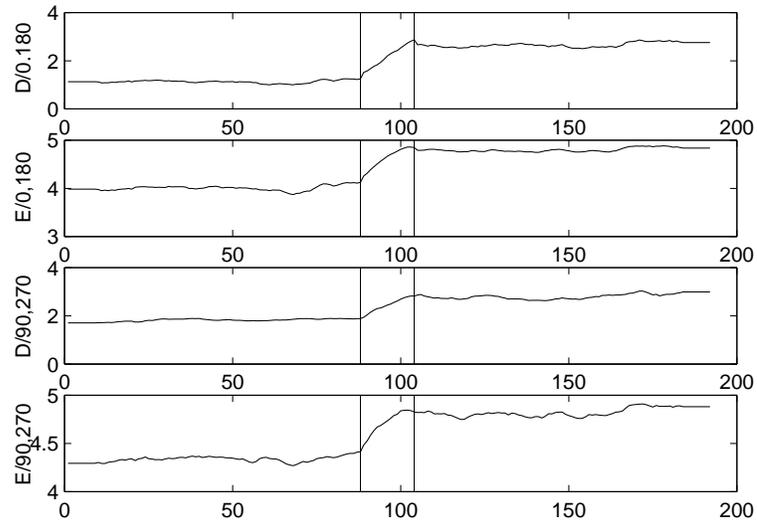
(a)



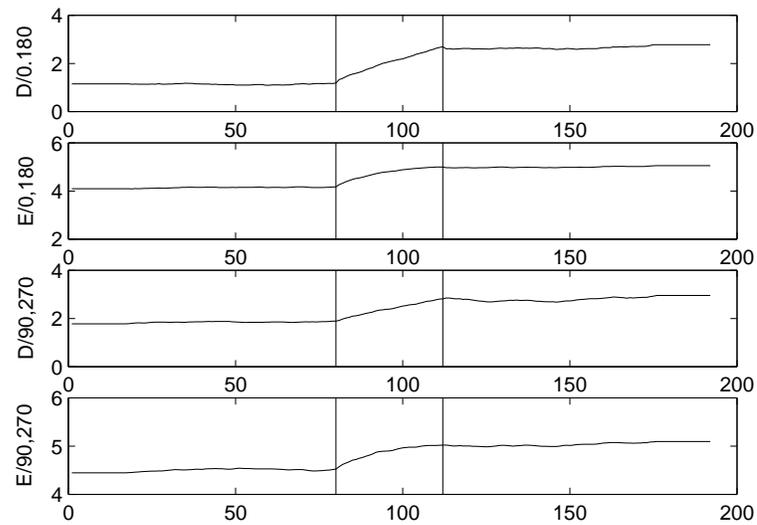
(b)

Figure 5.10: The averaged *GMRF* model parameters over 50 arbitrarily selected rows from Fig. 5.8 a (the area between the two straight lines is the boundary area).

(a)  $n = 16$ . (b)  $n = 32$ .

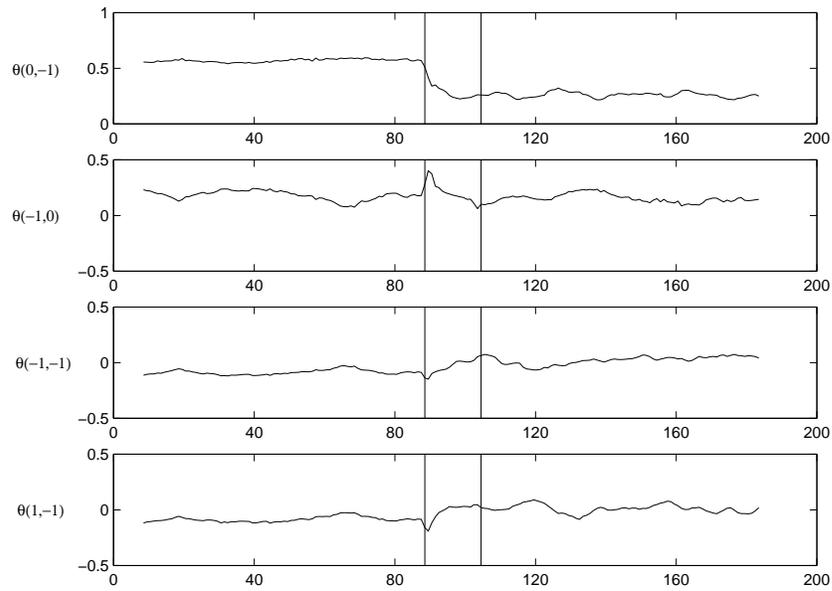


(a)

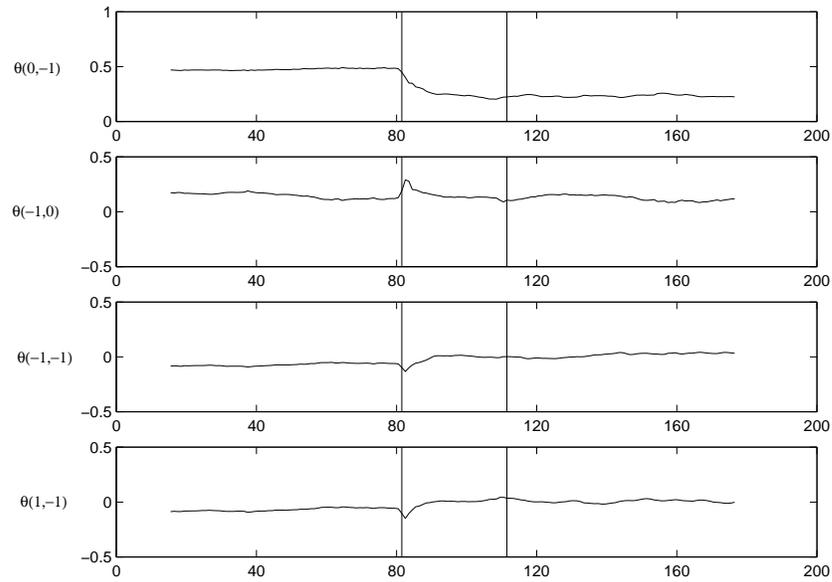


(b)

Figure 5.11: The averaged *GLCP* texture features over 50 arbitrarily selected rows from Fig. 5.8 b (the area between the two straight lines is the boundary area). (a)  $n = 16$ . (b)  $n = 32$ .

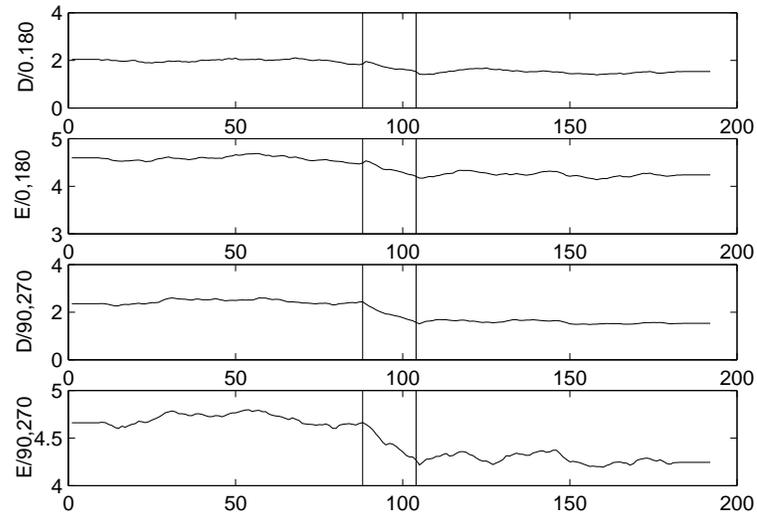


(a)

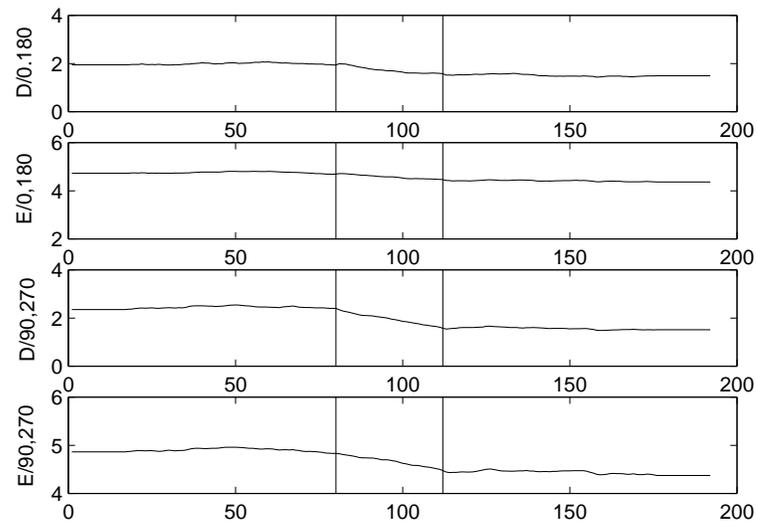


(b)

Figure 5.12: The averaged *GMRF* model parameters over 50 arbitrarily selected rows from Fig. 5.8 b (the area between the two straight lines is the boundary area).  $n = 16$ . (b)  $n = 32$ .

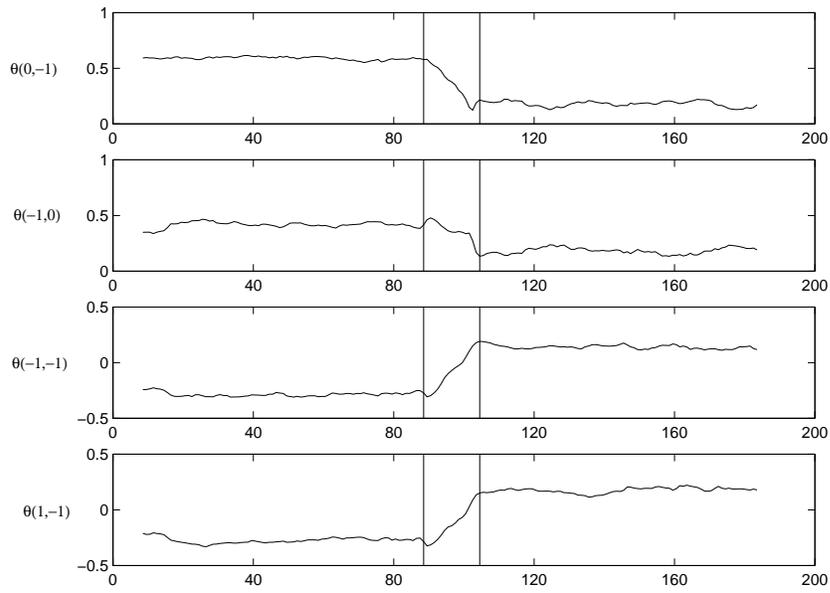


(a)

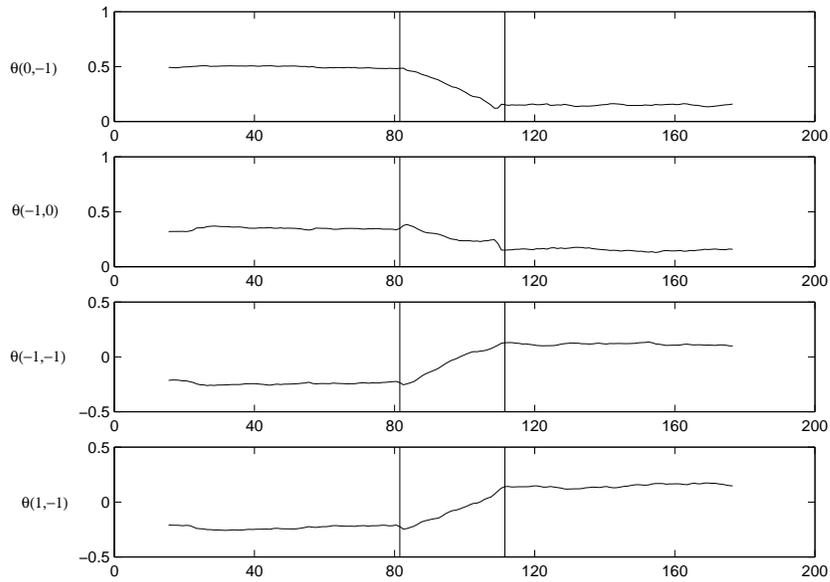


(b)

Figure 5.13: The averaged *GLCP* texture features over 50 arbitrarily selected rows from Fig. 5.8 c (the area between the two straight lines is the boundary area). (a)  $n = 16$ . (b)  $n = 32$ .



(a)



(b)

Figure 5.14: The averaged *GMRF* model parameters over 50 arbitrarily selected rows from Fig. 5.8 c (the area between the two straight lines is the boundary area).

(a)  $n = 16$  . (b)  $n = 32$ .

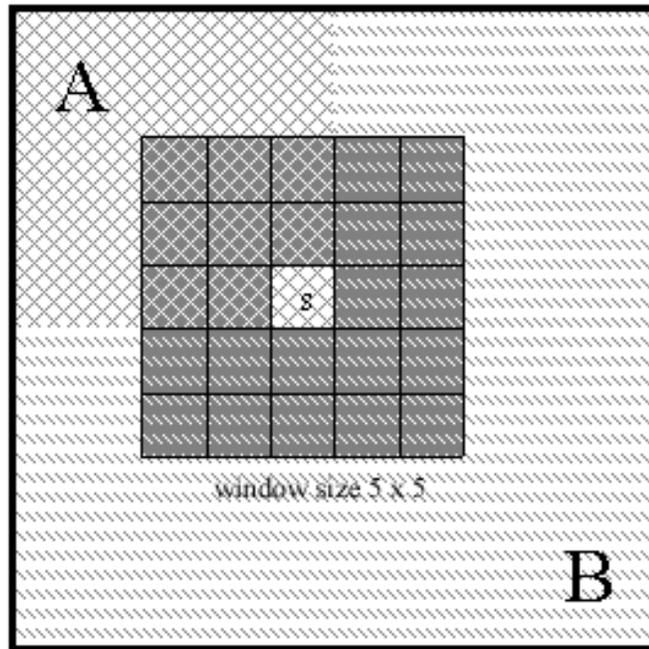


Figure 5.15: The estimated texture parameters of pixel  $s$  using  $5 \times 5$  window could be more similar to the parameters of texture B than A.

However, the better boundary obtained by the *MRF* in the first original image is not obvious in this case. Continuing to increase the complication of the boundary, the results from the *MRF* and *GLCP* methods do not appear similar any more. Fig. 5.18 a has a sinusoidal boundary with eleven periods which makes the boundary between two textures much more difficult to distinguish than the first two original images. The segmentation results using  $16 \times 16$  window size from the *GLCP* and *MRF* methods are shown in Fig. 5.18 c and d. Both methods do not have a successful segmentation. For the *GLCP* method, the bad segmentation concentrates on the boundary area, the left and right sides have the correct segmentation. But for the *MRF*, the bad segmentation happens all over the image. Fig. 5.18 e and f are the segmentation results using  $8 \times 8$  window size. The *GLCP* method produces

a much better segmentation than the *MRF* method.

Observing the segmentation results, one can see that for textures with simple boundaries, given a window, most of the time, there is only one texture type involved in this window. In this case, both methods can make a correct estimation and thus produce an accurate segmentation. When segmenting the images with complicated boundaries all over the image, the ability of the two methods may be damaged by the multi-texture windows, the segmentation results from both methods should not be good. To avoiding the multi-texture windows, small window size could be used. In this case, *GLCP* method probably produces a better segmentation result than the *MRF* method. This can be explained from the results of the first two research questions, i.e., as the window size decreases, the standard deviation of the estimated *GMRF* model parameters as well as its separability for textures becomes more inferior compared to the *GLCP* texture features.

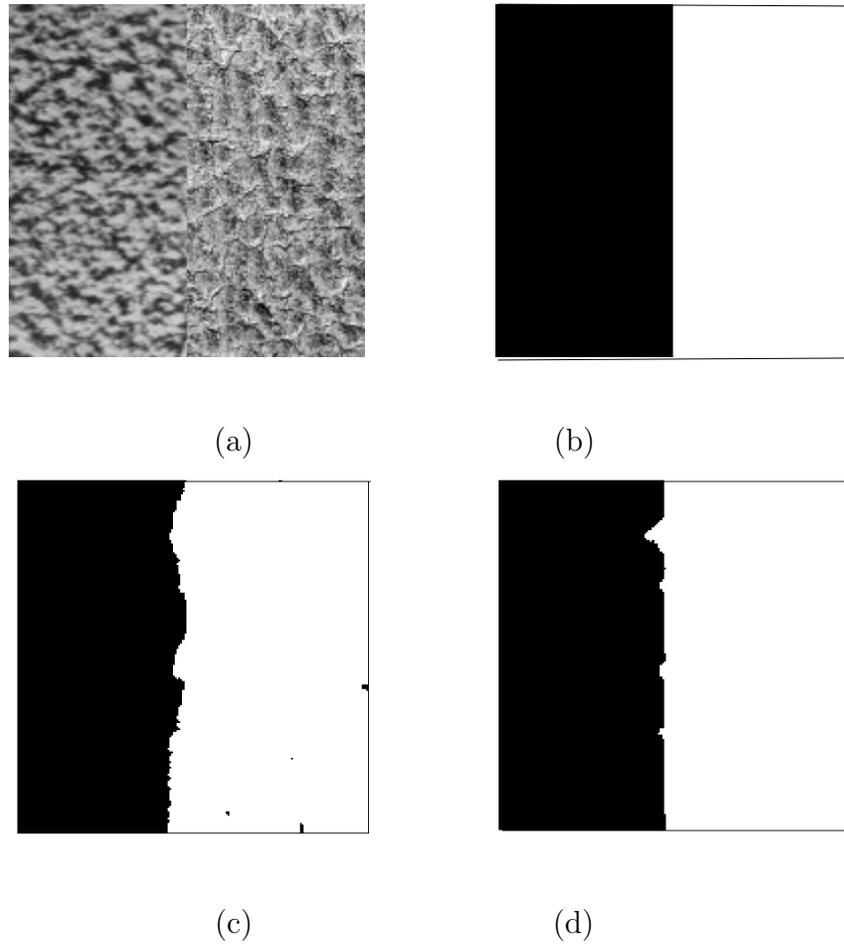


Figure 5.16: Segmentation of Brodatz texture image ( $n = 16$ ). (a) Original image. (b) True segmentation. (c) *GLCP* result. (d) *MRF* result.

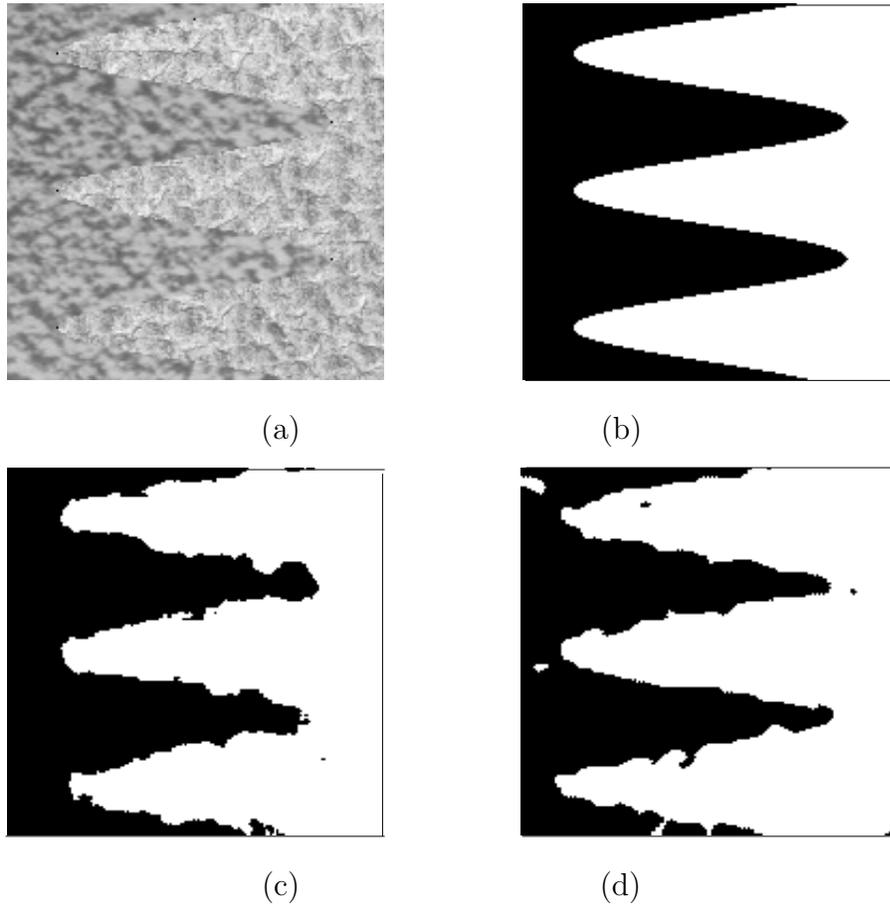


Figure 5.17: Segmentation of Brodatz texture image ( $n = 16$ ). (a) Original image. (b) True segmentation. (c) *GLCP* result. (d) *MRF* result.

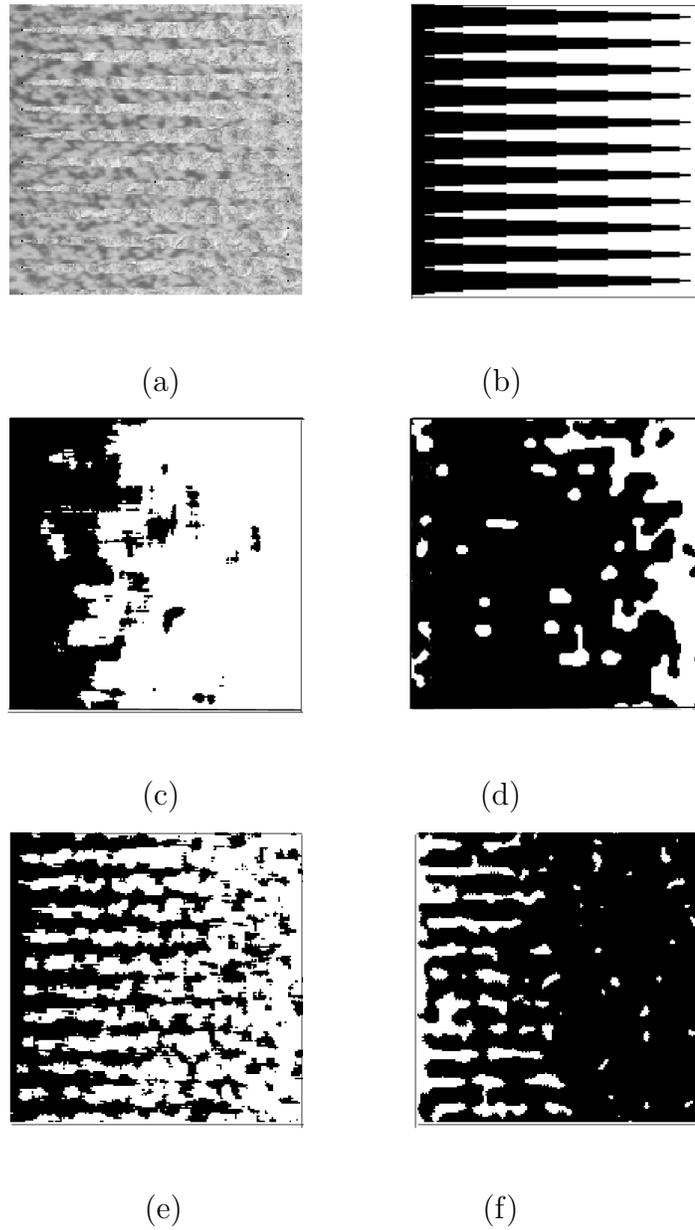


Figure 5.18: Segmentation of Brodatz texture image. (a) Original image. (b) True segmentation. (c) *GLCP* result ( $n = 16$ ). (d) *MRF* result ( $n = 16$ ). (e) *GLCP* result ( $n = 8$ ). (f) *MRF* result ( $n = 8$ ).

## 5.5 Image segmentation Results

In this section, the *GLCP* and *MRF* methods will be applied to synthetic, Brodatz and SAR sea ice images. Since the synthetic and Brodatz images are generated with different artificial boundaries, they can serve as the examples to demonstrate the theoretical concerns described in this thesis. The segmentation result of SAR sea ice image are demonstrated and discussed finally.

The first image is a sample of synthetic image Fig. 5.19 a. It contains two synthetic textures with a sinusoidal boundary of one period. The most complicated texture distribution of this image happens in the middle of the image, where given a window, two textures could be included with complicated boundary. Using a  $16 \times 16$  window, the *GLCP* has a much better segmentation result (c) than *MRF* (d). The *GLCP* can distinguish the two texture regions and also the exact boundary between the textures. However, the  $16 \times 16$  window size is not suitable for the *MRF* method, which can not separate the two texture regions. Using  $32 \times 32$  window size, both methods generate a better result than their  $16 \times 16$  window size results. The *GLCP* method has an improved segmentation in the texture regions, but the segmentation of the boundary area in the middle of the image is not as good as  $16 \times 16$  window size. The improvement of the *MRF* method is noticeable. Observing the segmentation result, the *MRF* method almost can distinguish the true boundary between the two textures.

The next segmentation examples will use Brodatz images. Compared with the synthetic textures image, more texture types will be involved as well as more complicated boundaries between textures. The first Brodatz image contains three Brodatz textures Fig. 5.20 a. The segmentation results using  $16 \times 16$  and  $32 \times 32$  window sizes of the *GLCP* and *MRF* methods are given in figure c, d, e and f. Basically,

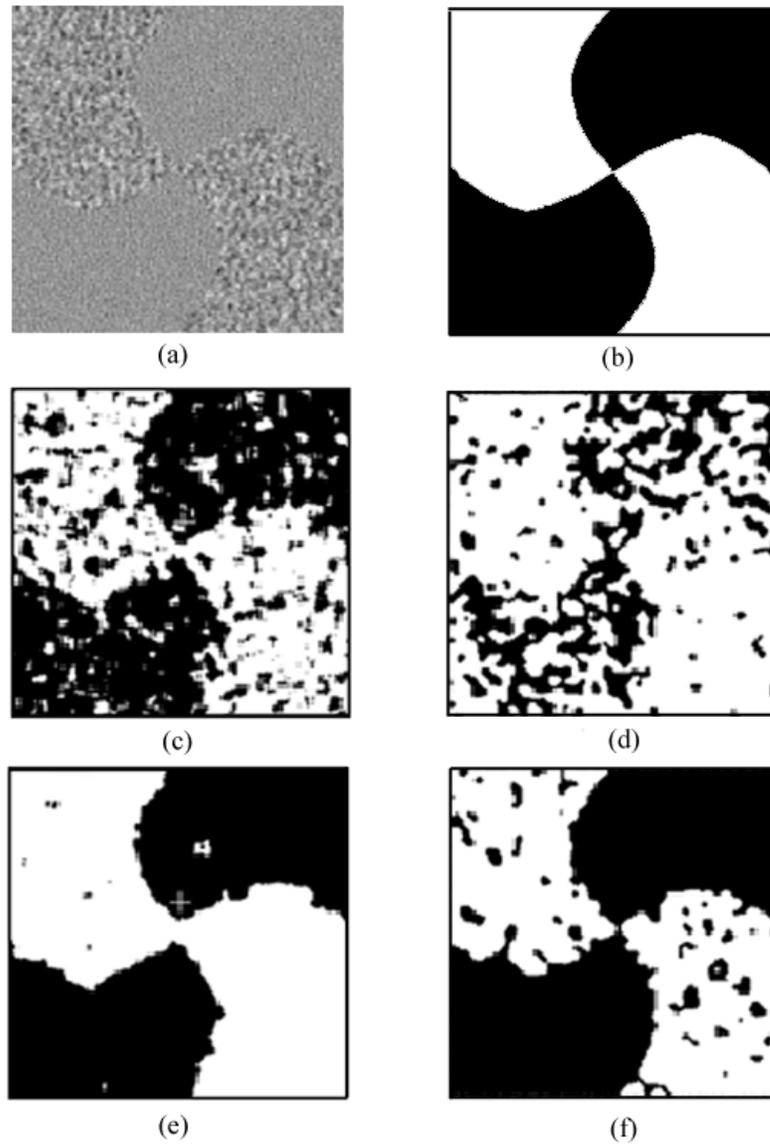


Figure 5.19: Segmentation of Synthetic texture image. (a) Original image. (b) True segmentation. (c) *GLCP* result ( $n = 16$ ). (d) *MRF* result ( $n = 16$ ). (e) *GLCP* result ( $n = 32$ ). (f) *MRF* result ( $n = 32$ ).

both methods are able to segment the image. Observing these results, some similarities exist with the synthetic image results. First, for the *GLCP* method, a  $16 \times 16$  window has a better segmentation along the boundary areas, whereas,  $32 \times 32$  has a tendency to confuse raffia texture (middle) with wood texture (top). Second, the *MRF* method does not produce a correct segmentation along the boundary area using  $16 \times 16$  window size, but with a  $32 \times 32$  window size, the segmentation result is improved considerably. For this image,  $16 \times 16$  window size is suitable for the *GLCP* method, and  $32 \times 32$  is good for the *MRF* method. In fact, if not considering the boundary area, both window sizes work well in the *GLCP* method. So, using the *GLCP* method, in order to obtain a correct segmentation in the boundary areas, among several possible selections, the window size should be as small as possible. Whereas, for the *MRF* method, the small window can not supply enough information to correctly estimate the texture models.

The second Brodatz texture image contain five Brodatz textures Fig. 5.21 a. Some of them (pigskin, raffia, water) have different textural resolutions, and the water, wood and raffia shows stronger texture pattern compared with the grass and pigskin textures. The five Brodatz textures with different patch size are mosaiced together with straight vertical and horizontal boundaries between each other. The  $16 \times 16$  window size does not work well for both methods. The *GLCP* method can not distinguish the raffia and pigskin textures but separate the wood texture into two classes. The *MRF* can not distinguish water and raffia textures. But with  $16 \times 16$  window size, both methods generate a basically correct boundaries between textures that can be separated. Using  $32 \times 32$  window size, again the *GLCP* method can not distinguish the raffia and pigskin, but the wood texture is identified correctly. The big window make the algorithm mis-classify the boundary region around certain textures as a separate class, and the two small texture patches

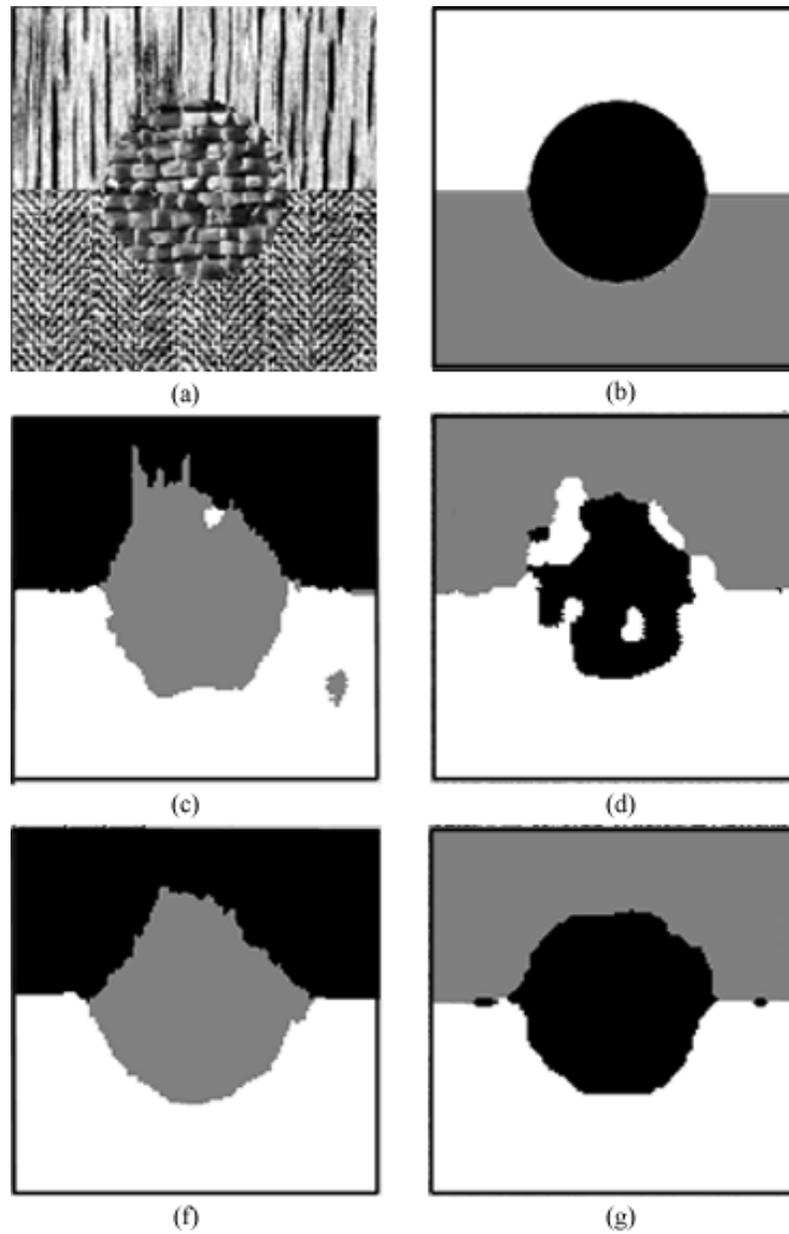


Figure 5.20: Segmentation of Brodatz texture image. (a) Original image. (b) True segmentation. (c) *GLCP* result ( $n = 16$ ). (d) *MRF* result ( $n = 16$ ). (e) *GLCP* result ( $n = 32$ ). (f) *MRF* result ( $n = 32$ ).

wood and raffia in the middle of the image are also grouped into the boundary class. Comparing the two segmentation images using *GLCP* method, one can see that the segmentation result using  $16 \times 16$  window size is better than using  $32 \times 32$  window size. Different situation happens in the *MRF* method, the better result comes from  $32 \times 32$  window size again as the previous images. The algorithm can separate all the texture types basically, and the boundary locations are very close to the true ones.

The last Brodatz image Fig. 5.22 a contains four textures separated with each other by sinusoidal instead of straight boundaries. The segmentation results are similar as Fig. 5.20. For both window size  $16 \times 16$  and  $32 \times 32$ , the *GLCP* method again can not separate pigskin and raffia textures and suffers in the boundary areas. *MRF* method has a better segmentation result using 32 window size.

Observing the segmentation results, if the textures can be separated by two methods, most of the time, the *MRF* generates better segmentation and boundaries than the *GLCP* method. The latter suffers from distinguishing the pixels along the boundary areas. The reason can be explained as follows. As mentioned in Chapter four, people's prior knowledge about the images can be involved into the segmentation process by probably designing the prior image models for the intensity and the label image. Based on the label image model, given a pixel, the most probable class label of this pixel is the one that most of its neighborhood pixels take. As an example of how the label image model helps to improve the segmentation results during each deterministic iteration mentioned in *ICM* algorithm in Section 4.4.4, Fig. 5.23 shows the segmentation results after one, five and fifteen iterations. One can see that great changes happened between the fifteen iterations. After the first iteration, it is hard to image that the algorithm could finally segment the image. But after five iterations, the right result basically appeared with some "noise"

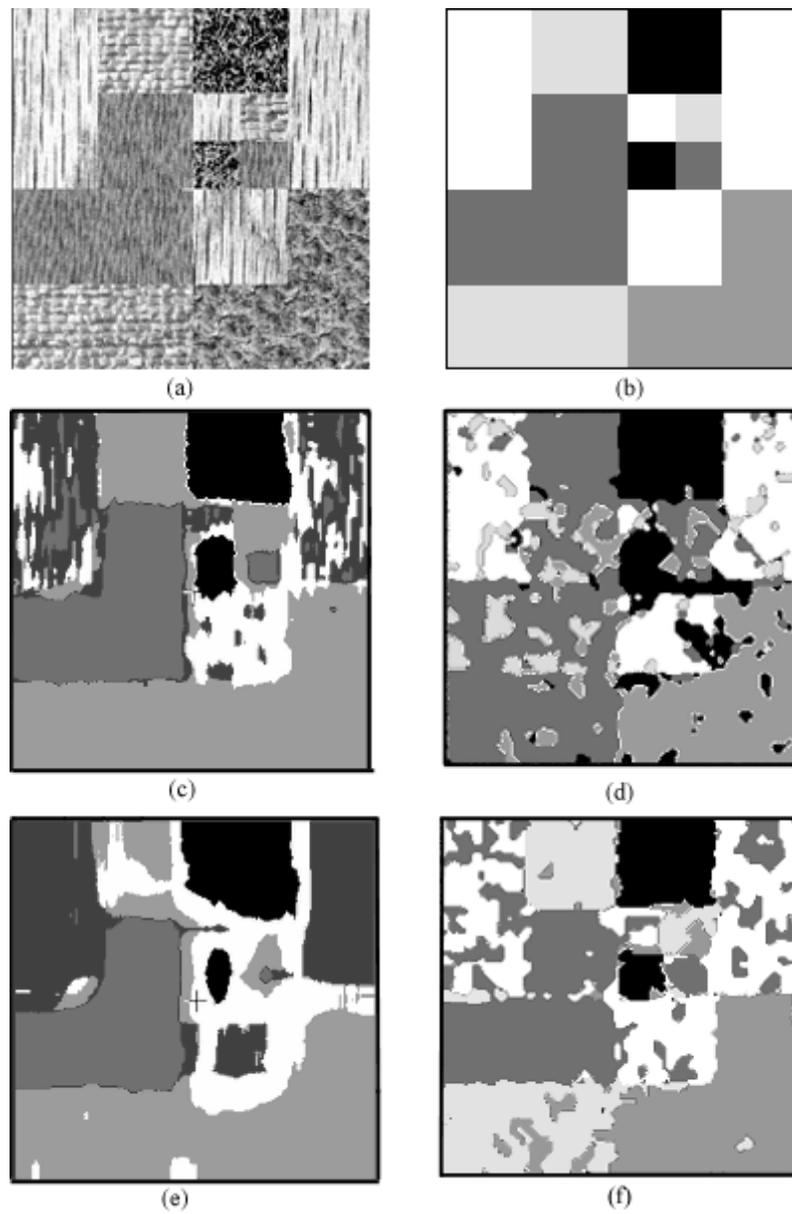


Figure 5.21: Segmentation of Brodatz texture image. (a) Original image. (b) True segmentation. (c) *GLCP* result ( $n = 16$ ). (d) *MRF* result ( $n = 16$ ). (e) *GLCP* result ( $n = 32$ ). (f) *MRF* result ( $n = 32$ ).

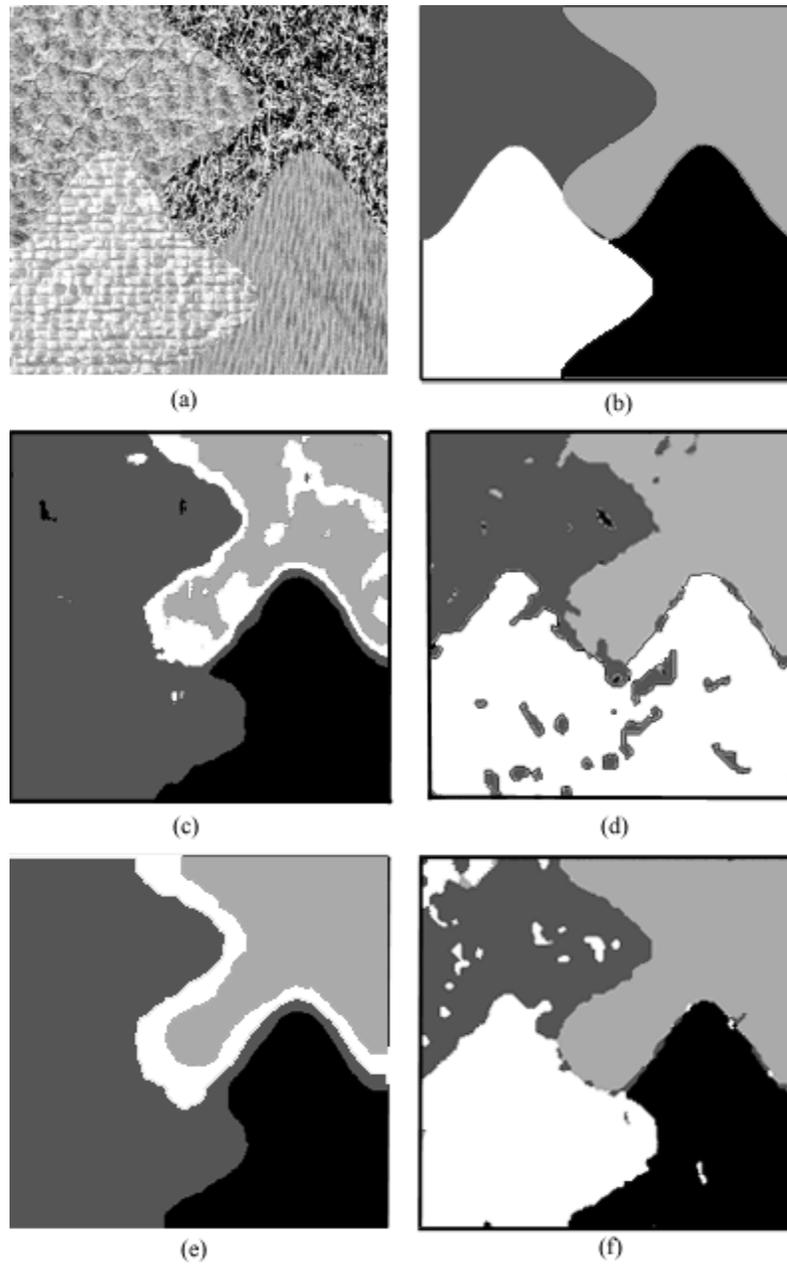


Figure 5.22: Segmentation of Brodatz texture image. (a) Original image. (b) True segmentation. (c) *GLCP* result ( $n = 16$ ). (d) *MRF* result ( $n = 16$ ). (e) *GLCP* result ( $n = 32$ ). (f) *MRF* result ( $n = 32$ ).

distributed in each texture type, and after fifteen iterations, the segmentation image is quite similar to the final result shown in Fig. 5.22 f, which was generated after thirty iterations. The iteration process can be regarded as a kind of correcting process guided by the prior models. Based on the segmentation result after the first iteration, the *ICM* algorithm will gradually update the pixel labels which are most likely to happen based on the image models. Given a region, if more than half of the pixels in this region take the right labels, the algorithm will finally make a correct segmentation. But if the first iteration generates a wrong initial segmentation, the segmentation result won't be correct.

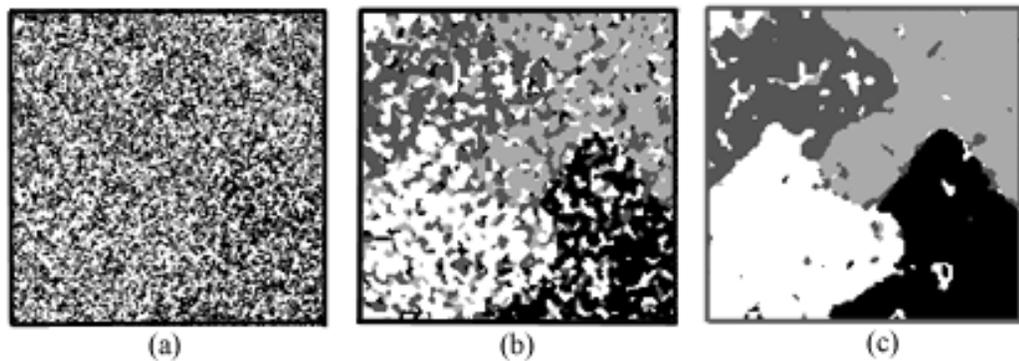


Figure 5.23: *MRF* Mid-segmentation results of Fig. 5.22. (a) After one iteration. (b) After five iterations. (c) After fifteen iterations.

From the segmentation results using different window sizes, one can see that both methods prefer larger window size to obtain a robust estimation. But larger window size may cause segmentation problem using the *GLCP* method in the boundary area, i.e., the true boundary between textures may be blurred, and sometimes, the pixels along the boundary areas could be distinguished as another texture class. To overcome this boundary problem, the window size should be as small

as possible for the *GLCP* method. Using the *MRF* method, small window size may ruin the texture model estimation and distinguish ability, from this point, the window size should be as big as possible. But given an image with complicated boundaries all over the image, large window size could also damage the estimated models based on the third research question. General speaking, the *MRF* has a more restricted limitation on the window size than the *GLCP* method. Table 5.4 shows the segmentation error of each test image.

Segmentation error of the test images				
	GLCP		MRF	
	$16 \times 16$	$32 \times 32$	$16 \times 16$	$32 \times 32$
Fig. 5.19	15.83 %	3.53 %	25.37 %	8.05 %
Fig. 5.20	4.29 %	4.08 %	8.83 %	3.03 %
Fig. 5.21	29.42 %	29.87 %	27.60 %	14.90 %
Fig. 5.22	24.16 %	21.82 %	19.89 %	3.60 %

Table 5.4: Segmentation error of the test images in section 5.5.

The last two segmentation results are from SAR sea ice images. Sea ice segmentation has been a challenging research topic for years, and there are few publications about SAR sea ice image segmentation using texture methods, especially using automated approaches. As Clausi mentioned [13], the unsupervised segmentation should be more suitable for SAR sea ice type identification than supervised segmentation. As introduced in Chapter 2, many different variables may influence the appearance of sea ice in a SAR image. The same ice type may not be identical from image to image, and even within the same image. In this case, the selected training samples in a supervised algorithm may not be sufficient to include all the class variability throughout the image.

The sea ice image shown in Fig. 5.24 is obtained from Barber *et al* [16]. The image is extracted from X-band, HH polarization, STAR-1 with a six meters resolution. Two ice types are included in the image: first year ice and multi-year ice. Because the sea ice texture has a finer resolution than Brodatz textures, the window size used in the *GLCP* method is  $13 \times 13$ . An  $16 \times 16$  window size is used in the *MRF* method considering the complicated boundaries and many small patches of multi-year ice in the image. The segmentation results from the *GLCP* and *MRF* methods are given in Fig. 5.25 and Fig. 5.27. Both methods can separate the two ice types. But comparing the two results, one can see the difference along the boundary area. As in the previous examples, the boundaries generated from the *MRF* method are more accurate than the *GLCP* method, the *GLCP* method tends to increase the distribution of multi-year ice in the image. This observation can be demonstrated more clearly using the segmentation edge map shown in Fig. 5.26 and Fig. 5.28. The edges in these two figures are extracted from the segmentation results. Visually observing the boundary locations in these two figures, one can see that the edges in Fig. 5.28 present more appropriately the boundary locations of different ice types. Barber *et al* used this image in supervised segmentation using statistical based texture analysis methods. Based on the above explanation about the shortcoming of supervised classification for sea ice image, it is not surprise to know from their paper that the classification accuracy of the training image was always higher than the test image.

The second sea ice image shown in Fig. 5.29 is obtained from the Canadian Ice Service [11]. This C-band HH Radarsat ScanSAR image was obtained on October 13, 1997 on Beaufort Sea. The image was captured at incidence angles of between 20 and 49 degrees with a pixel spacing of 100 meters. Multi-year ice, new ice and grey ice are contained in the image. The new ice which is in a dark tone can be

found in the middle of the image. The grey ice with a grey colour can be found in the top and bottom regions of the image. The multi-year ice with a brighter colour than grey ice, scatters among grey ice, can be located in the top region of the image, and a small amount can also be found in the bottom region. The *GLCP* method and *MRF* methods generate different segmentation results, and both methods do not perform as well as the first ice image. Generally speaking, the *GLCP* method Fig. 5.30 generates a better segmentation image than *MRF* method Fig. 5.32. For the new ice segmentation, Fig. 5.32 tends to confuse it with grey ice in the top part of the image. In distinguishing multi-year ice and grey ice, the two methods have obviously different tendency. The *MRF* method is too sensitive to the texture changes and the segmentation result looks more broken than *GLCP* method, which can also be demonstrated by the segmentation edge maps shown in Fig. 5.31 and 5.33 respectively. however, the *GLCP* method tends to confuse the multi-year ice with grey ice on the top region of the image. It enlarges the grey ice scale more than it should be.

Comparing the two ice images, one can notice that the pattern of each ice type in Fig. 5.24 is more strongly defined than the ice's pattern in Fig. 5.29. Also the textures pattern of first year and multi-year ice in Fig. 5.24 have a bigger difference than those in Fig. 5.29. Finally, the multi-year ice and grey ice in the latter image are mixed together most of the time. All of these could result a poorer model estimation and make the second image more difficult to segment using the *MRF* method.

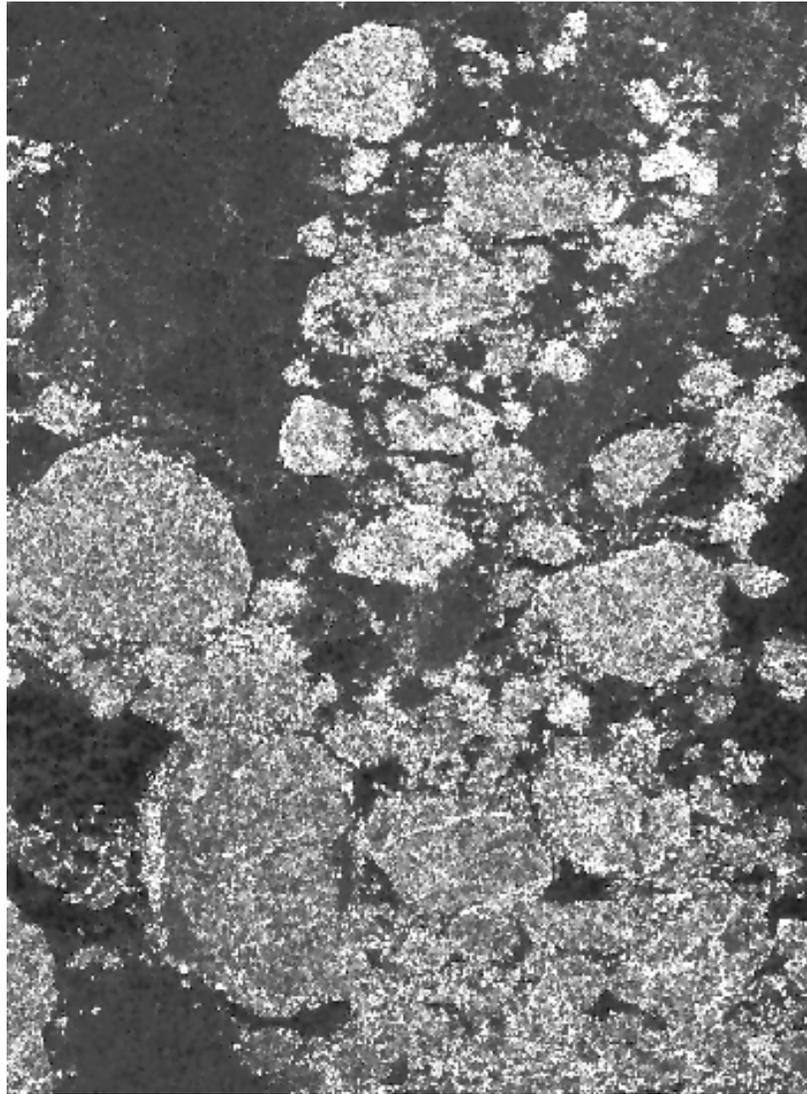


Figure 5.24: Aerial SAR image (sub-image of Figure 2(b) in [16])



Figure 5.25: *GLCP* segmentation result of Fig. 5.24

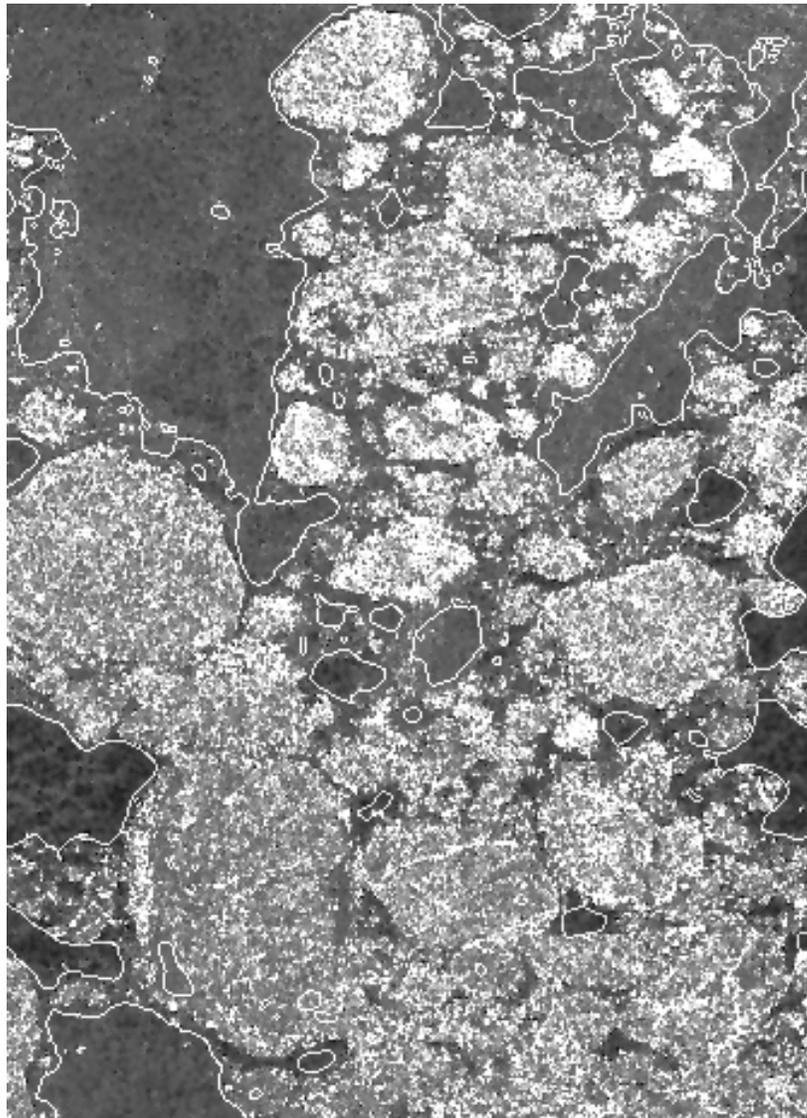


Figure 5.26: *GLCP* segmentation result edge map of Fig. 5.25.

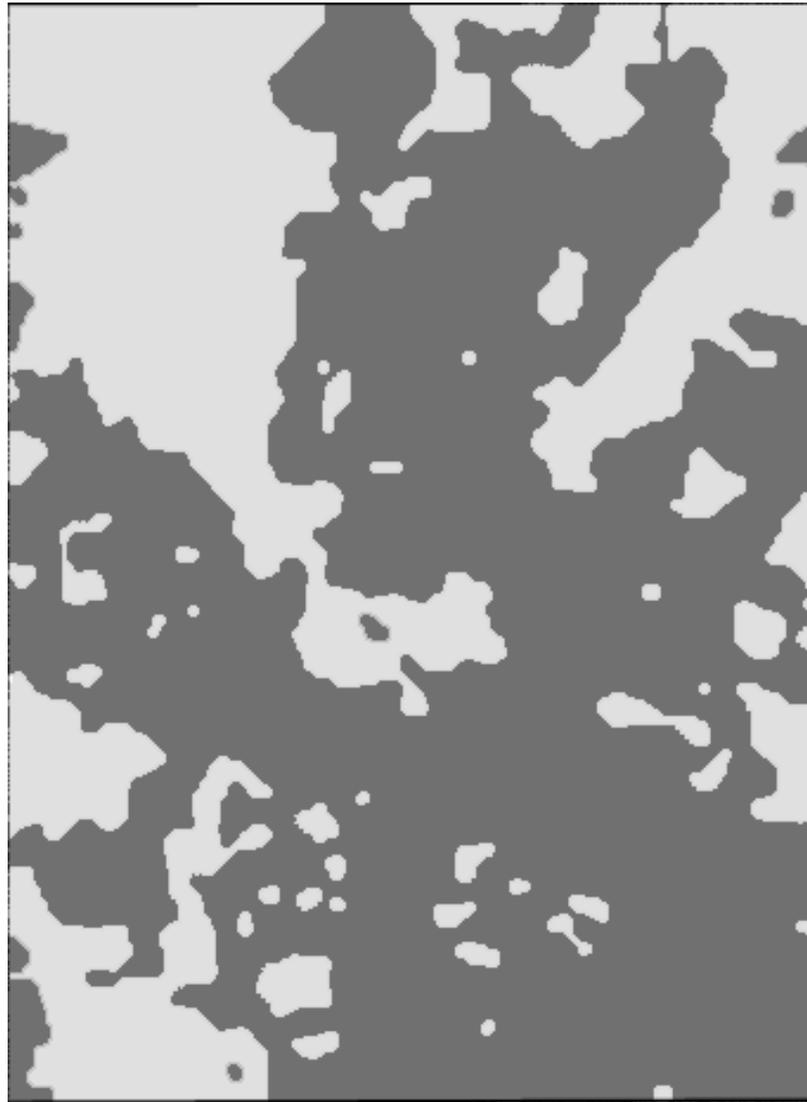


Figure 5.27: *MRF* segmentation result of Fig. 5.24

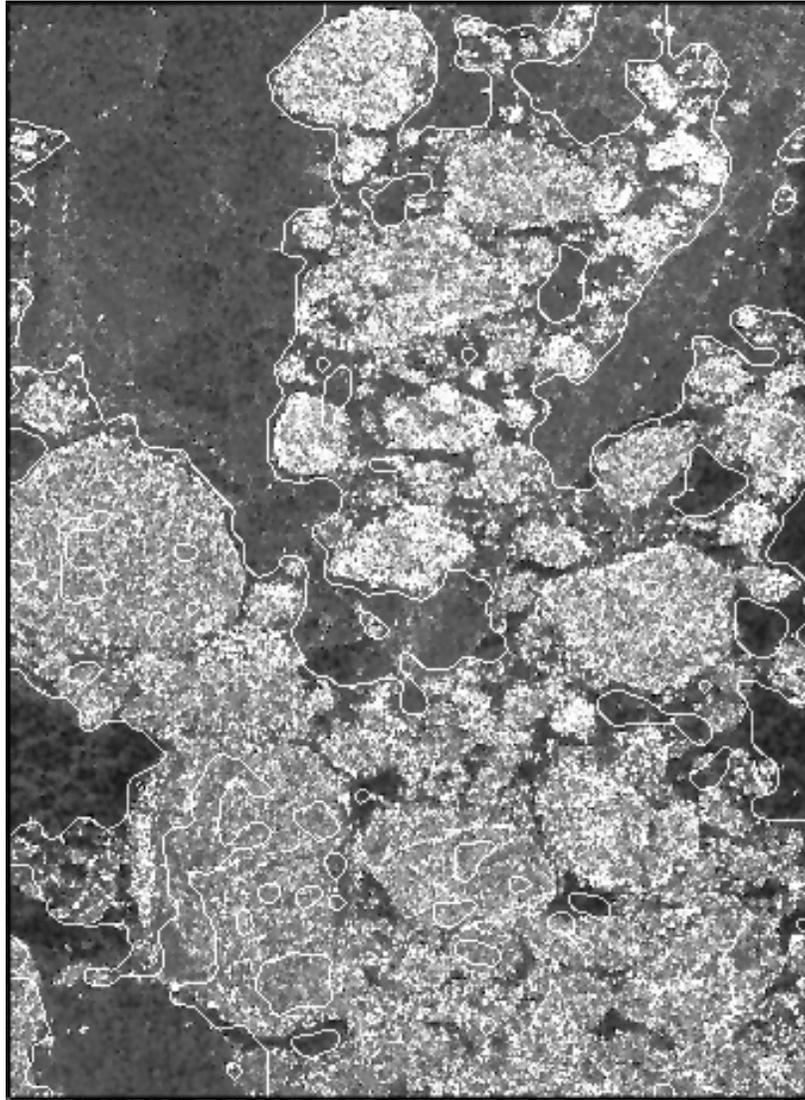


Figure 5.28: *MRF* segmentation result edge map of Fig. 5.27.

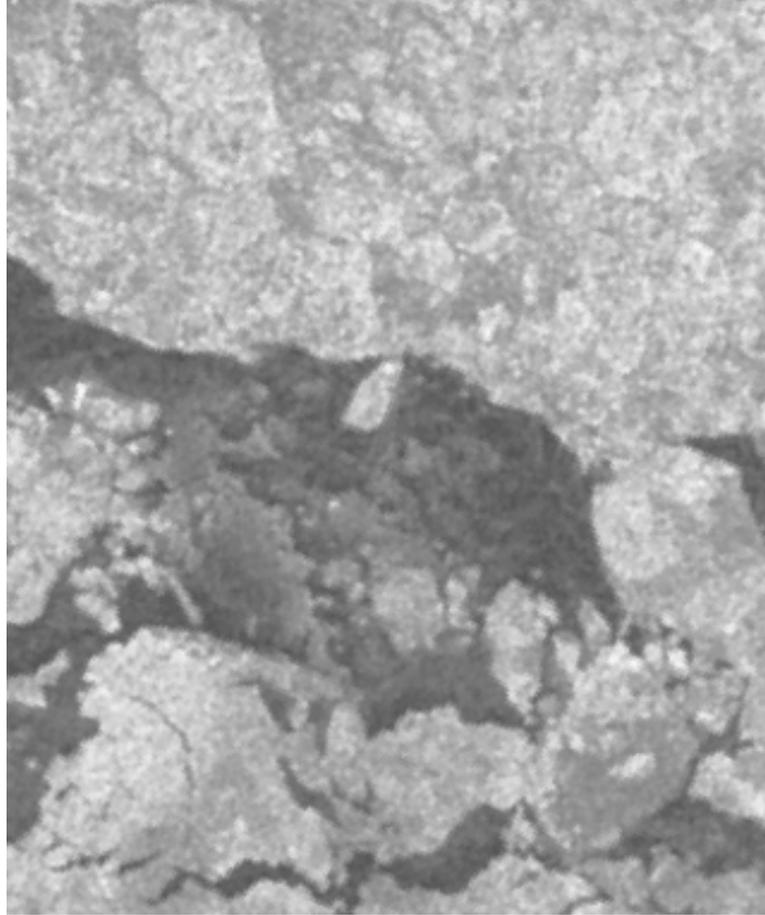


Figure 5.29: Radarsat SAR sea ice image obtained from Canadian Ice Service [11].

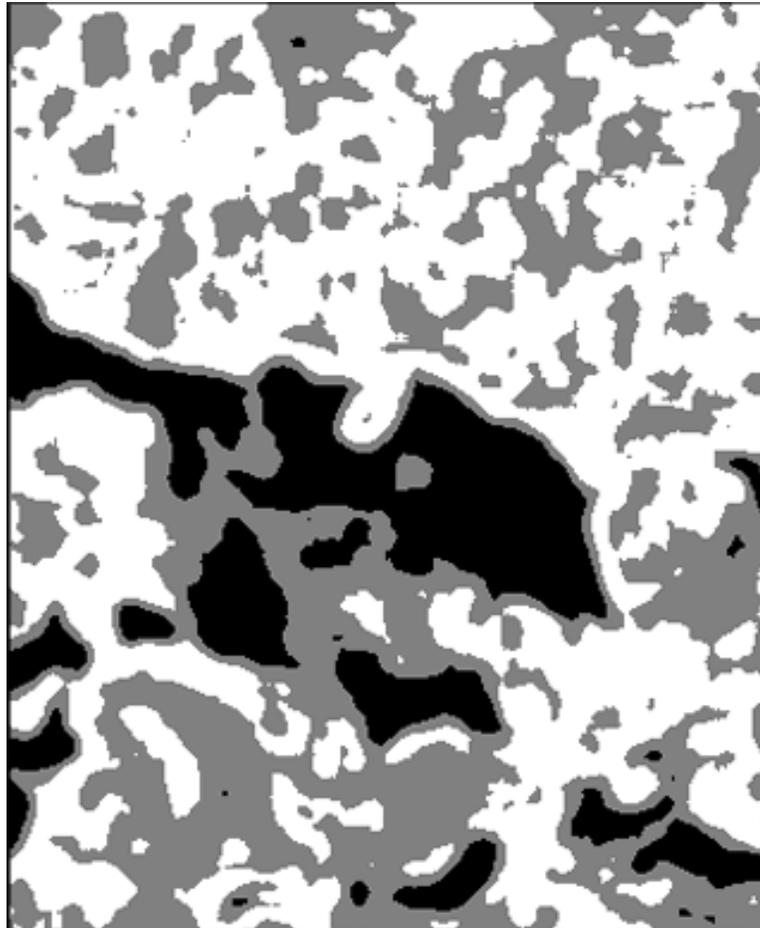


Figure 5.30: *GLCP* segmentation result of Fig. 5.29

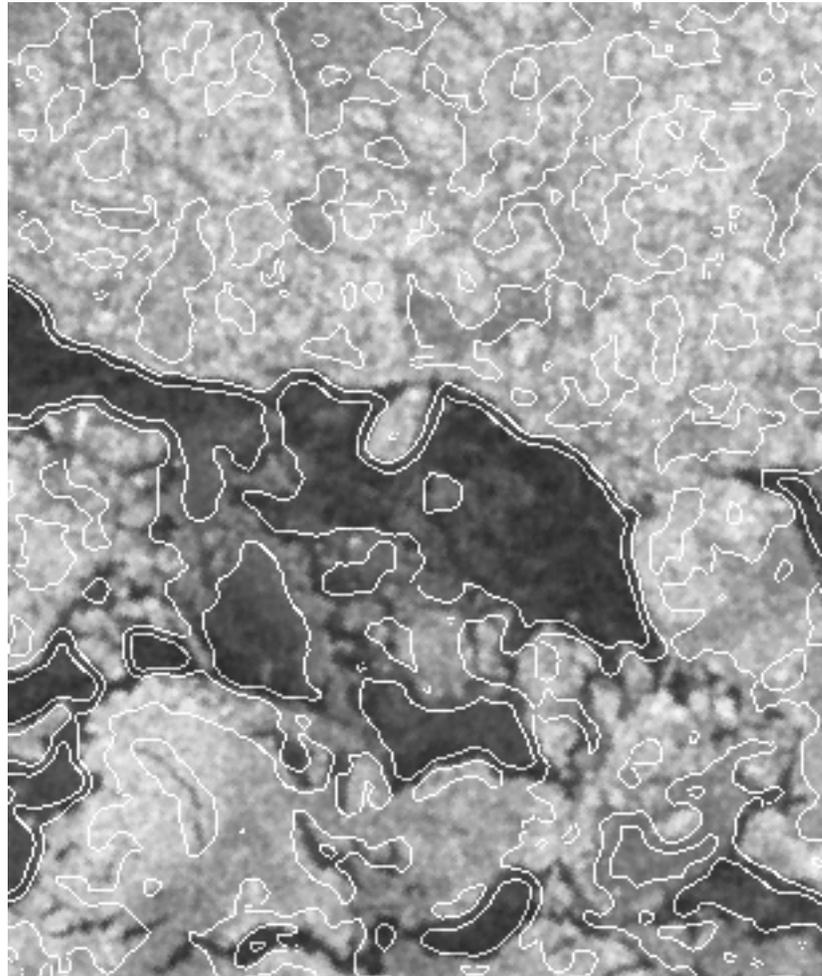


Figure 5.31: *GLCP* segmentation result edge map of Fig. 5.30.

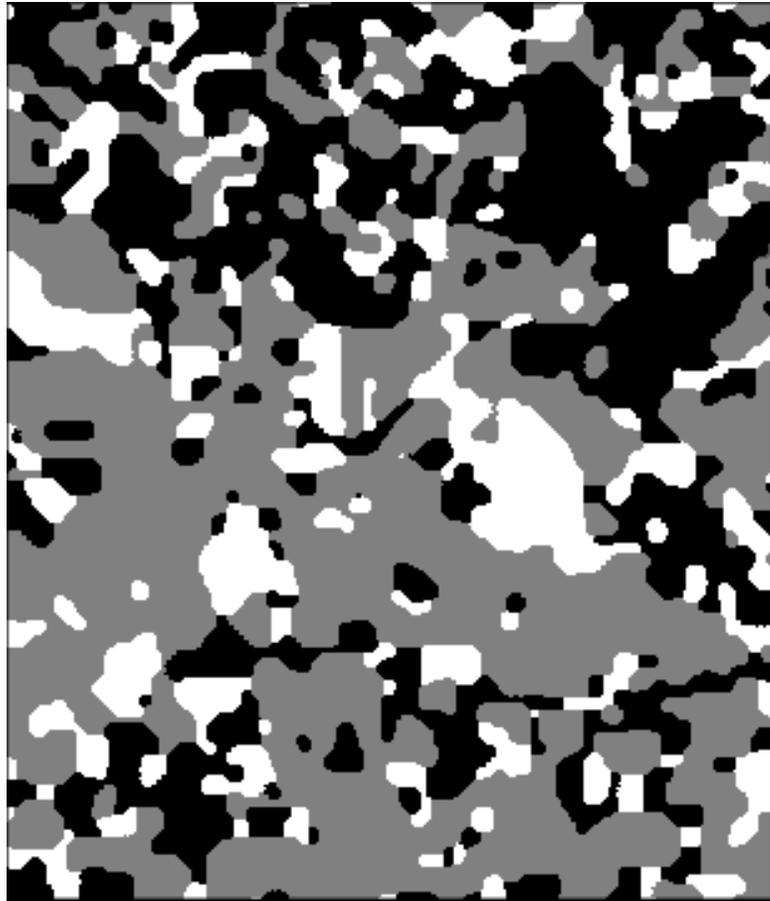


Figure 5.32: *MRF* segmentation result of Fig. 5.29

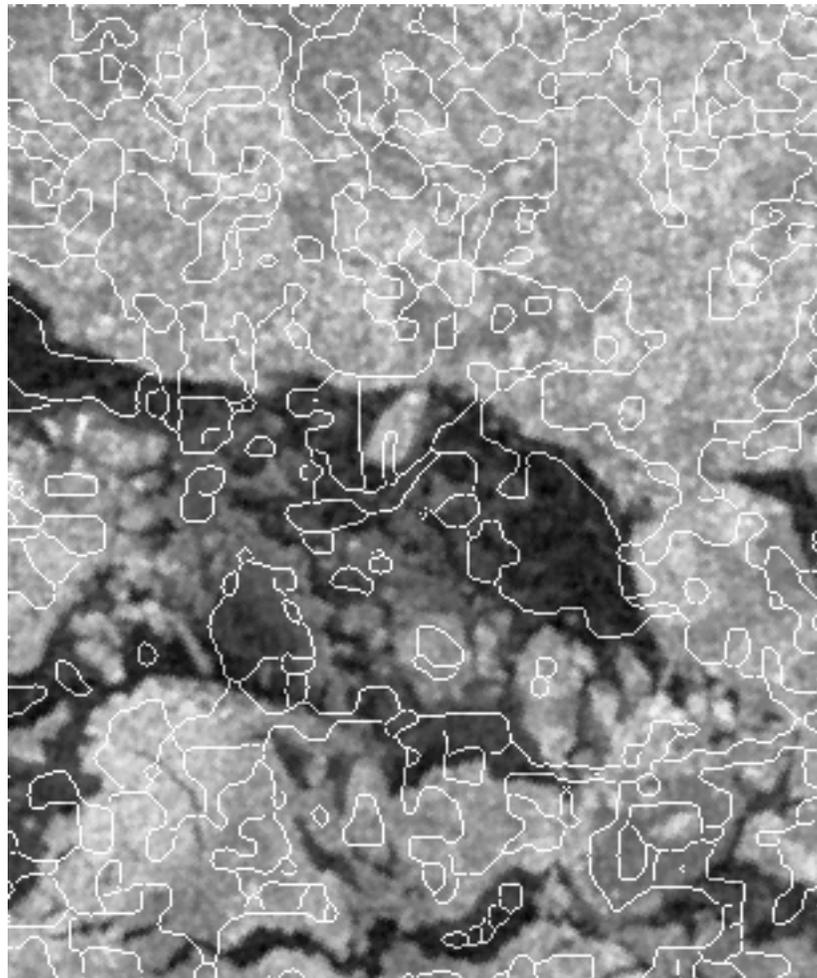


Figure 5.33: *MRF* segmentation result edge map of Fig. 5.32.

# Chapter 6

## Conclusions and Future Work

### 6.1 Conclusions

The topic of this thesis is to focus on texture image analysis and segmentation. The theory of the statistical-based *GLCP* and model-based *MRF* texture analysis techniques are introduced, the texture segmentation algorithms based on them are developed. For the comparison purposes, three research questions related to parameter estimations and the respective texture discrimination ability of the *GLCP* texture features and *GMRF* models are raised. A series of experiments are conducted and the results are investigated and analyzed. Finally, to illustrate the segmentation algorithms, three kinds of texture images, the synthetic, Brodatz, and SAR sea ice images, are tested.

Several conclusions can be obtained from the above work. These are list below.

- The *GLCP* and *MRF* are two different texture analysis techniques. The *GLCP* method does not have any assumptions on the analyzed textures. The texture statistics used in the *GLCP* method are the measurements that describe

the qualitative aspect of the texture. Whereas, the *MRF* regards a texture as a sample generated from a stochastic process, and the relationship of a pixel with its neighborhood can be described by a model. By controlling the parameters of the *MRF* model, one can generate textures with different structures. But there is no comparable methods to synthesize texture based on the *GLCP* texture features. Because not all textures in nature follow a model, the *GLCP* method could be applicable to more texture types than the *MRF* method.

- For both methods, the window size is a very important element in the segmentation process. As a function of decreasing window size, the standard deviation of the estimated *GMRF* model parameter deteriorates faster than *GLCP* texture features. Also the texture distinguishing ability of the *GMRF* model become inferior to the *GLCP* texture features. Because of the relative stability of the estimated *GLCP* texture features across different window sizes, the *GLCP* method performance is better with a smaller window size than the *MRF* method. The latter has more difficulties to obtain a robust model estimation using a small window size.
- Given a multi-texture window, each estimated *GMRF* model parameter from this window is a weighted linear combination of the corresponding model parameter of each texture. The same conclusion applies to the *GLCP* texture features. In the unsupervised segmentation process, when a window moves across the boundary areas between textures, the model parameters estimated from the multi-texture window dose not represent either texture. The poor estimations may damage the segmentation for both methods. For the *GLCP* method, the poor estimation will affect the segmentation in the boundary

areas, whereas for the *MRF* method, the poor estimation will affect the segmentation all over the image. One way to decrease this damage is to decrease the window size, so fewer windows will contain multi-textures. This solution works well for the *GCLP* method, but not for the *MRF* method based on the above conclusion.

- Observing the deterministic relaxation process, the label image model used in the current *MRF* segmentation algorithm prefers big texture patches. The small patches of textures scattering among other textures can be easily swallowed by its surrounding textures. This problem can not be easily solved because the algorithm can not distinguish which small patches should be removed because of the wrong segmentation and which one should be kept because there really exists a small patch of texture in the image.
- Base on the segmentation results, the *MRF* method has a more restricted limitation on the window size than the *GLCP* method. Useful guidelines for selecting the suitable window size for each method is as follows. For the *GLCP* method, the window size should be as small as possible because this is appropriate for locating the true boundary. For the *MRF* method, the window size should be as large as possible to obtain a robust model estimation.
- Besides the window size, there are also several other important elements in each method which could heavily effect the segmentation results. The quantization, displacement, orientation and texture statistics are important parameters in the *GLCP* method. The texture model, model size, and energy window size are the significant concerns in the *MRF* method.

## 6.2 Future Work

To improve the performance of the *GLCP* and *MRF* segmentation algorithm, several directions could be explored.

- As mentioned in section 6.1 to obtain a robust texture model estimation in *MRF* method, the window size should be as large as possible. But increasing the window size will cause the multi-texture window problem in the boundary regions, which will also damage the estimation. An intelligent window selection scheme could be considered so that the multi-texture windows can be excluded in the calculation. This will greatly increase the accuracy of the estimated texture models. For example, the estimated parameters from a window can be regarded as a point in the multi-dimensional feature space, if the point is beyond certain distance away from the cluster center, the window will not be included in the parameter estimation.
- Although there are many *MRF* texture classification algorithms published in the literature, the available image models are quite limited. Gauss-model is most commonly used for the intensity image. Only the pair-clique pixel relationship is considered in this model. It is necessary to investigate the possibility of using other texture models for sea ice image based on the further analysis and understanding on the sea ice texture structure.
- Different initial methods for label image should be explored for increasing the segmentation accuracy. The current initial label image is a uniformly distributed random image. The initial label image could be the preliminary segmentation results from some high speed segmentation algorithms, e.g. the

segmentation result using the first order methods in statistical texture analysis.

- An intelligent segmentation algorithm are also needed to developed for the *GLCP* method other than the currently used k-means approach. The main effort should be focused on the mis-classification along the boundary area.

# Appendix A

## Radarsat Image and Sea Ice Types

Launched in November 1995, RADARSAT is a sophisticated Earth observation (EO) satellite developed by Canada to monitor environmental change and the planet's natural resources [48].

### A.1 Image Modes of Radarsat

The orbit of Radarsat is shown in figure A.1. It can apply 25 image modes, each image mode corresponds one beam position shown in figure 2.2. Radarsat is a side-looking imaging system, and the captured image is on the right side of its track.

### A.2 Processing Levels of Radarsat Image

For each Radarsat beam mode, image products are processed to different levels of geometric accuracy and radiometric calibration. It is very important to know

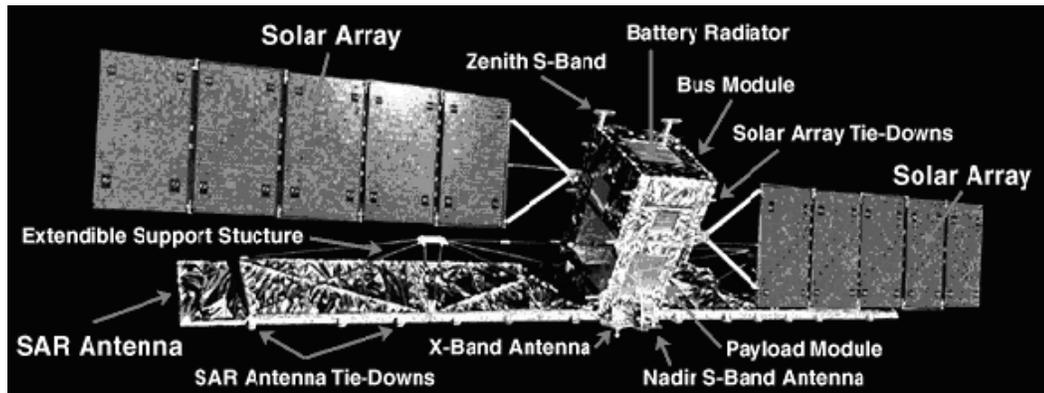


Figure A.1: Radarsat-1 (adapted from [48]).

this before one orders some data for a special research. There are seven types of Radarsat image products: signal data, single look complex, path image, path image plus, map image, precision map image, and orthorectified. The first two types are slant range image, and Path Image and Path Image Plus have already been converted to ground range image, the direction of the image is the same as the direction of the satellite track. The last three modes are already corrected and geocoded using the ground control points.

### A.3 Ice Types

Table A.3 lists the ice types commonly used in the preparation of the Canadian Ice Service products and publications [11].

Ice Types	Description
New ice	A general term for recently formed ice which includes frazil ice, grease ice, nilas, slush and shuga. These types of ice are composed of ice crystals which are only weakly frozen together (if at all) and have a definite form only while they are afloat.
Grey ice	Young ice 10-15 cm thick. Less elastic than nilas and breaks on swell. Usually rafts under pressure.
Grey-white ice	Young ice 15-30 cm thick. Under pressure it is more likely to ridge than to raft.
Thin-first year	First-year ice of not more than one winter's growth, 30-70 cm thick.
Medium first-year	First-year, ice 70-120 cm thick year.
Thick first-year	First-year ice over 120 cm thick.
Old year	Sea ice which has survived at least one summer's melt. Topographic features generally are smoother than first-year ice. May be subdivided into second-year ice and multi-year ice. Second-year ice: Old ice which has survived only one summer's melt. Multi-year ice: Old ice which has survived at least two summer's melt.

Table A.1: Sea ice types

# Bibliography

- [1] R. D. Abreu. Radar sea ice signatures: An operational primer. *Proceedings of a workshop on mapping and archiving of sea ice data-the expanding role of Radar*, pages 85–94, 2000.
- [2] D. G. Barber and E. F. LeDrew. Sar sea ice discrimination using texture statistics: A multivariate approach. *Photogrammetric Engineering and Remote Sensing*, 57(4):385–395, 1991.
- [3] S. A. Barker and P. J. W. Rayner. Unsupervised image segmentation using markov random field models. *Pattern Recognition*, 33(2000):587–602, 1999.
- [4] J. Besag. On the statistical analysis of dirty pictures. *J. Royal Statist. Soc. B*, 48:259–302, 1986.
- [5] J. E. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical society B*, 36(2):192–236, 1974.
- [6] Editorial Board. *Microwave Remote Sensing of Sea Ice: Preface*. Prentice Hall, Inc., 1992.

- [7] B. A. Burns and E. S. Kasischke. Extraction of texture information from sar data: Application to ice and geological mapping. *International Symposium on Remote Sensing of Environment, Second Thematic Conference*, pages 861–868, 1982.
- [8] K. R. Castleman. *Digital Image Processing*. Prentice Hall, Inc., 1996.
- [9] R. Chellappa. Two-dimensional discrete gaussian markov random field models for image processing. *Progress in Pattern Recognition 2*, pages 79–112, 1985.
- [10] R. Chellappa and S. Chatterjee. Classification of texture using gaussian markov random field. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(4):959–963, 1985.
- [11] CIS. <http://www.cis.ec.gc.ca/home.html>, 2001.
- [12] D. A. Clausi. Comparison and fusion of co-occurrence, Gabor, and MRF texture features for classification of SAR sea ice imagery. *Atmosphere-Ocean, in press*.
- [13] D. A. Clausi. *Texture Segmentation on SAR Ice Imagery*. PhD thesis, University of Waterloo, Canada, 1996.
- [14] G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):44–58, 1983.
- [15] CRYSYS. <http://www.crysys.uwaterloo.ca>, 2001.
- [16] D. G. Barber, M. E. Shokr, R. A. Fernandes, E. D. Soulis, D. G. Flett, and E. F. LeDrew. A comparison of second-order classifiers for sar sea ice distrimination. *Photogrammetric Engineering and Remote Sensing*, 59(9):1397–1408, 1993.

- [17] D. Haverkamp, L. K. Soh, and C. Tsatsoulis. A comprehensive, automated approach to determining sea ice thickness from sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 33:46–57, 1995.
- [18] H. Derin and W. S. Cole. Segmentation of textured images using gibbs random fields. *Computer Vision, Graphics and Image Processing*, 35:72–98, 1986.
- [19] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):39–55, 1987.
- [20] E. O. Lewis, C. E. Livingstone, C. Garrity, and H. R. Rossiter. *Remote Sensing of Sea Ice and Icebergs*. John Wiley Sons. Inc., 1994.
- [21] B. E. Goodison et al. *CRYSYS-Variability and Changes in the CRYospheric SYSTEM in Canada*. Atmospheric Environment Service, Environment Canada, 1999.
- [22] F. D. Carsey, R. G. Berry, and W. F. Weeks. *Microwave Remote Sensing of Sea Ice: Chapter 1*. American Geophysical Union, 1992.
- [23] P. Fieguth. *Multidimensional Signal Model and Estimation Course Notes*. Systems Design Engineering, University of Waterloo, ON, Canada, May 2000.
- [24] E. W. Montroll G. F. Newell. On the theory of the ising model of ferrogagnetism. *Rev. Modern Phys.*, 25:353–389, 1953.
- [25] M. M. Galloway. Texture classification using grey-level run lengths. *Computer Graphics and Image Processing*, (4):172–179, 1975.
- [26] L. Garand. Automated recognition of oceanic cloud patterns, i. methodology and application to cloud climatologu. *J. Climate*, 1:20–39, 1988.

- [27] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984.
- [28] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. *Proceedings of the International Congress of Mathematicians*, 1986.
- [29] G. L. Gimel'farb. *Image textures and Gibbs Random Fields*. Kluwer Academic Publishers, 1999.
- [30] S.-Y. Hsu. Texture-tone analysis for automated land use mapping. *Photogramm. Eng. Remote Sensing*, 44(11):1393–1404, 1978.
- [31] J. S. Weszka, C. R. Dyer, and A. Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6(4):269–285, 1976.
- [32] J. R. Jensen. *Introductory Digital Image Processing: A Remote Sensing Perspective*. Prentice Hall, Inc., 1986.
- [33] B. Julesz. Visual pattern discrimination. *IEEE Transactions on Information Theory*, IT-8:84–92, 1962.
- [34] R. L. Kashyap and R. Chellappa. Estimation and choice of neighbors in spatial-interaction models of images. *IEEE Transactions on Information Theory*, 29(1):61–72, 1983.
- [35] S. Krishnamachari and R. Chellappa. Multiresolution gauss-markov random field models for texture segmentation. *IEEE Transactions on Image Processing*, 6(2):251–267, Feb. 1997.

- [36] C. Tsatsoulis L. K. Soh. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2):780–795, 1999.
- [37] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.
- [38] B. S. Manjunath and R. Chellappa. Unsuperivsed texture segmentation using markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):478–482, 1991.
- [39] B. S. Manjunath, T. Simchony, and R. Chellappa. Stochastic and deterministic networks for texture segmentation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(6):1039–1049, 1990.
- [40] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):478–482, 1995.
- [41] D. K. Panjwani and G. Healey. Markov random field models for unsupervised segmentation of textured color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):478–482, Feb. 1991.
- [42] Q. A. Holmes, D. R. Nuesch, and R. A. Shuchman. Textural analysis and real-time classification of sea-ice types using digital sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 22(2):113–120, 1984.
- [43] R. A. Shuchman, C. C. Wackerman, A. L. Maffett, R. G. Onstott, and L. L. Sutherland. The discrimination of sea ice types using sar backscatter statistics. *International Geoscience and Remote Sensing Symposium*, 1:381–385, 1989.

- [44] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for images classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.
- [45] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. A Wiley-interscience Publication, 2001.
- [46] E. Hall R. Sutton. Texture measures for automatic classificaion of pulmonary disease. *IEEE Transactions on Computers*, C-21(7):667–676, May 1972.
- [47] C. A. Harlow R. W. Connors. Theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(3):204–222, May 1980.
- [48] Radarsat. <http://www.ccrs.nrcan.gc.ca/ccrs/tekrd/radarsat/rsate.html>, 2001.
- [49] A. Rosenfeld. *Markov Random Fields: Theory and Applications: Image Modeling during the 1980s: A Brief Overview*. New York: Academic, 1993.
- [50] F. F. Sabins. *Remote Sensing Principles and Interpretation*. W. H. Freeman and Company, New York, 1987.
- [51] M. R. Drinkwater W. F. Weeks. *Microwave Remote Sensing of Sea Ice, Charpter 2. Physical Properties of Sea Ice Relevant to Remote Sensing*. American Geophysical Union, 1992.
- [52] W. I. Rosenblum, C. Salvaggio, and J. R. Schott. Selection of optimal textural features for maximum likelihood image classification. Technical report, Rochester Institute of Techmology Center for Image Sciene, 1990.
- [53] J. A. Wiebe. Texture estimates of operational forestry parameters. Master’s thesis, University of Calgary, Canada, 1998.

- [54] G. Winkler. *Image analysis, random fields, and dynamic Monte Carlo methods : a mathematical introduction*. Berlin : Springer-Verlag, 1995.
- [55] C. S. Won and H. Derin. Unsupervised segmentation of noisy and textured images using markov random fields. *CVGIP: Graphical Models and Image Processing*, 54(4):308–328, July, 1992.
- [56] X. Descombes, M. Moctezuma, H. Maitre, and J. P. Rudant. Coastline detection by a markovian segmentation on sar images. *Signal Processing*, 55:123–132, 1996.
- [57] X. Descombes, R. D. Morris, J. Zerubia, and M. Berthod. Estimation of markov random field prior parameters using markov chain monte carlo maximum likelihood. *IEEE Transactions on Image Processing*, 8(7):954–962, 1999.