

ON PARAMETER ESTIMATION FOR A CLASS OF
MARKOV RANDOM FIELD IMAGE MODELS:

A Thesis

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Master Of Science

by

Sean Borman B.Sc. (Eng)

Ken D. Sauer, Director

Department of Electrical Engineering

Notre Dame, Indiana

April 1996

ON PARAMETER ESTIMATION FOR A CLASS OF
MARKOV RANDOM FIELD IMAGE MODELS:

Abstract

by

Sean Borman

Mathematical image models find common application as *a-priori* information in many image reconstruction and restoration problems. A broad class of models, Markov random fields, are well recognized for their convenient properties in these problems. The characteristic behavior of a MRF is determined by its set of model *parameters*. Appropriate choice or *estimation* of these parameters is necessary to ensure that the model is a faithful representation of the *a-priori* knowledge.

Estimation of the *clique weight parameters* of a particular class of MRF, the generalized Gaussian MRF is the primary topic investigated in this thesis. In the proposed approach, minimization of prediction error estimation (MPEE), we choose the clique weight parameters for which the penalized error between model predicted data and known realization data is a minimum. Procedures for minimization of this error are proposed and the performance of the MPEE method is evaluated.

Contents

LIST OF TABLES	v
LIST OF FIGURES	vii
ACKNOWLEDGEMENTS	x
1 INTRODUCTION	1
1.1 Quo Vadis ?	1
1.2 Thesis Organization	1
2 BACKGROUND	4
2.1 Image Models	4
2.2 Applications of Image Models	5
2.2.1 Stochastic Regularization and Bayesian Estimation	5
2.3 Markov Random Field Image Models	8
2.4 Common Markov Random Fields	14
2.4.1 Discontinuity Preserving MRF's	17
2.5 Model Parameter Estimation	22
3 SIMULATION OF RANDOM FIELDS	23
3.1 Introduction	23
3.2 Reason for Simulating MRF's	24
3.3 Methods for Simulating MRF's	25

3.3.1	Metropolis Algorithm	25
3.3.2	Gibbs Sampler	27
3.3.3	Clique Weight Conventions	28
3.4	Convergence Issues	29
3.5	Some Results of GGMRF Simulations	31
3.5.1	Simulation Convergence for Various GGMRF Realizations . . .	31
3.5.2	Realizations from GGMRF Simulations	31
4	PARAMETER ESTIMATION FOR THE GGMRF	41
4.1	Introduction	41
4.2	Parameter Estimation for MRF's	42
4.3	Difficulty of Parameter Estimation for MRF's	43
4.4	Some Existing Approaches to Parameter Estimation	44
4.5	The Absolute Value MRF and Weighted Order Statistic Filters	45
4.5.1	Graphical Solution for the AVMRF MAP Pixel Predictor . . .	48
4.6	Clique Weight Estimation by Minimization of Prediction Error	50
4.7	Choice of the Prediction Error Cost Function	52
4.8	Minimization of Prediction Error for the AVMRF	56
4.8.1	MAP Pixel Prediction Behavior as a Function of Clique Weight	58
4.8.2	Risk Behavior as a Function of Clique Weight	62
4.8.3	Quantization Effect	64
4.8.4	Generalization to Higher Order Predictor Models	65
4.8.5	A Problem with the AVMRF MAP Pixel Predictor	66
4.8.6	A Solution to the AVMRF MAP Pixel Predictor Problem . . .	69
4.8.7	Estimator Performance for Higher Order Models	70

4.8.8	Midpoint Value AVMRM MPE Estimator Performance	72
4.9	The GGMRM and Generalized Weighted Order Statistic Filters	77
4.9.1	Graphical Solution for the GGMRM MAP Pixel Predictor	79
4.9.2	AVMRM MAP Predictor Problem Revisited	80
4.10	Minimization of Prediction Error for the GGMRM	82
4.10.1	MAP Pixel Prediction Behavior as a Function of Clique Weight	86
4.10.2	Risk Behavior as a Function of Clique Weight	87
4.10.3	Risk Behavior as a Function of the GGMRM p Parameter	88
4.10.4	MPE by Gradient Descent	89
4.10.5	MPE by Simulated Annealing	94
4.10.6	Other Possible Optimization Approaches	101
4.11	MPEE for Other Classes of MRM	101
4.12	MRM Simulation Convergence Revisited.	102
5	CONCLUSIONS	105
5.1	Primary Research Contribution	105
5.2	Future Directions of Investigation	106
A	DERIVATION OF GRADIENT FOR MPE OPTIMIZATION	111
B	ABSOLUTE VALUE PREDICTOR MPE PERFORMANCE	116
	REFERENCES	121

List of Tables

4.1	Behavior of the solution to the MAP estimate of a pixel given its neighbors for varying values of a chosen independent weight w	62
4.2	Performance of a 1 st order, midpoint prediction, absolute value MPE estimator for various input MRF realizations.	73
4.3	Performance of a 2 nd order, midpoint prediction, absolute value MPE estimator for various input MRF realizations.	74
4.4	Performance of a 2 nd order, midpoint prediction, absolute value MPE estimator for various input MRF realizations.	75
4.5	Performance of 1 st and 2 nd order, midpoint prediction, absolute value MPE estimators for various input test images.	76
4.6	Performance of a 1 st order, GGMRF MPE estimator, with $p = 1.01$ for various input MRF realizations.	83
4.7	Performance of a 2 nd order, GGMRF MPE estimator, with $p = 1.01$ for various input MRF realizations.	84
4.8	Performance of a 2 nd order, GGMRF MPE estimator, with $p = 1.01$ for various input MRF realizations.	85
4.9	Performance of a 2 nd order, GGMRF MPE estimator, with $p = 1.01$ for various input test images.	85
4.10	Gradient descent clique weight vector estimates using a 1 st order, GGMRF MPE estimator for various input MRF realizations.	95
4.11	Estimates of clique weight vectors using simulated annealing optimization. The risk functions optimized resulted from both 1 st and 2 nd order GGMRF predictors, with $p = 1.3$, using the power cost functional with $\gamma = 0.05$. The input realizations are GGMRF's with $p = 1.3$	100
4.12	Grid searching GGMRF MPE estimate of clique weight parameters for realizations from a GGMRF simulation chain with true clique weight parameters $[\frac{1}{2} \ \frac{3}{8} \ 0 \ \frac{1}{8}]^T$, $p = 1.3$ and $\sigma = 0.2$. $\hat{\sigma}$ is the ML estimate of the σ parameter at each simulation iteration.	104
B.1	Performance of a 1 st order, absolute value MPE estimator for various input MRF realizations using absolute value cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.	117

B.2	Performance of a 1 st order, absolute value MPE estimator for various input MRF realizations using squared cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.	118
B.3	Performance of a 2 nd order, absolute value MPE estimator for various input MRF realizations using absolute value cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.	119
B.4	Performance of a 2 nd order, absolute value MPE estimator for various input MRF realizations using squared cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.	120

List of Figures

2.1	Neighbors of given order are specified as the set of pixels either partially or fully contained by circle of increasing radius centered on a given pixel.	10
2.2	Cliques of a first order neighborhood.	11
2.3	Cliques of a second order neighborhood.	11
2.4	Gaussian MRF clique activity penalty function (above) and its derivative (below).	16
2.5	Blake and Zisserman type clique activity penalty function (above) and its derivative (below) for threshold $\alpha = 0.75$. All clique activity greater than the threshold is penalized equally.	18
2.6	Huber type clique activity penalty function (above) and its derivative (below) for threshold $\alpha = 0.75$. Like Blake and Zisserman's penalty, this function is quadratic up to the threshold parameter, but thereafter increased spatial activity is penalized linearly.	19
2.7	Absolute value clique activity penalty function (above) and its derivative (below). There is no distinction between edges and smooth regions in this model.	20
2.8	Generalized Gaussian MRF clique activity penalty function (above) and its derivative (below). The GGMRF has, as its limiting cases, the Gaussian MRF and the AVMRF. The plot is for $\alpha = 1.2$	22
3.1	Notation for ordering the elements of the clique weight vector. Here we show a second order model, with the relevant clique weight vector elements labeled.	28
3.2	Convergence behavior of the maximum likelihood estimate of the model σ parameter given p for the simulation of realizations of a GGMRF with $\sigma = 0.2$ and $p = 1.1$, with clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$	30
3.3	Comparison of the convergence behavior of the maximum likelihood estimate of the model σ parameter given p for the simulation of realizations of a GGMRF with $\sigma = 0.2$ and $p = 1.1$, with clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$, for iid uniform noise initial condition and constant valued initial condition.	32
3.4	Comparison of the convergence behavior of the ML estimate of σ for various GGMRF simulations with $\sigma = 0.2$ and $1.1 \leq p \leq 1.9$, with clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$	32

3.5	Sequence of images in generation of 1 st order GGMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.	33
3.6	Sequence of images in generation of 1 st order GGMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.	34
3.7	Sequence of histogram equalized images in generation of 1 st order GGMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.	35
3.8	Sequence of histogram equalized images in generation of 1 st order GGMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.	36
3.9	Sequence of images in generation of 1 st order GGMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.	37
3.10	Sequence of images in generation of 1 st order GGMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.	38
3.11	Sequence of histogram equalized images in generation of 1 st order GGMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.	39
3.12	Sequence of histogram equalized images in generation of 1 st order GGMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.	40
4.1	Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is a unique value.	49
4.2	Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is an interval of values.	50
4.3	Cost functionals used for penalization of prediction error in MPE estimation. The cost functionals include the absolute value cost, squared cost, zero/one cost ($\alpha = 0.1$), power function cost ($\gamma = 0.1$), and piecewise linear cost ($\alpha = 0.1, \beta = 0.1$).	56
4.4	Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is an interval of values.	61
4.5	Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is a unique value.	61

4.6	Risk as a function of independent clique weight for a 1 st order predictor model. The MRF realization for which the prediction risk is computed has dimensions 128 pixels by 128 pixels, with clique weight vector $\frac{1}{6}[0\ 2\ 0\ 1]^T$. The square of the prediction error penalty was used.	64
4.7	Test images (128 pixels square) from which MRF clique weight parameters were estimated. (a)Aliased radial chirped sinusoid. (b)Hand segmented chest CT phantom. (c)Lenna. (d)Tiffany.	76
4.8	Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below) for a GGMRF.	81
4.9	Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below) for a GGMRF. The same neighborhood system led to a non-unique prediction when the data was assumed to be from an AVMRF.	81
4.10	Risk as a function of independent clique weight for a 1 st order GGMRF predictor model with $p = 1.3$. The input realization is a GGMRF with clique weight vector $\frac{1}{6}[0\ 2\ 0\ 1]^T$ and $p = 1.3$. The square of the prediction error penalty was used.	88
4.11	Risk as a function of independent clique weight for a 1 st order GGMRF predictor for various values of p . The input realization is the “lenna” test image. The square of the prediction error penalty was used. . . .	89
4.12	Risk function and the trajectory of a gradient descent search for a minimum of the function. Shown in lower frame is the derivative of the risk.	91
4.13	Risk function for which gradient descent optimization is ineffective. . .	93
4.14	Weight values accepted in a simulated annealing optimization of a complex risk functional associated with a 1 st order GGMRF. The circle denotes the optimal independent clique weight as found by annealing.	99
4.15	Risk function for a 1 st order, $p = 1.3$ GGMRF predictor, penalized using the power cost with $\gamma = 0.05$, on a realization of a 2 nd order GGMRF, $p = 1.3$, clique weight vector $[1\ 0\ 0\ 0]^T$	101
4.16	Weight values accepted in a simulated annealing optimization. In this case the annealing was performed too quickly with the result that the system “froze” in a local minimum (marked by the circle). The global minimum exists close to the weight value of 0.06.	102

ACKNOWLEDGEMENTS

Were it not for the generous support of the Fulbright Scholarship Program, you would not, in all likelihood, be reading this. Many thanks to Gill Jacot–Guillarmod, the Fulbright Program representative in South Africa, who was instrumental in the selection and placement of “Fulbrighters” throughout the U.S. I hope that I have justified the confidence that Gill showed in me. My gratitude is also extended to my IIE field representatives in Chicago, who were most helpful throughout the duration of my studies here.

My Alma Mater, the University of Cape Town, South Africa, has been very generous in its support of my overseas studies. My thanks to the Annel Trust Bursary and the Sonnenberg Scholarship programs for their financial assistance. In the postgraduate scholarships office, I wish to single out Ms. Cynthia Grindley for her unceasing help and cooperation in surmounting administrative hurdles. In addition, through Ms. Grindley’s email letters describing her weekend and vacation expeditions, I was able to continue to enjoy the majestic Cape mountains, albeit in a necessarily vicarious fashion.

The University of Notre Dame, very generously provided me with a full tuition scholarship during my first academic year here.

My thanks also to Professors Bauer, Stevenson and Huang who kindly agreed to review this work, and serve on my thesis examining committee.

Finally but definitely not least significantly, I wish to express gratitude to my academic advisor Professor Ken Sauer for his patient support of my work through both good, and sometimes very trying, times. I feel that though my initial academic background was perhaps not ideally suited to the work we tackled, working with Dr. Sauer has contributed towards my academic development well beyond my expectations. It is my sincere hope therefore that this body of work meets the very exacting standards of my advisor. Dr. Sauer also very generously supported my work through a research assistantship funded by a research grant he holds from the National Science Foundation.

Chapter 1

INTRODUCTION

1.1 Quo Vadis ?

In this thesis we shall focus on the problem of the *estimation* of the *clique weight parameters* for the generalized Gaussian Markov random field (GGMRF). We shall introduce the technique which we term *minimization of prediction error estimation* (MPEE) and study its application and performance. To reach this point we shall have to follow a route which will review necessary background information concerning Markov random fields in general and will also spend some time discussing simulation of MRF's in order to produce realization data which will be necessary to test the performance of the MPEE method.

1.2 Thesis Organization

The body of this thesis is logically divided into three main sections:

- Background

In chapter 2 we discuss the preliminary knowledge that is required to understand the estimation problem we will tackle. We begin this chapter by discussing image models in general, suggesting their usefulness in image restoration and reconstruction problems. This discussion includes a brief review of

the *Bayesian* framework for using image models in inverse problems. With a hint of the value of image models for representing *a-priori* information, we make formal mathematical acquaintance with the well known Markov random field (MRF). We follow this somewhat technical discussion with the introduction of a number of Markov fields well known in the literature, commenting on their relative merits and demerits. With sufficient familiarity with these models, we hint towards the looming estimation problem that will concern us in chapter 4.

- Simulation

The next major topic discussed in chapter 3, concerns simulation of MRF models. We make some initial comments concerning the reason why simulation is at all necessary and then introduce and discuss the framework common to many simulation algorithms, the *Metropolis method*. We follow this with a brief mention of the interesting problem of convergence testing. Of course we cannot leave the reader's curiosity concerning the true character of MRF's unsatisfied, so we include a number of example images which are realizations from the mathematically elegant Markov random field.

- Estimation

Having entertained ancillary topics for almost forty pages, we enter the realm of parameter estimation in chapter 4. We begin our journey with an introduction to the problem and an exposition of some of the characteristics which make it decidedly non-trivial. We briefly mention some of the existing approaches to estimation for the MRF before illuminating a connection between weighted median filtering and MAP estimation of a pixel for a particular type

of MRF, the absolute value MRF (AVMRF). This discussion will leave us sufficiently prepared to introduce our proposed estimation method, minimization of prediction error estimation (MPEE), for estimating the clique weight parameters for (initially) the AVMRF. After a few details and problems are considered we discuss a grid searching procedure for minimizing the objective function inherent in the MPEE method for the AVMRF. We show results from tests performed on realization data generated using the simulation methods discussed in chapter 3, as well as on real imagery. With the essential ideas of MPEE for the AVMRF understood, we suggest a generalization of the weighted median filter which enables us to extend the MPEE method to the generalized Gaussian MRF, the model that is of primary interest in this work. We show that it is not possible to effect MPEE optimization for the GGMRF using the grid searching procedure used for AVMRF MPEE, so we discuss alternative methods, namely, gradient descent optimization and simulated annealing. Performance of the MPEE method for the GGMRF is also discussed. We conclude the chapter with some speculation as to how MPEE may be applied to other classes of MRF's.

Finally, in chapter 5, we summarize and comment on what we have achieved in this work and suggest further directions of investigation.

Chapter 2

BACKGROUND

In this chapter we shall focus our attention on the background information required to understand the parameter estimation problem we hope to tackle. We begin by introducing the concept of an image model and briefly describe the usefulness of such models in image reconstruction, restoration and analysis. We follow this with an introduction to a very broad and useful class of image models known as random fields, before examining the Markov random field in particular. It is with the Markov random field (MRF) that we make closest acquaintance. We discuss a number of MRF types and identify desirable properties of these models. We then draw these ideas together and introduce the generalized Gaussian Markov random field (GGMRF) model which is the principle MRF under investigation in this thesis. We conclude this chapter with a prelude to the model parameter estimation problem which forms the main body of this work.

2.1 Image Models

Describing the essential character of images is, in general, a decidedly non-trivial task. This is, however, exactly what is required when attempting to formulate an *image model*. One common, intuitive image model suggests that an image may

be described as piecewise smooth areas of nearly constant intensity connected by discontinuities or edges. This seemingly crude model will nonetheless be sufficient for the work that we will undertake. More complex models are certainly possible but often lead to significant computational problems or are mathematically intractable.

Resulting from the broad range of approaches to image modeling, there is a significant literature on the subject. See for instance the reviews [2, 3, 33].

In order to gain some understanding of the value of even a simple image model such as the intuitive one we have proposed, we shall briefly examine one framework within which image models play an essential role.

2.2 Applications of Image Models

To attempt to describe every possible application for an image model would verge on the absurd. We shall, instead, introduce one application which, despite its simple appearance, is general enough to encompass a great many practical problems. The technique is known as stochastic regularization and relies on Bayesian estimation theory.

2.2.1 Stochastic Regularization and Bayesian Estimation

We will direct our attention to an *observation model* of the form,

$$\mathbf{Y} = \mathbf{HZ} + \mathbf{N}, \tag{2.1}$$

where \mathbf{Y} is the observed output, \mathbf{H} is some known linear transformation acting on the underlying data \mathbf{Z} and \mathbf{N} is additive noise. Our goal is to compute an estimate $\hat{\mathbf{Z}}$ of the value of the underlying data \mathbf{Z} given the observations \mathbf{Y} . An initial approach to obtaining an estimate of \mathbf{Z} would be to attempt direct inversion as,

$$\hat{\mathbf{Z}} = \mathbf{H}^{-1}\mathbf{Y}$$

$$= \mathbf{Z} + \mathbf{H}^{-1}\mathbf{N}. \quad (2.2)$$

The problem with this approach concerns the matrix \mathbf{H} . The conditioning of \mathbf{H} strongly affects the behavior of the solution [20]. Indeed, \mathbf{H} could even be singular, completely ruling out this simplistic approach for estimating \mathbf{Z} . Ill conditioning of the \mathbf{H} matrix is far less uncommon than might be assumed. Classic examples from machine vision and image processing include stereopsis and image restoration [24].

Hadamard [16] defines a problem as “well-posed” if its solution exists, is unique and depends continuously on the observation data. The inverse problem of estimating \mathbf{Z} violates the second of the Hadamard requirements if the matrix \mathbf{H} is singular, since many possible solutions exist. Multiple solutions may also exist if there is insufficient observation data.

The techniques known as *regularization* may, however, be used to tackle such ill-posed problems. There are two common approaches; deterministic and stochastic. Their common feature is that they both make use of *a-priori* information concerning \mathbf{Z} to impose a “cost” on the range of possible solutions. The *a-priori* knowledge is provided in the form of a *model*.

An example of the former approach to solving the problem is Tikhonov regularization [38]. Our attention will, however, focus on the second approach, stochastic regularization, which assumes probabilistic models for the underlying data process \mathbf{Z} and the noise process \mathbf{N} .

If we assume that both \mathbf{Z} and \mathbf{N} are derived from stochastic processes characterized by some probability density functions, we may adopt a Bayesian approach to estimating \mathbf{Z} [24]. The Bayesian approach for estimation of \mathbf{Z} requires maximization

of the *a-posteriori* probability $\mathcal{P}(\mathbf{Z}|\mathbf{Y})$ [39] :

$$\hat{\mathbf{Z}}_{MAP} = \arg \max_{\mathbf{Z}} \{\mathcal{P}(\mathbf{Z}|\mathbf{Y})\}. \quad (2.3)$$

Not surprisingly, this technique is called maximum *a-posteriori* (MAP) estimation.

If we apply Bayes' rule to equation 2.3, we find,

$$\hat{\mathbf{Z}}_{MAP} = \arg \max_{\mathbf{Z}} \left\{ \frac{\mathcal{P}(\mathbf{Y}|\mathbf{Z})\mathcal{P}(\mathbf{Z})}{\mathcal{P}(\mathbf{Y})} \right\}. \quad (2.4)$$

Since the maximization is over \mathbf{Z} , the solution $\hat{\mathbf{Z}}_{MAP}$ is unaffected by the denominator term $\mathcal{P}(\mathbf{Y})$ which is constant with respect to \mathbf{Z} , so we may simplify equation 2.4 to,

$$\hat{\mathbf{Z}}_{MAP} = \arg \max_{\mathbf{Z}} \{\mathcal{P}(\mathbf{Y}|\mathbf{Z})\mathcal{P}(\mathbf{Z})\}. \quad (2.5)$$

Since the logarithm is a monotonic, increasing function, we may equivalently solve,

$$\hat{\mathbf{Z}}_{MAP} = \arg \max_{\mathbf{Z}} \{\log \mathcal{P}(\mathbf{Y}|\mathbf{Z}) + \log \mathcal{P}(\mathbf{Z})\}. \quad (2.6)$$

As a consequence of the application of Bayes' rule, we have separated out the conditional probability $\mathcal{P}(\mathbf{Y}|\mathbf{Z})$ familiar in ML estimation [5], as well as the term $\mathcal{P}(\mathbf{Z})$, which is called the *a-priori probability* or simply the *prior*. This term contains all the knowledge we have on the underlying process \mathbf{Z} . Notice that equation 2.6 differs from the ML estimation approach only in the sense that it contains the prior. This is, however, a very significant difference, since the inclusion of this knowledge in the solution process adds a “bias” term which favors solutions which are consistent with the model. This additional information is what makes for a “regularized” solution.

In regularized solutions to inverse problems such as the one we have considered, often the primary difficulty is in the computational optimization which maximizes the *a-posteriori* density.

Though the focus of this thesis is on parameter estimation for a class of image model, the motivation for this estimation work is the potential for improved Bayesian image reconstruction and restoration resulting from the use of more realistic *a-priori* image models. In particular, we estimate parameters for a class of image models called Markov random fields, which we shall now introduce.

2.3 Markov Random Field Image Models

In the following sections we will examine Markov random field image models in some detail. An introduction to the subject may be found in [21]. A random field is essentially a probability measure on the space of images. We will refine this definition in later sections, but simply put, a random field specifies which “types” of images are more likely than others. Random fields are *stochastic models* characterized by a *cumulative distribution function* (CDF). Assuming that the corresponding *probability density function* (PDF) exists, we shall examine the PDF of various models to gain some understanding of how we may control model parameters to encourage successful representation of typical image properties. It is this ability to control the parameters of the image model that allows us to use a random field as an effective model of prior knowledge.

Random Field Nomenclature

As our interest lies in *computational processing* of image data, we are restricted to computation on finite data sets. We shall therefore limit this discussion to random fields defined on finite, countable point sets. We shall assume that the cardinality of this set is M . This set may be any set of M points, the most familiar example, to those involved in the processing of images, being a regular 2-dimensional lattice.

Independent, however, of the spatial distribution of these points, we may always define an ordering of the points or *sites*. We denote this ordered set of M sites as $\mathcal{S} = \{S_i : 1 \leq i \leq M\}$. As a concrete example, a lexicographic ordering of the pixels of an image will map the two dimensional lattice to the point set \mathcal{S} .

Associated with every $S_i \in \mathcal{S}$ there is a random variable Z_i . The ordered set of Z_i 's forms a random vector \mathbf{Z} . A particular value of a random variable or random vector will be identified using lower case. Here we shall consider only continuous-valued random variables, but discrete-valued RV's are also common. We define a sample space Ω based on \mathcal{S} and a Borel set Ψ on Ω . Finally we may specify \mathcal{P} , a probability measure on Ψ . With these three elements defined, we have specified a valid *probability space*. Our emphasis will be on the definition of the probability measure \mathcal{P} .

Neighbors

Considering the point set \mathcal{S} defined above, we may define for each $S_i \in \mathcal{S}$, a set of *neighbors* \mathcal{N}_i . The neighbors of a site in a lattice system are typically the geometric neighbors of the site. We define a *neighborhood system* \mathcal{N} as the set of neighbors \mathcal{N}_i of each S_i [13]. Note that a site cannot be its own neighbor. Furthermore, if S_j is a neighbor of S_i , that is $S_j \in \mathcal{N}_i$, then this implies that S_i is also a neighbor of S_j , or $S_i \in \mathcal{N}_j$. At the boundaries of the image we have the choice of imposing a “free boundary” where image boundary sites have fewer neighbors, or other constraints, such as periodic repetition (toroidal boundary).

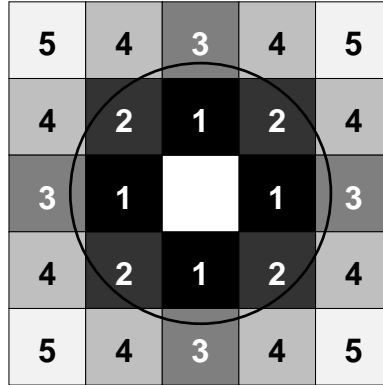


Figure 2.1: Neighbors of given order are specified as the set of pixels either partially or fully contained by circle of increasing radius centered on a given pixel.

Model Order Determination

Since MRF's are characterized by their neighborhood structure, it is useful to define a *model order* which describes which sites are neighbors of any given site. Figure 2.1 shows the numbering system that will be used in this text. From the figure, it should be clear that the if the four-neighbors of a site are the only neighbors of a given site, then we denote this as a first order model. If the eight-neighbors are used, we have a second order model since models of increasing order are defined to include all the sites that constitute models of lower order. In the general case, we denote models of increasing order by including all sites which are either partially or fully contained by a circle of increasing radius centered on the origin of the center site. Each increase in the number of sites included by this expanding circle denotes a unit increase in the model order. Thus we will have 4, 8, 12, 20 and 24 sites in models of order 1, 2, 3, 4 and 5 respectively.

Cliques

A clique is a set of sites, all pairs of which are mutual neighbors. Individual sites are, by definition, also cliques. To clarify this definition, the reader is referred to

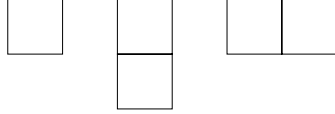


Figure 2.2: Cliques of a first order neighborhood.

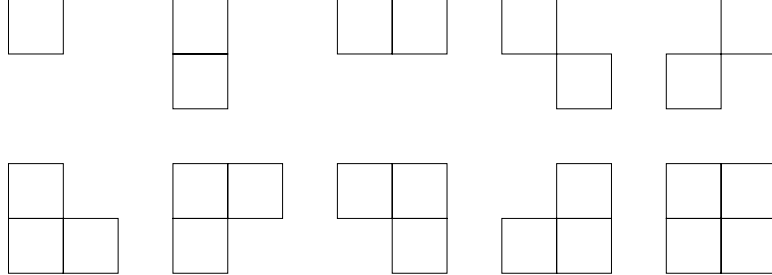


Figure 2.3: Cliques of a second order neighborhood.

figures 2.2 and 2.3, which show the clique types corresponding to first and second order neighborhoods respectively. Since higher order neighborhoods contain all sites of neighborhoods of lower order, cliques of higher order also contain all cliques of lower order. We denote the clique of S_i as \mathcal{C}_i and the set of all cliques as \mathcal{C} .

Markov Random Field Definition

The vector \mathbf{Z} , the ordered collection of random variables of site values, is a Markov random field with respect to its neighborhood system [13] provided that for every $S_i \in \mathcal{S}$ and realization ω in the sample space Ω ,

$$\mathcal{P}(\mathbf{Z} = \omega) > 0 \quad \forall \omega \in \Omega \quad (2.7)$$

$$\mathcal{P}(Z_i = z_i \mid Z_k = z_k, k \neq i) = \mathcal{P}(Z_i = z_i \mid Z_k = z_k, k \in \mathcal{N}_i). \quad (2.8)$$

It is the second of these equations that lends the name “Markov” to the otherwise ordinary random field definition, due to the characteristic that the conditional density of the single random variable Z_i , conditioned on the values of all the remaining variables Z_k , $i \neq k$, is a function of only those variables which are *neighbors* of S_i . Equation 2.8 describes the *local characteristics* of the Markov model. In the

following section, we shall see that the specification of the local characteristics of the MRF determines the complete form of the MRF. This is one of the main reasons why the MRF is so useful and convenient in many applications. The question now arises as to the exact form of the joint PDF of \mathbf{Z} .

MRF / Gibbs Distribution Equivalence

It has been shown [11, 13] that there is an equivalence between a Markov random field and a Gibbs distribution. More specifically, the joint density of the site value random variables Z_i must necessarily be a Gibbs distribution. Technically, the random vector \mathbf{Z} associated with the sites \mathcal{S} and the neighborhood system \mathcal{N} is a MRF if and only if it is distributed according to,

$$\mathcal{P}(\mathbf{Z}) = \frac{1}{K_p} \exp \left\{ -\frac{1}{\beta} \sum_{c \in \mathcal{C}} V_c(\mathbf{Z}) \right\}. \quad (2.9)$$

This result, commonly known as the Hammersley Clifford theorem, comes to our rescue in determining the form of the joint PDF. In this equation, the scaling constant K_p is known as the the *partition function* or “Zustandsumme” in German, meaning literally *sum over states*, since the value of the constant is found by integrating the un-scaled PDF expression over all possible states of the argument \mathbf{Z} [17]. Often in the literature, the partition function is denoted as Z , but we refrain from this choice in order to avoid confusion with the data term \mathbf{Z} . Considering the dimensionality of all but the smallest MRF’s, the seemingly innocuous partition function is, in practice, very difficult to compute. It is this fact that so complicates estimation problems involving MRF’s. The parameter β may be considered akin to the *temperature* parameter of the Gibbs distribution encountered in thermodynamics. $V_c(\cdot)$, called the *clique function*, is a function of the MRF cliques. As before, \mathcal{C}

denotes the set of all image cliques. The summation of clique function terms gives the *energy* of the field.

It will be useful for us to further refine our definition of the MRF PDF above to introduce the concept of the *clique activity function*, the *clique activity penalty function* and the *clique weight*. We may restate equation 2.9 in the form,

$$\mathcal{P}(\mathbf{Z}) = \frac{1}{K_p} \exp \left\{ -\frac{1}{\beta} \sum_{c \in \mathcal{C}} w_c \cdot \rho_{c,\alpha}(\mathcal{A}_c(\mathbf{Z})) \right\}. \quad (2.10)$$

where w_c is the *clique weight*, $\rho_{c,\alpha}(\cdot)$ is the *clique activity penalty function* and \mathcal{A}_c is the *clique activity function*.

The clique activity penalty function is a function of the clique c as well as the parameter α which governs the shape of the penalty incurred on clique “activity”. Note that we have the relationship $V_c(\cdot) = w_c \cdot \rho_{c,\alpha}(\mathcal{A}(\cdot))$. The advantage of this notation is that it separates the logically independent functions which determine spatial activity and which penalizes it. Also it clearly identifies the clique weight parameters of the model.

The *clique activity function* \mathcal{A}_c measures the “activity” of a clique and often takes the form of a discrete approximation to the first (or higher) derivative of the local data. In image processing, this function is used to provide a measure of local image smoothness.

The *clique activity penalty function* $\rho_{c,\alpha}$ penalizes the clique activity in some systematic manner. For instance, if smoothness is a desirable property of the model, then high clique activity may be penalized more heavily than lower activity. The parameter α controls the shape of the penalty function. We will have more to say about the activity penalty function in later sections.

Finally the *clique weight* w_c determines the relative importance of a particular

clique's contribution to the overall energy function.

Though it is possible to define w_c for every clique, this is rarely done. For example, in image restoration or reconstruction problems, where MRF's are often used as homogeneous image priors, w_c is constant for each clique type. Thus referring to figure 2.2, showing the cliques types for a first order model, we have three distinct w_c values. For a second order model, whose clique types are shown in figure 2.3, there would be a maximum of ten. A common class of MRF, known as *pairwise interaction* MRF, has $w_c = 0$ for all cliques which do not contain exactly two sites.

This reduced complexity also extends to the clique activity penalty function, which, in general, may be defined differently for all cliques, as denoted in equation 2.10. In a homogeneous field, however, there is a single function for each type of clique. Furthermore, when we have a pairwise interaction process, the clique activity penalty function is the same for all cliques, allowing us to simplify $\rho_{c,\alpha}$ to dependence on just the parameter α . We may then write the PDF of a pairwise interaction, homogeneous MRF as,

$$\mathcal{P}(\mathbf{Z}) = \frac{1}{K_p} \exp \left\{ -\frac{1}{\beta} \sum_{c \in \mathcal{C}_2} w_i \cdot \rho_\alpha (\mathcal{A}_c(\mathbf{Z})) \right\}. \quad (2.11)$$

Where i numbers each unique type of clique pair in the set of pair cliques \mathcal{C}_2 .

2.4 Common Markov Random Fields

In this section we shall discuss a number of common Markov random field models. This discussion is not intended to be an exhaustive review; for this, the reader is referred to [9, 11, 23, 28]. Instead, we shall focus on a narrow class of MRF's, in particular, homogeneous, pairwise interaction MRF's, which are the concern of this text. This section should impart to the reader a better understanding of the Markov

random field and why, in particular, we focus on a particular model, the generalized Gaussian Markov random field (GGMRF).

The Gaussian MRF

Perhaps the best known example of a MRF is the Gaussian Markov random field (GMRF). It has been utilized extensively by image processing researchers and, despite having been technically eclipsed by more capable edge preserving models (to be discussed in section 2.4.1), it continues to enjoy some popularity.

The GMRF is characterized by a square law clique activity penalty function as,

$$\rho(x) = x^2. \tag{2.12}$$

Notice that there is no control parameter α in this model. A plot of this penalty function and its derivative are displayed as figure 2.4. The derivative function is akin to the “influence function” in robust statistics [18], and is useful in understanding the effect of the penalty function.

Recalling the form of the probability density of a Markov random field, it should be immediately evident that the GMRF is little more than a multivariate Gaussian distribution. The most significant problem with the Gaussian MRF model is the fact that it is not realistic for images. While the basic idea of regularization is to alleviate the ill-posed nature of the original problem through the inclusion of *a priori* knowledge, we do require this *a-priori* information to be as realistic as possible. The Gaussian MRF model is effective in modeling smooth data, but edges are heavily penalized. A model which is characterized by excessive edge penalization is contrary to the basic idea of images being essentially piecewise smooth regions joined by discontinuities or edges, therefore when using the Gaussian MRF as a

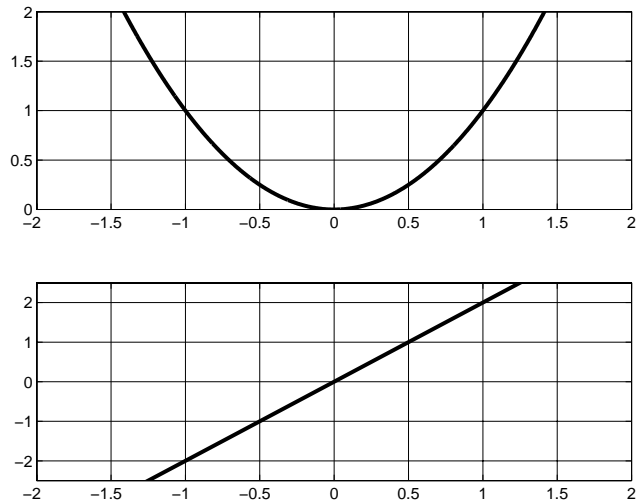


Figure 2.4: Gaussian MRF clique activity penalty function (above) and its derivative (below).

priori information for regularized image reconstruction, edge reconstruction is heavily degraded. Since much of the information in an image is contained in edges, discouraging their reconstruction is clearly disadvantageous.

Since the *raison d'être* of image models is to capture the essential characteristics of images, there has been considerable effort exerted towards finding image models which favor local smoothness while at the same time accommodating the presence of edges. It is important that the reader be cognizant that these two requirements are essentially conflicting. It is not possible, using simple MRF models, to fully satisfy both requirements. We face the choice of using either a relatively simple, computationally attractive model, with the disadvantage of having to compromise in the attempt to satisfy the conflicting requirements of local smoothness and edge preservation, or to increase the complexity of the model and deal with increased computational burden. Both these approaches appear in the literature. An interesting example of the latter is Geman and Geman's combined point and line process

model [13]. We are most concerned with the former approach and will examine a number of well known models designed to address the problem of edge preservation. It is well worth mentioning that the interesting problem of edge preserving image models, despite having been subject to extensive scrutiny, is far from being put to rest.

2.4.1 Discontinuity Preserving MRF's

As discussed in section 2.3, the manner in which local spatial variation is penalized through the choice of the *clique activity penalty function* $\rho_\alpha(\cdot)$. In this section we shall review a number of choices for ρ_α , which focus on edge preservation.

Apart from the models we shall discuss, many more may be found in the literature. References to a number of these models may be found in [7].

Blake and Zisserman's MRF

An early attempt to determine a MRF model which allowed edges while maintaining the necessary property of local smoothness was made by Blake and Zisserman [6] who introduced an activity penalty function of the form,

$$\rho_\alpha(x) = \begin{cases} x^2, & |x| \leq \alpha, \\ \alpha^2, & |x| > \alpha. \end{cases} \quad (2.13)$$

This function, shown with its derivative in figure 2.5, is quadratic up to the threshold α , but there is no additional cost incurred for increased local spatial activity beyond this threshold level. Blake and Zisserman view an edge as any clique activity which exceeds the threshold parameter α . All such edges are penalized equally. Their penalty function is intuitively appealing, but unfortunately the non-convexity of this function may result in unsatisfactory behavior when using this model as a prior in reconstruction problems [7].

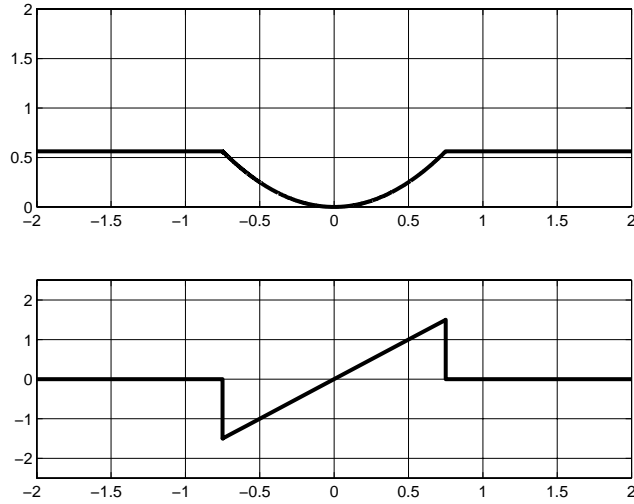


Figure 2.5: Blake and Zisserman type clique activity penalty function (above) and its derivative (below) for threshold $\alpha = 0.75$. All clique activity greater than the threshold is penalized equally.

The Huber MRF

The Huber MRF championed by Stevenson *et. al.* [34,36] harks back to the work of Huber in robust statistics [18]. This model is similar to that of Blake and Zisserman, with the exception that from the threshold α , a linear penalty increase is incurred for increasing spatial activity. The model is characterized by,

$$\rho_{\alpha}(x) = \begin{cases} x^2, & |x| \leq \alpha, \\ \alpha^2 + 2\alpha(|x| - \alpha), & |x| > \alpha. \end{cases} \quad (2.14)$$

The Huber penalty and its derivative, are shown in figure 2.6. The modification of the tails of the Huber penalty function compared with the Blake and Zisserman penalty ensures that the Huber function is both convex and possesses a continuous first derivative. Thus, the Huber function avoids the numerical pitfalls associated with the Blake and Zisserman model. It is interesting to note that for $\alpha = 0$ the Huber function coincides with the absolute value function and for $\alpha \rightarrow \infty$, it tends to the Gaussian model. By adjusting the threshold parameter, a wide range of

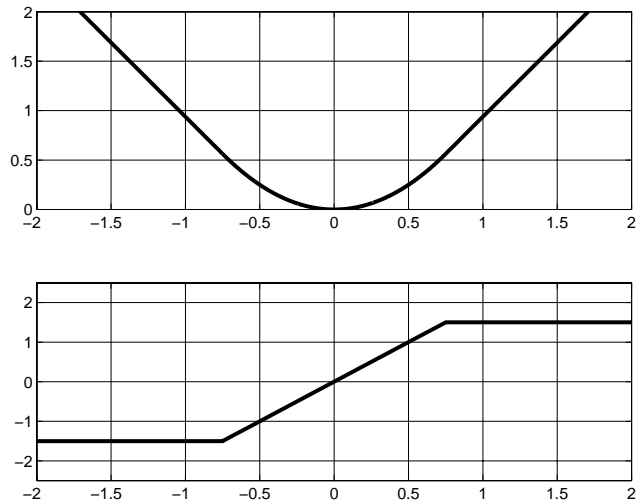


Figure 2.6: Huber type clique activity penalty function (above) and its derivative (below) for threshold $\alpha = 0.75$. Like Blake and Zisserman’s penalty, this function is quadratic up to the threshold parameter, but thereafter increased spatial activity is penalized linearly.

behavior, from very smooth Gaussian data to excellent edge preservation is possible.

The Absolute Value MRF

We have seen that the absolute value clique activity penalty function is reached as an extreme case of the Huber Model. The absolute value penalty levies linearly increasing cost for all increasing spatial activity, as evident from figure 2.7. Mathematically the clique activity penalty function is of the form,

$$\rho(x) = |x|. \quad (2.15)$$

The absolute value clique activity penalty function [26] characterizes the absolute value MRF (AVMRF). Notice that, unlike the Blake and Zisserman and Huber models, the AVMRF penalty function treats all spatial activity in the same manner. There are no regions of spatial activity which are considered to be characteristically

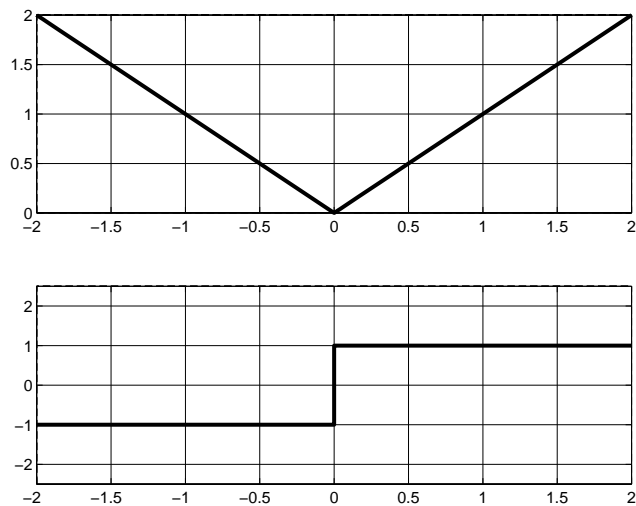


Figure 2.7: Absolute value clique activity penalty function (above) and its derivative (below). There is no distinction between edges and smooth regions in this model.

smooth or edge data. In practice, the AVMRF is highly effective in modeling data which contains edges [7]. A problem with the AVMRF model, however, is that of a discontinuous first derivative at the origin, making optimization problems involving this function potentially troublesome [7].

Desirable MRF Properties

Thus far we have examined a number of MRF's and have identified the following desirable characteristics.

1. Convexity of $\rho_\alpha(x)$

In stochastically regularized solutions to inverse problems as discussed in section 2.2.1, choosing a MRF prior for $\mathcal{P}(\mathbf{Z})$ with a convex clique activity penalty, along with certain restrictions on \mathbf{H} and $\mathcal{P}(\mathbf{N})$ [7], leads to a convex optimization problem for finding the MAP estimate.

2. Symmetry of $\rho_\alpha(x)$

Symmetry ensures that rising and falling regions in the data, (spatial activity) are penalized equally.

3. Reduced penalty for edges relative to quadratic

Places less penalty on the presence of edges in the data relative to the quadratic.

4. Continuity of $\frac{d}{dx}\rho_\alpha(x)$

A discontinuous clique activity penalty function makes for difficult maximization of the the *a-posteriori* distribution.

Generalized Gaussian MRF

A model championed by Bouman and Sauer [7] and the focus of this thesis, is the *generalized Gaussian Markov random field* (GGMRF). This model also derives from the field of robust statistics, in particular that of generalized Gaussian noise. The clique activity penalty function for this model is of the form,

$$\rho_\alpha(x) = |x|^\alpha, \quad 1 \leq \alpha \leq 2, \quad (2.16)$$

In addition to meeting all the desirable MRF properties discussed, the GGMRF model also possesses the characteristic of scalability, discussed in [7]. It is on this model that we shall focus our attention for the remaining chapters. Figure 2.8 shows a plot of the GGMRF clique activity penalty function and its derivative. Often we shall, as in the literature, use the letter p instead of the Greek α to denote the GGMRF shape parameter.

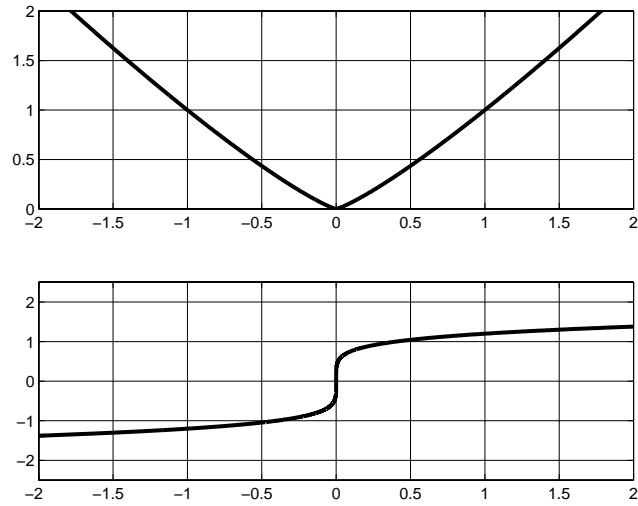


Figure 2.8: Generalized Gaussian MRF clique activity penalty function (above) and its derivative (below). The GGMRF has, as its limiting cases, the Gaussian MRF and the AVMRF. The plot is for $\alpha = 1.2$.

2.5 Model Parameter Estimation

The problem of model parameter estimation for pairwise interaction MRF's has been the focus of some recent research [10, 37]. It is in this field that this thesis makes its the primary contribution. We shall concentrate our efforts on the problem of estimating the clique weight parameters of a generalized Gaussian Markov Random field which best matches given realization data. Before we do this though, we shall first examine the problem of simulation of Markov random fields, which will provide both a deeper understanding of MRF's and the important ability to generate test data for the MRF parameter estimation procedures we shall examine in chapter 4.

Chapter 3

SIMULATION OF RANDOM FIELDS

3.1 Introduction

Having concluded our introduction to random fields and MRF's in particular, we turn our attention to the problem of *simulation* of random fields. We shall show why seemingly trivial questions such as, "What is the mean of a given MRF," are difficult to answer. We shall show that we are forced to utilize *iterative simulation* to attempt to answer questions like the one just posed. The reader is referred to the review [31], and [14] for further examples on the simulation of various Markov random fields. In this chapter, we will concentrate purely on the generalized Gaussian Markov random field, but the techniques discussed are quite general and are thus widely applicable.

In section 3.2 we show why we are forced to simulate MRF's even when seeking ordinarily easily found statistics. We continue in section 3.3 to review some methods for the simulation of MRF's. We follow this with a brief discussion on convergence issues pertaining to these iterative simulation methods and conclude this chapter in section 3.5 where some results of the simulation of GGMRF's are shown.

3.2 Reason for Simulating MRF's

Recall from section 2.3 that a MRF is characterized by a PDF defined for a set of sites \mathcal{S} and a neighborhood system \mathcal{N} . Now consider, as a simple example, the problem of determining the mean of a random field, assuming we know all parameters of the distribution, including the partition function. To find the mean $\mu_{\mathbf{z}}$ we must compute,

$$\mu_{\mathbf{z}} = \int_{\Omega} \mathbf{z} \cdot \mathcal{P}(\mathbf{z}) d\mathbf{z}. \quad (3.1)$$

Equation 3.1 at first appears quite benign, but for practically all MRF's of interest, (those with a large number of sites), the dimensionality of the sample space Ω makes it infeasible to apply direct methods, such as this one, to compute statistics of the field. Furthermore, often we do not have complete information to enable us to fully specify the PDF of the random field. Specifically, we are referring to a lack of knowledge of the value of the partition function, the normalization constant for the Markov random field PDF.

It is natural to ask if we have indeed reached an impasse or whether there is an alternative strategy for finding realizations from, and statistics of, the field. The answer to this question is that there is indeed a technique for finding realizations from the field, namely *iterative simulation*. Once we are in a position to simulate a particular MRF PDF, we may obtain statistics by assuming ergodicity and then computing ensemble averages as time averages from the simulation chain.

Although there exist a number of incarnations of iterative simulation methods [13, 15], many share a common ancestor in the *Metropolis algorithm*. As a result, we shall examine the Metropolis method in some detail, so that an understanding of the alternative methods may be gained by simply noting their differences from the

prototype.

3.3 Methods for Simulating MRF's

3.3.1 Metropolis Algorithm

Here we discuss the *Metropolis algorithm* proposed in [25]. The basic idea of the algorithm is to construct a Markov chain in time which converges to an equilibrium distribution which we choose to be that of the desired MRF PDF. Thus, if we begin the simulation from any initial state, we know that provided we run the simulation chain *long enough*, the statistical behavior of the chain will converge to that of the desired MRF. At this stage, the chain will have the same statistical behavior as the desired MRF distribution.

More specifically, denote the desired PDF as π . Assume it is possible to construct a Markov chain which has π as its equilibrium distribution. If we denote the distribution of the chain after n simulation iterations as π_n then we know that $\pi_n \rightarrow \pi$ as n increases. We assumed it was possible to construct such a Markov chain. The question is how to achieve this.

Consider two states i and j , with probabilities π_i and π_j respectively. Ripley and Kirkland [31] show that if we choose a symmetric, irreducible transition matrix Q and make the transition from state i to state j with probability $\min(1, \pi_j/\pi_i)$, (otherwise remaining at state i), then we can construct a new transition matrix P according to,

$$p_{ij} = \min\left(1, \frac{\pi_j}{\pi_i}\right) q_{ij} \quad \text{for } i \neq j \quad (3.2)$$

$$p_{ii} = q_{ii} + \sum_{j \neq i} \max\left(0, 1 - \frac{\pi_j}{\pi_i}\right) q_{ij}, \quad (3.3)$$

which ensures that

$$\pi_i p_{ij} = \pi_j p_{ji} \quad \text{for } i \neq j. \quad (3.4)$$

This definition of the transition matrix P is, according to [19, 31], sufficient to ensure that π is the equilibrium distribution of the chain. This result ensures that it is possible to sample from the desired MRF PDF, by sampling from the simulation chain once the chain has run for sufficient duration. We now outline a strategy for implementing the Metropolis algorithm to simulate a MRF.

1. Choose an initial state S_0 . The initial state may be chosen by assigning random values to every pixel in the realization image.
2. Given any state S_i , propose a new state S_{i+1} . Typically this new state is determined by choosing some random increment which is added to state S_i . In the case of an image, we add a random value to a single pixel in the image yielding the the proposed state S_{i+1} .
3. Accept state S_{i+1} with probability $\min\left(1, \frac{\mathcal{P}(S_{i+1})}{\mathcal{P}(S_i)}\right)$, else remain in state S_i . Thus, if the the probability of the proposed state S_{i+1} is higher than that of the present state S_i , then we automatically choose the proposed state, otherwise we choose it with probability equal to the ratio of the proposed state probability to the present state probability.
4. Return to step 2, for some other pixel in the image.

The order in which we update image pixels is unimportant, as convergence is guaranteed. Typically though, successive raster scan passes are utilized. We shall have more to say on the issue of convergence in section 3.4. However, once the chain

has converged, images will be produced by the chain according to the PDF of the desired MRF.

The reader should note that a very convenient result of the ratio of probabilities test for acceptance is that the partition functions of the numerator and denominator probabilities cancel. This implies that in order to simulate a MRF, knowledge of the value of the partition function is unnecessary. Furthermore, the Markov property of the MRF (the conditional probability of a pixel given all other image pixels, depends only on that pixel's neighbors) ensures that the probability ratio need only be computed using the update pixel's neighbors, saving substantial computation time.

3.3.2 Gibbs Sampler

A simulation method which has received considerable attention is the *Gibbs sampler* proposed by Geman and Geman [13]. This method differs from the standard Metropolis algorithm in that instead of choosing the proposed state completely randomly, the state is chosen from the conditional density of the pixel given the value of its neighbors. This results in state transition matrices which do not behave in the usual fashion, but Geman and Geman show that their method converges considerably faster than the regular Metropolis method, as the Gibbs sampler has zero probability of rejecting a proposed update. The Gibbs sampler is nevertheless identical to the Metropolis algorithm in the sense that it produces a chain whose equilibrium distribution is π .

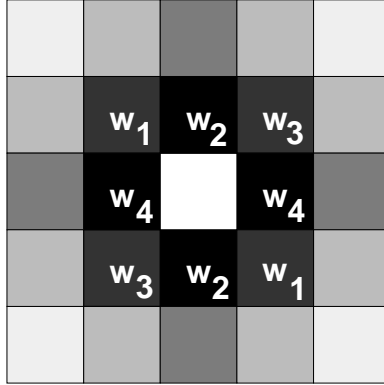


Figure 3.1: Notation for ordering the elements of the clique weight vector. Here we show a second order model, with the relevant clique weight vector elements labeled.

3.3.3 Clique Weight Conventions

As discussed in section 2.3, we have a set of clique weights in our definition of the MRF. It is possible to reduce the number of clique weights from the total number of cliques, to simply the number of clique types, if homogeneity is assumed. Since we deal with only pairwise interaction MRF's this further reduces the number of clique weights. Additionally, if we consider only neighborhoods of order n as described in section 2.3 then we can very succinctly describe the set of clique weights as a vector of the clique weights for each neighbor, beginning with the top left most neighbor relative to the center pixel, preceding in a raster scan through all the remaining neighbors. Due to the homogeneity of the MRF, clique weights associated with neighbors symmetrically positioned about the center pixel have equal value. Figure 3.1 should clarify this situation. For conciseness, we do not write these repeated weight values in the clique weight vector, since they are implied.

Thus, a first order model has only 2 elements in the clique weight vector. A second order model will have 4 and so on. With this convention behind us, we may now examine the convergence behavior for various GGMRF models.

Lastly, we adopt the convention that the sum of all the clique weights is unity (the sum of the elements of the abbreviated form clique weight vector is therefore one half). We are entitled to do this since it is only the ratio of the clique weights that effects the relative influence of a particular clique as compared with another. Any scaling of the complete weight set may be absorbed into the GGMRF σ parameter.

3.4 Convergence Issues

Before closing this chapter with some example realizations of GGMRF's we must address the question of convergence. Earlier we discovered that by using the Metropolis type algorithms, the distribution of simulation chain converges to the distribution of the desired MRF as n , the number of simulation iterations, increases. The question we must thus ask ourselves is, *how long must the simulation run before we have converged to the equilibrium distribution ?*

Ripley and Kirkland [31], suggest a method for testing for convergence. They show that convergence of a suitable parameter estimated from the chain of realizations, implies convergence of the chain itself to the equilibrium distribution. In their work, they demonstrate the estimation of a parameter denoting long distance correlation, using pseudolikelihood and asymptotic maximum likelihood (ML) methods.

Since this investigation concentrates on the Generalized Gaussian MRF model, we shall discuss convergence testing for this model alone. Bouman and Sauer in [8], derived the result that a *closed form* maximum likelihood estimate of the scale parameter σ of a GGMRF may be found given p .

In the simulation process, both the σ and p parameters are known for the desired equilibrium distribution, suggesting a convenient way of checking for convergence. At each iteration of the simulation chain, compute the ML estimate of the σ pa-

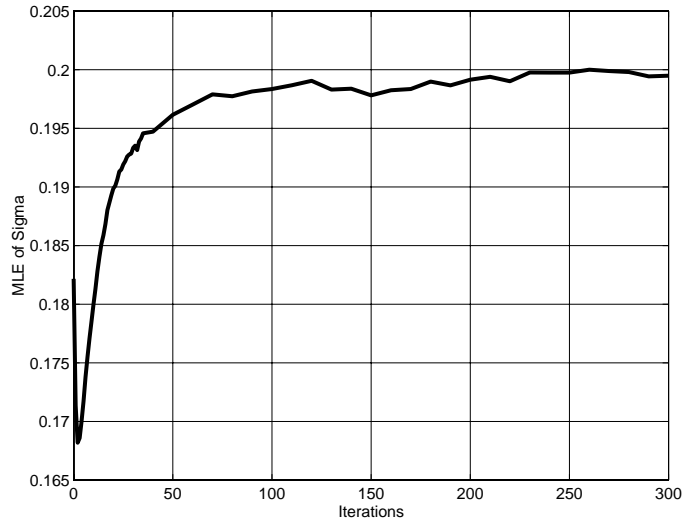


Figure 3.2: Convergence behavior of the maximum likelihood estimate of the model σ parameter given p for the simulation of realizations of a GGMRF with $\sigma = 0.2$ and $p = 1.1$, with clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$.

parameter, given the known p and check for convergence to the known σ parameter of the equilibrium distribution. According to [31], as the estimated parameter (in our case σ) converges to its equilibrium value, we can be reasonably confident that the realizations generated by the chain are “typical” of the MRF PDF, suggesting that the distribution of the chain is close to that of the desired MRF. Unfortunately, we have thus far been unable to prove that convergence of σ implies convergence of the chain. Later, in section 4.12 we shall argue that for the purposes for which we require simulation realizations, monitoring the convergence of the ML estimate $\hat{\sigma}$, appears sufficient.

Figure 3.2 shows such a scenario, where the ML estimate of the parameter σ is estimated from realizations of the simulation chain.

3.5 Some Results of GGMRF Simulations

In this section we show the results of simulation of the Generalized Gaussian MRF model. The simulation method is the Metropolis algorithm, with successive raster scans through all the image pixels.

3.5.1 Simulation Convergence for Various GGMRF Realizations

As discussed in section 3.4, the convergence behavior of the simulation Markov chain may be gauged from the convergence behavior of a parameter estimated from the chain. Figure 3.2 shows the convergence behavior of the σ parameter as estimated from the GGMRF simulation sequence. In this simulation, the chain was initialized to independent, uniformly distributed noise. In figure 3.3 however, we compare the convergence for a simulation which was begun using all zero values. Notice that the initial estimate for the σ parameter for this simulation is zero. This is not unexpected since σ is a measure akin to the variance in the Gaussian MRF distribution. Figure 3.4 compares the convergence rates for varying values of the GGMRF p parameter, using the uniform noise initial condition. Notice that convergence is slower for lower p .

3.5.2 Realizations from GGMRF Simulations

Realizations from a GGMRF model starting with uniform iid noise are shown in figures 3.5, 3.6 and with histogram equalization in figures 3.7 and 3.8, so as to make the large scale structure of the field realization more evident. Realizations from a GGMRF whose simulation was started with a constant valued initial condition are shown in figures 3.9, 3.10 and again, with histogram equalization in

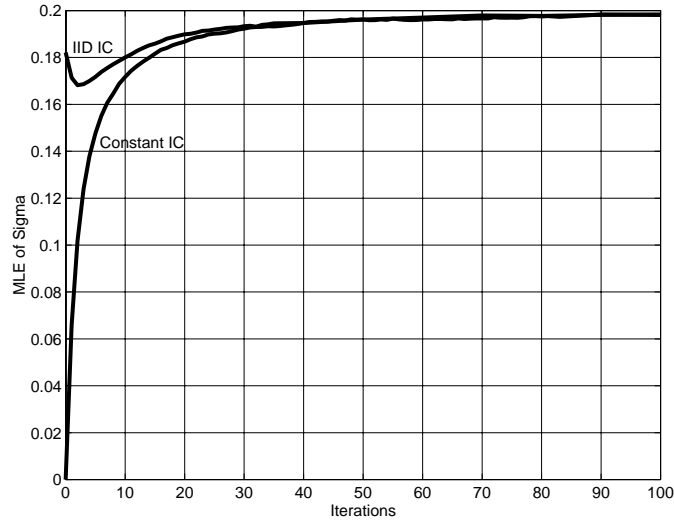


Figure 3.3: Comparison of the convergence behavior of the maximum likelihood estimate of the model σ parameter given p for the simulation of realizations of a GGMRF with $\sigma = 0.2$ and $p = 1.1$, with clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$, for iid uniform noise initial condition and constant valued initial condition.

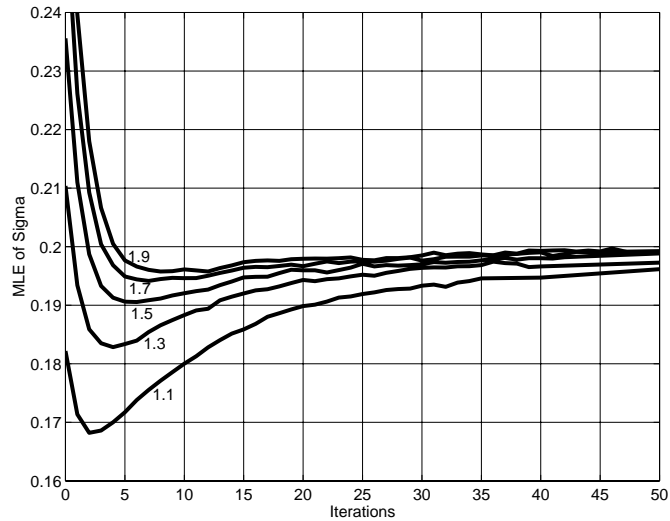
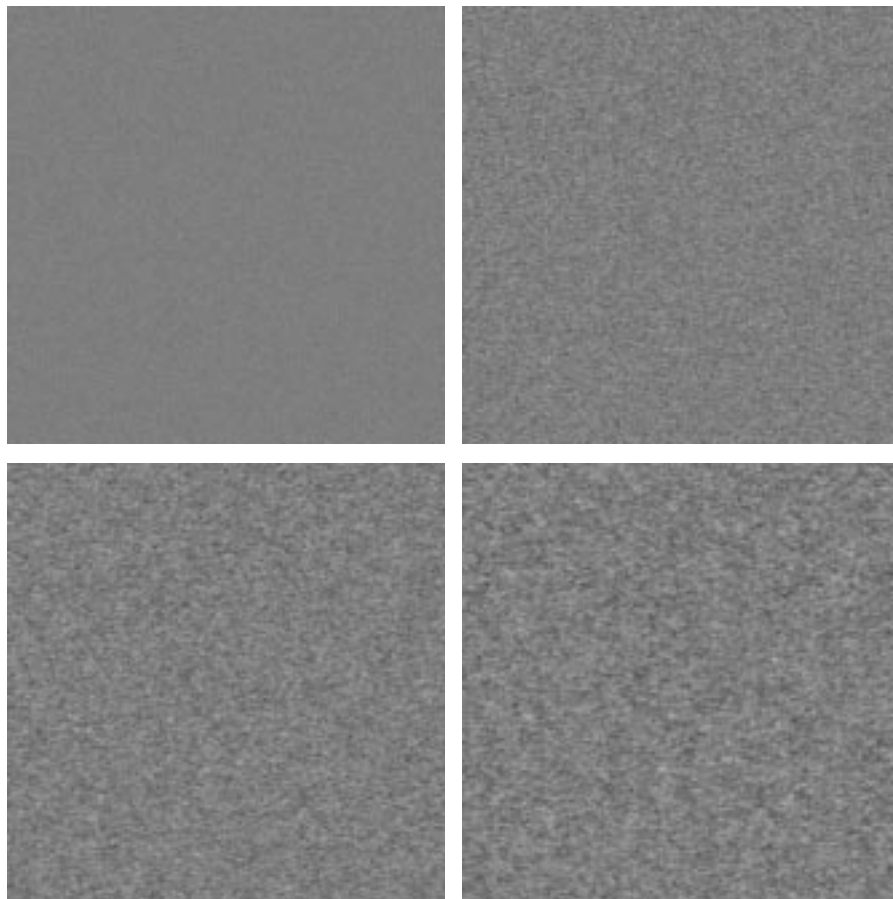


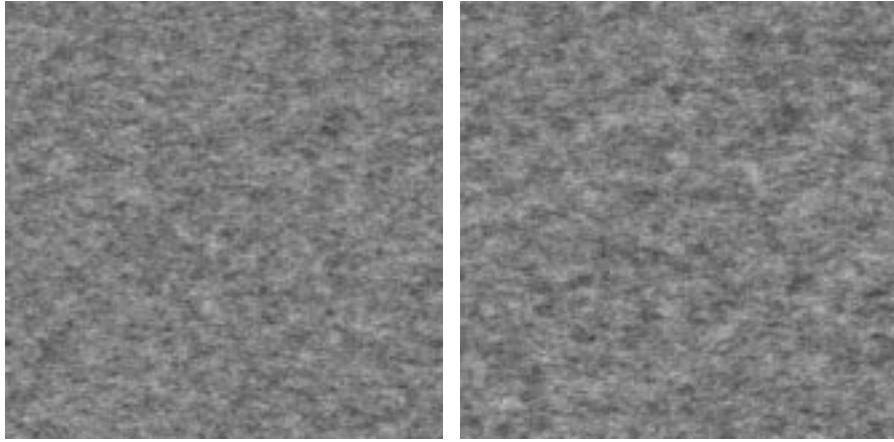
Figure 3.4: Comparison of the convergence behavior of the ML estimate of σ for various GGMRF simulations with $\sigma = 0.2$ and $1.1 \leq p \leq 1.9$, with clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$.



a	b
c	d

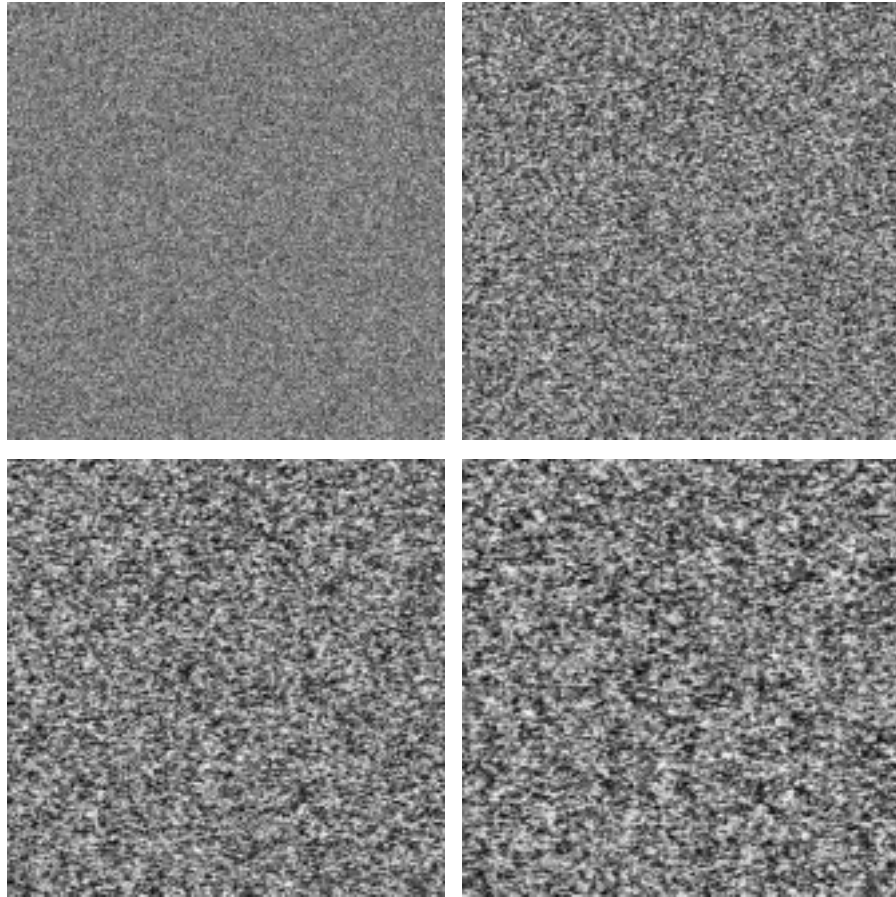
Figure 3.5: Sequence of images in generation of 1st order GMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.

figures 3.11 and 3.12.



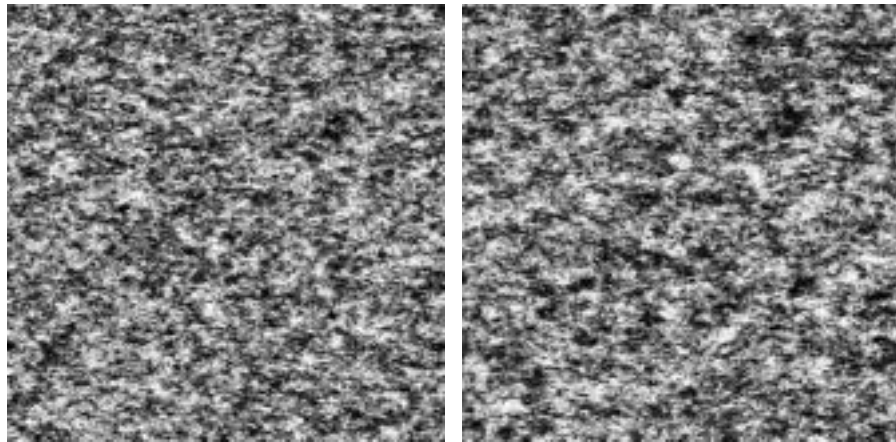
a **b**

Figure 3.6: Sequence of images in generation of 1st order GGMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.



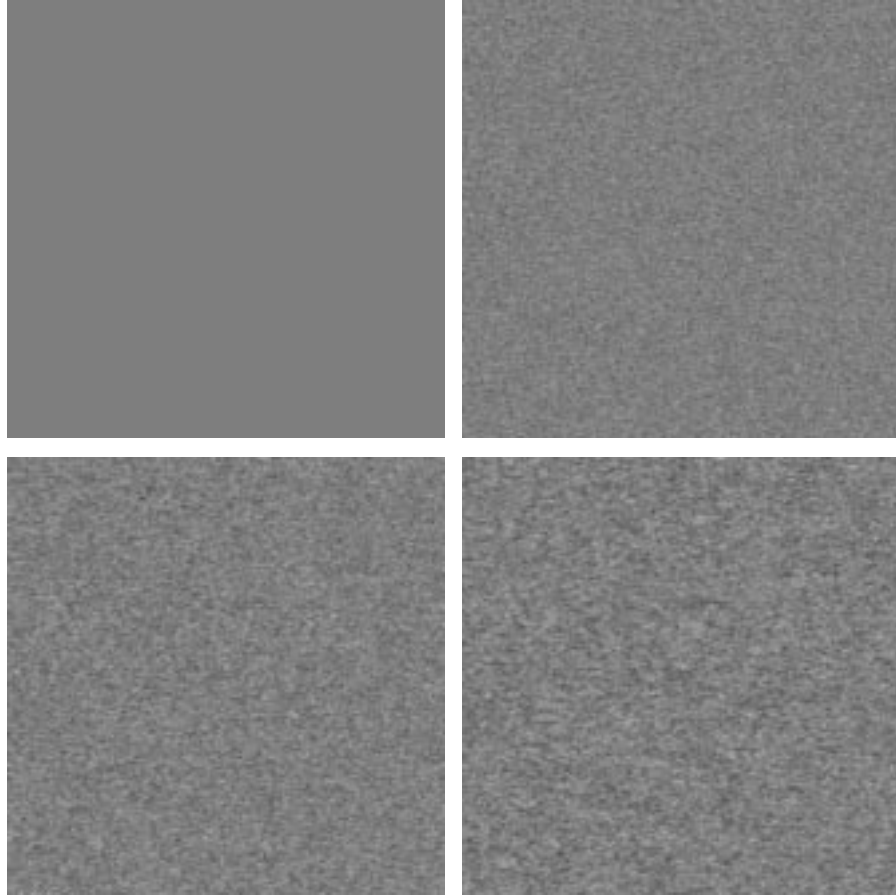
a	b
c	d

Figure 3.7: Sequence of histogram equalized images in generation of 1^{st} order GMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.



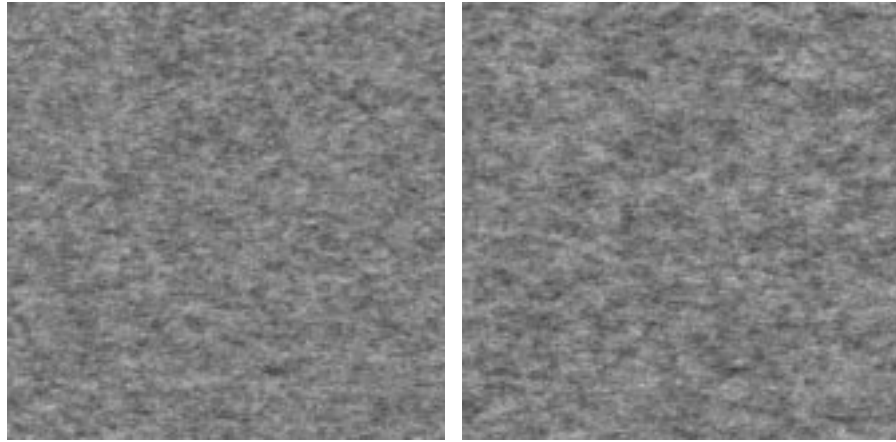
a **b**

Figure 3.8: Sequence of histogram equalized images in generation of 1st order GGMRF realization with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.



a	b
c	d

Figure 3.9: Sequence of images in generation of 1st order GGMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.



a|b

Figure 3.10: Sequence of images in generation of 1st order GMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.

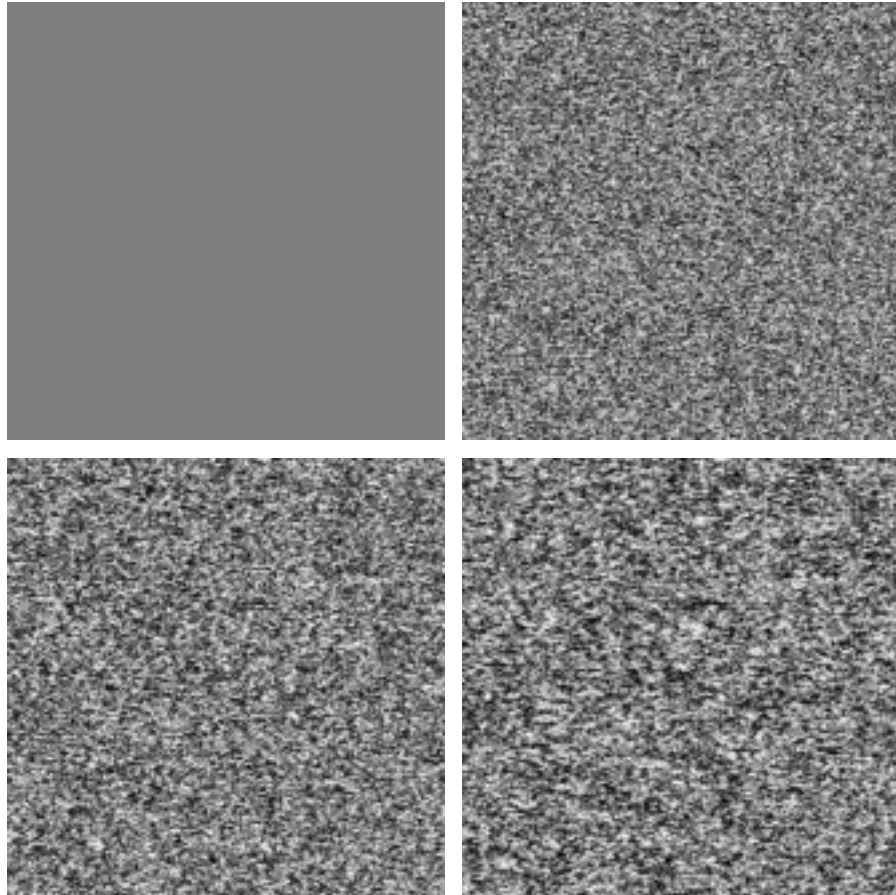
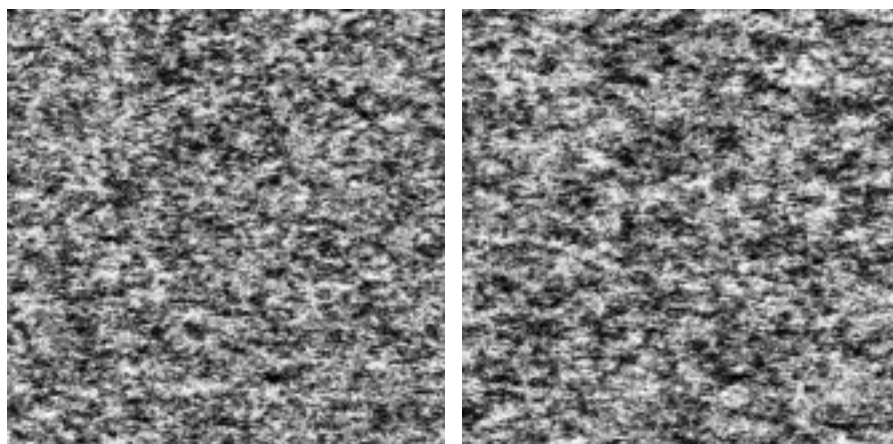


Figure 3.11: Sequence of histogram equalized images in generation of 1st order GGMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) Initial guess. (b) 20 iterations. (c) 50 iterations. (d) 100 iterations.



a|b

Figure 3.12: Sequence of histogram equalized images in generation of 1st order GGMRF realization from constant valued initial condition with $\sigma = 0.2$, $p = 1.1$ and clique weight vector $\frac{1}{8} \cdot [1 \ 3]^T$. (a) 400 iterations. (b) 800 iterations.

Chapter 4

PARAMETER ESTIMATION FOR THE GGMRF

4.1 Introduction

In this chapter, we shall first introduce the general problem of estimation of MRF parameters, before continuing, in section 4.3, to briefly explain why this problem is a non-trivial one. We follow this with a brief discussion of a few of the existing approaches to parameter estimation for MRF's, before investigating in section 4.5, the link between the absolute value MRF and a weighted order statistic filter. Specifically, we show that for the absolute value prior, maximization of the *a-posteriori* distribution of a pixel given its neighbors, is equivalent to weighted order statistic prediction of the pixel value from its neighbor values. With this knowledge, we move on to section 4.6 where we propose our method for estimation of the clique weight vector from realizations of a MRF. In section 4.8 we apply our “minimization of prediction error” (MPE) method to estimation of the clique weight vector for the absolute value MRF, discovering many characteristics of the method along the way.

Having examined the clique weight parameter estimation algorithm for the absolute value prior, we introduce, in section 4.9, the idea of a generalized weighted order statistic predictor. The equivalence between MAP estimation of a pixel for the

generalized Gaussian MRF and the generalized weighted order statistic predictor is then demonstrated, which enables us to extend the principle of parameter estimation by minimization of prediction error to the case of the generalized Gaussian MRF in section 4.10.

This innovation leads to further questions as to the manner in which to minimize the prediction error functional in order to estimate GGMRF clique weight vectors. Two approaches are examined in section 4.10, namely gradient descent, which attempts to minimize the prediction error by descending the prediction error functional and simulated annealing, a stochastic gradient descent method. We conclude the section and this crux chapter with some tentative speculation as to other possible methods for solving the optimization problem inherent in parameter estimation by MPE and by introducing the possibility of extending the MPE to other classes of Markov random fields.

4.2 Parameter Estimation for MRF's

As we have seen in chapter 2, MRF's tend to have a number of parameters which may be adjusted to obtain differing field behavior. It is not an uncommon situation to obtain data known, or at least assumed, to be a realization from a MRF, the exact parameters of which are, however, unknown. In *parameter estimation*, we seek to estimate the parameters of the unknown MRF model from the observed data. The parameters which are often of interest, are the scale parameter of the model, the shape parameter, and in this thesis, the clique weight parameters. Note that if the observed data is some image, then estimating parameters under the assumption that the data derives from a MRF is, in effect, finding the parameters of the MRF which best fits the observed data.

4.3 Difficulty of Parameter Estimation for MRF's

Having discussed the essentials of Markov random fields in chapter 2, the question now arises as to why it is difficult to estimate parameters for MRF's since, according to the Hammersley Clifford theorem, a MRF is nothing more than an exponential distribution, in which case there is a substantial body of knowledge where estimation is concerned [5].

Pickard [27], made the following illuminating comment on the parameter estimation problem: "Likelihood inference is, therefore, apparently straightforward. The requisite normalizing constants, however, are obstreperous." This is the essence of the matter. It is true that the PDF of a Markov random field is of a simple form, but we quickly run into the problem of the partition function which hinders direct attempts at parameter estimation. Consider for example a direct approach of ML estimation for a parameter of the MRF. Recall from chapter 2, equation 2.11 that we may write the MRF in the form,

$$\mathcal{P}(\mathbf{Z}) = \frac{1}{K_p} \exp \left\{ -\frac{1}{\beta} \sum_{c \in \mathcal{C}_2} w_i \cdot \rho_\alpha (\mathcal{A}_c(\mathbf{Z})) \right\}. \quad (4.1)$$

Consider as a specific example, the problem of ML estimation of the weight parameter w_1 . We set up the ML approach as,

$$\hat{w}_1 = \arg \max_{w_1} \{ \mathcal{P}(\mathbf{Z}|w_1) \}, \quad (4.2)$$

which may be reduced to finding,

$$\hat{w}_1 = \arg \max_{w_1} \left\{ \ln \left(\frac{1}{K_p} \right) - \frac{1}{\beta} \sum_{c \in \mathcal{C}_2} w_1 \cdot \rho_\alpha (\mathcal{A}_c(\mathbf{Z})) \right\}. \quad (4.3)$$

Ostensibly, this problem appears innocuous. However, once it is noted that the constant K_p , the *partition function*, is an integral over all states of the field and is thus also a function of w_1 , the difficulty of a direct approach becomes apparent.

Often, since the partition function is unknown and is generally infeasible to compute, many common estimation approaches do not meet with success. Other approaches must be taken.

4.4 Some Existing Approaches to Parameter Estimation

In this section we mention some approaches to the parameter estimation problem for MRF's. This is meant to be but a brief mention of some references and techniques. We concentrate on pairwise interaction processes and often specifically convex models.

The book by Stoyan and Stoyan [37], covers geometrical statistics in general, but has useful sections on Gibbs random fields and statistical inference for those fields. Besag [4] with its attendant discussion, is a useful reference for both the MAP restoration problem as well as ideas on parameter estimation. Pickard [27], gives a detailed introduction to the parameter estimation problem and presents some results for the celebrated binary Ising model, which hails from statistical physics and the attempt to explain spontaneous magnetization.

A recent attempt to simultaneously estimate both the scale and the shape parameter of the Huber MRF was reported by Schultz et. al. in [35]. Their approach hinges on the ability to approximate the partition function of the MRF. These authors have not as yet attempted to apply their methods to clique weight parameter estimation.

In section 3.4, we utilized the results of the paper by Bouman and Sauer [8] who showed that the closed form ML estimate of the scale parameter for the GGMRF may be found given the shape parameter p . This work has recently been extended

[32] to estimating both the scale and shape parameters for the GGMRF.

It is interesting to note that few attempts have been made to estimate clique weight parameters for MRF's. It appears that, in general, authors who use MRF's in Bayesian estimation problems, (image reconstruction and image restoration for example) choose the clique weight parameters on an ad-hoc basis. By estimating these parameters from realistic sample solution imagery, we expect that an improvement in Bayesian estimator performance will be possible as a result of the use of more appropriate and realistic image models.

4.5 The Absolute Value MRF and Weighted Order Statistic Filters

In this section we gain some insight into the relationship between the absolute value MRF (AVMRF) model, and a *weighted order statistic* (WOS) filter, or *weighted median filter*.

Recall from section 2.3 that the AVMRF has a PDF of the form

$$\mathcal{P}(\mathbf{Z}) = \frac{1}{K_p} \exp \left[-\frac{1}{\beta} \sum_{c \in \mathcal{C}_2} w_i \cdot \rho_\alpha (\mathcal{A}_c(\mathbf{Z})) \right], \quad (4.4)$$

where $\rho_\alpha(\cdot) = |\cdot|$, where i numbers each unique type of clique pair in the set of pair cliques \mathcal{C}_2 . Given such a MRF, consider the problem of finding the MAP estimate of a single pixel's value given the values of its neighbors. MAP estimation requires the *maximization of the a-posteriori distribution* $\mathcal{P}(Z_k | \mathbf{Z} \setminus Z_k)$, that is, the conditional PDF of Z_k given all image pixels excepting Z_k itself. This *a-posteriori* distribution may, after applying the Markov property of the MRF, that is, $\mathcal{P}(Z_k | \mathbf{Z} \setminus Z_k) = \mathcal{P}(Z_k | Z_j \text{ for } j \in \mathcal{N}_k)$, be reduced to,

$$\mathcal{P}(Z_k | \mathbf{Z} \setminus Z_k) = \frac{1}{K_p'} \exp \left[-\frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_i |Z_k - Z_j| \right], \quad (4.5)$$

where K_p' is a new normalization constant. Thus finding the MAP estimate of the k^{th} pixel's value given all other image pixel values, is identical (by virtue of the Markov property of the MRF) to finding the MAP estimate of the k^{th} pixel's value given only its neighbors. The MAP estimate of the k^{th} pixel's value, denoted as \hat{Z}_k , requires the solution of the following equation:

$$\begin{aligned}
\hat{Z}_k &= \arg \max_{\zeta} \left\{ \frac{1}{K_p'} \exp \left[-\frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j| \right] \right\} \\
&= \arg \max_{\zeta} \left\{ \ln \left(\frac{1}{K_p'} \right) - \frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j| \right\} \\
&= \arg \max_{\zeta} \left\{ -\frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j| \right\} \\
&= \arg \max_{\zeta} \left\{ - \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j| \right\} \\
&= \arg \min_{\zeta} \left\{ \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j| \right\} \tag{4.6}
\end{aligned}$$

It is also interesting to note [7] that in the MAP estimation framework discussed in section 2.2.1, under the assumption of additive, white, Laplacian noise in the image observation model and the use of the AVMRF prior, the sequential pixel updates used to find the MAP estimate are equivalent in form to a weighted median filter.

We shall now take what appears to be a detour and examine *weighted order statistic* or *weighted median* filters. At the end of this detour, it is hoped that the reader will be enriched by the understanding of the equivalence between a weighted order statistic filter and the MAP predictor of a pixel's value in equation 4.6.

In the most familiar instance of a weighted median filter, we find the filter output as the median value of a list of numbers constituted by replication (the multiplicity determined by the filter weights) of each of the input data values. This definition of a

WOS filter fails when there is an even number of values in the list (since the median is no longer a unique value), or when the weights are not integer values, or have no common factors (since we cannot replicate a number a non-integer number of times). We may avoid these inconveniences if the WOS filtering problem is formulated as one of finding the position of the minimum of a function of the data and the weights, as shown in equation 4.7 below. In this equation, we assume that there are K data values x_i and associated weights w_i where $1 \leq i \leq K$ and K is any integer greater than zero. We assume from here on that $w_i \geq 0$, so as to ensure that the right hand side of 4.7 is a convex function.

$$x_{wos} = \arg \min_x \sum_{i=1}^K w_i |x - x_i| \quad (4.7)$$

Some thought will convince the reader that the two approaches to WOS filtering yield the same results in familiar situations, but that the minimization formulation is indeed more general. To assist in understanding this concept, it may help to initially assume that $w_i = 1, 1 \leq i \leq K$ and that K is odd. Under these conditions, equation 4.7 is identical to a non-weighted median filter of an odd number of values. Equation 4.7 is then a superposition of K absolute value functions with the origin of the i^{th} absolute value function shifted to x_i . Assume initially that the data values x_i are mutually distinct (It is an easy matter to remove this restriction once the basic idea is understood). It should be clear (a simple sketch should verify this) that under these circumstances, the x value at which the superposition of absolute value functions reaches its minimum will be at that data value which has as many data points to the left of it, as to the right. This value is of course the median. With this understood, it is an easy matter to extend the concept to the general case.

Notice that the minimization definition of the WOS filter resolves the problem of

even numbers of arguments since an interval can satisfy equation 4.7. Furthermore, there is no problem presented by the appearance of weights which are non-integer values. Finally, this definition of a WOS filter will be of value to us when we consider the case of the GGMRF, where we will introduce a “generalized” weighted order statistic filter, which includes equation 4.7 as a special case, in the same fashion as the GGMRF contains, for $p = 1$, the AVMRF.

We are now in a position to illuminate the relationship between the AVMRF and the WOS filter. Examine equation 4.6, which is the MAP estimate of a pixel value given its neighbors for an AVMRF. Compare this equation to equation 4.7, which is the defining equation for a weighted order statistic filter. Indeed the two are identical in form.

Since we consider equation 4.6 to be a predictor of the value of a pixel, we are equally entitled to view 4.7 in a similar light, given the equivalence of form between the two equations. Henceforth, we shall discuss the weighted order statistic or median *predictor*.

4.5.1 Graphical Solution for the AVMRF MAP Pixel Predictor

In this section, we show graphical examples of how to compute the MAP estimate of a pixel given its neighbors, assuming an absolute value MRF. As we discussed in the previous sections, computing the MAP pixel estimate involves the solution of equation 4.6, which is a minimization of the negative logarithm of the conditional PDF of the pixel given its neighbors. The upper graphs in figures 4.1 and 4.2 are example plots of the negative logarithm of the conditional PDF (with scaling factors removed) for two distinct AVMRF neighborhoods. Neighbor values are shown as

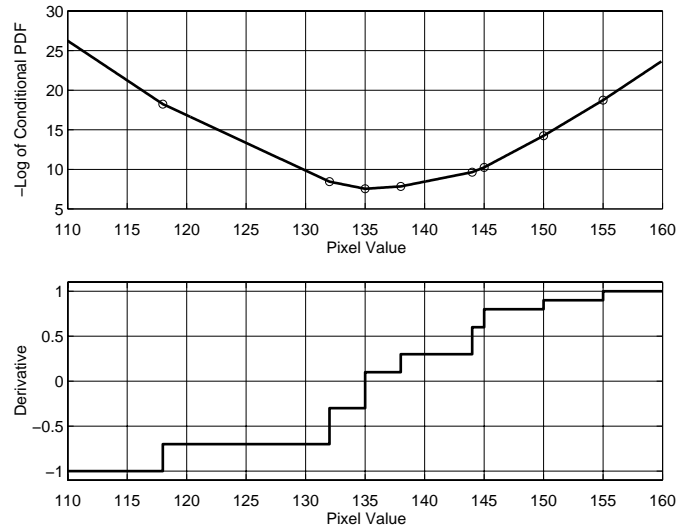


Figure 4.1: Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is a unique value.

circles in the conditional PDF plot. The weight value associated with each neighbor is proportional to the change in slope of the conditional PDF plot at the given neighbor value. To assist in the minimization process, we also plot the *derivative* of the negative logarithm of the conditional PDF (the lower plots in figures 4.1 and 4.2). Notice the simple form of the derivative functions. The neighbor values may be read off as the positions of the jump discontinuities whose heights are proportional to the associated weight values. MAP estimation of a pixel given its neighbors thus reduces to solving for the zeroes of the derivative function. In figure 4.1, this solution is unique, whereas in figure 4.2 the solution is *not* unique, but assumes a range of values, all of which are equally “optimal” in the sense that they satisfy equation 4.6.

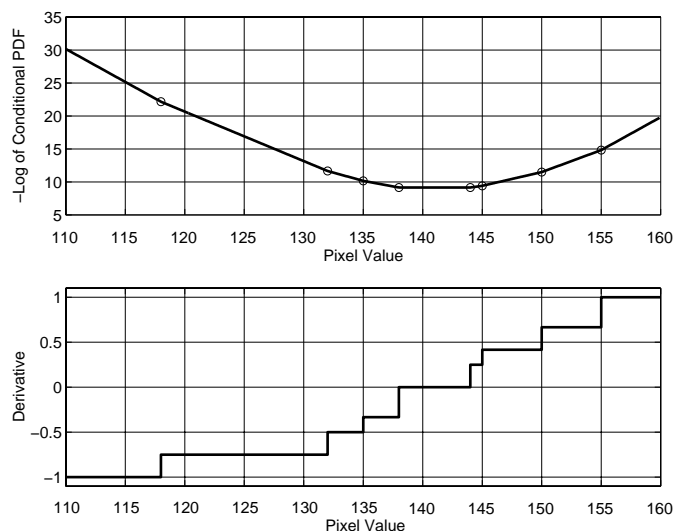


Figure 4.2: Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is an interval of values.

4.6 Clique Weight Estimation by Minimization of Prediction Error

The question we will now address is, *given a single image observation, is it possible to estimate the clique weight parameters of the absolute value MRF that best fits the data ?*

We have shown that the MAP choice of a pixel given its neighbors may be viewed as a *prediction* based on the known neighbors and known model parameters. Consider now the scenario where the *model clique weight parameters are unknown, while the neighbor values, as well as the value of the pixel which is to be predicted from its neighbors, are known.* This is indeed the situation we have when we are given a realization from an AVMRF whose clique weight parameters are unknown. We have knowledge of the value of *every* pixel in the realization image.

Would it not be useful to attempt to seek the parameters of the weighted median

predictor which tends to minimize the error between the weighted median prediction of a pixel and the actual value of that pixel (known from the realization data)? Indeed, this is the approach we have chosen, seeking the values of the clique weight parameters which minimize the error between the MAP prediction of a pixel given its neighbors and the known value of that pixel from the realization. We call this optimization problem “*minimization of prediction error*” (MPE). We use the term “*minimization of prediction error estimation*” (MPEE) to refer to the use of the MPE principle for estimation.

The MPEE procedure which we propose was born of an investigation into techniques for selecting weight values for weighted order statistic filters which are optimally matched to a given data set. This avenue of investigation was initially pursued when it became clear that there was an intimate relationship between the AVMRP and weighted median filters. The well documented approach by Nuevo, Gabbouj, Lin, Yang and Yin [12, 40–44] is to minimize the error between the output of the weighted median filter and the given data set, under some set of “structural constraints” . The structural constraints specify characteristics of the root signal of the weighted median filter, that is, the type of signal features which are preserved under median filtering. Their approach leads to a constrained minimization problem, which can often be reduced to a linear programming problem.

Returning to MPEE, we begin our exposition by formalizing the procedure. We denote the known realization image vector as Z and the predicted image vector as \hat{Z} , which is a function of the model parameters. Since our interest is primarily in the weight parameters, we shall denote \hat{Z} as $\hat{Z}(W)$. We may thus describe the MPEE

approach to finding the optimal weights W_{opt} as,

$$W_{opt} = \arg \min_W \{ \mathcal{E} [\mathcal{C} (\hat{Z}(W) - Z)] \}, \quad (4.8)$$

where \mathcal{E} is the expectation operation and \mathcal{C} is a cost function applied to the pixel prediction error. In the following section, we shall discuss various choices for this cost function in greater detail.

4.7 Choice of the Prediction Error Cost Function

In this section we address the question of the choice of the cost function \mathcal{C} applied to the pixel prediction error, in equation 4.8. There is a great deal of freedom in the choice of the prediction error cost function, so we present various of the alternatives investigated. The cost functions we considered were chosen on the basis of their equivalence with existing methods, as well as for their attractiveness for our particular problem.

Certain of these cost functionals, particularly the absolute value cost and the square cost have found application in the problem of optimal weight selection for WOS filters [40, 42–44]. These authors do not offer strong justification for their choice of the absolute value cost function, but utilize the square cost functional for computational convenience after showing that, assuming binary valued data, the square and absolute value cost functionals are equivalent.

Figure 4.3 shows a number of candidates for the role of the prediction error cost function, each of which is discussed in the sections that follow. Note that in our discussion, the pixel prediction error is denoted as ϵ .

Absolute Value Cost

If we choose the cost \mathcal{C} as,

$$\mathcal{C}(\epsilon) = |\epsilon|, \quad (4.9)$$

then after substitution into equation 4.8, we have the estimation optimization of the form,

$$W_{opt} = \arg \min_W \left\{ \mathcal{E} \left(\left| \hat{Z}(W) - Z \right| \right) \right\}, \quad (4.10)$$

This should be recognized as the familiar estimation procedure which minimizes the mean absolute error (MMAE). Assuming ergodicity, replacing the ensemble average with the spatial average over the point set \mathcal{S} , we may write this as,

$$W_{opt} = \arg \min_W \left\{ \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left| \hat{Z}_i(W) - Z_i \right| \right\}, \quad (4.11)$$

where $\hat{Z}_i(W)$ is the MAP prediction for the i^{th} pixel value. $|\mathcal{S}|$ is the cardinality of the point set \mathcal{S} .

Square Cost

Another common choice for a cost functional is,

$$\mathcal{C}(\epsilon) = |\epsilon|^2. \quad (4.12)$$

For this squared cost functional, the problem of estimating the clique weights becomes,

$$W_{opt} = \arg \min_W \left\{ \mathcal{E} \left(\left| \hat{Z}(W) - Z \right|^2 \right) \right\}. \quad (4.13)$$

This is identical to minimizing the mean of the squared error, the familiar MMSE estimation technique. Again, under the ergodicity assumption,

$$W_{opt} = \arg \min_W \left\{ \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left| \hat{Z}_i(W) - Z_i \right|^2 \right\}. \quad (4.14)$$

Zero / One Cost

An interesting choice for the prediction error cost function is the zero/one cost function,

$$\mathcal{C}(\epsilon) = \begin{cases} 0, & |\epsilon| \leq \alpha, \\ 1, & |\epsilon| > \alpha. \end{cases} \quad (4.15)$$

This cost function has no penalization for MAP pixel predictions \hat{Z}_i which fall a distance α or less from the known realization pixel value Z_i , but penalizes all predictions which fall outside of this range equally, with unity cost. Once again, we may write out the clique weight estimation procedure as,

$$W_{opt} = \arg \min_W \left\{ \mathcal{E} \left(\left\{ \begin{array}{l} 0, \quad |\hat{Z}(W) - Z| \leq \alpha, \\ 1, \quad |\hat{Z}(W) - Z| > \alpha. \end{array} \right\} \right) \right\}, \quad (4.16)$$

and assuming ergodicity,

$$W_{opt} = \arg \min_W \left\{ \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \left\{ \begin{array}{l} 0, \quad |\hat{Z}_i(W) - Z_i| \leq \alpha, \\ 1, \quad |\hat{Z}_i(W) - Z_i| > \alpha. \end{array} \right\} \right\}. \quad (4.17)$$

So far we have described our pixel predictor as the MAP pixel predictor. This is only true if we use the *true* clique weight parameters. Since we do not know the true parameters (we are trying to estimate them), what we are in effect doing is formulating predictors with the equivalent form as the MAP pixel predictor and testing their performance using various cost functions. The MAP pixel predictor, however, is exactly that predictor which yields minimum risk using the zero/one cost function for $\alpha \rightarrow 0$ [5]. We are therefore assured that if we find a consistent estimator of minimum risk using the zero/one cost function as $\alpha \rightarrow 0$, we have found the MAP estimator and therefore the exact clique weights. We therefore seek a consistent estimator for $\alpha \rightarrow 0$. Unfortunately it is not possible to achieve this since no matter how much we increase the data size, as we make $\alpha \rightarrow 0$, the variance of the prediction error does not tend to zero.

Piecewise Linear Cost

One obvious problem with the zero/one cost function is that its derivative is zero almost everywhere. This is a major drawback if one is interested in a directed search through the weight space, such as would be the case with a gradient descent type search for the optimal clique weights (to be discussed later in section 4.10.4). In order to avoid the problem of having no information as to the direction in which the estimate improves, while at the same time maintaining an approximation to the characteristics of zero/one cost function, the following, piecewise linear cost function was tested.

$$\mathcal{C}(\epsilon) = \begin{cases} \frac{1}{\alpha}|\epsilon|, & |\epsilon| \leq \alpha, \\ \beta|\epsilon| + (1 - \alpha\beta), & |\epsilon| > \alpha. \end{cases} \quad (4.18)$$

Like the zero/one cost function, the piecewise linear cost function has a central region and an outer region. Prediction errors ϵ which fall in the range $|\epsilon| \leq \alpha$ (the central region) are subject to a linearly increasing penalty from a minimum of zero for no prediction error, to unity for $|\epsilon| = \alpha$. The slope of the central region linearly increasing penalty is solely determined by the choice of α . A prediction error falling in the outer region for which $|\epsilon| > \alpha$ incurs a linearly increasing (slope determined by β) penalty with a minimum of unity as $|\epsilon| \rightarrow \alpha$. In order to make the piecewise linear cost function a reasonable approximation to the zero/one cost function, we must choose small values for both α and β .

Power Cost

A problem with the piecewise linear approximation to the zero/one cost function is that its derivative is discontinuous. This problem is addressed by the introduction

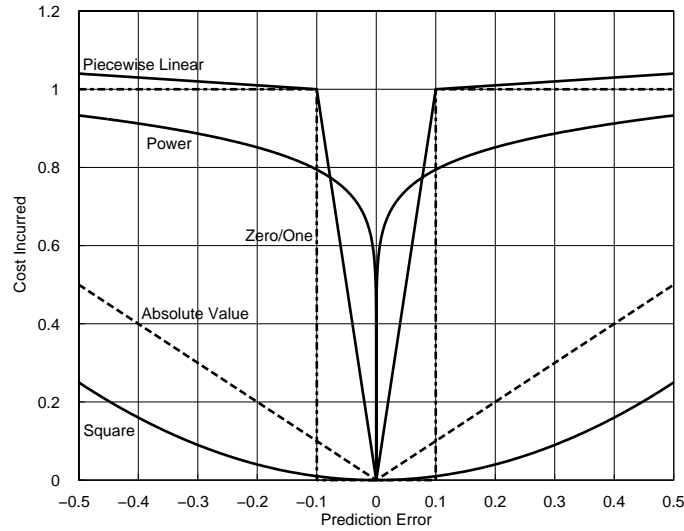


Figure 4.3: Cost functionals used for penalization of prediction error in MPE estimation. The cost functionals include the absolute value cost, squared cost, zero/one cost ($\alpha = 0.1$), power function cost ($\gamma = 0.1$), and piecewise linear cost ($\alpha = 0.1, \beta = 0.1$).

of the power law cost function described by:

$$\mathcal{C}(\epsilon) = |\epsilon|^\gamma, \quad 0 < \gamma < 1. \quad (4.19)$$

As $\gamma \rightarrow 0$, the power cost function is an increasingly good approximation to the zero/one cost functional. The derivative of the power cost function is well behaved everywhere except at the origin. This function is better suited to a gradient descent type approach to solving the optimization for the clique weights. Use of the power cost function leads to the following optimization problem:

$$W_{opt} = \arg \min_W \left\{ \mathcal{E} \left(\left| \hat{Z}(W) - Z \right|^\gamma \right) \right\} \quad (4.20)$$

4.8 Minimization of Prediction Error for the AVMRP

The minimization of prediction error principle was initially applied to the problem of estimating the clique weight parameters for an absolute value MRF. The choice

to attempt estimation for the AVMRF in particular, was made on the basis of our understanding of the form of the MAP predictor of a pixel given its neighbors for this model. In MPEE for the AVMRF, we attempt to apply equation 4.8 to realizations of an AVMRF. Equation 4.8 apparently requires that we search the space of all possible clique weights to find the set of weights which results in the lowest *risk*. By risk, we refer to the expectation of the penalized prediction error (see appendix A for a precise definition). In effect, this search amounts to testing various predictors in an attempt to find the one which results in the lowest risk. The predictor(s) which possesses the property of having the lowest risk amongst their peers is deemed optimal in the MPE sense.

It will be necessary to gain an understanding as to how risk is affected by a change in the choice of clique weights of the model under test. Without an understanding of this sort, it becomes necessary to search the entire clique weight space, which is clearly infeasible. Let us, therefore, attempt to gain an understanding of the behavior of the risk as a function of the choice of clique weights for the predictor model $\hat{Z}(W)$.

First, notice that the total risk is a weighted summation of the penalized prediction error for each pixel in the image. It is thus possible to gain an understanding of the global behavior of the risk by examining the behavior of the solution to each neighborhood prediction. We are therefore concerned with the manner in which the prediction error varies as a function of the choice of clique weights for a single neighborhood prediction. Understanding this mechanism for a single, general neighborhood, allows us to predict the type of behavior that can be expected when the contribution of all neighborhoods is taken into account when computing the total prediction risk.

4.8.1 MAP Pixel Prediction Behavior as a Function of Clique Weight

As a tutorial example to assist in understanding the behavior of the prediction and hence the prediction error for a single neighborhood, we will consider the case of a first order predictor. In the first order neighborhood, the MAP prediction of the center pixel's value given its neighbors is a function of the values of the pixel's 4-neighbors, and the 4 clique weights associated with those neighbors. The reader is referred to figures 4.4 and 4.5 for graphical examples of two such neighborhoods.

In these figures, the values of the neighbor pixels are marked as circles in the upper plot of the negative log of the conditional PDF of the pixel given its neighbors. Notice that in figure 4.4 the optimal prediction is an interval of pixel values. This is also evident in the associated derivative plot where the minimum of the negative log conditional PDF corresponds to the range of the zero equality of the derivative. In figure 4.5 however, the optimal prediction is unique.

The numbering of the clique weights in the graphs follow the ordering of the pixel values, not our usual convention of numbering unique clique types. The smallest pixel value is labeled x_1 with corresponding weight w_1 up to the largest, x_4 with weight w_4 . For a given set of clique weights, different choice of neighborhoods will result in a different ordering of the clique weights. This should be clear by comparing figures 4.4 and 4.5 where the clique weights are unchanged, but their ordering is affected by the ordering of the neighbor pixel values. This is a result of the fact that the actual clique weights are associated with the *position* of the neighbor relative to the pixel to be estimated (see section 2.3).

The clique weights also occur in symmetric pairs, as the MRF is stationary. There are 4 neighbors x_i and 4 clique weights w_i , or 2 clique weight pairs. Also it

is required that the sum of the clique weights is unity. For a set of clique weights to be valid, they must conform to these constraints.

As an example of formulating a consistent weight set, consider figure 4.4. In the figure $w_1 = w_4$ and $w_2 = w_3$ by symmetry. Also the sum of all the weights is seen to be unity.

Consider now the effect of varying, say, w_1 in the figure. A first observation is that w_4 must follow suite, since $w_4 = w_1$. Satisfaction of the unity sum of weights normalization requires that the remaining weights, w_2 and w_3 , must make up the difference between $w_1 + w_4$ and unity. Furthermore, since w_2 and w_3 are a symmetry pair and therefore equal, they must be adjusted equally when ensuring that the sum of the weights is unity.

Let us now turn our attention to the actual problem of determining the behavior of the MAP pixel prediction value as a function of the clique weights. In figures 4.4 and 4.5 the clique weights are explicitly marked and are seen to be proportional to the height of the jump discontinuities in the derivative plots.

With reference to figure 4.4, consider the effect on the MAP prediction as the value of w_1 is varied under the constraints discussed above.

Some thought will show that the MAP prediction (the values for which the derivative function is zero) will remain the same as long as $0 \leq w_1 < 0.5$. Applying similar logic shows that when $w_1 = 0.5$, the solution interval ranges from x_1 to x_4 . For the particular configuration of neighborhood shown in figure 4.4, varying any weight will not affect the solution unless the weight changes to, or from, 0.5. For any other values, the solution remains the same. Already it appears that it may be unnecessary to check the solution for every possible weight configuration since the solution remains constant over predictable ranges of the weights.

The example in figure 4.4 is, however, something of a special case since the weights occur in an ordering starting with w_1 , $w_2 = 0.5 - w_1$, which means that the 2nd pair of weights have the relationship $w_3 + w_4 = 0.5$. This is of course not the only possible configuration. There is another “interesting” possibility for a first order neighborhood, that is, to have a set of pixel values which results in the weight ordering $w_1, w_2 = w_1, w_3 = 0.5 - w_1, w_4 = w_3 = 0.5 - w_1$. This situation is shown in figure 4.5. For this set of neighbors there are a greater number of possible solutions.

In figure 4.5, we have $w_1 = w_2 < 0.25$, which results in a unique solution to the optimization problem at x_3 . From the figure, it should be clear that provided $0 < w_1 < 0.25$, the solution will remain at x_3 . Consider now what occurs if $w_1 = 0$. In this case the solution becomes the interval between x_3 and x_4 . If we take $w_1 = 0.25$, then the solution is the region between x_2 and x_3 . With $0.25 < w_1 < 0.5$, the solution exists uniquely at x_2 . Finally if we choose $w_1 = 0.5$ then the solution is the interval from x_1 to x_2 .

From the two example figures 4.4 and 4.5, it is possible to draw some conclusions concerning the general behavior of the solution as a function of the clique weights for this first order neighborhood. These conclusions are claimed to be general for the first order model since the two examples considered actually encompass all possible orderings of the clique weights. The behavior of the solution to the MAP pixel choice, as a function of a single varying independent weight, is summarized in table 4.1. From this table, it should be clear that since the solution does not vary for all changes of the independent clique weight w , it is unnecessary to compute the MAP pixel prediction for all values of the weights. Indeed, we need only examine the solution at $w = 0, 0.25, 0.5$ and also at any single point inside the interval $0 < w < 0.25$, and the interval $0.25 < w < 0.5$. Since we have complete freedom to choose which points in

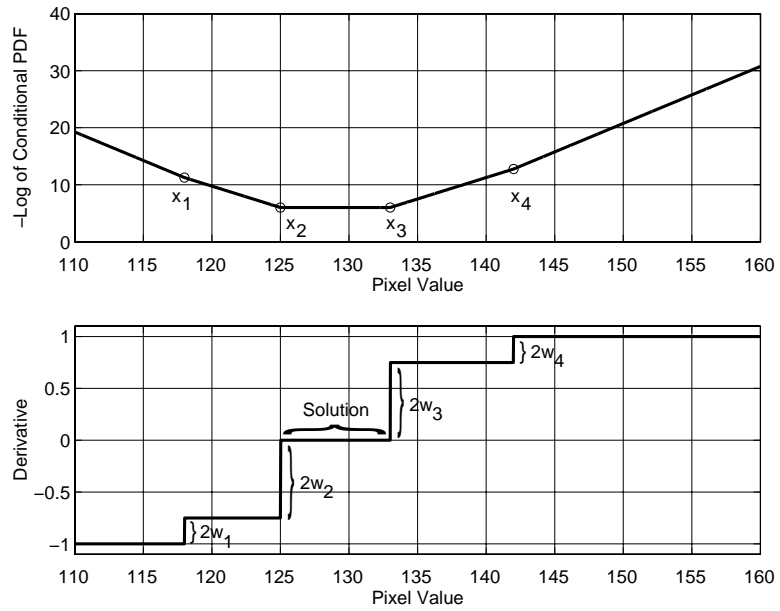


Figure 4.4: Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is an interval of values.

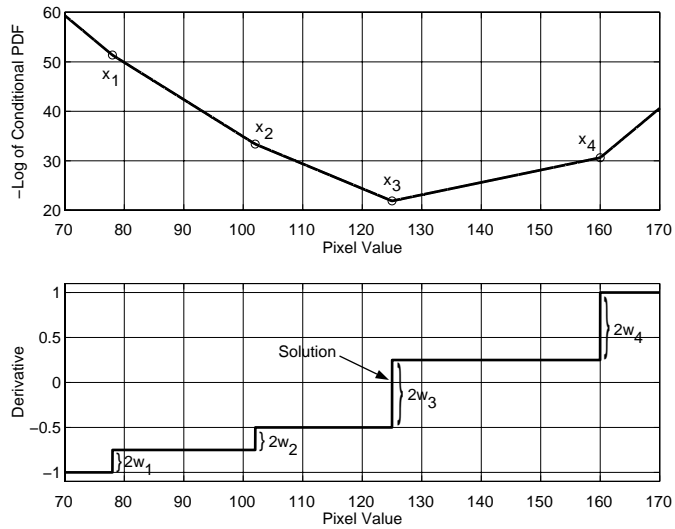


Figure 4.5: Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below), in an instance where the optimal prediction is a unique value.

the intervals to choose, we will choose points that are convenient. If we take $w = \frac{k}{2N}$, where N is the number of neighbors, and k is an integer with $0 \leq k \leq N$, then we have a simple set of points $\{0, 0.125, 0.25, 0.375, 0.5\}$, which *completely* samples all possible behavior of the MAP predictor solution. We tentatively suggest that the spacing $\frac{k}{2N}$ is general for all orders of predictors. We shall return to this idea later.

Weight Value	Solution Behavior
$w = 0$	Possible change
$0 < w < 0.25$	Constant
$w = 0.25$	Possible change
$0.25 < w < 0.5$	Constant
$w = 0.5$	Possible change

Table 4.1: Behavior of the solution to the MAP estimate of a pixel given its neighbors for varying values of a chosen independent weight w .

4.8.2 Risk Behavior as a Function of Clique Weight

Knowledge of the behavior of the MAP pixel prediction of a single pixel as a function of the clique weights brings us some distance to understanding the behavior of the *risk* of the complete image prediction as a function of the clique weights. Recall that we seek to utilize equation 4.8 to estimate the “optimal” (in the MPE sense) clique weights to match a given realization. This equation is restated below.

$$W_{opt} = \arg \min_W \left\{ \mathcal{E} \left[\mathcal{C} \left(\hat{Z}(W) - Z \right) \right] \right\}, \quad (4.21)$$

In appendix A we show how, assuming ergodicity we may replace the ensemble average with a spatial average as,

$$W_{opt} = \arg \min_W \left\{ \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathcal{C} \left(\hat{Z}_i(W) - Z_i \right) \right\}. \quad (4.22)$$

Where \mathcal{S} is the set of all image pixels. In this computational formula, we define the risk similar to expected cost as,

$$\mathcal{R} \triangleq \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathcal{C} \left(\hat{Z}_i(W) - Z_i \right). \quad (4.23)$$

The risk is computed as a weighted average of penalized prediction errors, where the penalty is levied by the cost function \mathcal{C} on the error between the value of the MAP prediction of the i^{th} pixel and the actual value of the i^{th} pixel in the realization image. The choice of this penalty function was discussed in section 4.7.

Since we understand the behavior of the MAP prediction for a single neighborhood, we should be able to predict the form of the risk as a function of the clique weights. Since the MAP prediction of any single pixel given its neighbors remains constant over large variations of the clique weights, it should be clear that provided the cost function \mathcal{C} is relatively well behaved (see section 4.7), the risk as defined in equation 4.23, will also exhibit large constant valued regions.

An example of the behavior of the risk as a function of the independent weight parameter of a first order prediction model may be seen in the plot of figure 4.6. This figure confirms our suspicions that the behavior of the risk function closely mimics the behavior of single neighborhood predictions. We observe constant valued regions of risk for the same range of the independent clique weight as we previously observed constant MAP pixel predictions in single neighborhoods. Furthermore, we observe changes in the risk functional only, and exactly, at those points which we identified earlier as transition points for the value of the MAP pixel prediction.

We now move on to address the implications of the behavior of the risk function on the MPE solution.

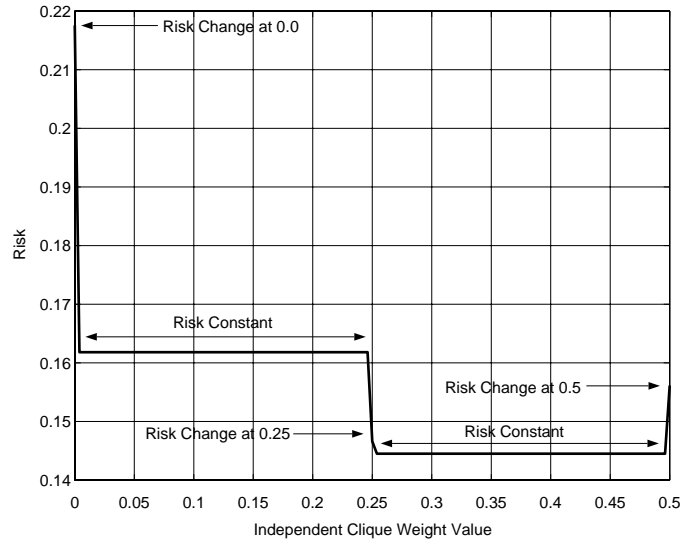


Figure 4.6: Risk as a function of independent clique weight for a 1st order predictor model. The MRF realization for which the prediction risk is computed has dimensions 128 pixels by 128 pixels, with clique weight vector $\frac{1}{6}[0 \ 2 \ 0 \ 1]^T$. The square of the prediction error penalty was used.

4.8.3 Quantization Effect

As discussed in the previous section, the risk function to be minimized in MPEE of the clique weight parameters exhibits constant valued regions. These regions of constant risk imply that all choices of weights which result in the given constant risk value are equally good choices. No particular clique weight set is any better a choice (in a minimum risk sense) than any other clique weight set if the sets have identical risk. If we find, as is the case in figure 4.6, that the risk function reaches a minimum for a range of clique weights, then any one of these clique weight sets is optimal. We term this “range of optimal solutions” behavior the *quantization effect*, since the optimal solution for the clique weights is *quantized* into regions of equal risk. The quantization effect places a limitation on the precision of our estimate for the clique weight parameters.

Recall from section 4.8 that in order to solve for the MAP estimate of a pixel

given its neighbors we only needed to find the prediction value for a discrete number of points in the range of possible weights. The same effect is present in searching for the optimal set of weights in equation 4.22. This is evident in the risk plot in figure 4.6, where it is unnecessary to test the risk at every weight value. In fact, as with the neighbor prediction, we need only check the weights at intervals of $\frac{k}{2N}$. Therefore, we need not search the entire weight space as initially appeared inevitable at the outset of section 4.8. This is possible due to the well understood structure of the MPE risk function for the absolute value MRF. Without this simplification, it is not clear whether or not it would be possible to solve the optimization problem for the absolute value MRF, since there is no gradient information available for a directed search, due to the presence of either discrete point or constant risk interval solutions for the optimal clique weights.

Assuming the use of the convenient $\frac{k}{2N}$ grid search of the clique weight space, it is important not to forget that if the minimum risk solution is found on a weight with $w = \frac{k}{2N}$, where k is odd (a weight which is the center of an interval of optimal weight values), then this solution is *not unique*. There is a range of weight values which satisfy the MPE criterion, namely $0 \leq \frac{k-1}{2N} < w < \frac{k+1}{2N} \leq \frac{1}{2}$. This is simply another way of describing the quantization effect.

4.8.4 Generalization to Higher Order Predictor Models

In the previous sections we utilized the tutorial example of a first order predictor model, but did not address the general case of higher order predictor models. It was claimed, however, that the $\frac{k}{2N}$ search spacing was valid for predictors of higher order. Essentially identical arguments to those used to derive the simpler first order results may be used to demonstrate that the discrete search technique is valid for the case

of higher order predictor models. The $\frac{k}{2N}$ searching strategy is still valid, but in the case of higher order models, there will be more than just a single independent clique weight variable. The previously one-dimensional search, becomes a search on grids in higher dimensional space. The principle, however, is exactly the same, though the computational cost will naturally increase with the power of the dimensionality of the search.

4.8.5 A Problem with the AVMRF MAP Pixel Predictor

Thus far, we have discussed the absolute value predictor model at some length and in theory, all appears well. However, in this section, we shall show that there is a fundamental, though not initially obvious, problem with the AVMRF MAP pixel predictor, which leads to degenerate results when this predictor is used in the estimation of the clique weight parameters by the MPE method. To introduce the problem, the reader is invited to carefully examine tables B.1 through B.4 in appendix B, which display the results of clique weight estimation by our minimization of prediction error optimization procedure using the AVMRF MAP pixel predictor with various cost functionals. The input images to the estimation tests are MRF simulation images (see chapter 3) whose clique weight parameters are known by design.

The reader will notice that although the clique weights in the input MRF simulation images vary significantly, the estimated clique weights are always either 0.0 or 0.5. A careful examination of the tables will show that the estimates *do* tend to follow the general trend of the input MRF clique weights despite the lack of variation from the fixed 0 or 0.5 behavior in the clique weight estimate. It appears therefore, that the clique weight estimates may be vaguely correct in the sense that

they capture the qualitative trend of the input MRF model parameters, but the quantitative behavior is far from satisfactory in that a degenerate “all or nothing” assignment of available clique weight appears to be the trend.

It is hoped that the reader is, at this stage, somewhat perplexed and is asking his or herself why this curious behavior is observed when, in theory, all seemed fine. Indeed, this was initially the author’s predicament.

In order to solve this quandary, we shall have to follow the clues in the simulation tables. The essential point to notice from the tables is that all the available clique weight (unity by convention) is apportioned to a single symmetric pair, rather than being distributed amongst various neighbors as would be expected for input MRF’s which we know do not follow this “all or nothing” type of behavior.

Knowing that MPE estimation of the clique weight vector hinges on the MAP predictions of pixels given their neighbors (the clique weight estimate is exactly the vector of weights associated with the lowest risk MAP pixel predictor for the image), it is wise to closely examine the behavior of the absolute value MAP pixel predictor in the context of choosing the set of weights which tend to minimize the prediction error. Given the “all or nothing” behavior of the clique weight estimates in the tables, it would seem prudent to endeavor to understand the behavior of the MAP pixel prediction if a symmetric weight pair is allotted the *total* available clique weight of unity (0.5 each). To assist in this regard, the reader may find it useful to refer to figures 4.1, 4.2, 4.4 and 4.5, and consider the effect on the predictor solution if a pair of clique weights is assigned a value of 0.5 each. It should be clear from these figures that forcing a pair of clique weights to each have value 0.5 almost always creates a non-unique MAP pixel prediction value since, in this situation, the MAP pixel prediction is an interval. We say almost always since there are pathological

cases where this is not true; consider for example the case where all the neighbor pixels are the same value. Since these cases are rare for ordinary images, we shall not dwell on them.

In the case of a non-unique MAP pixel prediction, every pixel value which falls in the prediction interval is assigned zero cost since all values in the interval are equally good solutions to the MAP pixel prediction problem. Since the risk associated with a particular set of clique weights is nothing more than a weighted sum of the costs for each neighborhood prediction, the lowest risk will be associated with a clique weight vector which leads to interval type MAP pixel predictions. If this is still unclear, notice that those clique weight vectors which lead to unique MAP pixel predictions will incur non-zero prediction error for *all* prediction values which are not the exact solution. Thus we expect that the clique weight vector of lowest risk, (optimal in a MPE sense) will be a vector which exhibits the “all or nothing” behavior we observed from our simulation results.

We have come some distance in explaining the observed behavior of the clique weight estimates. We shall now discuss the qualitative behavior of the estimates.

For input MRF realizations where not all the clique weights are equal, MPE estimation using the AVMRF MAP predictor will tend to place the full clique weight in the position which best approximates the principal orientation of the MRF clique weights. This fact is evident from an examination of the tables in appendix B. For MRF’s with little or no directional bias, we see that the behavior of the estimate is less reliable. See for example the tables for the instances where the simulation MRF’s have equal clique weights (no directional bias). We know that the true solution has equally distributed clique weights amongst the neighbors, but the AVMRF predictor MPE estimation algorithm places all the weight on a particular pixel pair. The result

is an unreliable estimate. The particular weight configuration chosen and thus the particular neighbor orientation to which the full clique weight is assigned, is of course still on the basis of minimum risk. The problem is that the minimum risk solution is highly sensitive to small fluctuations in the distribution of neighbor pixel values, since a number of candidate solutions will yield extremely similar or even equal risk. This leads to misleading, or ambiguous estimates.

We conclude, therefore, that *direct application of the AVMRf MAP pixel predictor, which allows interval solutions, will yield qualitatively correct, but quantitatively degenerate, estimates of the input MRF clique weights.* Furthermore, we have learned that the behavior of the estimator is unreliable for input MRF's which have little or no directional bias.

In this section we have shown that there are problems with the use of the AVMRf MAP pixel predictor for MPE estimation of the clique weight vector. We will propose two solutions to this problem. The first, rather ad-hoc solution, is discussed in the next section. The second, more satisfying but more complex solution, will require us to generalize our predictor model and will be introduced in section 4.9.

4.8.6 A Solution to the AVMRf MAP Pixel Predictor Problem

The degenerate behavior of the clique weight estimates stem from interval type solutions to the AVMRf MAP pixel prediction used in MPE estimation. In this section we propose a simple, somewhat ad-hoc, solution to this problem that nevertheless yields significantly improved clique weight vector estimates. We propose that in the case where the solution to the MAP pixel prediction for a neighborhood is an interval, instead of using the actual interval as a zero cost region, we choose the *midpoint*

of the interval as the *unique* prediction value for that neighborhood. The rationale for this solution is as follows: since we know that the interval type solutions to the AVMRF MAP pixel prediction lead to the degenerate clique weight vector estimate behavior discussed in the previous section, we simply disallow such solutions, forcing the prediction to be unique, with the expectation of securing better behaved estimates as a result.

The natural criticism of the midpoint value approach to correcting the AVMRF MAP pixel predictor problem is that it is somewhat arbitrary and has no obvious claim to optimality. With this objection in mind, let us examine the actual performance of the midpoint value MAP pixel predictor to ascertain whether practice holds true to our expectation of improved estimator performance, or if the apparent arbitrariness of the proposed solution is reflected in a lack of improved estimator performance.

4.8.7 Estimator Performance for Higher Order Models

A consequence of increasing the order of the predictor model in MPEE is an increase in the precision of the estimate of the clique weight parameters. This effect may be understood by noting that as the order of the model increases, so does N , which in turn results in a smaller inherent grid search spacing $\frac{1}{2^N}$. The increased complexity of the predictor model allows for the modeling of more complex risk functions, which match better the true characteristics of the MRF realization from which parameters are to be estimated.

It is important to realize that the improvement in performance as the predictor model order is increased does not imply that we gain information where there was none previously. Consider, for example, the problem of estimating the clique weights

for the first order AVMRP with clique weights $\frac{1}{6}[1\ 2]^T$. The elements of the clique weight vector represent the pair clique weights in the north and west orientations. As usual, the constraints of clique weight symmetry and the unity sum of the clique weights uniquely specify the complete clique weight vector. Using a first order midpoint value AVMRP predictor model, we find that the optimal MPE estimate is $\frac{1}{8}[1\ 3]^T$. This is the minimum risk solution. From our discussion on the quantization effect, we expect the range of solutions for say, the north oriented clique weight to be $\frac{0}{8} < w_{North} < \frac{2}{8}$.

If we repeat the same exercise using the second order predictor in the estimator, we find not one minimum risk solution, but three, namely, $\frac{1}{16}[0\ 1\ 0\ 7]^T$, $\frac{1}{16}[0\ 2\ 0\ 6]^T$ and $\frac{1}{16}[0\ 3\ 0\ 5]^T$. Taking into account the second order quantization effect for the combination these three intervals yields a range of values for the north oriented clique weight as, $\frac{0}{16} < w_{North} < \frac{4}{16}$, which yields no improvement over the the first order case.

If, however, we have a true second order MRF realization with the clique weights of say, $\frac{1}{6}[1\ 0\ 1\ 1]^T$ (pair clique weights in the north-west, north, north-east, and west orientations), then our clique weight estimate using a first order model yields $\frac{1}{8}[1\ 3]^T$, with quantization over $\pm\frac{1}{8}$ whereas second order predictor estimation yields a risk minimum at $\frac{1}{16}[2\ 0\ 3\ 3]^T$, with a quantization over $\pm\frac{1}{16}$.

The second order predictor yields a tighter quantization region, with the result that the MPE estimate of the clique weights is closer to the known true value, if indeed the original data derives from a second order model. We expect that in higher dimensions the same effects will come into play. If the data model order is higher than the predictor model order in the estimator, then a gain in estimate precision may achieved by increasing the predictor order. The disadvantage of increasing

the predictor model order is increased computation. Once multiple minimum risk solutions appear in the search grid, we assume that we have exceeded the model order of the original data.

4.8.8 Midpoint Value AVMRF MPE Estimator Performance

In tables 4.2 through 4.4 we show the results of performance tests of the midpoint value, AVMRF MPE estimator for first and second order predictors, on various MRF simulation realizations. The prediction error was penalized by the squared cost function. The tables show the clique weight ratios of the input realization data, the actual clique weights for the input realizations and the clique weight estimates, in a form which corresponds to the shape of the neighborhood geometry. Note that the estimation results are not exact but must be interpreted with due consideration of the quantization effect discussed in section 4.8.3. In addition to the MRF simulation examples, table 4.5 shows the estimates for the clique weights for the test images shown in figure 4.7.

Note that in the case of second order predictor estimates of first order MRF's we discovered in section 4.8.7 that there can exist multiple minimum risk solutions. In the tables where this is the case, we show all of these solutions. The reader should compare the results of the estimates using a second order predictor with those using a first order predictor with consideration of the quantization effect, to realize that the second order predictor estimates for a first order model provide no increase in the precision of the estimate.

From the test data shown, it should be clear to the reader that the midpoint value version of the AVMRF MAP pixel predictor in MPE estimation of the clique weight parameters is successful, despite it being a rather ad-hoc solution to the

Ratio			True Clique Weights			Estimated Clique Weights		
0	0	0	0.0	0.0	0.0		0.0	
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0		0.0	
0	1	0	0.0	0.5	0.0		0.5	
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0		0.5	
0	0	1	0.0	0.0	0.5		0.25	
0		0	0.0		0.0	0.25		0.25
1	0	0	0.5	0.0	0.0		0.25	
1	0	0	0.5	0.0	0.0		0.25	
0		0	0.0		0.0	0.25		0.25
0	0	1	0.0	0.0	0.5		0.25	
0	1	0	0.00	0.25	0.00		0.25	
1		1	0.25		0.25	0.25		0.25
0	1	0	0.00	0.25	0.00		0.25	
0	2	0	0.000	0.333	0.000		0.375	
1		1	0.167		0.167	0.125		0.125
0	2	0	0.000	0.333	0.000		0.375	
0	3	0	0.000	0.375	0.000		0.375	
1		1	0.125		0.125	0.125		0.125
0	3	0	0.000	0.375	0.000		0.375	
0	7	0	0.0000	0.4375	0.0000		0.5	
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000		0.5	
1	0	1	0.167	0.000	0.167		0.125	
1		1	0.168		0.168	0.375		0.375
1	0	1	0.167	0.000	0.167		0.125	
1	1	1	0.167	0.167	0.168		0.375	
0		0	0.000		0.000	0.125		0.125
1	1	1	0.168	0.167	0.167		0.375	
1	1	1	0.125	0.125	0.125		0.25	
1		1	0.125		0.125	0.25		0.25
1	1	1	0.125	0.125	0.125		0.25	
1	2	3	0.05	0.10	0.15		0.25	
4		4	0.20		0.20	0.25		0.25
3	2	1	0.15	0.10	0.05		0.25	

Table 4.2: Performance of a 1st order, midpoint prediction, absolute value MPE estimator for various input MRF realizations.

Ratio			True Clique Weights			Estimated Clique Weights		
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0		0	0.0		0.0	0.0		0.0
1	0	0	0.5	0.0	0.0	0.5	0.0	0.0
1	0	0	0.5	0.0	0.0	0.5	0.0	0.0
0		0	0.0		0.0	0.0		0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0	1	0	0.00	0.25	0.00	0.00	0.25	0.00
1		1	0.25		0.25	0.25		0.25
0	1	0	0.00	0.25	0.00	0.00	0.25	0.00
0	1	0	0.000	0.062	0.0000	0.0	0.0	0.0
7		7	0.437		0.4375	0.5		0.5
0	1	0	0.000	0.062	0.0000	0.0	0.0	0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1	0	1	0.167	0.000	0.167	0.1250	0.0000	0.1875
1		1	0.168		0.168	0.1875		0.1875
1	0	1	0.167	0.000	0.167	0.1875	0.0000	0.1250
1	1	1	0.167	0.167	0.168	0.1250	0.1875	0.1875
0		0	0.000		0.000	0.0000		0.0000
1	1	1	0.168	0.167	0.167	0.1875	0.1875	0.1250
1	1	1	0.125	0.125	0.125	0.125	0.125	0.125
1		1	0.125		0.125	0.125		0.125
1	1	1	0.125	0.125	0.125	0.125	0.125	0.125
1	2	3	0.05	0.10	0.15	0.0625	0.1250	0.1250
4		4	0.20		0.20	0.1875		0.1875
3	2	1	0.15	0.10	0.05	0.1250	0.1250	0.0625
4	3	2	0.20	0.15	0.10	0.1875	0.1250	0.1250
1		1	0.05		0.20	0.0625		0.0625
2	3	4	0.10	0.15	0.20	0.1250	0.1250	0.1875

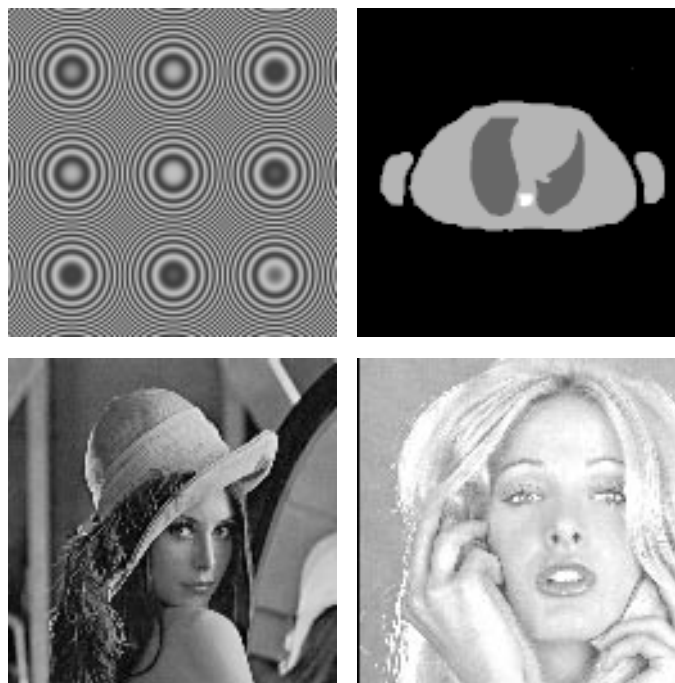
Table 4.3: Performance of a 2nd order, midpoint prediction, absolute value MPE estimator for various input MRF realizations.

Ratio			True Clique Weights			Estimated Clique Weights		
0	1	0	0.000	0.167	0.000	0.0000	0.1875	0.0000
2		2	0.333		0.333	0.3125		0.3125
0	1	0	0.000	0.167	0.000	0.0000	0.1875	0.0000
as above						0.0000	0.1250	0.0000
						0.3750		0.3750
						0.0000	0.1250	0.0000
						0.0000	0.0625	0.0000
						0.4375		0.4375
						0.0000	0.0625	0.0000
0	2	0	0.000	0.333	0.000	0.0000	0.4375	0.0000
1		1	0.167		0.167	0.0625		0.0625
0	2	0	0.000	0.333	0.000	0.0000	0.4375	0.0000
as above						0.0000	0.3750	0.0000
						0.1250		0.1250
						0.0000	0.3750	0.0000
						0.0000	0.3125	0.0000
						0.1875		0.1875
						0.0000	0.3125	0.0000
0	1	0	0.000	0.125	0.000	0.0000	0.1875	0.0000
3		3	0.375		0.375	0.3125		0.3125
0	1	0	0.000	0.125	0.000	0.0000	0.1875	0.0000
as above						0.0000	0.1250	0.0000
						0.3750		0.3750
						0.0000	0.1250	0.0000
						0.0000	0.0625	0.0000
						0.4375		0.4375
						0.0000	0.0625	0.0000
0	3	0	0.000	0.375	0.000	0.0000	0.4375	0.0000
1		1	0.125		0.125	0.0625		0.0625
0	3	0	0.000	0.375	0.000	0.0000	0.4375	0.0000
as above						0.0000	0.3750	0.0000
						0.1250		0.1250
						0.0000	0.3750	0.0000
						0.0000	0.3125	0.0000
						0.1875		0.1875
						0.0000	0.3125	0.0000

Table 4.4: Performance of a 2nd order, midpoint prediction, absolute value MPE estimator for various input MRF realizations.

Test Image	1 st Order Estimate			2 nd Order Estimate		
chirp		0.25		0.1875	0.0625	0.1875
	0.25		0.25	0.0625		0.0625
		0.25		0.1875	0.0625	0.1875
phantom		0.125		0.0625	0.1250	0.0625
	0.375		0.375	0.2500		0.2500
		0.125		0.0625	0.1250	0.0625
lenna		0.375		0.0000	0.3125	0.1250
	0.125		0.125	0.0625		0.0625
		0.375		0.1250	0.3125	0.0000
tiffany		0.375		0.0625	0.3125	0.0625
	0.125		0.125	0.0625		0.0625
		0.375		0.0625	0.3125	0.0625

Table 4.5: Performance of 1st and 2nd order, midpoint prediction, absolute value MPE estimators for various input test images.



a	b
c	d

Figure 4.7: Test images (128 pixels square) from which MRF clique weight parameters were estimated. (a)Aliased radial chirped sinusoid. (b)Hand segmented chest CT phantom. (c)Lenna. (d)Tiffany.

degenerate behavior of the original AVMRF MAP pixel predictor.

4.9 The GGMRF and Generalized Weighted Order Statistic Filters

As mentioned in section 4.8.5, there is a second approach to resolving the problem of degenerate MPE estimate behavior for the AVMRF MAP pixel predictor. For this second solution, we will generalize the MAP pixel predictor model, providing us with a solution to the problems we encountered with the AVMRF MAP pixel predictor, as well as a framework within which to effect clique weight estimation for realizations of the generalized Gaussian MRF.

In section 4.5 we began by investigating the relationship between the AVMRF and a WOS filter. We saw that finding the MAP prediction of a pixel given its neighbors was equivalent to solving a weighted order statistic predictor problem (see equations 4.6 and 4.7). Following a similar line of reasoning, consider the generalized Gaussian MRF with PDF,

$$\mathcal{P}(\mathbf{Z}) = \frac{1}{K_p} \exp \left[-\frac{1}{\beta} \sum_{c \in \mathcal{C}_2} w_i |\mathcal{A}_c(\mathbf{Z})|^p \right]. \quad (4.24)$$

In this equation, p is the GGMRF exponent and i numbers each unique type of clique pair in the set of pair cliques \mathcal{C}_2 . As before, we consider the problem of finding the MAP estimate of a single pixel's value given the values of its neighbors, that is, finding the value of the k^{th} pixel such that the *a-posteriori* distribution $\mathcal{P}(Z_k | \mathbf{Z} \setminus Z_k)$ is maximized. The *a-posteriori* distribution may be reduced to,

$$\mathcal{P}(Z_k | \mathbf{Z} \setminus Z_k) = \frac{1}{K_p'} \exp \left[-\frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_i |Z_k - Z_j|^p \right], \quad (4.25)$$

where K_p' is a new normalization constant. MAP estimation of the k^{th} pixel's value,

denoted as \hat{Z}_k , requires us to solve,

$$\begin{aligned}
\hat{Z}_k &= \arg \max_{\zeta} \left\{ \frac{1}{K_p'} \exp \left[-\frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j|^p \right] \right\} \\
&= \arg \max_{\zeta} \left\{ \ln \left(\frac{1}{K_p'} \right) - \frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j|^p \right\} \\
&= \arg \max_{\zeta} \left\{ -\frac{1}{\beta} \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j|^p \right\} \\
&= \arg \max_{\zeta} \left\{ - \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j|^p \right\} \\
&= \arg \min_{\zeta} \left\{ \sum_{j \in \mathcal{N}_k} w_j |\zeta - Z_j|^p \right\} \tag{4.26}
\end{aligned}$$

In section 4.5, we showed that the last line in equation 4.6 was equivalent to the weighted order statistic filter (or in our context a predictor) of equation 4.7. In a similar vein, we interpret the final line in equation 4.26 as the implementation of what we term a *generalized* weighted order statistic (GWOS) predictor defined as,

$$x_{gvos} = \arg \min_x \sum_{i=1}^K w_i |x - x_i|^p. \tag{4.27}$$

We use the term *generalized*, since we now include the GGMRF parameter p , with $1 \leq p \leq 2$. Equation 4.27 is, therefore, a more general form of the WOS filter described by equation 4.7. It should be immediately evident that the GWOS equation includes the WOS equation as a special case for $p = 1$. If we accept the definition of the generalized weighted order statistic predictor, then solving the MAP pixel prediction problem for the GGMRF is equivalent to solving the GWOS predictor equation for the given neighborhood. It remains now for us to examine the GWOS predictor equation in greater detail.

4.9.1 Graphical Solution for the GGMRF MAP Pixel Predictor

We now examine graphical examples which demonstrate how to compute the MAP estimate of a pixel given its neighbors for a GGMRF using the generalized weighted order statistic predictor. As we saw in the previous section, computing the MAP pixel prediction involves the solution of equation 4.26, which is a minimization of the negative logarithm of the conditional PDF of the pixel given its neighbors. We discovered that this equation is equivalent in form to the GWOS predictor equation 4.27.

The upper graphs in figures 4.8 and 4.9 are example plots of the negative logarithm of the conditional PDF (with scaling factors removed) for two distinct neighborhoods in a realization of a generalized Gaussian MRF with $p = 1.3$. Neighbor values are shown as circles in the conditional PDF plot. Notice that the neighbor values and weights used in these example plots are identical to those used in figures 4.1 and 4.2, so as to enable comparison between the behavior of the WOS and GWOS predictors.

Whereas in the AVMRF case of section 4.5.1 the weight value associated with each neighbor is proportional to the change in slope of the conditional PDF plot at the given neighbor value, the same cannot be said for the generalized Gaussian MRF conditional PDF plots since the GGMRF conditional PDF is composed of a summation of *non-linear* functions.

To assist in the minimization process, we also plot the derivative of the negative logarithm of the conditional PDF (the lower plots in figures 4.8 and 4.9). Unlike their AVMRF counterparts, the derivative functions do not exhibit a simple form. We expect to observe a range of behavior, determined by the value of the GGMRF

parameter p , from the AVMRF for $p = 1$ (resulting in the now familiar staircase function derivative) up to a perfectly smooth parabola through all the neighbor points for $p = 2$. Shown in the figures is an intermediate case for $p = 1.3$. Except for the $p = 1$ extreme, it is not possible to directly read off the values of the neighborhood weights from the derivative functions. Also the asymptotic behavior of the GGMRF conditional PDF derivative plot no longer follows the ± 1 limits observed for the AVMRF MAP predictor. The actual neighbor values though, are still visible, as the points for which the second derivative of the conditional PDF of the GGMRF is infinite.

As was the case with the AVMRF MAP predictor, MAP estimation of a pixel given its neighbors for the GGMRF reduces to solving for the zeroes of the derivative function (effected using an iterative numerical method such as binary bisection). However, *if we restrict $1 < p \leq 2$, that is, if we exclude the absolute value case, the MAP prediction of a pixel given its neighbors is always unique.* As an example of this, compare figure 4.2, for which the AVMRF MAP pixel prediction is an interval of pixel values, with figure 4.9 which is the same neighborhood, but now using the GGMRF MAP pixel prediction with $p = 1.3$. As $p \rightarrow 1$, the behavior of the plots in figure 4.9 tend towards those in 4.2, but not until $p = 1$ is the solution non-unique.

4.9.2 AVMRF MAP Predictor Problem Revisited

The uniqueness of the GGMRF model MAP pixel prediction for $1 < p \leq 2$ is the key to our second solution to the problem we encountered with MPE estimation of the clique weights for an AVMRF model. Recall that we discovered that the AVMRF MPE estimation problems were due to the non-unique solution to the MAP pixel prediction for the AVMRF model. By choosing a value of p close to unity and using

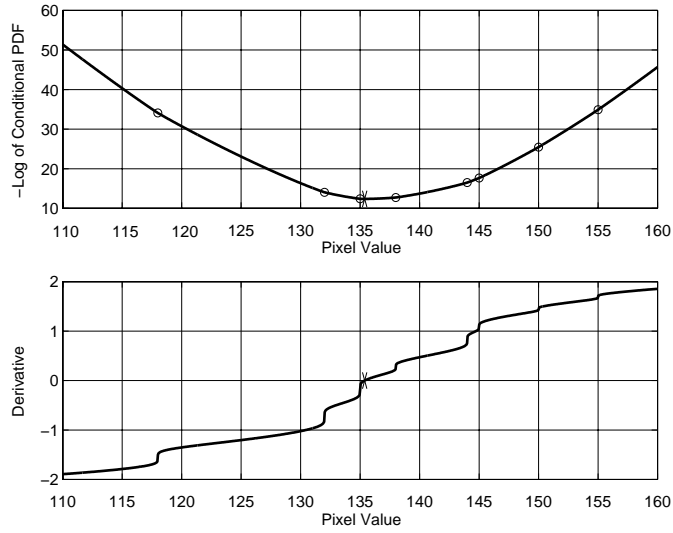


Figure 4.8: Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below) for a GMRF.

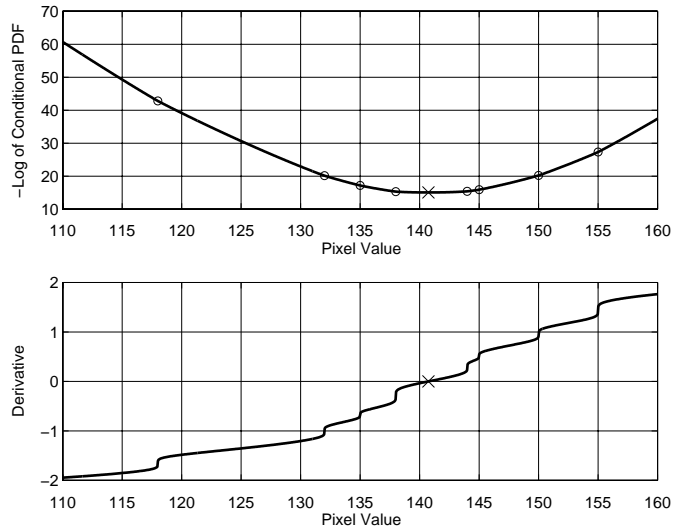


Figure 4.9: Negative logarithm of the conditional PDF of a pixel given its neighbors (above) and its derivative (below) for a GMRF. The same neighborhood system led to a non-unique prediction when the data was assumed to be from an AVMR.

the GGMRF MAP predictor, we closely approximate the behavior of the AVMRF MAP predictor while avoiding the problem of non-unique solutions inherent in the AVMRF MAP predictor. Instead of resorting to the ad-hoc solution proposed in section 4.8.6, we may now use the GGMRF MAP predictor for p close to unity, in place of AVMRF MAP predictor in the MPE grid searching procedure of section 4.8.

Tables 4.6, through 4.8 show the results of the tests using a GGMRF MAP pixel predictor with $p = 1.01$ on realizations of various absolute value MRF's. The results from these tests closely match those found using the midpoint value AVMRF MPE procedure of section 4.8.8. Unlike the AVMRF predictor case, the second order predictor model estimates of first order realization are *unique*. Using the GGMRF predictor for p close to unity tends to remove the problem of multiple solutions with equal risk. Also shown, in table 4.9, are clique weight estimates using the GGMRF predictor for the test images shown in figure 4.7. These results are identical to the results found using the midpoint AVMRF MPE method, shown in table 4.5. The square of the prediction error cost function was used in these tests.

4.10 Minimization of Prediction Error for the GGMRF

Earlier, we investigated MPE estimation for the AVMRF. In this section we will attempt a similar approach for MPE estimation for the GGMRF with the full range of model shape parameters, $1 < p \leq 2$. We have already spoken at length about the case of $p = 1$.

Previously we introduced the technique we use for clique weight estimation by MPE. Recall that the relevant equation was,

$$W_{opt} = \arg \min_W \left\{ \mathcal{E} \left[\mathcal{C} \left(\hat{Z}(W) - Z \right) \right] \right\}, \quad (4.28)$$

Ratio			True Clique Weights			Estimated Clique Weights		
0	0	0	0.0	0.0	0.0		0.0	
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0		0.0	
0	1	0	0.0	0.5	0.0		0.5	
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0		0.5	
0	0	1	0.0	0.0	0.5		0.25	
0		0	0.0		0.0	0.25		0.25
1	0	0	0.5	0.0	0.0		0.25	
1	0	0	0.5	0.0	0.0		0.25	
0		0	0.0		0.0	0.25		0.25
0	0	1	0.0	0.0	0.5		0.25	
0	1	0	0.00	0.25	0.00		0.25	
1		1	0.25		0.25	0.25		0.25
0	1	0	0.00	0.25	0.00		0.25	
0	2	0	0.000	0.333	0.000		0.375	
1		1	0.167		0.167	0.125		0.125
0	2	0	0.000	0.333	0.000		0.375	
0	3	0	0.000	0.375	0.000		0.375	
1		1	0.125		0.125	0.125		0.125
0	3	0	0.000	0.375	0.000		0.375	
0	7	0	0.0000	0.4375	0.0000		0.5	
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000		0.5	
1	0	1	0.167	0.000	0.167		0.125	
1		1	0.168		0.168	0.375		0.375
1	0	1	0.167	0.000	0.167		0.125	
1	1	1	0.167	0.167	0.168		0.25	
0		0	0.000		0.000	0.25		0.25
1	1	1	0.168	0.167	0.167		0.25	
1	1	1	0.125	0.125	0.125		0.25	
1		1	0.125		0.125	0.25		0.25
1	1	1	0.125	0.125	0.125		0.25	
1	2	3	0.05	0.10	0.15		0.25	
4		4	0.20		0.20	0.25		0.25
3	2	1	0.15	0.10	0.05		0.25	

Table 4.6: Performance of a 1st order, GGMRF MPE estimator, with $p = 1.01$ for various input MRF realizations.

Ratio			True Clique Weights			Estimated Clique Weights		
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0		0	0.0		0.0	0.0		0.0
1	0	0	0.5	0.0	0.0	0.5	0.0	0.0
1	0	0	0.5	0.0	0.0	0.5	0.0	0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0	1	0	0.00	0.25	0.00	0.00	0.25	0.00
1		1	0.25		0.25	0.25		0.25
0	1	0	0.00	0.25	0.00	0.00	0.25	0.00
0	1	0	0.0000	0.0625	0.0000	0.0	0.0	0.0
7		7	0.4375		0.4375	0.5		0.5
0	1	0	0.0000	0.0625	0.0000	0.0	0.0	0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1	0	1	0.167	0.000	0.167	0.1250	0.0000	0.1875
1		1	0.168		0.168	0.1875		0.1875
1	0	1	0.167	0.000	0.167	0.1875	0.0000	0.1250
1	1	1	0.167	0.167	0.168	0.1250	0.1875	0.1875
0		0	0.000		0.000	0.0000		0.0000
1	1	1	0.168	0.167	0.167	0.1875	0.1875	0.1250
1	1	1	0.125	0.125	0.125	0.125	0.125	0.125
1		1	0.125		0.125	0.125		0.125
1	1	1	0.125	0.125	0.125	0.125	0.125	0.125
1	2	3	0.05	0.10	0.15	0.0625	0.1250	0.1250
4		4	0.20		0.20	0.1875		0.1875
3	2	1	0.15	0.10	0.05	0.1250	0.1250	0.0625
4	3	2	0.20	0.15	0.10	0.1875	0.1250	0.1250
1		1	0.05		0.05	0.0625		0.0625
2	3	4	0.10	0.15	0.20	0.1250	0.1250	0.1875

Table 4.7: Performance of a 2nd order, GGMRF MPE estimator, with $p = 1.01$ for various input MRF realizations.

Ratio			True Clique Weights			Estimated Clique Weights		
0	1	0	0.000	0.167	0.000	0.0000	0.1250	0.0000
2		2	0.333		0.333	0.3750		0.3750
0	1	0	0.000	0.167	0.000	0.0000	0.1250	0.0000
0	2	0	0.000	0.333	0.000	0.0000	0.3750	0.0000
1		1	0.167		0.167	0.1250		0.1250
0	2	0	0.000	0.333	0.000	0.0000	0.3750	0.0000
0	1	0	0.000	0.125	0.000	0.0000	0.0625	0.0000
3		3	0.375		0.375	0.4375		0.4375
0	1	0	0.000	0.125	0.000	0.0000	0.0625	0.0000
0	3	0	0.000	0.375	0.000	0.0000	0.4375	0.0000
1		1	0.125		0.125	0.0625		0.0625
0	3	0	0.000	0.375	0.000	0.0000	0.4375	0.0000

Table 4.8: Performance of a 2nd order, GGMRF MPE estimator, with $p = 1.01$ for various input MRF realizations.

Test Image	1 st Order Estimate			2 nd Order Estimate		
chirp		0.25		0.1875	0.0625	0.1875
	0.25		0.25	0.0625		0.0625
		0.25		0.1875	0.0625	0.1875
phantom		0.125		0.0625	0.1250	0.0625
	0.375		0.375	0.2500		0.2500
		0.125		0.0625	0.1250	0.0625
lenna		0.375		0.0000	0.3125	0.1250
	0.125		0.125	0.0625		0.0625
		0.375		0.1250	0.3125	0.0000
tiffany		0.375		0.0625	0.3125	0.0625
	0.125		0.125	0.0625		0.0625
		0.375		0.0625	0.3125	0.0625

Table 4.9: Performance of a 2nd order, GGMRF MPE estimator, with $p = 1.01$ for various input test images.

where \mathcal{E} is the expectation operation and \mathcal{C} is a cost function applied to the pixel prediction error. In section 4.9 we found the form of the MAP pixel $\hat{Z}(W)$ predictor for the GGMRF, so we are now in a position to examine ways in which to solve the new optimization problem required for estimation of the clique weights for the GGMRF.

4.10.1 MAP Pixel Prediction Behavior as a Function of Clique Weight

Our earlier approach to MPE estimation of the clique weight vector for AVMRF realizations relied on the fact that we could reduce the size of the search space implicit in equation 4.28 as a result of the particular behavior of the MAP pixel prediction as a function of the clique weights. Let us now consider the possibility of this type of approach for use with the GGMRF MAP pixel predictor.

In section 4.8.1 we used a tutorial example to show that the behavior of the MAP prediction of a pixel given its neighbors was not an everywhere continuous function of the clique weights. For the GGMRF MAP pixel predictor, a reference to figures 4.8 and 4.9 will confirm that for $1 < p \leq 2$ the MAP pixel prediction of a pixel given its neighbors will, under ordinary (non-pathological) circumstances, be unique and will vary as a continuous function of the clique weights. There certainly will be no convenient ranges of clique weight values for which the MAP pixel prediction will be constant, or single values where the prediction is likely to change, as was the case for the AVMRF MAP pixel prediction.

In the following section we examine the consequences of the behavior of the GGMRF MAP pixel prediction as a function of the clique weights, on the the MPE risk functional.

4.10.2 Risk Behavior as a Function of Clique Weight

Previously, we defined the *risk* associated with a particular choice of clique weight vector, in computational form as,

$$\mathcal{R} \triangleq \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathcal{C} \left(\hat{Z}_i(W) - Z_i \right). \quad (4.29)$$

The risk is simply a weighted average of penalized MAP pixel prediction errors for each neighborhood. Since the GGMRF MAP pixel prediction exhibits no convenient, predictable dependence on the choice of the clique weight vector, there is little that can be said concerning the form of the risk functional. This is in stark contrast to the case for the AVMRF, where, in section 4.8.2, the structure of the risk functional was determined and explicitly utilized in the risk minimization procedure.

Since we do not expect to find the sort of step function dependence of risk versus clique weights as was the case for the AVMRF, the question arises as to what we *do* find. An example of a risk function for a first order GGMRF realization is illustrated in figure 4.10.

This and other example risk functions used in this work, are derived from MRF simulation images generated using the methods discussed in chapter 3. In order to ensure that the MRF realizations used are representative of the field PDF, convergence of the σ parameter was required to within a tolerance of 10^{-3} of the equilibrium σ value, as described in section 3.4. In addition, a minimum number of simulation iterations (typically 50) was required. Typical numbers of iterations to acceptable convergence therefore range from a minimum of 50 to several hundred for slowly converging simulation chains.

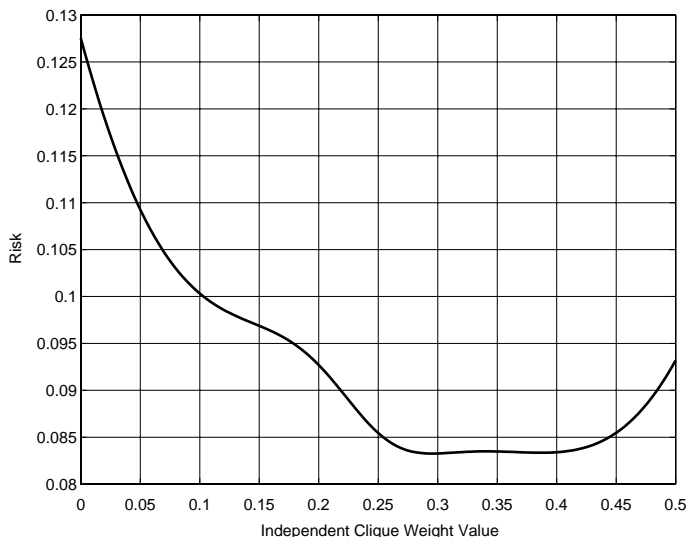


Figure 4.10: Risk as a function of independent clique weight for a 1st order GGMRF predictor model with $p = 1.3$. The input realization is a GGMRF with clique weight vector $\frac{1}{6}[0 \ 2 \ 0 \ 1]^T$ and $p = 1.3$. The square of the prediction error penalty was used.

4.10.3 Risk Behavior as a Function of the GGMRF p Parameter

Earlier in section 4.9.2 we discussed how the uniqueness of the GGMRF MAP pixel prediction provided a second solution to the problem encountered with the non-unique, interval type predictions for the AVMRF. We relied on the fact that as $p \rightarrow 1$, the GGMRF tends towards the AVMRF. Using a graphical example, we now show how one particular risk function, derived from a sample MRF realization, varies as a function of the GGMRF p parameter.

In figure 4.11 we show plots of the risk function for the lenna test image, using a first order GGMRF predictor model and the square of the prediction error cost penalization, with varying values of the p parameter. Notice that as $p \rightarrow 1$, the behavior tends towards the step-like risk functions we found using the AVMRF predictor. We may, however, choose p to be close to unity and still have risk functions

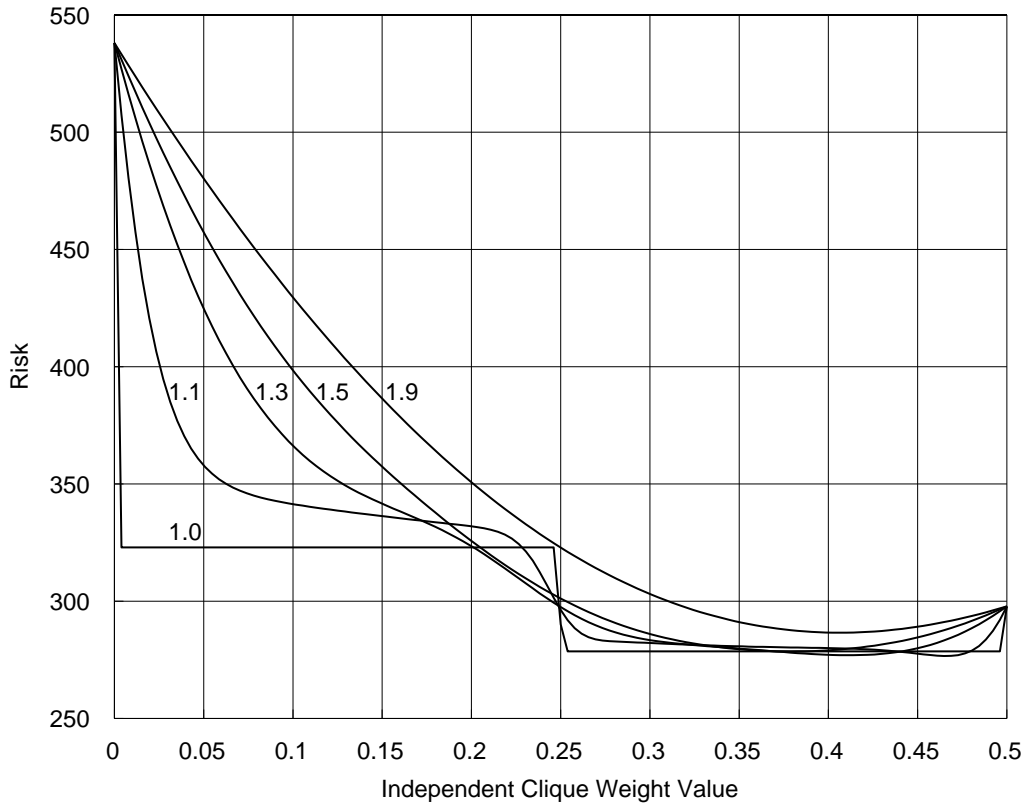


Figure 4.11: Risk as a function of independent clique weight for a 1st order GGMRF predictor for various values of p . The input realization is the “lenna” test image. The square of the prediction error penalty was used.

which do not become flat over any interval of clique weight values. This implies that we do not encounter the quantized behavior of the prediction.

4.10.4 MPE by Gradient Descent

Notice that our example risk functional associated with the GGMRF MAP pixel predictor shown in figure 4.10, is smooth and well behaved. This is to our advantage since we must minimize this risk functional in order to find the MPE estimate of the clique weight vector. A well know approach to solving optimization problems of this sort is *gradient descent* [20]. The principle is simple; given some initial starting point on the risk functional, follow the direction of steepest descent until the gradient

tends to zero. At this point we have reached (at least) a local minimum.

Given the agreeable behavior of our example risk functional for the GGMRF MPE estimation problem, an attempt was made to apply the gradient descent method to the problem of locating the position of the minimum of the risk functional.

Expressions for the Gradient

Since we seek a method to descend the “surface” of the risk functional, we require expressions which provide information concerning the rate of change of the risk in any direction of change of the clique weights. Recall that the risk functional is defined as,

$$\mathcal{R} \triangleq \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} c(\hat{Z}_i(W) - Z_i). \quad (4.30)$$

Since we require information to direct the descent of this risk functional, we must find the rate of change of the risk with respect to a change in each clique weight. This is simply $\frac{\partial \mathcal{R}}{\partial W_k}$, the partial derivative of \mathcal{R} with respect to the k^{th} weight W_k .

In appendix A we show how to compute $\frac{\partial \mathcal{R}}{\partial W_k}$. Since the expression is lengthy, the reader is referred to the the full derivation in the appendix. Once we have an expression for $\frac{\partial \mathcal{R}}{\partial W_k}$ for each W_k we may construct the gradient vector $\nabla_{\mathcal{R}}$ whose k^{th} element $[\nabla_{\mathcal{R}}]_k \triangleq \frac{\partial \mathcal{R}}{\partial W_k}$. We denote the risk gradient evaluated for some position W in the weight space as $\nabla_{\mathcal{R}}(W)$.

Gradient Descent Algorithm

Having found the form of the gradient of the risk as a function of the clique weight parameters, we present the gradient descent algorithm.

1. Initialize

Set $M = 0$

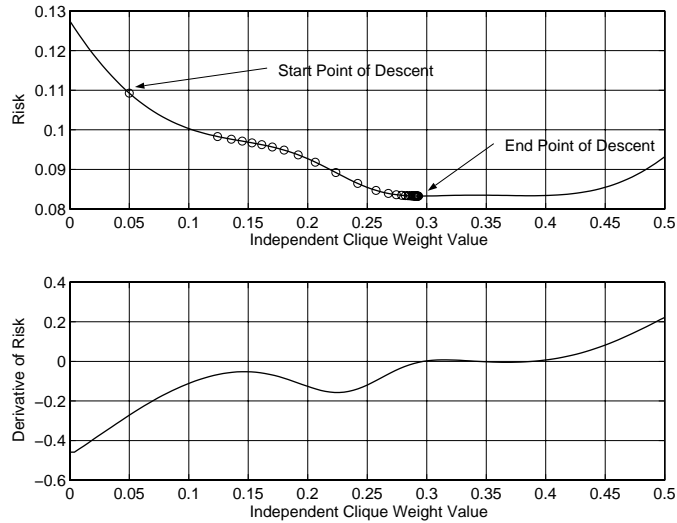


Figure 4.12: Risk function and the trajectory of a gradient descent search for a minimum of the function. Shown in lower frame is the derivative of the risk.

Set W^M , the initial guess for the minimum risk W .

Find the expression for $[\nabla_{\mathcal{R}}]_k = \frac{\partial \mathcal{R}}{\partial W_k}$

2. Update

$$W^{M+1} = W^M - \alpha \cdot \nabla_{\mathcal{R}}(W^M)$$

$$M \leftarrow M + 1$$

3. Convergence Check

if $\|\nabla_{\mathcal{R}}(W^M)\| < \tau$ then stop, else return to 2.

In this algorithm, α is some positive step size and τ is a threshold to determine when the algorithm has converged sufficiently. $\|\cdot\|$ is some norm.

Figure 4.12 shows a graphical example of the operation of the gradient descent method. Given an initial guess the algorithm steps “downhill” until the magnitude of the gradient falls within the stopping criterion threshold.

In our work using gradient descent to minimize risk functionals, both the step size and the convergence threshold were determined experimentally. Note that it is sometimes possible to choose highly efficient step sizes given some knowledge of the character of $\mathcal{R}(W)$ [20]. This avenue was not, however, pursued since there are more pressing problems concerning gradient descent which we shall discuss shortly.

A Problem with Gradient Descent MPEE

Our example risk plot in figure 4.10, not only serves to suggest the gradient descent method, but also suggests the reason why gradient descent optimization of the risk functional will necessarily fail – local minima. Since gradient descent blindly follows the direction of steepest descent of the risk functional, it is all but guaranteed to find a local minimum (rare cases exist where this is not necessarily true). This local minimum found, need not necessarily be the *global minimum*. Figure 4.10 is evidence that this is not merely an academic problem. This function actually has *two* local minima, which implies that the estimate of the “optimal” clique weight vector found by a gradient descent optimization will depend on the initial starting point of the search. This is clearly undesirable.

Figure 4.10 results from the use of a squared penalty function in computing the risk. The problem of local minima becomes much more serious when we consider a cost function such as the zero/one cost, or a close approximation thereof. Recall that these cost functions, especially the zero/one cost, led to desirable estimator properties. See for example figure 4.13, which is the risk plot resulting from the use of a power cost function, with $\gamma = 0.05$. For this risk functional, gradient descent optimization is clearly infeasible.

If, however, we are able to choose an initial starting point for the gradient descent

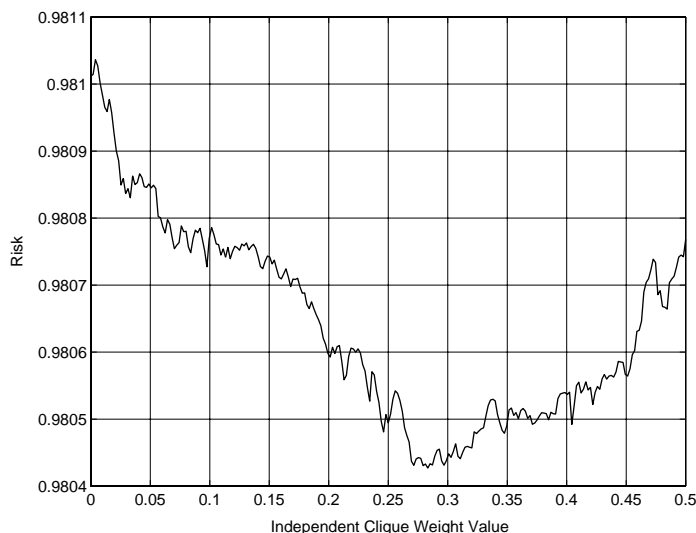


Figure 4.13: Risk function for which gradient descent optimization is ineffective.

method that is “close” to the true solution, we reduce the possibility of finding an incorrect local minimum solution. In section 4.9.2, we discussed an approach for estimating the clique weight parameters for the AVMRF model, using a GGMRF predictor, with p close to unity. We may use these results as an initial starting point for the gradient descent optimization of the risk. Even with the utilization of this educated initial guess in an attempt to ameliorate the problem of finding merely locally minimum risk solutions, there is still no guarantee that gradient descent will terminate at the global minimum of the risk functional.

The gradient descent approach to solving the MPE problem for the GGMRF appeared initially to be quite promising, but as we have seen, the problem of local minima precludes reliable application of this approach. Furthermore, we demonstrated these problems in a first order model - it should be expected that substantially more complex behavior of the risk function will be observed for higher order models, making the problem of local minima more serious yet.

Performance of the Gradient Descent MPEE

Despite the problems associated with the gradient descent method, we venture, in table 4.10, results of gradient descent risk function optimizations as applied to known parameter MRF realizations. The square of the prediction error cost, which tends to result in relatively well behaved risk functionals, was used for this work. The initial guesses used are the results of grid searching MPE optimization passes, as discussed in the previous section. In spite of the problems with gradient descent, the performance nevertheless appears satisfactory given the good initial guesses. These results are comparable to those found in sections 4.8.8 and 4.9.2.

4.10.5 MPE by Simulated Annealing

As we discovered in the previous section, the presence of local minima in the risk function for realistic MRF realizations implies that gradient descent provides no guarantee of finding the globally minimum risk clique weight vector as required for MPE. For this reason, it is necessary to examine a more robust form of optimization, which has the ability to avoid such local minima. *Simulated annealing* is such a method.

Simulated Annealing

In this section we briefly review the method of *simulated annealing* for solving optimization problems. *Annealing* is a thermal process whereby a highly ordered, low energy state is achieved in a solid lattice, after melting and very careful cooling back to the solid state.

Earlier, in section 3.3.1 we introduced the Metropolis algorithm [25] which is a computational procedure for simulating the evolution of a lattice system in a heat

True Clique Weights			Estimated Clique Weights Square Cost Functional			Estimated Clique Weights Power Cost Functional		
0.0	0.0	0.0		0.001414			0	
0.5		0.5	0.498586		0.498586	0.5		0.5
0.0	0.0	0.0		0.001414			0	
0.0	0.5	0.0		0.499882			0.5	
0.0		0.0	0.000118		0.000118	0		0
0.0	0.5	0.0		0.499882			0.5	
0.0	0.0	0.5		0.260479			0.249852	
0.0		0.0	0.239521		0.239521	0.250148		0.250148
0.5	0.0	0.0		0.260479			0.249852	
0.5	0.0	0.0		0.245851			0.135741	
0.0		0.0	0.254149		0.254149	0.364259		0.364259
0.0	0.0	0.5		0.245851			0.135741	
0.00	0.25	0.00		0.252171			0.233276	
0.25		0.25	0.247829		0.247829	0.266724		0.266724
0.00	0.25	0.00		0.252171			0.233276	
0.000	0.333	0.000		0.382457			0.321800	
0.167		0.167	0.117543		0.117543	0.178200		0.178200
0.000	0.333	0.000		0.382457			0.321800	
0.000	0.167	0.000		0.096219			0.164676	
0.333		0.333	0.403781		0.403781	0.335324		0.335324
0.000	0.167	0.000		0.096219			0.164676	

Table 4.10: Gradient descent clique weight vector estimates using a 1st order, GGMRF MPE estimator for various input MRF realizations.

bath to *thermal equilibrium* [1]. A solid may be modeled as a lattice system, and for every state of the lattice system, there is an associated *energy*.

The Metropolis algorithm works as follows:

1. Choose an initial state S_0 .
2. Given S_i , propose S_{i+1} .
3. Accept S_{i+1} with probability $\min \left\{ 1, \exp \left(-\frac{E_{i+1}-E_i}{k_B T} \right) \right\}$.
4. Return to step 2

E_i and E_{i+1} are the energies associated with states S_i and S_{i+1} respectively. k_B is the Boltzmann constant and T is the temperature of the heat bath.

Running the Metropolis for a sufficient length of time ensures that the lattice system reaches *thermal equilibrium* with the heat bath at temperature T . At this stage, the probability of states of the lattice follow a Boltzmann, or Gibbs, distribution [1]. If we denote the state of the lattice in vector form as \mathbf{X} , we may write the PDF of the lattice states in thermal equilibrium as,

$$\mathcal{P}(\mathbf{X} = S_i) = \frac{1}{K_p(T)} \exp \left(\frac{-E_i}{k_B T} \right). \quad (4.31)$$

E_i is the energy of state S_i and $K_p(T)$ is the familiar partition function.

The above formulation of the Metropolis algorithm may appear slightly different from the one in section 3.3.1, but when it is noted that the PDF for the MRF is a Gibbs distribution, the equivalence is obvious.

Now, if the temperature parameter is carefully lowered such that at each temperature the lattice system is allowed to reach thermal equilibrium, then the Boltzmann distribution in equation 4.31 becomes progressively more “peaked”, with the result

that only lower energy states have significant probability of being observed. Eventually, the lattice is “frozen” in its lowest energy state.

In *Simulated annealing*, introduced by Kirkpatrick, Gellat and Vecchi [22], the annealing is not physical, but a computational equivalent of the statistical mechanics that model the physical annealing process we have just discussed. Their far reaching insight was to apply the annealing principle to *optimization* problems. If the parameter space of an optimization problem is identified with the state space of a thermal system (like our solid in a heat bath), and the risk function of the optimization is identified with the energy of the physical system, then we may view the process of annealing, which isolates the *minimum energy state* in a physical system as equivalent to isolating the *minimum risk state* in the optimization problem.

The distinguishing feature of optimization by simulated annealing is that the risk is *not* monotonically decreased. In any state, there is always a finite probability that a proposed state of *higher* risk will be accepted. This behavior lends simulated annealing the name “stochastic descent”, since it behaves like gradient descent in that it has a propensity to follow the direction of decreasing risk, but *stochastic* since there is always a *probability* that the algorithm will choose an *increase* in risk. This implies that simulated annealing optimization will not “get stuck” in local minima. If we begin the anneal at a high “temperature” (in the optimization paradigm there is no analog to temperature, so some authors [1] refer to T as a control parameter), the algorithm will accept wide ranging transitions, exploring the large scale structure of the risk functional, but as we reduce T , the range of accepted transitions tends to be reduced. Eventually as we “freeze” the system, if we have performed the annealing slowly enough, we end with the lowest risk state.

A troublesome aspect of simulated annealing concerns the choice of the rate

at which to reduce the temperature parameter. If T is decreased too quickly, the system will settle into only a locally optimal state, whereas decreasing T too slowly, though it will yield the correct result, will be computationally inefficient. Some attention has been paid to this problem [1, 13] and although *annealing schedules* that guarantee convergence to the global minimum risk have been proposed, their rate of convergence is so slow as to be of only theoretical value. In practice, the choice of a suitable annealing schedule is usually an ad-hoc one.

Performance of the Annealing MPEE

Recall that in figure 4.13 we showed an example of a risk function which exhibited numerous local minima. In figure 4.14 we show how simulated annealing succeeds in finding the global minimum of even this risk functional. Each point in the figure is a weight value accepted in the annealing process.

Proposed weight values were determined by random selection from a uniform distribution over all feasible weights, with due respect to the symmetry and unity sum constraints.

The initial “temperature” of the anneal was initialized to a value where all proposed weight choices were consistently accepted. This approximately corresponds to the “boiling” state of the lattice system, where disorder is great. The T parameter was exponentially decreased, with $T_{n+1} = \alpha \cdot T_n$, where n denotes the annealing iteration. The choice of α was an ad-hoc one, based on the experience gained from testing. A typical value used in these anneals is $\alpha = 0.99$.

Table 4.11 shows the performance of the annealing MPE estimator for various input MRF realizations. Notice that we experience the same difficulties using a first order predictor on second order realization data as we observed earlier. The second

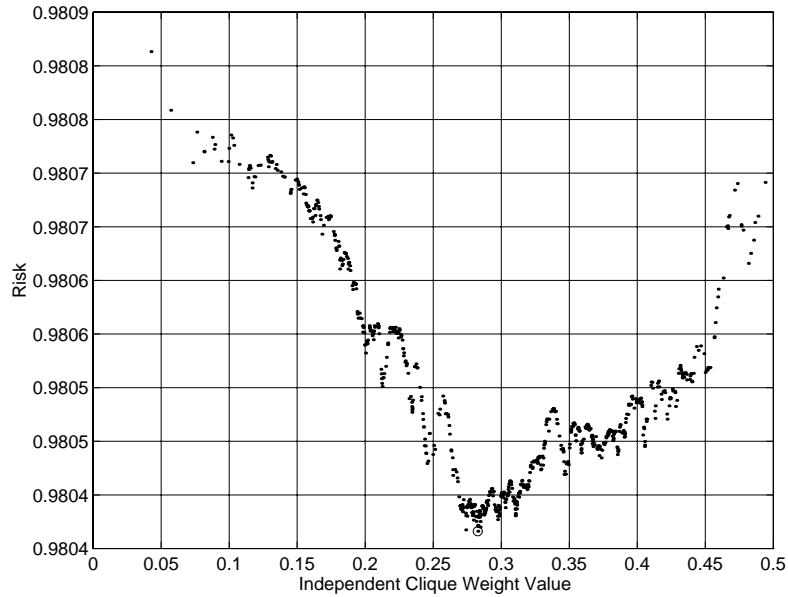


Figure 4.14: Weight values accepted in a simulated annealing optimization of a complex risk functional associated with a 1st order GGMRF. The circle denotes the optimal independent clique weight as found by annealing.

order predictor anneals do, however, produce excellent results despite the complexity of the risk functionals associated with the cost functionals which approximate the zero/one cost.

Problems with Simulated Annealing

Simulated annealing is a very computationally costly procedure and although it can be proved [13] that it converges given a suitable annealing schedule, the problem of selecting an annealing schedule which balances the need to have a solution in a reasonable amount of time, versus the guarantee of the correctness of the solution, is a difficult one. As an example, we show in figure 4.15 a risk function which was to be optimized using simulated annealing. The weight values visited during the anneal are shown in figure 4.16. Comparing the two figures, it is clear that the solution found by the anneal is not the minimum risk solution. The anneal was not performed sufficiently slowly, with the result that the state in which the optimization

True Clique Weights			Estimated Clique Weights 1 st Order Predictor			Estimated Clique Weights 2 nd Order Predictor		
0.0	0.0	0.0		0.000092		0.001698	0.001511	0.006154
0.5		0.5	0.499908		0.499908	0.490637		0.490637
0.0	0.0	0.0		0.000092		0.006154	0.001511	0.001698
0.0	0.5	0.0		0.499601		0.001895	0.494863	0.001988
0.0		0.0	0.000399		0.000399	0.001253		0.001253
0.0	0.5	0.0		0.499601		0.001988	0.494863	0.001895
0.0	0.0	0.5		0.282798		0.003673	0.006817	0.486816
0.0		0.0	0.217202		0.217202	0.002694		0.002694
0.5	0.0	0.0		0.282798		0.486816	0.006817	0.003673
0.5	0.0	0.0		0.205937		0.494635	0.003386	0.000238
0.0		0.0	0.294063		0.294063	0.001741		0.001741
0.0	0.0	0.5		0.205937		0.000238	0.003386	0.494635
0.000	0.333	0.000		0.375580		0.001570	0.338218	0.000196
0.167		0.167	0.124420		0.124420	0.160016		0.160016
0.000	0.333	0.000		0.375580		0.000196	0.338218	0.001570
0.000	0.167	0.000		0.192979		0.025032	0.101202	0.030804
0.333		0.333	0.307021		0.307021	0.342962		0.342962
0.000	0.167	0.000		0.192979		0.030804	0.101202	0.025032

Table 4.11: Estimates of clique weight vectors using simulated annealing optimization. The risk functions optimized resulted from both 1st and 2nd order GGMRF predictors, with $p = 1.3$, using the power cost functional with $\gamma = 0.05$. The input realizations are GGMRF's with $p = 1.3$.

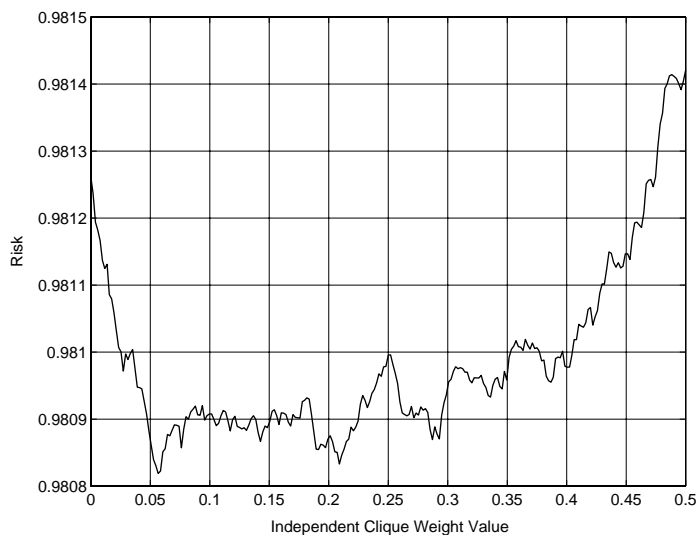


Figure 4.15: Risk function for a 1st order, $p = 1.3$ GGMRF predictor, penalized using the power cost with $\gamma = 0.05$, on a realization of a 2nd order GGMRF, $p = 1.3$, clique weight vector $[1\ 0\ 0\ 0]^T$.

“froze” was not the global minimum.

4.10.6 Other Possible Optimization Approaches

The guarantees of convergence to the true solution using simulated annealing makes few assumptions concerning the nature of function to be optimized. If it were possible to obtain knowledge as to the general behavior limits of the risk functional, it may be possible to formulate hybrid stochastic gradient descent methods which are better suited to yielding solutions in a reasonable amount of time, as a result of the incorporation of *a-priori* knowledge. Work of this nature would be naturally be applicable to optimization problems in general.

4.11 MPEE for Other Classes of MRF

In this study, we have concentrated on the application of MPEE to the GGMRF in particular. However, it should be noted that the technique may be applied to other

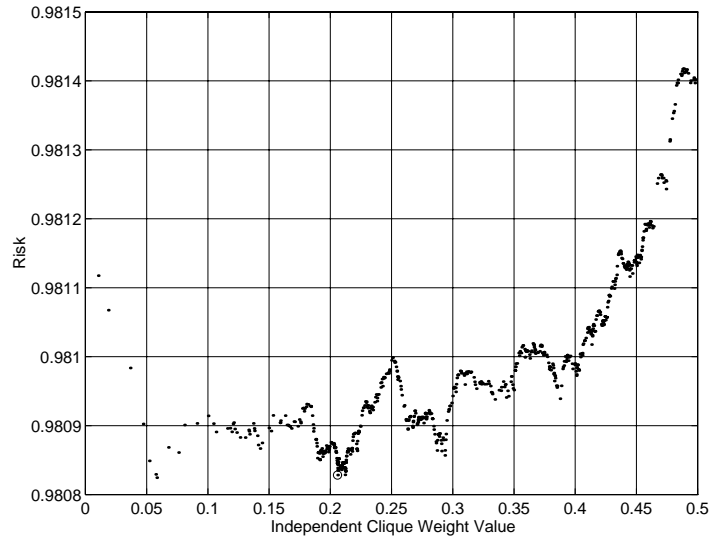


Figure 4.16: Weight values accepted in a simulated annealing optimization. In this case the annealing was performed too quickly with the result that the system “froze” in a local minimum (marked by the circle). The global minimum exists close to the weight value of 0.06.

models as well. Consider, for example, the application of the the MPE method to the problem of clique weight estimation for the Huber MRF. In this case the predictor would take on a slightly different form. It would be the superposition of Huber MRF influence functions rather than the AVMRf or GGMRf influence functions with which we have become so familiar.

4.12 MRF Simulation Convergence Revisited.

Throughout this chapter, our MPE estimation methods have been tested using realizations from MRF simulation chains with “known” parameters. As discussed in chapter 3, the parameters are in fact “known” only if the simulation chain is allowed to run for a duration sufficient to ensure that the equilibrium distribution of the MRF is reached. In section 3.4 we touched briefly on the topic of convergence testing. Though we were unable to prove that convergence of the estimated σ parameter ($\hat{\sigma}$) implies convergence of the chain itself, we suggested that we had

reason to believe that the actual convergence testing method we used (the number of simulation iterations to “convergence” is the number of iterations required to bring the estimated σ value to within 10^{-3} of the equilibrium σ value, or 50, whichever is greater) is sufficient for purpose of testing our MPE estimation methods.

In table 4.12, we show the results of grid searching GGMRF MPE estimates of realizations from a simulation chain of a second order GGMRF. Of particular note is the fact that the MPE estimates settle to the correct value early in the chain and early on in the convergence of $\hat{\sigma}$; far earlier, in fact, than the typical minimum number of realizations we require for “acceptable” convergence of the chain.

This observation, though certainly not proof of the convergence of the chain for converging $\hat{\sigma}$, does, however, lend some credence to the idea that the MRF chains have been allowed to run for sufficient duration for our purposes.

Iteration	$\hat{\sigma}$	Estimated Clique Weights			
Init	0.211272	0.1250	0.1250	0.1250	0.1250
001	0.198846	0.1875	0.1250	0.0625	0.1250
002	0.191456	0.1875	0.1875	0.0625	0.0625
003	0.189348	0.1875	0.1875	0.0625	0.0625
004	0.189380	0.2500	0.1875	0.0000	0.0625
005	0.189464	0.2500	0.1875	0.0000	0.0625
006	0.187994	0.2500	0.1875	0.0000	0.0625
007	0.189104	0.2500	0.1875	0.0000	0.0625
008	0.190819	0.2500	0.1875	0.0000	0.0625
009	0.191112	0.2500	0.1875	0.0000	0.0625
010	0.192238	0.2500	0.1875	0.0000	0.0625
020	0.197699	0.2500	0.1875	0.0000	0.0625
030	0.194499	0.2500	0.1875	0.0000	0.0625
040	0.196506	0.2500	0.1875	0.0000	0.0625
050	0.196873	0.2500	0.1875	0.0000	0.0625
060	0.197603	0.2500	0.1875	0.0000	0.0625
070	0.199310	0.2500	0.1875	0.0000	0.0625
080	0.199366	0.2500	0.1875	0.0000	0.0625
090	0.199259	0.2500	0.1875	0.0000	0.0625
100	0.198167	0.2500	0.1875	0.0000	0.0625
200	0.198856	0.2500	0.1875	0.0000	0.0625
300	0.199282	0.2500	0.1875	0.0000	0.0625
400	0.199002	0.2500	0.1875	0.0000	0.0625
500	0.200697	0.2500	0.1875	0.0000	0.0625

Table 4.12: Grid searching GGMRF MPE estimate of clique weight parameters for realizations from a GGMRF simulation chain with true clique weight parameters $[\frac{1}{2} \frac{3}{8} 0 \frac{1}{8}]^T$, $p = 1.3$ and $\sigma = 0.2$. $\hat{\sigma}$ is the ML estimate of the σ parameter at each simulation iteration.

Chapter 5

CONCLUSIONS

5.1 Primary Research Contribution

We began this thesis with an introduction to the theory of Markov random fields, continued with MRF simulation and then focussed our attention on parameter estimation for a particular class of MRF, the generalized Gaussian MRF. It is in this area that this the thesis makes it's primary research contribution. We proposed and implemented a scheme for estimation of the clique weight parameters of the generalized Gaussian Markov random field, using the technique we call *minimization of prediction error estimation* or MPEE.

Optimal clique weight parameters for the GGMRF are estimated by minimizing the weighted error or *risk* between a prediction, based on observed data, and the known realization data itself. The predictor model derives from the MAP estimate of an image pixel given its neighbors for the given MRF model, and is shown to be equivalent in form to what we call a *generalized weighted order statistic* (GWOS) predictor. Performance tests of the MPEE algorithm were performed, with favorable results.

At the conclusion of this work, it should be noted our MPE method, and the statistical technique known as “cross validation” [29] have much commonality. Cross

validation has recently been applied to the problem of median filter mask selection [30], a problem quite closely allied to our clique weight estimation problem for the GGMRF. It does not, however, appear that cross-validation has as yet been applied to estimation of MRF clique weight parameters as was achieved in this work using MPEE.

5.2 Future Directions of Investigation

This work has certainly spawned more questions than it has answered. There are numerous avenues which we have identified as potentially fruitful directions of investigation, to both resolve outstanding problems and to further the understanding and efficacy of the MPEE method.

MRF Simulation

- **Theoretical Issues**

- Optimally efficient MRF simulation.

Since iterative simulation is a computationally costly procedure, investigating the possibility of “optimal” algorithms for MRF simulation appears justified. Some attention [13, 15] has been paid to the problem of the inefficiency of the Metropolis algorithm due to sample rejection, but there appears to be scope for the investigation of *optimal* algorithms for iterative simulation of MRF’s.

- Convergence testing for MRF simulation.

In section 3.4 we briefly discussed the issue of convergence testing, that is, determining at what stage of the MRF simulation the statistics of the

chain have converged to those of the desired MRF. Our primary reference [31] provides arguments to suggest that convergence of a parameter estimated from the realization chain guarantees convergence of the chain itself to the desired statistical behavior. Unfortunately, the exact details of what type of parameter to utilize, is unclear. The question we thus pose is one of determining what class of parameters estimated from the simulation chain guarantee convergence of the simulation chain to the desired equilibrium distribution.

- **Computational Issues**

- Performance enhancement.

As mentioned in the theoretical problems, iterative simulation is a computationally costly procedure. The problem is especially pronounced when attempting to simulate large MRF realization images, with substantial neighborhood sizes. There is certainly room for improvement over simple Metropolis updates of pixels [13]. One possible approach which could perhaps improve performance would be iterative simulation at various scales. This would naturally lead to the requirement for a theoretical proof of the convergence of the method.

MPE Estimation

- **Theoretical Issues**

- Formal proof of generalization of grid searching to higher dimensions.

In section 4.8.4 we suggest that in grid searching MPEE for the AVMRf, a grid with spacings $\frac{k}{2N}$ is sufficient to find the optimal clique weight set

even in higher dimensions. A formal proof of this result, probably using an inductive argument, should be investigated.

- Choice of optimal cost functions in risk computation.

In section 4.7 we considered a number of cost functions used in the computation of the prediction risk. We showed that the zero/one cost function was optimal in that risk minimization under this cost functional yields the *exact* MRF clique weights. Unfortunately, this result is only asymptotically true, that is, as the amount of sample data tends to infinity. For typically sized realization data sets, choosing a zero/one cost function with $\alpha \rightarrow 0$ results in most predictions being penalized, and risk functionals with high variance which are difficult to optimize.

Although the zero/one cost functional is asymptotically optimal, it may not necessarily (and is probably unlikely to be) the optimal cost functional for finite data sets (this should be shown). Indeed we do not even know how the performance of the zero/one cost function MPEE degrades as a function of the size of the realization data set and the parameter α .

- Determining optimality criteria for cost functions.

The criteria required to determine “optimality” of the cost functional in MPEE are not as obvious as one may think. For example, two important properties of a candidate cost functional may be guarantees of :

1. desirable statistical properties of the MPE estimator.
2. conveniently optimized risk functionals.

There may, of course be additional requirements. Unfortunately, these requirements could be (indeed are likely to be) conflicting. Therefore, de-

termining the “optimality criterion” for choosing a cost functional is not obvious. Clearly we wish our estimate to be “correct”, but if issues such as the feasibility of optimizing complex risk functionals need be considered, we find that we must address the question as to what characteristics make a cost function “optimal”.

- Statistical behavior of MPE estimators.

Perhaps the most important question that should be asked and answered, concerns the statistical properties of MPE estimators. A theoretical analysis of the MPEE method should be undertaken.

- Extending MPEE to other MRF’s and other parameters.

In section 4.11, we mentioned the possibility of extending the MPEE method to other classes of MRF’s. The feasibility of applying MPEE method to estimating other MRF parameters should also be investigated.

- **Computational Issues**

- Alternative algorithms for risk minimization.

In section 4.10.6 we suggest that methods, other than the ones that we have discussed, for finding the minimum risk clique weight vector should be investigated.

- Computational cost of MPEE.

An analysis of the computational cost required to find the MPEE estimate would be a useful topic of investigation.

Applications

The primary focus of this thesis, clique weight parameter estimation for the GGMRF, was motivated by the belief that the use of more realistic image models as *a-priori* knowledge in image reconstruction and restoration problems will lead to improved performance of these methods. At this juncture, we are in a position to test whether this belief was justified.

We will begin testing to determine whether or not performance gains are evident when utilizing MRF's, whose clique weight parameters are determined using the MPPE method from sample solution imagery, as *a-priori* knowledge for Bayesian tomographic reconstruction. Previously, the clique weights of the MRF prior were typically chosen on an ad-hoc basis. Naturally, it is our hope that our approach will prove successful.

Appendix A

DERIVATION OF GRADIENT FOR MPE OPTIMIZATION

In this appendix we derive the results required for the gradient descent solution of the optimization of the risk function to determine the MPE optimal clique weight parameters. We derive here the gradient formulæ for the general case of unspecified cost function. In the following we shall denote vectors in uppercase and elements from those vectors using subscripts. The following variables are used.

1. Z is the realization image vector.
2. Z_i denotes the value of the i^{th} pixel of the realization vector.
3. W is a vector of clique weights.
4. W_k denotes the value of the k^{th} weight.
5. W_{opt} is the vector of optimal weights.
6. $\hat{Z}(W)$ is the predicted image vector, a function of the weight vector W . Note that \hat{Z} is also a function of other parameters, however, since only the dependence on W is relevant to the derivation that follows, we denote the prediction as merely $\hat{Z}(W)$.

7. \hat{Z}_i denotes the value of the i^{th} predicted pixel.
8. \mathcal{S} is the MRF point set as discussed in section 2.3
9. $|\mathcal{S}|$ is the cardinality of the point set \mathcal{S} .
10. \mathcal{C} is the cost function applied to the pixel prediction error.
11. \mathcal{E} is the expectation operation.
12. \mathcal{R} is the risk of associated with the prediction given the weights W .

The essential equation for the estimation of the clique weight parameters using the minimization of prediction error (MPE) principle is,

$$W_{opt} = \arg \min_W \left\{ \mathcal{E} \left[\mathcal{C} \left(\hat{Z}(W) - Z \right) \right] \right\}. \quad (\text{A.1})$$

Assuming ergodicity, we replace the expectation operation by a spatial average over the realization image obtaining the computation form of the above as,

$$W_{opt} = \arg \min_W \left\{ \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathcal{C} \left(\hat{Z}_i(W) - Z_i \right) \right\}. \quad (\text{A.2})$$

In this computational formula, we define the risk similar to expected cost as,

$$\mathcal{R} \triangleq \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \mathcal{C} \left(\hat{Z}_i(W) - Z_i \right). \quad (\text{A.3})$$

In the minimization of prediction error algorithm, we seek to minimize this risk, with respect to the weights W . A necessary condition for the existence of a minimum of this functional is that the partial derivatives of \mathcal{R} with respect to W_k be zero. If we are to attempt a gradient descent approach to finding these points, we are obliged to find expressions for the partial derivatives of the risk with respect to the weights.

$$\frac{\partial \mathcal{R}}{\partial W_k} = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \frac{\partial}{\partial W_k} \left[\mathcal{C} \left(\hat{Z}_i(W) - Z_i \right) \right] \quad (\text{A.4})$$

$$= \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} c'(\hat{Z}_i(W) - Z_i) \cdot \frac{\partial}{\partial W_k} (\hat{Z}_i(W) - Z_i) \quad (\text{A.5})$$

$$= \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} c'(\hat{Z}_i(W) - Z_i) \cdot \frac{\partial}{\partial W_k} (\hat{Z}_i(W)). \quad (\text{A.6})$$

We may of course compute the gradient of \mathcal{R} with respect to W by computing the partial derivative for each element of W . From equation A.6 it is clear that the slope of the risk functional in the direction of the k^{th} clique weight separates into two convenient terms, the first being the derivative of the cost functional, and the second, the partial derivative of the prediction value as a function of the k^{th} weight. This decoupling is expedient since a change in the form of the cost functional is easily accommodated when computing the gradient of the risk. We have still to determine the form of the term $\frac{\partial}{\partial W_k} (\hat{Z}_i(W))$. In order to facilitate this endeavor, we define the following notational conventions.

Recall from section 4.9 that the predicted pixel value is the output of a generalized weighted median filter. Let us denote this output as Y . The prediction Y is computed by generalized weighted median filtering of the $2N$ neighbors of pixel Z_i . For sake of notational convenience, we collect these neighbor values into a single vector X , such that pairs of neighbors symmetrically placed about pixel i (pixels which, due to the homogeneity property of the GGMRF, share a common clique weight value) occupy positions $2j - 1$ and $2j$ in the X vector, for some $1 \leq j \leq N$. We denote as W_j , the value of the clique weight corresponding to the pair X_{2j-1} and X_{2j} . This ordering allows us to write the generalized median prediction problem as,

$$\hat{Z}_i = \arg \min_Y \sum_{j=1}^N W_j (|Y - X_{2j-1}|^p + |Y - X_{2j}|^p) \quad (\text{A.7})$$

The solution is found as the values of Y for which the following holds:

$$\sum_{j=1}^N \left\{ W_j |Y - X_{2j-1}|^{p-1} \cdot \text{sgn}(Y - X_{2j-1}) + W_j |Y - X_{2j}|^{p-1} \cdot \text{sgn}(Y - X_{2j}) \right\} = 0. \quad (\text{A.8})$$

We seek $\frac{\partial Y}{\partial W_k}$ which we may find using implicit partial differentiation, that is by treating $Y = f(W_j)$. This yields,

$$\frac{\partial}{\partial W_k} \sum_{j=1}^N W_j \left\{ |Y - X_{2j-1}|^{p-1} \cdot \text{sgn}(Y - X_{2j-1}) + |Y - X_{2j}|^{p-1} \cdot \text{sgn}(Y - X_{2j}) \right\} = 0$$

We extract the k^{th} term from the summation to give,

$$\begin{aligned} \frac{\partial}{\partial W_k} \sum_{j=1, j \neq k}^N W_j \left\{ |Y - X_{2j-1}|^{p-1} \cdot \text{sgn}(Y - X_{2j-1}) + |Y - X_{2j}|^{p-1} \cdot \text{sgn}(Y - X_{2j}) \right\} + \\ \frac{\partial}{\partial W_k} W_k \left[|Y - X_{2k-1}|^{p-1} \cdot \text{sgn}(Y - X_{2k-1}) + |Y - X_{2k}|^{p-1} \cdot \text{sgn}(Y - X_{2k}) \right] = 0. \end{aligned}$$

Now apply implicit differentiation to find,

$$\begin{aligned} \sum_{j=1, j \neq k}^N W_j (p-1) \left\{ |Y - X_{2j-1}|^{p-2} \frac{\partial Y}{\partial W_k} + |Y - X_{2j}|^{p-2} \frac{\partial Y}{\partial W_k} \right\} + \\ \left[|Y - X_{2k-1}|^{p-1} \cdot \text{sgn}(Y - X_{2k-1}) + W_k (p-1) |Y - X_{2k-1}|^{p-2} \cdot \frac{\partial Y}{\partial W_k} + \right. \\ \left. |Y - X_{2k}|^{p-1} \cdot \text{sgn}(Y - X_{2k}) + W_k (p-1) |Y - X_{2k}|^{p-2} \cdot \frac{\partial Y}{\partial W_k} \right] = 0, \end{aligned}$$

since

$$\begin{aligned} \frac{\partial}{\partial W_k} \left[W_k |Y - X|^{p-1} \cdot \text{sgn}(Y - X) \right] &= |Y - X|^{p-1} \cdot \text{sgn}(Y - X) + \\ &W_k \cdot \frac{\partial}{\partial W_k} \left\{ |Y - X|^{p-1} \cdot \text{sgn}(Y - X) \right\} \\ &= |Y - X|^{p-1} \cdot \text{sgn}(Y - X) + \\ &W_k (p-1) |Y - X|^{p-2} \cdot \frac{\partial Y}{\partial W_k}. \end{aligned}$$

We may now collect terms to reconstitute the summation to find,

$$\begin{aligned} \sum_{j=1}^N W_j (p-1) \frac{\partial Y}{\partial W_k} \left\{ |Y - X_{2j-1}|^{p-2} + |Y - X_{2j}|^{p-2} \right\} + \\ |Y - X_{2k-1}|^{p-1} \cdot \text{sgn}(Y - X_{2k-1}) + |Y - X_{2k}|^{p-1} \cdot \text{sgn}(Y - X_{2k}) = 0. \end{aligned}$$

Finally we may easily solve for $\frac{\partial Y}{\partial W_k}$ to find,

$$\frac{\partial Y}{\partial W_k} = \frac{|Y - X_{2k-1}|^{p-1} \cdot \text{sgn}(Y - X_{2k-1}) + |Y - X_{2k}|^{p-1} \cdot \text{sgn}(Y - X_{2k})}{(1-p) \sum_{j=1}^N W_j \{|Y - X_{2j-1}|^{p-2} + |Y - X_{2j}|^{p-2}\}}. \quad (\text{A.9})$$

We have thus found the form of the term $\frac{\partial}{\partial W_k} \left(\hat{Z}_i(W) \right)$ in equation A.6. All that is required is to simply evaluate the derivative of the cost function, after which we may compute the gradient required for descent of the risk functional.

Appendix B

ABSOLUTE VALUE PREDICTOR MPE PERFORMANCE

Ratio of Clique Weights			True Clique Weights			Estimated Clique Weights		
0	0	0	0.0	0.0	0.0		0.0	
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0		0.0	
0	1	0	0.0	0.5	0.0		0.5	
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0		0.5	
0	0	1	0.0	0.0	0.5		0.5	
0		0	0.0		0.0	0.0		0.0
1	0	0	0.5	0.0	0.0		0.5	
1	0	0	0.5	0.0	0.0		0.5	
0		0	0.0		0.0	0.0		0.0
0	0	1	0.0	0.0	0.5		0.5	
0	1	0	0.00	0.25	0.00		0.5	
1		1	0.25		0.25	0.0		0.0
0	1	0	0.00	0.25	0.00		0.5	
0	2	0	0.000	0.333	0.000		0.5	
1		1	0.167		0.167	0.0		0.0
0	2	0	0.000	0.333	0.000		0.5	
0	3	0	0.000	0.375	0.000		0.5	
1		1	0.125		0.125	0.0		0.0
0	3	0	0.000	0.375	0.000		0.5	
0	7	0	0.0000	0.4375	0.0000		0.5	
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000		0.5	
1	0	1	0.167	0.000	0.167		0.0	
1		1	0.168		0.168	0.5		0.5
1	0	1	0.167	0.000	0.167		0.0	
1	1	1	0.167	0.167	0.168		0.5	
0		0	0.000		0.000	0.0		0.0
1	1	1	0.168	0.167	0.167		0.5	
1	1	1	0.125	0.125	0.125		0.5	
1		1	0.125		0.125	0.0		0.0
1	1	1	0.125	0.125	0.125		0.5	
1	2	3	0.05	0.10	0.15		0.0	
4		4	0.20		0.20	0.5		0.5
3	2	1	0.15	0.10	0.05		0.0	

Table B.1: Performance of a 1st order, absolute value MPE estimator for various input MRF realizations using absolute value cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.

Ratio of Clique Weights			True Clique Weights			Estimated Clique Weights		
0	0	0	0.0	0.0	0.0		0.0	
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0		0.0	
0	1	0	0.0	0.5	0.0		0.5	
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0		0.5	
0	0	1	0.0	0.0	0.5		0.5	
0		0	0.0		0.0	0.0		0.0
1	0	0	0.5	0.0	0.0		0.5	
1	0	0	0.5	0.0	0.0		0.5	
0		0	0.0		0.0	0.0		0.0
0	0	1	0.0	0.0	0.5		0.5	
0	1	0	0.00	0.25	0.00		0.5	
1		1	0.25		0.25	0.0		0.0
0	1	0	0.00	0.25	0.00		0.5	
0	2	0	0.000	0.333	0.000		0.5	
1		1	0.167		0.167	0.0		0.0
0	2	0	0.000	0.333	0.000		0.5	
0	3	0	0.000	0.375	0.000		0.5	
1		1	0.125		0.125	0.0		0.0
0	3	0	0.000	0.375	0.000		0.5	
0	7	0	0.0000	0.4375	0.0000		0.5	
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000		0.5	
1	0	1	0.167	0.000	0.167		0.0	
1		1	0.168		0.168	0.5		0.5
1	0	1	0.167	0.000	0.167		0.0	
1	1	1	0.167	0.167	0.168		0.5	
0		0	0.000		0.000	0.0		0.0
1	1	1	0.168	0.167	0.167		0.5	
1	1	1	0.125	0.125	0.125		0.5	
1		1	0.125		0.125	0.0		0.0
1	1	1	0.125	0.125	0.125		0.5	
1	2	3	0.05	0.10	0.15		0.0	
4		4	0.20		0.20	0.5		0.5
3	2	1	0.15	0.10	0.05		0.0	

Table B.2: Performance of a 1st order, absolute value MPE estimator for various input MRF realizations using squared cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.

Ratio of Clique Weights			True Clique Weights			Estimated Clique Weights		
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0		0	0.0		0.0	0.0		0.0
1	0	0	0.5	0.0	0.0	0.5	0.0	0.0
0		0	0.0		0.0	0.0		0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0	1	0	0.00	0.25	0.00	0.0	0.5	0.0
1		1	0.25		0.25	0.0		0.0
0	1	0	0.00	0.25	0.00	0.0	0.5	0.0
0	2	0	0.000	0.333	0.000	0.0	0.5	0.0
1		1	0.167		0.167	0.0		0.0
0	2	0	0.000	0.333	0.000	0.0	0.5	0.0
0	3	0	0.000	0.375	0.000	0.0	0.5	0.0
1		1	0.125		0.125	0.0		0.0
0	3	0	0.000	0.375	0.000	0.0	0.5	0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1	0	1	0.167	0.000	0.167	0.5	0.0	0.0
1		1	0.168		0.168	0.0		0.0
1	0	1	0.167	0.000	0.167	0.0	0.0	0.5
1	1	1	0.167	0.167	0.168	0.5	0.0	0.0
0		0	0.000		0.000	0.0		0.0
1	1	1	0.168	0.167	0.167	0.0	0.0	0.5
1	1	1	0.125	0.125	0.125	0.5	0.0	0.0
1		1	0.125		0.125	0.0		0.0
1	1	1	0.125	0.125	0.125	0.0	0.0	0.5
1	2	3	0.05	0.10	0.15	0.0	0.0	0.0
4		4	0.20		0.20	0.5		0.5
3	2	1	0.15	0.10	0.05	0.0	0.0	0.0

Table B.3: Performance of a 2nd order, absolute value MPE estimator for various input MRF realizations using absolute value cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.

Ratio of Clique Weights			True Clique Weights			Estimated Clique Weights		
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
1		1	0.5		0.5	0.5		0.5
0	0	0	0.0	0.0	0.0	0.0	0.0	0.0
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0		0	0.0		0.0	0.0		0.0
0	1	0	0.0	0.5	0.0	0.0	0.5	0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0		0	0.0		0.0	0.0		0.0
1	0	0	0.5	0.0	0.0	0.5	0.0	0.0
0		0	0.0		0.0	0.0		0.0
0	0	1	0.0	0.0	0.5	0.0	0.0	0.5
0	1	0	0.00	0.25	0.00	0.0	0.5	0.0
1		1	0.25		0.25	0.0		0.0
0	1	0	0.00	0.25	0.00	0.0	0.5	0.0
0	2	0	0.000	0.333	0.000	0.0	0.5	0.0
1		1	0.167		0.167	0.0		0.0
0	2	0	0.000	0.333	0.000	0.0	0.5	0.0
0	3	0	0.000	0.375	0.000	0.0	0.5	0.0
1		1	0.125		0.125	0.0		0.0
0	3	0	0.000	0.375	0.000	0.0	0.5	0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1		1	0.0625		0.0625	0.0		0.0
0	7	0	0.0000	0.4375	0.0000	0.0	0.5	0.0
1	0	1	0.167	0.000	0.167	0.5	0.0	0.0
1		1	0.168		0.168	0.0		0.0
1	0	1	0.167	0.000	0.167	0.0	0.0	0.5
1	1	1	0.167	0.167	0.168	0.0	0.5	0.0
0		0	0.000		0.000	0.0		0.0
1	1	1	0.168	0.167	0.167	0.0	0.5	0.0
1	1	1	0.125	0.125	0.125	0.0	0.5	0.0
1		1	0.125		0.125	0.0		0.0
1	1	1	0.125	0.125	0.125	0.0	0.5	0.0
1	2	3	0.05	0.10	0.15	0.0	0.0	0.0
4		4	0.20		0.20	0.5		0.5
3	2	1	0.15	0.10	0.05	0.0	0.0	0.0

Table B.4: Performance of a 2nd order, absolute value MPE estimator for various input MRF realizations using squared cost function penalization of the interval solution of the MAP pixel prediction of a pixel given its neighbors.

Bibliography

- [1] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*. Wiley, 1989.
- [2] N. Ahuja and A. Rosenfeld, “Mosaic models for textures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, pp. 1–11, Jan. 1981.
- [3] N. Ahuja and B. J. Schachter, “Image models,” *Computing Surveys*, vol. 13, pp. 373–397, Dec. 1981.
- [4] J. Besag, “On the statistical analysis of dirty pictures,” *J. Roy. Statist. Soc. B*, vol. 48, no. 3, pp. 259–302, 1986.
- [5] P. J. Bickel and A. K. Doksum, *Mathematical Statistics*. San Francisco: Holden-Day, 1977.
- [6] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: The MIT Press, 1987.
- [7] C. A. Bouman and K. Sauer, “A generalized Gaussian image model for edge-preserving MAP estimation,” *IEEE Trans. Image Processing*, vol. 2, no. 3, pp. 296–310, 1993.
- [8] C. A. Bouman and K. Sauer, “Maximum likelihood scale estimation for a class of Markov random fields,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Adelaide, Australia), 1994.
- [9] G. R. Cross and A. Jain, “Markov random field texture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, pp. 25–39, Jan. 1983.
- [10] P. J. Diggle, T. Fiksel, P. Grabarnik, Y. Ogata, D. Stoyan, and M. Tanemura, “On parameter estimation for pairwise interaction point processes,” *International Statistical Review*, vol. 62, no. 1, pp. 99–117, 1994.
- [11] R. C. Dubes and A. Jain, “Random field models in image analysis,” *Journal of Applied Statistics*, vol. 16, no. 2, pp. 131–164, 1989.

- [12] M. Gabbouj and E. Coyle, “Minimum mean absolute error stack filtering with structural constraints and goals,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSp-38, pp. 955–968, June 1990.
- [13] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, no. 6, pp. 721–741, 1984.
- [14] A. J. Gray, J. W. Kay, and D. M. Titterington, “An empirical study of the simulation of various models used for images,” *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 16, no. 5, pp. 507–513, 1994.
- [15] P. J. Green and X. Han, “Metropolis methods, Gaussian proposals and anti-thetic variables,” in *Stochastic Models, Statistical Methods, and Algorithms in Image Processing* (A. F. P. Barone and M. Piccione, eds.), Lecture Notes in Statistics, 74, Springer Verlag, 1992.
- [16] J. Hadamard, *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. New Haven, CT: Yale University Press, 1923.
- [17] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.
- [18] P. J. Huber, “Robust smoothing,” in *Robustness in Statistics* (R. L. Launer and G. N. Wilkinson, eds.), pp. 33–47, Academic Press, 1979.
- [19] F. P. Kelley, *Reversibility and Stochastic Networks*. Chichester: Wiley, 1979.
- [20] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*. Pacific Grove, CA: Brooks/Grove Publishing Company, 1991.
- [21] R. Kindermann and J. L. Snell, *Markov Random Fields and Their Applications*. Providence, Rhode Island: American Mathematical Society, 1980.
- [22] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671–680, 13 May 1983.
- [23] M. R. Luetzgen, W. C. Karl, A. S. Willsky, and R. R. Tenney, “Multiscale representations of Markov random fields,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3377–3396, Dec. 1993.
- [24] J. Marroquin, S. Mitter, and T. Poggio, “Probabilistic solution of ill-posed problems in computational vision,” *J. Am. Statist. Assoc.*, vol. 82, no. 397, pp. 76–89, 1987.
- [25] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *Journal of Physical Chemistry*, vol. 21, pp. 1087–1092, June 1953.

- [26] L. Onural, M. B. Alp, and M. I. Güreli, "Gibbs random field model based weight selection for the 2-D adaptive weighted median filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 831–837, Aug. 1994.
- [27] D. K. Pickard, "Inference for discrete Markov fields: the simplest nontrivial case," *J. Am. Statist. Assoc.*, vol. 82, no. 397, pp. 90–96, 1987.
- [28] W. Qian and D. M. Titterton, "Multidimensional Markov chain models for image textures," *Journal of the Royal Statistical Society B*, vol. 53, no. 3, pp. 661–674, 1991.
- [29] S. J. Reeves, "A cross-validation framework for solving image restoration problems," *J. Visual Commun. Image Representation*, vol. 3, no. 4, pp. 433–445, 1992.
- [30] S. J. Reeves, "On the selection of median structure for image filtering," *IEEE Trans. Circuits and Systems II*, vol. 42, pp. 556–558, Aug. 1995.
- [31] B. D. Ripley and M. D. Kirkland, "Iterative simulation methods," *Journal of Computational and Applied Mathematics*, vol. 31, pp. 165–172, 1990.
- [32] S. A. Saquib, C. A. Bouman, and K. D. Sauer, "ML Parameter Estimation for Markov Random Fields, with Applications to Bayesian Tomography," *Submitted to IEEE Trans. Image Proc.*
- [33] B. J. Schachter and N. Ahuja, "Random pattern generating processes," *Computer Graphics and Image Processing*, vol. 10, pp. 95–114, 1979.
- [34] R. R. Schultz and R. L. Stevenson, "Stochastic modeling and estimation of multispectral image data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Adelaide, South Australia), pp. V–373 to V–376, 1994.
- [35] R. R. Schultz, R. L. Stevenson, and A. Lumsdaine, "Maximum likelihood parameter estimation for non-Gaussian prior signal models," in *Proc. IEEE Int. Conf. Image Processing*, (Austin, TX), November 1994.
- [36] R. L. Stevenson, B. E. Schmitz, and E. J. Delp, "Discontinuity preserving regularization of inverse visual problems," *IEEE Trans. Syst. Man Cybern.*, vol. 24, no. 3, pp. 455–469, 1994.
- [37] D. Stoyan and H. Stoyan, *Fractals, Random Shapes and Point Fields*. West Sussex, England: Wiley, 1994.
- [38] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington, D.C.: V. H. Winston & Sons, 1977.
- [39] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, 1968.

- [40] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Nuevo, "Optimal weighted median filters under structural constraints," in *Proc. of IEEE Int. Symp. on Circuits and Systems*, vol. 1, (Chicago, Illinois, USA), pp. 942–945, 1993.
- [41] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Nuevo, "Optimal weighted median filtering under structural constraints," *IEEE Transactions on Signal Processing*, vol. 43, pp. 591–604, Mar. 1995.
- [42] R. Yang, L. Yin, M. Gabbouj, and Y. Nuevo, "2-D optimal weighted median filters," in *IEEE Workshop on visual signal processing and communications*, (North Carolina, USA), 1992.
- [43] L. Yin, "Stack filter design: A structural approach," *IEEE Transactions on Signal Processing*, vol. 43, pp. 831–840, Apr. 1995.
- [44] L. Yin, J. Astola, and Y. Nuevo, "Adaptive weighted median filtering under the mean absolute error criterion," in *Proc. 1991 IEEE Wkshp. Vis. Sig. Proc. and Comm.*, (Hsingchu, Taiwan), pp. 184–187, June 1991.