ON THE ANALYSIS OF IMAGE DATA USING SIMULTANEOUS INTERACTION MODELS

Michael Grunkin

LYNGBY 1993
Ph.D. THESIS
NO. 67

ISSN 0908-3456

Trykt af IMSOR - DTH
Bogbinder Hans Meyer

This document was formatted with LaTeX.

HIPS and HIPS-2 are trademarks of SharpImage Software, New York, and The Turing Institute, Glasgow, UK.

CART is a trademark of California Statistical Software, Inc., Lafayette, California.

S-PLUS is a registered trademark of Statistical Sciences Inc., Seattle, Washington.

HP-UX and HP Apollo 9000/750 are registered trademarks of Hewlett-Packard Company.

Some of the work in this thesis has previously been published in Carstensen, Grunkin, & Conradsen (1991).

Preface

This thesis has been prepared at the Institute of Mathematical Statistics and Operations Research (IMSOR), Technical University of Denmark, in partial fulfillment of the requirements for the degree of Ph.D. in engineering.

The general framework of the thesis is statistics and digital image analysis. It is implied that the reader has a basic knowledge of these areas.

The treatment of the subjects is by no means exhaustive, but is intended to provide insight on texture description and modelling. This is done by going through selected theory and examples.

Lyngby, August 1993

Michael Grunkin

Acknowledgements

During the period of this work I have recieved great help from many persons and it would be impossible to thank them all. In particular I would like to thank the following people:

Professor Knut Conradsen for his guidance, encouragement and valuable comments throughout this project. I would also like to express my appreciation for his willingness to encourage and support my (sometimes wild) ideas and projects.

Bjarne Ersbøll for being a person I could always come to with problems - professional and personal. I greatly appreciate his assistance.

Professor Roland Wilson, Warwick, for all his valuable comments and suggestions for which I am very grateful.

I also wish to thank my other colleagues at IMSOR. In particular my officemate Jens Michael Carstensen for his moral support, valuable comments, numerous interesting discussions and our occasional fishing trips which I look forward to resume after this. Henrik Madsen, Ken Sejling and Henning T. Søgaard for our countless discussions. Especially I would like to thank Ken Sejling for his comments during the preparation of this thesis.

My good friends Allan Schefte and David Choleva were very helpful in the last critical moments of the preparation of this thesis.

Also, and in particular, I would like to thank Susanne Valentin for keeping me alive during this work. Without her kind thoughtfullnes I would probably, among other things, have starved to death.

My mother and my brother for their love, moral support and for always showing interest in my work.

Finally I wish to thank my father, Gregor Grunkin, to whom this thesis is dedicated. While he lived he always inspired me to pursue this type of work and, had he lived today, he would probably have taken a vivid interest in this work.

Resumé

Kapitel 1 placerer afhandlingen indenfor rammerne af digital billedbehandling og textur analyse.

Kapitel 2 et kort overblik over 2-D signal behandling og system teori og definerer væsentlige begreber benyttet i denne afhandling. Også grundlæggende begreber i relation til ikke-lineær optimering diskuteres.

Kapitel 3 omhandler den to-dimensionale diskrete Fourier transformation og diskuterer en række problemer i forbindelse med Fourier analysen. Der fokuseres i særlig grad på "spectral leakage" samt valg og design af passende 2-D data vinduer.

Kapitel 4 omhandler den parametriske beskrivelse af texturer under anvendelse af kausale Simultane AutoRegressive Moving Average (SARMA) modeller. Der introduceres en simpel notation til beskrivelse af modelorden. Estimation i sådanne modeller betragtes og der foreslås en 'backforecasting' metode for SARMA modeller. Problemet omkring model identifikation behandles og der foreslås en ny 'backwards' eliminations metode. Endelig betragtes forskellige metoder til simulation i denne klasse af modeller. Herefter følger to case-studies. Først betragtes rene SAR modeller (både NSHP og QP modeller). Disse benyttes i forbindelse med modellering af udvalgte Brodatz texturer.

Optimale modeller (i BIC forstand) identificeres og estimeres. Herigennem sammenlignes NSHP og QP modeller. Evalueringen af de estimerede modeller afsluttes med et omfattende simulations studie, hvor de identificerede modeller benyttes til at simulere de modellerede texturer, hvorved en visuel vurdering af modellerne opnås. Den ikke-lineære CLS estimator sammenlignes med ULS estimatoren for SARMA modeller. Det illustreres hvorledes texturer kan modelleres med færre parametre gennem anvendelse af blandede SARMA modeller.

Kapitel 5 beskæftiger sig med ikke-kausalitets problematikken i relation til estimation af ikke-kausale torus SAR modeller. Metoderne benyttet i kapitel 4 kan ikke benyttes her da LS estimatoren ikke er konsistent for denne type af modeller. Dette faktum belyses gennem simulation og estimation. Den approximative ML estimator, introduceret af Kashyap & Chellappa (1983), studeres gennem simulation og estimation, og det vises at denne estimator generelt ikke er anvendelig i praksis. Derfor betragtes den eksakte ML estimator for torus SAR modellerne. Simulations og estimations studier viser at den ikke-lineære estimation af likelihood funktionen giver særdeles gode resultater men kan være overordentligt vanskelig. En ny approximativ estimator introduceres. Simulations studierne udført i dette kapitel synes at indikere at der optræder fase-overgangs fænomener i denne type af modeller. Disse fænomener minder i høj grad om de der optræder for binære Markov felter.

Kapitel 6 giver et overblik over klassiske spektral estimations metoder. Der fokuseres i særlig grad på brugen af kausale og ikke-kausale SARMA modeller i forbindelse med parametrisk spektral estimation. Disse spectral estimatorer benytted til at estimere sinusbølger i hvid støj.

Kapitel 7 gennemår en række statistiske stikprøve funktioner beregnet på basis af givne spektral estimater. Desuden introduceres to nye statistiske stikprøve funktioner. Deres evne til at klassificere et antal udvalgte texturer undersøges. Dette efterfølges af et case-study hvor spektrale stikprøve funktioner benyttes

til at måle de texturelle forandringer for textiler der undergår enzymatisk behandling.

Contents

Chapter 1

Introduction

1.1 Texture analysis

> *Where order in variety we see, and where, though all things differ, all agree.*
>
> Alexander Pope, Essay on Man, 1735.

It is of importance to consider textural properties in diverse areas as medicine, industrial quality inspection, remote sensing etc.

In spite of the ubiquity of texture in the world surrounding us, and our intuitive understanding of this concept, we still do not have a formal definition of texture.

Of course Alexander Pope was not thinking about texture when he wrote the above quoted lines in 1735. Nevertheless these lines comprise an essential part of our notion of texture, as being the information in data which is not expressed

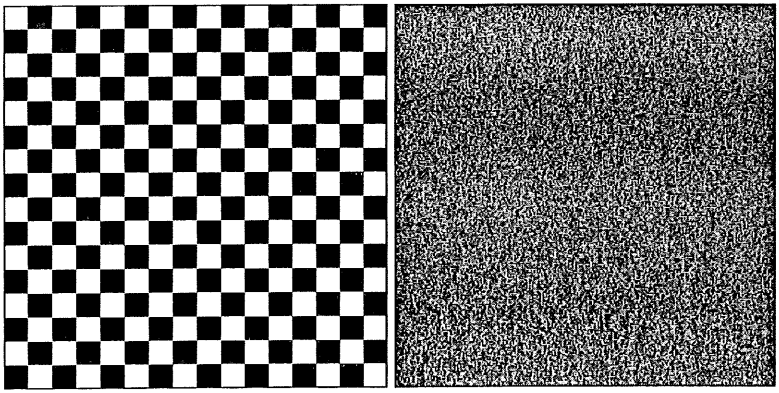

Figure 1.1. *The texture to the left (checkerboard) is completely deterministic and may be described completely from the knowledge of a small sub-pattern. The texture to the right is a strictly chaotic, or random, texture where the pixels are mutually uncorrelated. The noise is Gaussian and this type of noise is often referred to as Gaussian white noise.*

by means of the local intensity (e.g. pixel values), but rather as a regularity in the 'local' variations in intensity.

Textures can range from complete order, through complexity to complete chaos - but are characterized by the same 'type' and/or 'degree' of order, complexity or chaos within the considered region. Figure 1.1 shows a strictly ordered, or deterministic, texture (a checkerboard) which is completely described from the knowledge of a small sub-pattern. In the same figure, a strictly random (or chaotic) texture is shown. Most observable textures lie between these two extremes.

Throughout this thesis we shall often refer to the Brodatz textures, (Brodatz (1966)) which is a photographic collection of 112 textures. This collection has

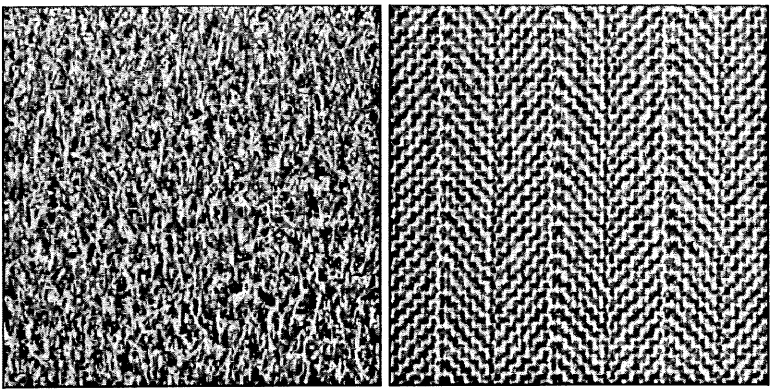

Figure 1.2. *The texture to the left is a picture of a grass lawn taken from the Brodatz album (D009). Such a texture is referred to as a stochastic texture. To the right a semideterministic texture is shown. This is a herringbone weave from the Brodatz album (D017).*

become a standard reference for researchers in the field of texture analysis. Some of these textures will be referred to as random or stochastic textures, and the assumption will be made that these can be described in terms of their first and second order moments. An example of such a texture is given in figure 1.2 (Grass lawn, D009). Other textures, as f.ex. textiles, are very close to being deterministic. It lies in the very nature of such textiles, however, that the weaves cannot be *exactly* deterministic. Such textures are referred to as being semideterministic. An example of such a texture is also given in figure 1.2 (Herringbone weave, D017).

Texture analysis is concerned with the extraction of information from samples of the texture, typically obtained as a digital image consisting of a rectangular grid of picture elements referred to as *pixels*. The two main approaches to texture analysis are the structural and the statistical. The structural approach

assumes that the texture is composed of one or more primitives called texture elements. The spatial distribution of the texture elements is determined by a placement rule. The shortcomings of this approach become conspicuous when considering stochastic textures, where the identification of texture elements and placement rules may seem a little forced. The statistical approach does not rely on the description of textures in terms of primitives and placement rules and is, in this respect, more general. The latter approach is taken in this thesis, and there will be drawn on a general set of statistical and signal processing tools.

The goals of texture analysis are typically either classification of a given texture as belonging to a given class, or segmentation of a textured area into regions characterized by different textural properties. For general reviews on texture analysis the reader is referred to e.g. Haralick (1979), van Gool, Dewaele, & Oosterlinck (1985), Tomita & Tsuji (1990), Rao (1990).

1.2 Signal processing and random fields

We shall mainly consider stochastic textures, which are viewed as realizations of a random field. The field of texture analysis is indeed interdisciplinary, and techniques from the statistical analysis of spatially distributed data and two-dimensional signal processing are often applied. Thus the terms *random field*, *texture* and *signal* are used interchangeably throughout the thesis.

Many approaches may be taken in the description and extraction of information from such random fields. In many applications the spectrum is a powerful tool in the description of textural properties (see e.g. Weszka, Dyer, & Rosenfeld (1976)). For a reliable classifier, however, an accurate estimate of the spectrum is required. The "high-resolution" performance of some of the one-dimensional

(1-D) spectral estimators has promoted an interest in two-dimensional (2-D) versions of these spectral estimators. For a comprehensive exposition of 1-D spectral estimation the reader is referred to Priestly (1981) or Kay (1988).

The successful use of AutoRegressive Moving Average models as a vehicle for parametric spectral estimation in 1-D has motivated attempts to extend these methods to 2-D (see f.ex. Cadzow & Ogino (1981), Tjøstheim (1981), Kashyap (1984), Sharma & Chellappa (1986) and Zou & Liu (1988)).

The extension of these methods is, however, not straightforward due to fundamental differences in 1-D and 2-D discrete systems theory. One of the fundamental problems discussed in this thesis is the problem of causality.

As noted by Kay (1988):

> *"The high computational load of the 2-D spectral estimation methods has resulted in their application to only small data set, which for the most part consists of a few sinusoids in white noise. Suffice it to say that 2-D spectral estimation continues to be an area primarily for research."*

The major advantage of considering stochastic models for a given textural phenomenon is that, apart from being a parsimonious representation that allows for a physical interpretation of the parameters, such models are generative. This means that the visual properties of a fitted model may be evaluated by simulating realizations of the fitted model. This approach introduces important problems such as model identification and estimation which are addressed in this thesis. The generative property also allows us to gain insight into the nature of these models.

1.3 Outline of the thesis

Chapter 2 gives a short overview of two-dimensional signal processing and systems theory, defining important terms and concepts used throughout the thesis. Finally, basic concepts in relation to non-linear optimization are discussed.

Chapter 3 is concerned with the discrete 2-D Fourier transform, some problems related to the Fourier analysis are discussed. In particular the focus is on spectral leakage. The choice and design of proper 2-D windows are discussed in some detail.

Chapter 4 deals with the parametric description of textures using causal Simultaneous AutoRegressive Moving Average (SARMA) models. A simple and intuitive notation, describing the order of a given model, is introduced. The estimation of such models is considered, and a back-forecasting method for 2-D SARMA models is presented. The problem of model identification is considered, and a new backwards elimination method is presented. Finally different simulation methods are considered. This is followed by two case studies. First pure SAR models (Non-Symmetrical HalfPlanes (NSHP) and Quarter Planes (QP)) are used for the description of selected stochastic Brodatz textures. Optimal models (in the BIC sense) are identified and estimated and NSHP models are compared with QP models. The evaluation of the estimated models are completed with an extensive simulation study, where the identified models are used to simulate the modelled textures, thus allowing for an assessment of the visual properties of the fitted model. The properties of the non-linear conditional estimator are evaluated and compared to the unconditional (back-forecasting) estimator previously introduced for SARMA models. The parsimony of the SARMA models compared to the pure SAR models is illustrated.

Chapter 5 is dealing with the problem of non-causality in relation to the estimation of toroidal SAR models. The approach taken in chapter 4 is not applicable here, as the Least Squares estimator is inconsistent in the non-causal case. This fact is illustrated through simulation and estimation. The approximative estimator, introduced by Kashyap & Chellappa (1983), is studied and shown to be inadequate for most practical purposes. Therefore the exact likelihood for toroidal SAR models is considered and shown to yield satisfactory results. Simulation and estimation studies show that the non-linear estimation problem in relation to the Maximum Likelihood estimation may indeed be complicated. A new approximative estimator is presented. The simulation studies carried out in this chapter seems to indicate, that phase-transition phenomena, similar to the ones encountered in binary Markov Random Field, are occurring in non-causal SAR models.

Chapter 6 gives an overview of classical methods for spectral estimation. The main focus is on the use of causal and non-causal SARMA models for parametric spectral estimation. The spectral estimators are used to estimate harmonics superimposed on white noise.

Chapter 7 gives a review of some of the different textural features, that one may extract from a given estimate of the power spectrum. Two new features are presented. The performance of spectral texture features is studied in terms of their ability to discriminate between 12 selected Brodatz textures. The performance is measured in terms the cross- validated misclassification probability, obtained using Classification And Regression Trees (CART), and the complexity of the obtained classification tree. The study is carried out using different types of spectral estimates. This is followed by a case study where Fourier based features are used to measure the textural changes that cotton textiles undergo during cellulase enzymatic treatment.

Chapter 2

Prerequisites

A two-dimensional signal can be modelled as a function of two independent variables. An *analog* signal is a signal where the independent as well as the dependent variables ranges over a continuum of values. When the signal is defined only over a discrete set of independent variables, we have a *discrete* signal. When also the dependent variable is restricted to vary over a discrete set of values, we have a *digital* signal.

In our case we see a photographic image as a continuous signal which we sample in a rectangular grid of uniformly spaced pixels. The intensity of the image is also discretized into a fixed number of gray levels, and thus we obtain a digital image.

Often the sampled signal (or image) is interpreted as a sample of a two-dimensional random process. Such a random process may be conceived as the output of a linear shift invariant system driven by a noise process.

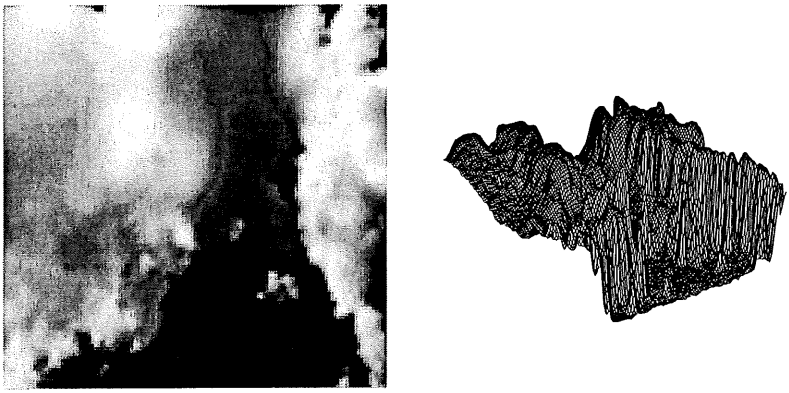

Figure 2.1. *Image of clouds as a gray level intensity plot to the left and as a perspective plot to the right.*

2.1 Signals

A 2D discrete signal (often referred to as a sequence) will be denoted as a function of two independent variables $x(m,n)$. For non-integer values of (m,n) the function $x(m,n)$ is undefined. The value of $x(m,n)$ represents the amplitude of the signal at (m,n) as illustrated in figure 2.1.

The impulse sequence, $\delta(m,n)$, is defined as:

$$\delta(m,n) = \begin{cases} 1 \ , & m = n = 0 \\ 0 \ , & \text{Otherwise} \end{cases} \tag{2.1}$$

and any sequence, $x(m,n)$, can be represented as a linear combination of shifted impulses.

$$x(m,n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(k,l)\delta(m-k, n-l) \tag{2.2}$$

Naturally the sequences we deal with are sampled over a finite region. Rather than restrict the domain of definition, we simply assume that the values of the samples outside the finite region are all equal to zero, and refer to these as finite extent sequences.

A *separable* sequence is any sequence that can be expressed in the form:

$$x(m,n) = x_1(m) \cdot x_2(n) \tag{2.3}$$

Despite the fact that only a very few signals encountered in practice are separable, they play an important role in 2-D signal processing.

Another important class of sequences are *periodic* sequences. A sequence is said to be periodic with period $M \times N$ if it satisfies

$$\begin{aligned} x(m,n+N) &= x(m,n) \\ x(m+M,n) &= x(m,n) \end{aligned} \tag{2.4}$$

When a sequence (or an image) repeats itself in every direction, as illustrated is figure 2.2, it is called doubly periodic or toroidal.

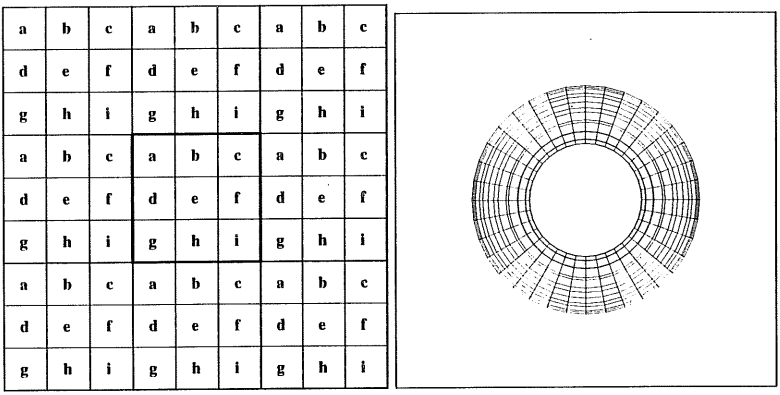

Figure 2.2. *The central image to the left, with pixels labelled* $a, b, \cdots, i$*, is repeated in every direction. Such an image is called doubly periodic or toroidal. This assumption is often made about images for dealing with boundary problems. The periodic extension of the image in all directions is equivalent to wrapping the central image into a torus shape which is shown to the right.*

2.2 Sampling of signals

When sampling from an image an appropriate sampling rate must be chosen carefully. If the sampling rate is too low, *aliasing* may occur in the DFT. To avoid aliasing the image should be sampled with a sampling rate (u_s, v_s) at least two times higher than the highest frequencies, (u_{max}, v_{max}), in the image i.e. the Nyquist rate:

$$u_s = \frac{1}{\Delta m} \geq 2\, u_{max} \tag{2.5}$$

$$v_s = \frac{1}{\Delta n} \geq 2\, v_{max} \tag{2.6}$$

where Δm and Δn are the sampling separations in the m and n directions, respectively.

This is the 2-D sampling theorem, and is a straightforward extension of the 1-D result.

2.3 Linear dynamic systems

A system is represented by an input-output relationship as depicted in figure 2.3. The system T relates an input $x(m,n)$ to an output $y(m,n)$

$$y(m,n) = T\left[x(m,n)\right] \tag{2.7}$$

A system may be characterized by a number of properties. Some common assumptions made about systems are discussed below.

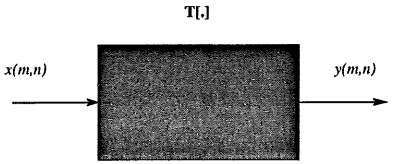

Figure 2.3. *A system* $\boldsymbol{T}$. $x(m,n)$ *represents the input sequence and* $y(m,n)$ *the output.*

2.3.1 Linear Shift-Invariant systems

In order to simplify the characterization and analysis of a system, two assumptions are typically made:

- Linearity:

  $T\left[ax_1(m,n) + bx_2(m,n)\right] = aT\left[x_1(m,n)\right] + bT\left[x_2(m,n)\right] = ay_1(m,n) + by_2(m,n)$

- Shift Invariance:

  $T\left[x(m-k,n-l)\right] = y(m-k,n-l)$

If the sequence in equation 2.2 is input to a linear system we get the output

$$y(m,n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(k,l)T\left[\delta(m-k,n-l)\right] \tag{2.8}$$

The system response to an impulse is referred to as the *impulse response* and is denoted:

$$h(m,n) = T\left[\delta(m,n)\right] \tag{2.9}$$

As the Shift-Invariance property is assumed for the system, we also have:

$$h(m-k,n-l) = T\left[\delta(m-k,n-l)\right] \tag{2.10}$$

and equation 2.8 is rewritten as the *convolution sum*

$$y(m,n) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} x(k,l)h(m-k,n-l) = x(m,n) * h(m,n) \tag{2.11}$$

Such a system is referred to as a Linear Shift-Invariant (LSI) system. An LSI system is completely characterized by its impulse response, and the output is obtained as a *convolution* of the input sequence with the impulse response. Convolution possesses the convenient properties of commutativity, associativity and distributivity.

A system is stable in the bounded-input-bounded-output (BIBO) sense if

$$\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} |h(k,l)| < \infty \tag{2.12}$$

Testing for stability of 2-D systems is not simple in general. The reader is referred to O'Conner & Huang (1981), Dudgeon & Mersereau (1984) or Lim (1990). Tjøstheim (1981) suggests simulation as a simple way to test for stability of a system. White noise is sent through the system. If the output grows without bounds, the system is not stable. Simulation is treated in more detail in chapter 4.

2.3.2 Causality

In the study of 1-D systems a system is said to be causal if the output cannot precede the input or, in more technical terms, the impulse response $h(n)$ is zero for $n < 0$. This is a natural constraint for systems, where the independent variable is time. Causality is, however, not a natural constraint in two dimensions.

The notion of causality can be extended to two dimensions, by defining a system where the impulse response h(m,n) has non-zero values only in a particular region, which is referred to as the region of support. The terms *unilateral* and *recursively computable* are often used about this type of system. This is, strictly speaking, more correct than causal, but is has become a convention to use the latter.

Two dimensional causal systems will typically be either a *quadrant support system* (QP) or a (non symmetrical) *half plane support system* (NSHP). We can refer to four QP- and eight NSHP regions of support. These are depicted in figure 2.4 and figure 2.5 respectively and defined by a notation (adopted from O'Conner & Huang (1981)) given below:

$$\begin{aligned} QP(+,+) &= \{(i,j) \in \mathcal{N} : i \geq 0 \wedge j \geq 0\} \\ QP(-,+) &= \{(i,j) \in \mathcal{N} : i \leq 0 \wedge j \geq 0\} \\ QP(+,-) &= \{(i,j) \in \mathcal{N} : i \geq 0 \wedge j \leq 0\} \\ QP(-,-) &= \{(i,j) \in \mathcal{N} : i \leq 0 \wedge j \leq 0\} \end{aligned} \tag{2.13}$$

To help the definition of NSHP regions we also define:

$$
\begin{aligned}
QP^*(+,+) &= \{(i,j) \in \mathcal{N} : i > 0 \wedge j > 0\} \\
QP^*(-,+) &= \{(i,j) \in \mathcal{N} : i < 0 \wedge j > 0\} \\
QP^*(+,-) &= \{(i,j) \in \mathcal{N} : i > 0 \wedge j < 0\} \\
QP^*(-,-) &= \{(i,j) \in \mathcal{N} : i < 0 \wedge j < 0\}
\end{aligned}
\tag{2.14}
$$

to be used in the definition of the NSHP regions:

$$
\begin{aligned}
NSHP(\oplus,+) &= QP(+,+) \cup QP^*(-,+) \\
NSHP(+,\oplus) &= QP(+,+) \cup QP^*(+,-) \\
NSHP(\ominus,+) &= QP(-,+) \cup QP^*(+,+) \\
NSHP(-,\oplus) &= QP(-,+) \cup QP^*(-,-) \\
NSHP(\oplus,-) &= QP(+,-) \cup QP^*(-,-) \\
NSHP(+,\ominus) &= QP(+,-) \cup QP^*(+,+) \\
NSHP(\ominus,-) &= QP(-,-) \cup QP^*(+,-) \\
NSHP(-,\ominus) &= QP(-,-) \cup QP^*(-,+)
\end{aligned}
\tag{2.15}
$$

Another important class of systems are the non-causal system. In this case the impulse response has support on the whole plane. This, in turn, means that the system *is not* recursively computable. Such systems are considered in chapter 5. The analysis of non-causal systems is in general far more complicated than the analysis of causal systems.

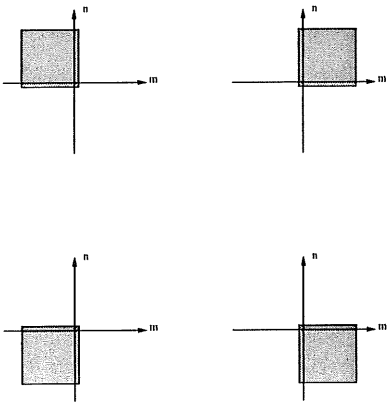

Figure 2.4. *Regions of support for Quarter-Plane causal systems.*

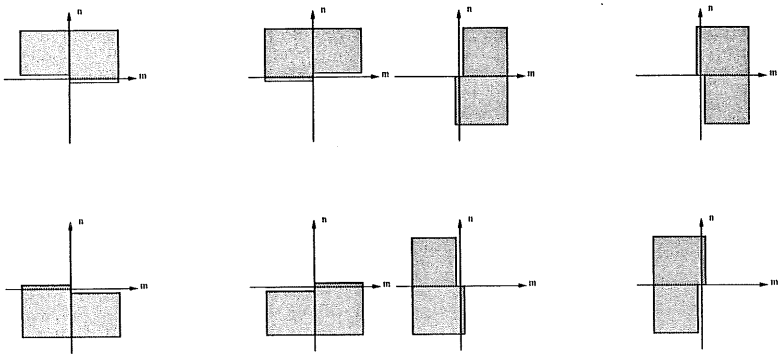

Figure 2.5. *Regions of support for Non-Symmetrical Half-Plane causal systems.*

2.3.3 IIR systems

An important class of systems is the so called Infinite Impulse Response systems. Often this class of systems is restricted to the recursive systems (see chapter 4). However, we shall also consider non-recursive, or non-causal, IIR systems (see chapter 5). The input-output relations for an IIR system is typically described through a difference equation

$$x(s) + \sum_{r \in N_{AR}} \phi_r x(s+r) = \sum_{r \in N_{MA}} \theta_r \epsilon(s+r) + \epsilon(s) \tag{2.16}$$

where the coefficient arrays θ and ϕ have a finite support. Modelling a system this way is often called Simultaneous AutoRegressive Moving Average (SARMA) modelling. The order of the difference equation depends on the region of support for the array of θ and ϕ. Since a given coefficient array can have any shape, the order of the difference equation is typically related to shapes like ellipses or rectangles. This is discussed in greater detail in chapter 4.

The *system function* is obtained through the 2-D z-transform of equation 2.16

$$H(z_r, z_c) = \frac{X(z_r, z_c)}{E(z_r, z_c)} = \frac{1 + \sum_{(k,l) \in N_{MA}} \theta_{k,l} z_r^{-k} z_c^{-l}}{1 + \sum_{(k,l) \in N_{AR}} \phi_{k,l} z_r^{-k} z_c^{-l}} \tag{2.17}$$

The *frequency response function* is obtained for $z_r = exp(-2\pi i \cdot f_r)$ and $z_c = exp(-2\pi i \cdot f_c)$, where $(f_r, f_c) \in \left[-\frac{1}{2}\ \frac{1}{2}\right]$, which is equivalent to the Fourier transform of the *impulse response function*.

2.4 Optimization of non-linear functions

In many situations, the coefficient arrays of a SARMA model must be estimated from a finite set of observed data. This typically involves the minimization (or maximization) of a given scalar function of the parameters, $S(\underline{\Theta} \mid x(s), s \in \Omega)$. This scalar function is referred to as the objective functional. A number of approaches to this problem have been developed. One of the most recent methods for non-linear optimization is the so called Dennis-Moré method, which is a modified BFGS method (see Dennis & Schnabel (1983)). As optimization is central to most of the estimation problems considered in this thesis, the principles of this method are briefly outlined here.

2.4.1 Approximations of the gradient

In all non-linear optimization algorithms, it is necessary to have some knowledge about the gradient of the objective functional. In case of the SARMA models, a closed form expression for the gradients cannot be obtained and consequently these must be calculated numerically. Depending upon the gradient approximation used, the objective functional must be evaluated *at least* $p+1$ times, where p is the number of parameters in the model.

Let $\underline{e}_i$ denote the i'th unit vector in parameter space, and $\delta > 0$ is a (sufficiently) small number. Three commonly used gradient approximations (Kaj Madsen & Hansen (1991)) are given below:

Definition 1 *The Forward Difference Approximation (FDA) is defined as*

$$\frac{\partial S(\underline{\Theta})}{\partial \Theta_i} \approx \frac{S(\underline{\Theta} + \delta \underline{e}_i) - S(\underline{\Theta})}{\delta} \tag{2.18}$$

Definition 2 *The Backwards Difference Approximation (BDA) is defined as*

$$\frac{\partial S(\underline{\Theta})}{\partial \Theta_i} \approx \frac{S(\underline{\Theta}) - S(\underline{\Theta} - \delta \underline{e}_i/2)}{\delta/2} \tag{2.19}$$

Definition 3 *The Central Difference Approximation (CDA) is defined as*

$$\frac{\partial S(\underline{\Theta})}{\partial \Theta_i} \approx \frac{S(\underline{\Theta}) - S(\underline{\Theta} + \delta \underline{e}_i)/3 - 4S(\underline{\Theta} - \delta \underline{e}_i/2)/3}{\delta} \tag{2.20}$$

The quality of these approximations is, not surprisingly, dependent upon the choice of δ, and naturally the CDA is far more stable than the two other approximations, but also more expensive in terms of computation. The stability of these approximations is an important issue, as inaccuracies might ruin the convergence of any gradient based optimization technique. In fact, the author has experienced some cases where convergence could not be obtained using the FDA or BDA (for any values of δ). In these cases the use of CDA gave valid results.

2.4.2 Variable metric methods

The optimization techniques considered in this section are often referred to as *variable metric* methods, because the updating formulas, for e.g. the Hessian matrix, can be derived using a scaling of the variable space that is different at every iteration (Dennis & Schnabel (1983)).

The problem is stated as a minimization of a nonlinear function $S : \mathbb{R}^p \rightarrow \mathbb{R}$ with respect to a vector of parameters $\underline{\Theta}$:

$$\min_{\underline{\Theta}} S(\underline{\Theta}) \tag{2.21}$$

The method is based on the assumption that near the minimum, the objective functional, $S(\underline{\Theta})$, can be approximated by a quadratic form:

$$S(\underline{\Theta}) \approx c - \underline{g}^T\underline{\Theta} + \frac{1}{2}\underline{\Theta}^T H \underline{\Theta} \tag{2.22}$$

where H is the *Hessian matrix* and $\underline{g}$ is the vector of gradients.

At the optimum $\underline{\Theta}^\star$, we have $H\underline{\Theta}^\star = \underline{g}$. Moving away from the optimum, however, the relation $H\underline{\Theta}_i = \nabla S(\underline{\Theta}_i) + \underline{g}$ must be considered. The index, i, is denoting the iteration number. If $\underline{\Theta}^\star$ is replaced with $\underline{\Theta}_{i+1}$ and the above expressions are subtracted, the following relation is obtained:

$$\underline{\Theta}_{i+1} = \underline{\Theta}_i - H(\underline{\Theta}_i)^{-1}\nabla S(\underline{\Theta}_i) \tag{2.23}$$

where $H(\underline{\Theta}_i)$ is an Hessian matrix, or an approximation to it, in the i'th iteration step. The above iteration scheme provides the general framework for several optimization strategies, that differ in the way the Hessian matrix (approximation) is updated. A comprehensive review of different methods is given in Dennis & Schnabel (1983). The method recommended in this book, as the best procedure for general non linear problems is the so called BFGS (Broyden, Fletcher, Goldfarb and Shanno) method, where an approximation to the *inverse* Hessian matrix is obtained through the recursive updating given below:

$$B_{i+1} = \frac{\underline{y}_i\underline{y}_i^T}{\underline{y}_i^T\underline{s}_i} - \frac{B_i\underline{s}_i\underline{s}_i^T B_i}{\underline{s}_i^T B_i \underline{s}_i} \tag{2.24}$$

where $\underline{s}_i$ is the difference between the present and the previous estimate of the parameters. Similarly $\underline{y}_i$ is the difference between the gradient vectors, i.e.

$$\underline{s}_i = \underline{\Theta}_i - \underline{\Theta}_i \tag{2.25}$$

$$\underline{y}_i = \nabla S(\underline{\Theta}_i) - \nabla S(\underline{\Theta}_{i-1}) \tag{2.26}$$

This method belongs to a group of methods called Quasi-Newton methods, due to the fact that the procedure is basically a modified Newton-Raphson procedure.

Now, the updating formula is expressed in terms of the equations 2.23 and 2.24:

$$\underline{\Theta}_{i+1} = \underline{\Theta}_i - B_i \nabla S(\underline{\Theta}_i) \tag{2.27}$$

When a proper search direction is determined in this way, some type of line search is applied to minimize the objective functional in this direction. Instead of performing a very expensive exact line search, in order to localize the exact minimum of the objective functional in the given direction, we shall be satisfied if we can just decrease the value of the objective functional. This is called a *soft line search.* The idea is to use the modified updating formula

$$\underline{\Theta}_{i+1} = \underline{\Theta}_i - \lambda_i \, B_i \nabla S(\underline{\Theta}_i) \tag{2.28}$$

where λ_i is used to insure that the next iterate decreases the value of $S(\underline{\Theta})$. This is done by successively decreasing λ_i until $S(\underline{\Theta}_{i+1}) < S(\underline{\Theta}_i)$.

Using the identity matrix I as initial value for the inverse Hessian matrix, B_0, it can be shown that the successive iterates, B_i, remain positive definite (Dennis & Schnabel (1983)).

The method yields very reliable approximations to the inverse Hessian matrix, which in turn gives us a good estimate of the variance-covariance matrix of the parameter estimates, thus allowing for statistical inference regarding the model parameters.

Chapter 3

Fourier Methods

In almost every area of science and technology, signals must be processed to facilitate the extraction of information. Thus, the development of signal processing techniques is of great importance. These techniques usually take form of a transformation of a signal into another signal that is in some sense more desirable than the original.

The Fourier transform is a powerful tool for image analysis and has many important applications. Examples are given in e.g. Grunkin (1993) where the Fourier transform is used to restore images corrupted by periodic noise. In chapter 7 textural features, extracted from the Fourier power spectrum of different Brodatz textures, are used for classification.

3.1 The Discrete Fourier Transform

The discrete 2D Fourier Transform and the inverse transform of an image $\{x(m,n)\}$, $m = 0, ..., M-1$ and $n = 0, ..., N-1$, are defined as:

$$X(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) \exp\left(-i2\pi(\frac{um}{M} + \frac{vn}{N})\right) \tag{3.1}$$

$$x(m,n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} X(u,v) \exp\left(i2\pi(\frac{um}{M} + \frac{vn}{N})\right) \tag{3.2}$$

where $u = 0, ..., M-1$ and $v = 0, ..., N-1$. The image is assumed to be one period of an endless number extending in every direction - i.e. doubly periodic as shown in figure 2.2.

In the literature some prefer to distribute the scaling constant over the transform and its inverse, while still others employ a reversal in the sign of the kernels.

We are throughout this thesis concerned with real signals $\{x(m,n)\}$. The Fourier transform of a real signal is generally complex:

$$X(u,v) = R(u,v) + i \cdot I(u,v) \tag{3.3}$$

where $R(u,v)$ and $I(u,v)$ are real- and imaginary parts of the transform. It is often convenient to express the transform in 3.1 in the exponential form

$$X(u,v) = |X(u,v)| \exp\left(i \cdot \phi(u,v)\right) \tag{3.4}$$

where

$$|X(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)} \tag{3.5}$$

is the magnitude function, also called the spectrum of x(m,n) (not to be confused with the power spectrum). The phase angle is given as

$$\phi(u,v) = tan^{-1}(\frac{I(u,v)}{R(u,v)}) \tag{3.6}$$

As the 2D Discrete Fourier Transform is separable it can be implemented by successively performing $M \times N$ 1D Fast Fourier Transforms (FFT), provided that both M and N can be written as integer powers of 2. This restriction is mainly due to the fact that most standard library FFT routines are for the radix 2 transform.

3.2 Fourier properties

The discrete Fourier transform and its inverse are doubly periodic with period $M \times N$ as shown in figure 2.2:

$$X(u,v) = X(u+M,v) = X(u,v+N) = X(u+M,v+N) \tag{3.7}$$

This is easily proved by direct substitution of the variables in equation 3.1.

The Fourier transform of a real signal exhibits *Hermitean symmetry*:

$$X(u,v) = X^*(-u,-v) \tag{3.8}$$

$$|\ X(u,v)\ | = |\ X(-u,-v)\ | \tag{3.9}$$

Hence all properties of the Fourier components can be found by considering only one half plane of the Fourier transform.

Another important property of the Fourier transform is the *translation property*, which is stated as:

$$x(m,n)\exp\left[2\pi i(u_0 m/M + v_0 n/N)\right] \leftrightarrow X(u-u_0, v-v_0) \tag{3.10}$$

$$x(m-m_0, n-n_0) \leftrightarrow X(u,v)\exp\left[2\pi i(um_0/M + vn_0/N)\right] \tag{3.11}$$

where $\leftrightarrow$ is used to indicate the relation between a signal and its Fourier transform. Hence the Fourier transform is translated by the vector (u_0, v_0), in the frequency domain, if the signal is multiplied with an exponential as indicated and vice versa. This property is exploited in order to obtain the so called *optical transform*, where the DC component of the transform is placed in the center of the image - i.e. at $(M/2, N/2)$ as shown in figure 3.1. The exponential term, with which the image should be multiplied, is written as:

$$\exp\left[2\pi i(m \cdot M/2 + n \cdot N/2)\right] = \exp\left[\pi i(m+n)\right] = (-1)^{(m+n)} \tag{3.12}$$

The *optical transform* emerges when the Fourier transform is performed optically using different diffraction gratings. This form is often preferred because its interpretation is easier.

It is an important property of the Fourier transform that *convolution* in the spatial domain corresponds to multiplication in the frequency domain. Assume that $x(m,n) \leftrightarrow X(u,v)$ and $y(m,n) \leftrightarrow Y(u,v)$. The *periodic convolution* is defined as:

$$x(m,n) \star y(m,n) = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} x(m,n) \cdot y(m-k, n-l) \tag{3.13}$$

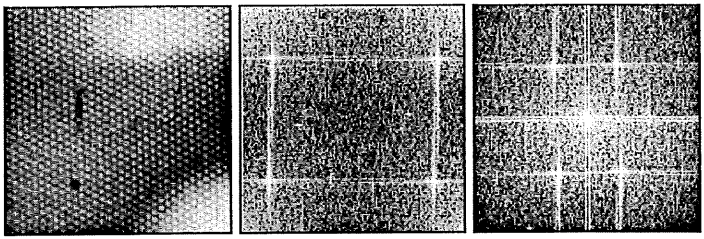

Figure 3.1. *The leftmost image is transformed using the DFT using the DFT standard transform. The result is shown in the middle. In the optical transform, shown to the right, the spectrum is shifted such that the DC value is placed in the middle of the image. Note the very pronounced striping in the image which is due to spectral leakage. The spectral leakage is caused by discontinuities in the periodic extension of the left image (i.e. at the edges of the image).*

The properties of convolution and multiplication is stated as:

$$x(m,n) \star y(m,n) \leftrightarrow X(u,v) \cdot Y(u,v) \tag{3.14}$$

$$x(m,n) \cdot y(m,n) \leftrightarrow X(u,v) \star Y(u,v) \tag{3.15}$$

Many other important properties of the Fourier transform may be listed. For a general exposition on the subject (and proofs) the reader is referred to Oppenheim & Schafer (1975) for 1-D Fourier analyses and Dudgeon & Mersereau (1984) or Lim (1990) for 2-D Fourier analysis.

3.3 Spectral leakage and windows

The harmonic estimates we obtain through the DFT are $M \times N$ uniformly spaced samples of the associated periodic spectra. Unfortunately, in many practical situations, the elegance of this approach must be compromised in order to obtain meaningful results. One such compromise consists of applying windows multiplicatively to the sampled data set. This is equivalent to convolution of the Fouriertranform with the frequency response of the chosen window. This will typically result in a smoothing of the spectral samples.

Signals with frequencies other than those of the basis set are not periodic in the observation window. The periodic extension of a signal not commensurate with its natural period exhibits discontinuities at the boundaries of the observation window. The discontinuities are responsible for spectral contributions over the entire basis set. This phenomenon is known as spectral leakage (see e.g. Harris (1978)) and is illustrated in figure 3.1 where the very pronounced lines in the estimated power spectrum is the results of discontinuities in the periodic extension of the image.

Windows are weighting functions, which may be applied to data in order to reduce the spectral leakage associated with the finite observation interval (or observation window). The window is applied multiplicatively to reduce the discontinuity at the boundaries of the periodic extension of the image.

In figure 3.2, a window is applied to the image shown in figure 3.1. The spectral leakage is reduced significantly. The estimate of the power spectrum is also smoother which is a side effect of the windowing scheme. This is discussed further in the next section.

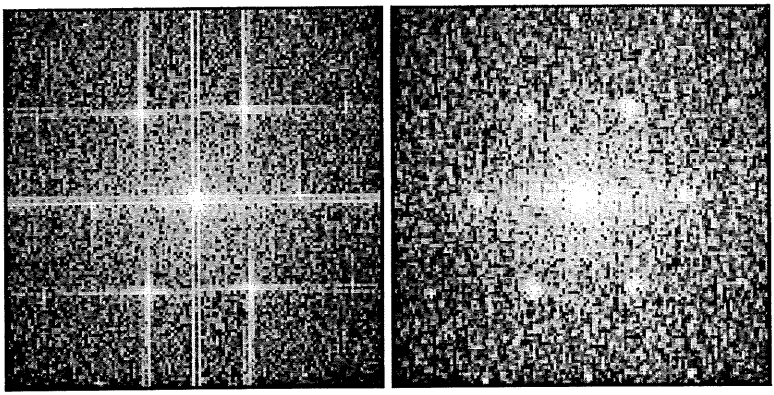

Figure 3.2. *The image to the left, is transformed using the DFT optical transform. The very pronounced striping in the image is due to spectral leakage. The spectral leakage is caused by discontinuities in the periodic extension of the image (which is shown in figure 3.1). A multiplicative window is used prior to the transform and the spectral leakage is removed as seen in the right image.*

3.4 Designing 2-D windows

The application of windowing schemes is important in may applications. As illustrated in the previous multiplicative data windows may be applied to avoid the effects of spectral leakage. Windows are also used to obtain a smoother estimate of the power spectrum than the periodogram estimate (which is an inconsistent estimate of the power spectrum).

The problem of choosing an appropriate shape for the 2D window is similar to the 1-D problem. The Fourier transform of the window (the frequency window) has a big central peak and small sidelobes. Using a multiplicative data window in the spatial domain is equivalent to convolve the Fourier transform of the signal with the frequency response of the window. Hence, if the frequency response of the window has large sidelobes, "ringing" will occur around discontinuities in the Fourier spectrum. When windows are used for spectral estimation, the frequency response of the window must be such that the spectral estimate remains positive. A sufficient (but not necessary) condition is that the frequency response of the window is positive.

Typically 2-D windows are designed from 1-D windows. There are two ways by which this is commonly done. The first method forms a 2D window with a square or rectangular region of support by taking the outer product of two 1D windows.

$$w_R(m,n) = w_1(|m - M/2|) \cdot w_2(|n - N/2|) \tag{3.16}$$

This method is described in e.g. Priestly (1981). Such windows are referred to as *separable windows*. The second method, introduced by Huang (1972), forms a window by sampling a circularly rotated, 1D, continuous window function:

$$w_C(m,n) = w(\sqrt{(m - M/2)^2 + (n - N/2)^2}) \tag{3.17}$$

It seems to be, at least empirically, well established that if the 1D windows are good, the 2D windows are good as well (See Huang (1972), Dudgeon & Mersereau (1984)).

Several 1-D windows have been investigated by e.g. Harris (1978). A few of these are listed below

- Rectangular (or truncation) window

$$w(n) = \begin{cases} 1 \ , & |n| \leq T/2 \\ 0 \ , & \text{otherwise} \end{cases} \tag{3.18}$$

- Bartlett (Fejer or triangular) window

$$w(n) = \begin{cases} 1 - |n|/T \ , & |n| \leq T/2 \\ 0 \ , & |n| > T/2 \end{cases} \tag{3.19}$$

- Tukey-Hanning window

$$w(n) = \begin{cases} \frac{1}{2}(1 + cos(\pi n/T)) \ , & |n| \leq T/2 \\ 0 \ , & |n| > T/2 \end{cases} \tag{3.20}$$

- Tukey-Hamming window

$$w(n) = \begin{cases} 0.54 + 0.46 \cdot cos(\pi n/T) \ , & |n| \leq T/2 \\ 0 \ , & |n| > T/2 \end{cases} \tag{3.21}$$

- Tukey α-taper window

$$w(n) = \begin{cases} 1.0 & , \quad 0 \leq |n| \leq (1-\alpha)\frac{T}{2} \\ \frac{1}{2}\left[1.0 + cos\left(\pi \frac{n-(1-\alpha)\frac{T}{2}}{\alpha \frac{T}{2}}\right)\right] & , \quad |n| > T \end{cases} \tag{3.22}$$

- Kaiser-Bessel window

$$w(n) = \begin{cases} \frac{I_0\left[\pi\alpha\sqrt{1.0-\left(\frac{n}{T/2}\right)^2}\right]}{I_0[\pi\alpha]} & , \quad |n| \leq T/2 \\ 0 & , \quad \text{Otherwise} \end{cases} \tag{3.23}$$

where $I_0(x)$ is the modified Besssel function of the first kind.

T is the *truncation* point.

When choosing a window, a large, narrow mainlobe and small sidelobes are desired. For a fixed region of support there is a trade-off between these two quantities. This is known as the *uncertainty principle* of signal processing (see e.g. Papoulis (1962)).

It is a research study in its own right to evaluate the performance of the different windows abundantly listed in the literature. Harris (1978) investigated more than twenty 1-D windows, and concluded that the Kaiser-Bessel window was among the top-performers.

Empirical studies support that the circular- and rectangular support 2-D Kaiser-Bessel windows are also good windows compared to other circular- and rectangular support 2D windows designed in a similar way. The Kaiser-Bessel window is an approximation to the finite prolate spheroidal sequences studied by e.g. Wilson & Spann (1988), where a similar conclusion, regarding the merits of this type of windows, was reached. The α parameter allows us to control the trade-off between mainlobe width and sidelobe height in a simple

way. The circular- and rectangular support Kaiser-Bessel windows are shown for different values of α in figure 3.3 and figure 3.4. In figure 3.6 the circular support Kaiser frequency windows from figure 3.3 are shown. It is seen that, as α get larger, the side lobes get smaller, but the main lobe get wider. This means that we must accept ripples (or ringing effects) if we wish to avoid blurring and vice versa. For the sake of completeness, the frequency windows for the rectangular support Kaiser windows are shown in figure 3.7.

Also, the Tukey-taper window is studied. This window represents an attempt to smoothly set the data to zero at the boundaries while not affecting the main part of the data. As α goes to zero we get a truncation window, and as α goes to one we get the Tukey-Hanning window. The (circular support) window is shown for different values of α in figure 3.5. For the values of α shown in figure 3.8, the spectral window exhibits a confusing array of side lobe values arising from the two components of this window type.

For many practical purposes the choice of a specific window type is not of great importance. It is, however, of great importance to choose a window. It is the experience of this author that circular support windows is a better choice than rectangular support windows as the latter type tends to put an emphasis on directions already present in the image due to the sampling geometry (as seen in e.g. figure 3.7). The circular support Tukey-taper window were used by Grunkin (1993) to remove periodic noise from bandlimited images.

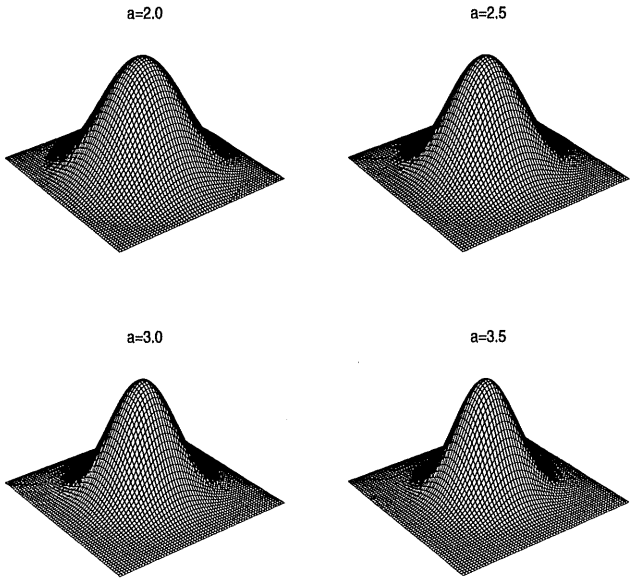

Figure 3.3. Spatial domain circular support Kaiser-Bessel windows. See figure 3.6 for the frequency response of the windows.

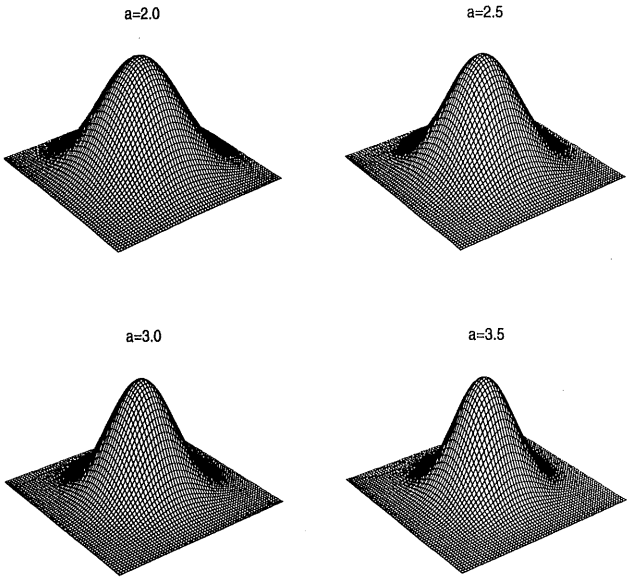

Figure 3.4. *Spatial domain rectangular support Kaiser-Bessel windows. See figure 3.7 for the frequency response of the windows.*

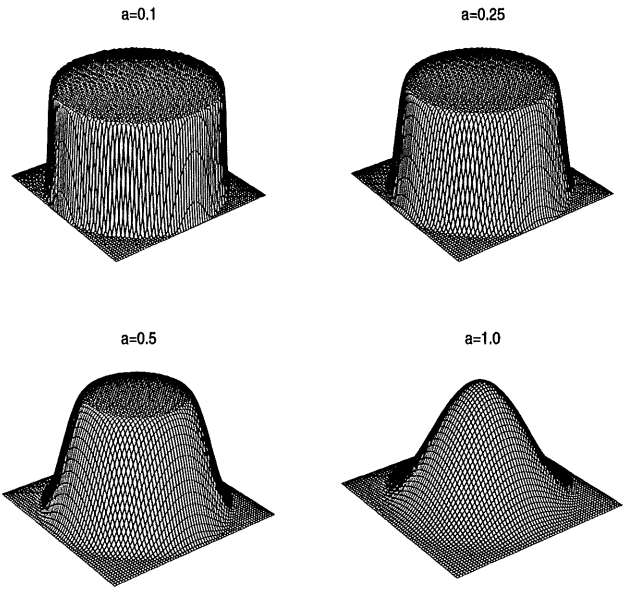

Figure 3.5. *Spatial domain circular support Tukey-taper windows. See figure 3.8 for the frequency response of the windows*

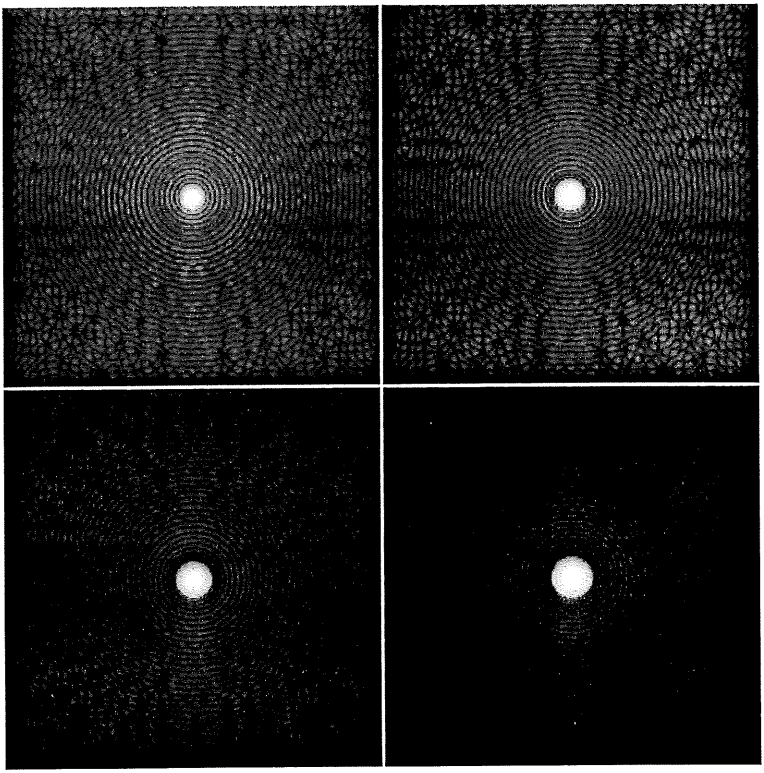

Figure 3.6. *Log-magnitude of circular support Kaiser frequency windows for (left to right, top down):* $\alpha = 2.0$, $\alpha = 2.5$, $\alpha = 3.0$ *and* $\alpha = 3.5$.

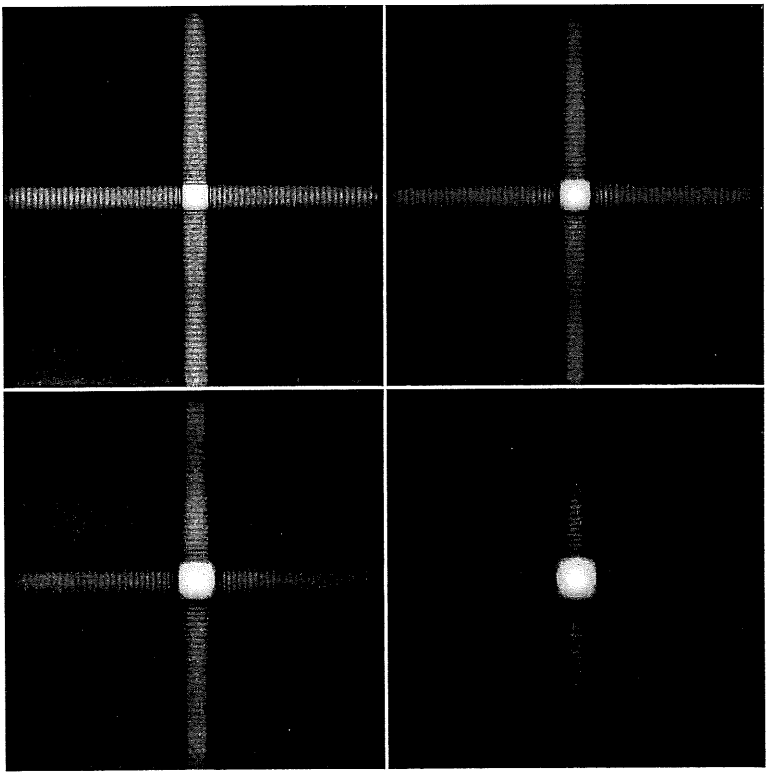

Figure 3.7. *Log-magnitude of rectangular support Kaiser frequency windows for (left to right, top down):* $\alpha = 2.0$, $\alpha = 2.5$, $\alpha = 3.0$ *and* $\alpha = 3.5$.

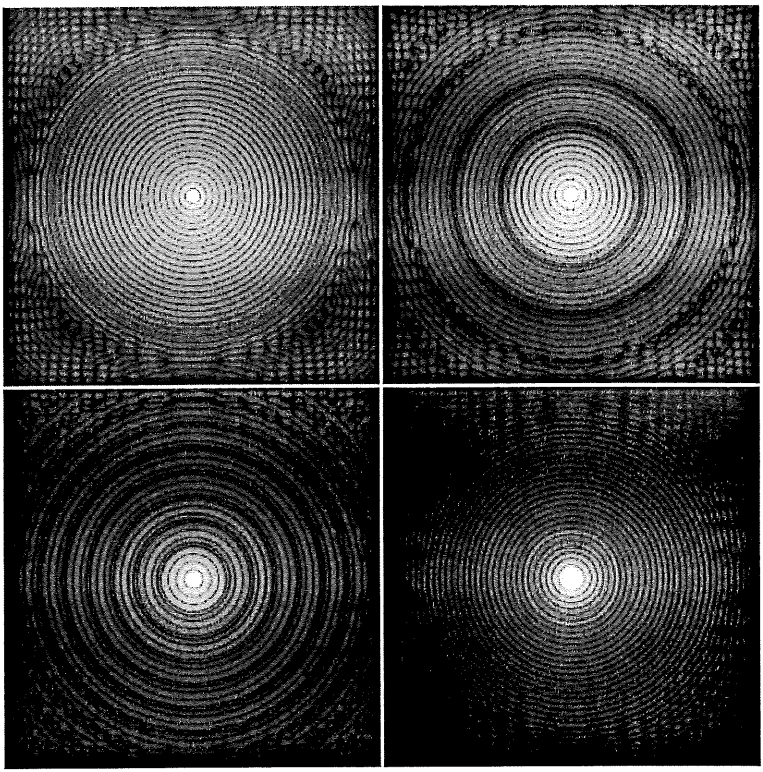

Figure 3.8. *Log-magnitude of circular support Tukey-taper windows for (left to right, top down):* $\alpha = 0.1$, $\alpha = 0.25$, $\alpha = 0.5$ *and* $\alpha = 1.0$.

3.5 The Fast Fourier Transform

One of the reasons that Fourier analysis is of such wide-ranging importance in digital signal- and image processing is the existence of efficient algorithms for computing the discrete Fourier transform. Direct calculation of the DFT in equation 3.1 would require a large number of floating point operations in the order of magnitude $(M \cdot N)^2$ multiplications and an equal number of additions. With the fast Fourier transform the computations are reduced dramatically to the order of magnitude $(M \cdot N)\log_2((M \cdot N))/2$ multiplications and $(M \cdot N)\log_2((M \cdot N))$ additions (using a row-column decomposition with 1-D FFT algorithm).

As the Fourier transform is separable, i.e. the 2-D Fourier transform is obtained as $M \cdot N$ successive 1-D Fourier transforms it is sufficient to discuss the 1-D fast Fourier transform here.

Traditionally the discovery of the Fast Fourier Transform is ascribed Cooley & Tukey (1965), although it is now known that Gauss used the Cooley-Tukey algorithm in the early nineteenth century (Heideman (1984)).

Most FFT algorithms, assumed to work on an N point sample, are exploiting two properties of the complex exponential $W_N = \exp(-2\pi i/N)$

$$W_N^{k(N-n)} = (W_N^{kn})^* \tag{3.24}$$

$$W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n} \tag{3.25}$$

which are known as the symmetry- and the periodicity properties respectively. Typically N is assumed to be a composite number, e.g. $N = 2^p$. The basic

principle of these algorithms is to decompose the computation of the DFT, of length N, into successively smaller DFT's.

Traditional approaches are the *Decimation in time* and *Decimation in frequency* FFT algorithms exploiting the properties in equation 3.24 and equation 3.25 respectively.

Even faster methods than these FFT algorithms exists today such as the *Winograd Fourier transform algorithm* (WFTA) and the *prime factor algorithm* (PFA).

For a detailed exposition on the topic the reader is referred to e.g. Oppenheim & Schafer (1975).

3.6 Periodogram and autocovariance

In 1906 Schuster proposed the modulus-squared of the finite Fourier transform as a vehicle for searching for hidden periodicities in data. This statistic was named the *Periodogram.* Even though the inconsistency of the periodogram-type statitistics is immediately apparent, when they are calculated from empirical functions, it is still applied in many different fields of natural sciences.

The periodogram estimate of the spectrum is given as

$$I(u,v) = \frac{1}{MN}\left|\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)\exp\left(-i2\pi(\frac{um}{M}+\frac{vn}{N})\right)\right|^2 \qquad (3.26)$$

The following important relation exists between the periodogram and the autocovariance function $\hat{\gamma}(k,l)$:

$$I(u,v) = \frac{1}{MN} \sum_{k=-(M-1)}^{(M-1)} \sum_{l=-(N-1)}^{(N-1)} \hat{\gamma}(k,l) \cdot \exp\left(-i2\pi(\frac{uk}{M} + \frac{vl}{N})\right) \tag{3.27}$$

where $\gamma(\hat{k}, l)$ is the biased, but positive definite, estimate of the auto covariance function. The relation holds for all $(k,l) \neq (0,0)$.

$$\hat{\gamma}(k,l) = \begin{cases} \frac{1}{(M)(N)} \sum_{i=0}^{M-1-k} \sum_{j=0}^{N-l-1} x(i,j) \cdot x(i+k, j+l) & k \geq 0, l \geq 0 \\ \frac{1}{(M)(N)} \sum_{i=0}^{M-1-k} \sum_{j=-l}^{N-1} x(i,j) \cdot x(i+k, j+l) & k \geq 0, l < 0 \end{cases} \tag{3.28}$$

for $k = -(M-1), \cdots, (M-1)$ and $l = -(N-1), \cdots, (N-1)$. The autocovariancefunction is symmetric around $(0,0)$ i.e

$$\begin{aligned} \gamma(k,l) &= \gamma(-k,-l) \\ \gamma(-k,l) &= \gamma(k,-l) \end{aligned} \tag{3.29}$$

Hence the periodogram may be written as:

$$I(u,v) = \gamma(0,0) + 2 \sum_{k=0}^{(M-1)} \sum_{l=-(N-1)}^{(N-1)} \gamma(k,l) \cdot cos\left(i2\pi(\frac{um}{M} + \frac{vn}{N})\right) \tag{3.30}$$

where $(k,l) \neq (0,0)$ in the summation.

As mentioned. the periodogram is an inconsistent estimate of the spectrum. When e.g. $x(m,n)$ is Gaussian white noise with variance σ_x^2. the properties of the periodogram is given in:

1. $I(u,v)$ are independent.

2. $\frac{2I(u,v)}{\sigma_x^2} \in \chi^2(2)$; $(u,v) \neq (0,0)$ and $(u,v) \neq (M/2, N/2)$.

3. $\frac{I(u,v)}{\sigma_x^2} \in \chi^2(1)$; $(u,v) = (0,0)$ and $(u,v) = (M/2, N/2)$.

(A proof may be found in e.g. Priestly (1981)).

Direct calculation of the autocovariance function, using equation 3.28 is not attractive. Therefore the autocovariance is often calculated using the FFT and exploiting the relation between the periodogram and the autocovariance function. When using Fourier transforms for this calculation wraparound errors may occur, especially for high lag indexes. This effect is due to the double periodic assumption of the Fourier transform. This can be avoided using windows or mean value padding of the image prior to the Fourier transform. The steps needed to perform this operation is outlined below:

- The image sequence $\{x(m,n)\}$ is padded with the average $\hat{\mu}_x$ thus creating a new image with the dimensions $2M \times 2N$ (of course the sequence must be multiplied with four (4) to maintain the correct DC value).
- The periodogram is obtained using the 2-D FFT.
- The autocovariance sequence is obtained as the inverse Fourier transform of the periodogram.

3.7 Conclusion

The definition, properties and implementation of the Discrete Fourier Transform (DFT) were briefly discussed. Spectral leakage and data windows were discussed and it was shown that the use careful of data windows reduces the effects of spectral leakage. The choice of appropriate data windows were discussed. It was shown that rectangular support windows put a strong emphasis on direction already present in the image due to the sampling geometry. Thus it is recommended to use circular support windows as described in this chapter.

The window functions described in this chapter were all implemented (in compliance with the *HIPS* standard) in the programming language *C*.

Chapter 4

Causal SARMA models

The ability to model stochastic spatial phenomena is very useful in diverse areas of application as e.g. classification, segmentation, restoration and spectral analysis. By using the class of interaction models known as Simultaneous AutoregRessive Moving Average (SARMA) models, which are generative (i.e. realizations of random fields obeying the model may be obtained through simulation), a subjective, visual evaluation of the fitted model is allowed for.

Whereas the time-series counterpart to these models is naturally causal, this notion is not natural in two dimensions. The non-causality, however, introduces a number of difficulties in e.g. the estimation (as we shall see in chapter 5). Therefore it is popular to impose an artificial directionality on the image, by choosing a causal (or unilateral) model [1].

The application of unilateral models is an attempt to generalize the very powerful ARMA time series models to two dimensions. When imposing a directionality in the image, by introducing a unilateral model, many important

[1] The terms causal and unilateral are used interchangeably.

results from time series analysis are directly applicable in two dimensions. See e.g. Box & Jenkins (1976) and Priestly (1981) for a comprehensive treatment of time series models.

In this chapter the unilateral SARMA models are investigated as a vehicle for describing observed textures. From a statistical point of view it is interesting to be able to model a stochastic phenomenon with only a very few parameters, thus obtaining, not only a substantial data reduction, but also allowing for a physical interpretation of the estimated parameters. We shall try to outline, to what extent observed textures can be modelled with this class of models. This introduces a number of well known problems such as model identification, estimation and validation, which are considered in this chapter.

4.1 The SARMA model

Assume a wide sense, zero mean, stationary stochastic process defined on a rectangular grid of pixel sites $\Omega = \{s = (s_1, s_2) : 0 \leq s_1 < M \ , \ 0 \leq s_2 < N\}$, obeying a SARMA model defined as

$$y_s + \sum_{r \in N_{AR}} \phi_r y_{s-r} = \epsilon_s + \sum_{r \in N_{MA}} \theta_r \epsilon_{s-r} \tag{4.1}$$

where $r = (r_r, r_c) \in N_{AR}$ or N_{MA}. N_{AR} and N_{MA} are the regions of support for the parameter arrays $\{\phi\}$ and $\{\theta\}$ respectively. We also refer to these sets as the neighborhood of the model. The noise process ϵ_s is i.i.d. $N(0, \sigma_\epsilon^2)$ or *white noise*. If $N_{MA} = \{\}$ we have a SAR model, and if $N_{AR} = \{\}$ we have a SMA model.

The corresponding system function (or transfer function) is given as

$$H(z_r, z_c) = \frac{X(z_r, z_c)}{E(z_r, z_c)} = \frac{1 + \sum_{(k,l)\in N_{MA}} \theta_{k,l} z_r^{-k} z_c^{-l}}{1 + \sum_{(k,l)\in N_{AR}} \phi_{k,l} z_r^{-k} z_c^{-l}} \tag{4.2}$$

The model is stationary if

$$\{1 + \sum_{(m,n)\in N_{AR}} \phi_{m,n} z_r^m z_s^n \neq 0\} \tag{4.3}$$

for all $|z_r| \geq 1$ and $|z_c| \geq 1$, i.e. all poles of the system function lie within the unit sphere.

To ensure invertibility we require

$$\{1 + \sum_{(m,n)\in N_{MA}} \theta_{m,n} z_r^m z_s^n \neq 0\} \tag{4.4}$$

for all $|z_r| \geq 1$ and $|z_c| \geq 1$, i.e. all roots of the system function lie within the unit sphere.

It is not a simple matter to test for stability in two dimensions. Some stability theorems and approaches may be found in e.g. O'Conner & Huang (1981), Dudgeon & Mersereau (1984) or Lim (1990). Stability criteria are quite cumbersome to apply in practice and the problem has not been pursued further in this thesis. Tjøstheim (1981), however, suggests simulation applied as a stability test for the estimated model. If the output grows without bounds, the estimated model is not stable. This approach has been used extensively throughout this thesis.

The (non-normalized) spectral density function of y_s is given as the squared modulus of the frequency response function multiplied by the noise variance:

$$P(f) = \sigma_\epsilon^2 \frac{\left|1 + \sum_{r \in N_{MA}} \theta_r exp(-2\pi i \cdot r^T \cdot f)\right|^2}{\left|1 + \sum_{r \in N_{AR}} \phi_r exp(-2\pi i \cdot r^T \cdot f)\right|^2}$$
$$= \sigma_\epsilon^2 \frac{\left(1 + \underline{\theta}^T \underline{C}(N_{MA})_f\right)^2 + \left(\underline{\theta}^T \underline{S}(N_{MA})_f\right)^2}{\left(1 + \underline{\phi}^T \underline{C}(N_{AR})_f\right)^2 + \left(\underline{\phi}^T \underline{S}(N_{AR})_f\right)^2} \tag{4.5}$$

where the spatial frequencies $(f_r, f_c) = f \in \Omega_f$, and

$$\Omega_f = \{f = (f_r, f_c) = (-\frac{1}{2} + \frac{r}{M}, -\frac{1}{2} + \frac{c}{N}) : \ (r, c) \in \Omega\} \tag{4.6}$$

The vector notation is defined below:

$$\begin{aligned}
\underline{\phi} &= \underline{col}\,[\phi_r, \ r \in N_{AR}] \\
\underline{\theta} &= \underline{col}\,[\theta_r, \ r \in N_{MA}] \\
\underline{C}(N.)_f &= \underline{col}\,[cos\,(\omega(f,r)), \ r \in N.] \\
\underline{S}(N.)_f &= \underline{col}\,[sin\,(\omega(f,r)), \ r \in N.] \\
\omega(f,r) &= 2\pi(r_r \cdot f_f + r_c \cdot f_c)
\end{aligned} \tag{4.7}$$

In time series analysis, and general signal processing, there has been a large research effort concentrated on developing efficient spectral estimators using, among others, ARMA modelling. An excellent review is given by Kay (1988). The successful use of ARMA modelling for parametric spectral estimation in 1-D, has motivated research in 2-D extensions of these methods. In chapter 6 SARMA models are used to obtain parametric spectral estimates of sequences of harmonic series embedded in white noise.

Causal models are typically either Quarter-Plane (QP) or Non-Symmetrical Half-Plane (NSHP) models. Such models have been investigated by e.g. Marzetta (1980), Jain (1981), Tou (1980), Tjøstheim (1981), Kashyap & Eom (1988) and Zhang & Cheng (1991) with applications as spectral estimation, texture classification and restoration. Apart from specifying the type of support for a given model, the *order* of the model must be specified. In time series analysis, the order of the model is given by the simple and intuitive notation $ARMA(p,q)$ where p is the order of the AR part, and q the order of the MA part. In two dimensions, however, the order of the model must be associated with a specific shape - typically a rectangle or an ellipse. By introducing a simple notation for these two shapes, we can be specific about the order of the model.

Definition 4 *A rectangle with side lengths* $2a+1$ *and* $2b+1$, *rotated by an angle* α *around the origin is denoted* $R(a,b,\alpha)$. *We now define*

- $R(a,b) = R(a,b,0)$
- $R(a) = R(a,a,0)$

Definition 5 *An ellipse with semi-major axis a and semi-minor axis b, rotated by an angle* α *around the origin, is denoted* $E(a,b,\alpha)$. *We now define*

- $E(a,b) = E(a,b,0)$
- $E(a) = E(a,a)$

Now the support of a given model is defined in terms of the set of (integer) lattice points contained in the intersection of one of the supports defined in equation 2.15 (NSHP) or 2.13 (QP) and a rectangle or an ellipse as defined above. Note that $(0,0)$ *does not* belong to the neighborhood of a model.

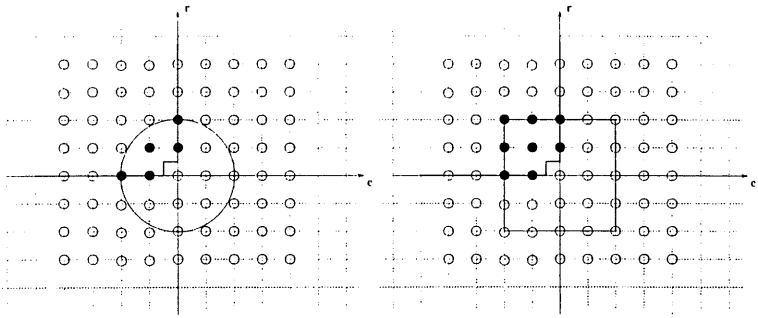

Figure 4.1. *$QP(+,-)$-models of order $E(2)$ and $R(2)$ respectively.*

This is illustrated in the figures 4.1 and 4.2. A SARMA process is completely specified in terms of the neighbor sets, the associated sets of parameters and a noise variance. Without loss of generality we will assume zero mean processes.

4.2 Simulating toroidal SARMA textures

Among the virtues of the random field models considered here is the fact that they are generative - i.e. it is easy to generate realizations (simulate) from the model. This is an important feature in a texture analysis context, where the "quality" of the fitted model may be assessed visually, by comparing a synthesized texture, obeying a fitted model, to the original texture. Although this is a subjective measure, and as such not a substitute for a formal model validation, it tends to be a good way to evaluate the fitted model. One should bear in mind that an optimal model, in a statistical sense, may not be an optimal model in a visual sense. Finally, simulation is sometimes the best, if not the only, way to investigate certain model characteristics that could not have been investigated analytically or in any other way.

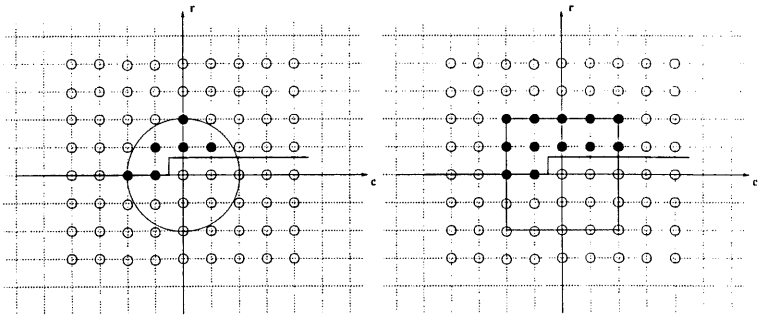

Figure 4.2. *$NSHP(+,\ominus)$ models of order $E(2)$ and $R(2)$ respectively.*

Two different ways of obtaining textures obeying a specified SARMA model are described. One method is based on relaxation, the other exploits the knowledge of the frequency response function.

The random generator used in all simulation experiments is the IMSL routine *rnnor* that generates independent Gaussian samples of unit variance using an inverse CDF method. This random generator is based on a linear multiplicative congruential method recommended by e.g. Ripley (see Ripley (1987) or Ripley (1990)).

4.2.1 Simulation by relaxation

We assume knowledge of the model parameters and the variance of the driving Gaussian (white) noise. Considering the model definition 4.1, we can calculate

$$w_s = \epsilon_s + \sum_{r \in N_{MA}} \theta_r \epsilon_{s-r} \tag{4.8}$$

once and for all, and then solve the equation

$$\Phi \underline{y} = \underline{w} \tag{4.9}$$

(where Φ is a sparse matrix) w.r.t. $\underline{y}$. This approach was suggested by e.g. V. Novikov & Valteris (1985), Kashyap (1981) or Woods (1972). This rather large system of equations is solved iteratively using e.g. the Gauss-Seidel method (relaxation). The algorithm is outlined in figure 4.3. In most cases, encountered in practice, the speed of convergence is increased a little by updating each pixel immediately in a single work array instead of storing both $\underline{y}^i$ and $\underline{y}^{i-1}$. As the model approaches non-stationarity, the convergence of this method slows down considerably. On the other hand, one may be interested in investigating the local properties of a non-stationary model. This can be achieved by just iterating a few times and stop before the output blows up numerically . This cannot be done with the frequency based method described in the next section, as singularities are introduced in the frequency response function.

4.2.2 Frequency domain simulation

When the parameters of a specific model are given, we can calculate the frequency response, $\mathcal{H}(f)$, for every spatial frequency $f \in \left[-\frac{1}{2}, \frac{1}{2}\right[\times \left[-\frac{1}{2}, \frac{1}{2}\right[$.

For a SARMA model we have the frequency response

$$\mathcal{H}(f) = \frac{\left(1 + \sum_{r \in N_{MA}} \theta_r exp(-2\pi i f^T r)\right)}{\left(1 + \sum_{r \in N_{AR}} \phi_r exp(-2\pi i f^T r)\right)} \tag{4.10}$$

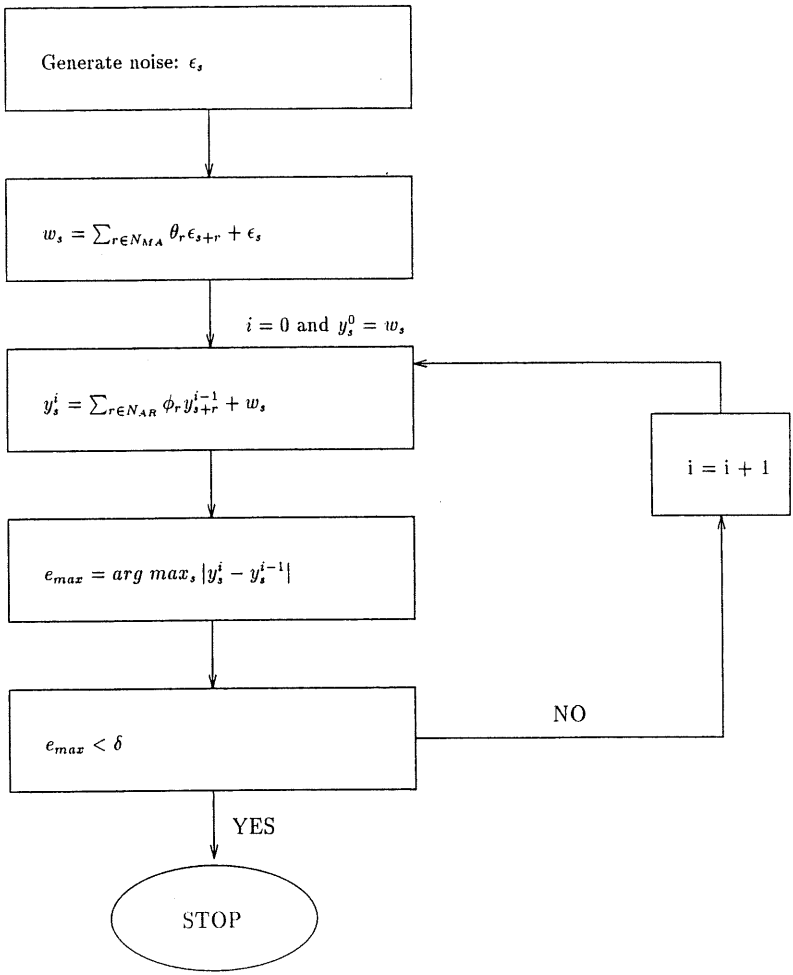

Figure 4.3. *Outline of an algorithm for simulating a SARMA process.*

Using the duality between convolution and multiplication in the spatial- and spatial frequency- domain respectively, we obtain

$$\mathcal{Y}(f) = \mathcal{H}(f) \cdot \mathcal{E}(f) \tag{4.11}$$

where $\mathcal{Y}(f)$ is the Fourier Transform of the texture with the desired properties, $\mathcal{H}(f)$ is the frequency response, and $\mathcal{E}(f)$ is the Fourier Transform of the generated noise (of variance σ_ϵ^2).

After performing an inverse discrete Fourier transform, we have a realization of a texture with the desired characteristics. Note that these textures are inherently torus due to the assumption of the discrete Fourier transform, namely that the image is repeating itself periodically in every direction. This method was suggested by Kashyap (1981).

The method outlined here is extremely sensitive close to non-stationarity (where, as discussed in the following sections, most natural textures tends to lie). This creates either singularities, or at least very large peaks in the frequency response function, that seriously degrade the visual impression of the simulated texture although salient features such as e.g. anisotropy are preserved.

4.3 Identification and Estimation of SARMA models

In the following it is assumed that a finite array of observations, $\{y_s, s \in \Omega\}$, is sampled on a rectangular grid from a zero mean, wide sense stationary and ergodic process (i.e. the time/space averages can replace the ensemble average). From this sample an appropriate model must be identified and the associated

parameters estimated. The estimation of SAR and mixed SARMA models are considered separately. The estimates of the SAR parameters are obtained directly using the Least Squares method. For mixed SARMA models the Conditional Least Squares (CLS) estimates are obtained using non-linear optimization. A back forecasting algorithm is proposed to obtain Unconditional Least Squares (ULS) estimates. Finally, a backwards elimination method for model identification is proposed.

4.3.1 LS-estimation in linear dynamic models

The SAR model in 4.12 is a special case of SARMA models, where N_{MA} is empty. This type of model is quite popular for a number of reasons. They are relatively flexible and, above anything else, it is simple to estimate the parameters. In fact, it is often preferred to use a high order SAR model for a given set of data, instead of using the more parsimonious SARMA model, where the estimator is strongly non-linear.

For the pure SAR model we have the model equation:

$$y_s + \sum_{r \in N_{AR}} \phi_r y_{s-r} = \epsilon_s \tag{4.12}$$

where, for estimation purposes, we shall only consider $s \in \Omega_I$, i.e. the *inner* set. A site, s, belongs to the inner set Ω_I, if all sites in $\underline{z}(s)$ (which is defined in 4.13) are contained in Ω.

An observation- and a parameter- vector is defined

$$\underline{z}(s) = \underline{col}\,[y_{s-r}\ r \in N_{AR}] \tag{4.13}$$

$$\underline{\Theta} = \underline{col}\,[\phi_r \; r \in N_{AR}]$$

and the one-step-ahead prediction is expressed in terms of these:

$$\hat{y}_s = -\sum_{r \in N_{AR}} \phi_r y_{s-r} = -\underline{\Theta}^T \underline{z}(s) \tag{4.14}$$

A function that measures the sum of squared *prediction errors* is defined for a given set of parameters and observations:

$$\begin{aligned} S_y(\underline{\Theta}) &= \sum_{s \in \Omega_I} \left(y_s + \sum_{r \in N_{AR}} \phi_r y_{s-r} \right)^2 \\ &= \sum_{s \in \Omega_I} \left(y_s + \underline{\Theta}^T \underline{z}(s) \right)^2 \end{aligned} \tag{4.15}$$

The Least Squares estimate is the set of parameters, $\hat{\underline{\Theta}}$, that minimizes $S(\underline{\Theta})$ and is obtained as:

$$\hat{\underline{\Theta}} = \left(\sum_{s \in \Omega_I} \underline{z}(s) \underline{z}(s)^T \right)^{-1} \left(- \sum_{s \in \Omega_I} y_s \underline{z}(s) \right) \tag{4.16}$$

As an estimate of $V[\epsilon_s] = \sigma_\epsilon^2$ we use

$$\hat{\sigma}_\epsilon^2 = \frac{1}{\underline{card}\Omega_I - \underline{card} N_{AR}} S(\hat{\underline{\Theta}}) \tag{4.17}$$

where $\underline{\text{card}}$ means the *cardinality* of the argument set of pixel sites.

The properties of the LS estimator is stated in:

Theorem 1 *If ϵ_s is Gaussian white noise, the LS estimator possesses the following properties*

i) $\hat{\underline{\Theta}}$ *is consistent.*

ii) $\hat{\underline{\Theta}}$ *is asymptotically efficient.*

iii) $\hat{\underline{\Theta}}$ *is asymptotically Gaussian with*

$$V\left[\hat{\underline{\Theta}}\right] \approx \sigma_\epsilon^2 \left[\sum_{s\in\Omega_I} \underline{z}(s)\underline{z}(s)^T\right]^{-1} \tag{4.18}$$

If ϵ_s is not Gaussian, property ii) is no longer valid.

Proof: See e.g. Box & Jenkins (1976) (or Madsen (1989) for a Danish reference).

In section 4.5 pure SAR models are used to model visual textures and the estimated models are evaluated visually through simulation.

4.3.2 The prediction error method

The residuals are, by assumption, an estimate of the underlying innovation process. This estimate is, of course, dependent upon the choice (estimate) of parameters. Therefore, when we use a model that incorporates interdependence between the noise variates and observations, the estimation problem becomes non-linear. As we saw in the preceding section, the LS estimates were obtained by minimizing the sum of squared residuals. The residuals were written in terms of the observations and the conditional expectation (prediction):

$$\epsilon_s = y_s - E\left[y_s | \underline{z}(s)\right] \tag{4.19}$$

This method is generalized to the SARMA process

$$y_s + \sum_{r \in N_{AR}} \phi_r y_{s-r} = \epsilon_s + \sum_{r \in N_{MA}} \theta_r \epsilon_{s-r} \tag{4.20}$$

where, again, the innovation process is assumed to be white Gaussian noise.

By choosing a unilateral model, a direction is imposed on the image, and the one-step-ahead prediction error is calculated *recursively* for a given set of parameters

$$\underline{\Theta} = \underline{col} \begin{bmatrix} \phi_r & , & r \in N_{AR} \\ \theta_r & , & r \in N_{MA} \end{bmatrix} \tag{4.21}$$

The recursive calculation procedure is illustrated in figure 4.4.

Given the one-step-ahead prediction

$$\hat{y}_s(\Theta) = -\sum_{r \in N_{AR}} \phi_r y_{s-r} + \sum_{r \in N_{MA}} \theta_r \hat{\epsilon}(\underline{\Theta})_{s-r} \tag{4.22}$$

where the notation $\hat{y}_s(\Theta)$ and $\hat{\epsilon}(\underline{\Theta})_{s-r}$ indicates that the prediction, as well as the prediction errors, depends upon the parameters.

The prediction error is obtained as

$$\hat{\epsilon}(\underline{\Theta})_s = y_s - \hat{y}_s(\Theta) \tag{4.23}$$

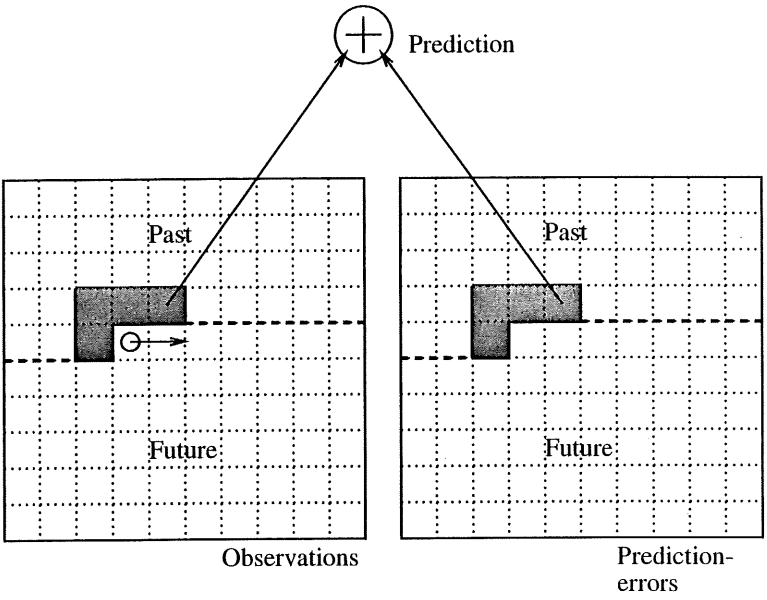

Figure 4.4. *This figure illustrates the recursive calculation of the one-step-ahead prediction error. First the one-step-ahead prediction is calculated as a weighted sum of observations and "old" prediction errors. Then the new prediction error is calculated, and used in the next prediction. The model illustrated is a* $NSHP(\oplus,+) - SARMA(R(1),R(1))$ *model.*

and again the function $S_y(\underline{\Theta})$ is defined in terms of the squared prediction errors

$$S_y(\underline{\Theta}) = \sum_{s \in \Omega_I} \hat{\epsilon}(\underline{\Theta})_s^2 \tag{4.24}$$

This function, also referred to as the objective functional, is minimized w.r.t. the parameters.

The properties of the estimator is given in

Theorem 2 *If ϵ_s is Gaussian white noise, the prediction error estimator possesses the following properties*

i) $\hat{\underline{\Theta}}$ *is consistent and asymptotically efficient.*

ii) $\hat{\underline{\Theta}}$ *is asymptotically Gaussian with* $E\left[\hat{\underline{\Theta}}\right] = \Theta$ *and the dispersion of the parameters*

$$V\left[\hat{\underline{\Theta}}\right] \approx 2\sigma_\epsilon^2 H^{-1} \tag{4.25}$$

where the Hessian *matrix* $\boldsymbol{H}$ *is determined by*

$$\{h_{kl}\} = \frac{\partial^2 S_y(\underline{\Theta})}{\partial\theta_k \partial\theta_l} \Big|_{\underline{\Theta}=\hat{\underline{\Theta}}} \tag{4.26}$$

and measures the curvature of $S_y(\underline{\Theta})$ at the optimum.

Proof: See e.g. Box & Jenkins (1976).

It can be shown that the prediction error estimates, under the Gaussian assumption, are equal to the Maximum Likelihood estimates (see e.g. Box & Jenkins (1976))

It is important to note that this method, ideally, requires an infinite number of "earlier" observations. There are two ways to deal with this problem referred to as Conditional Least Squares (CLS) or Unconditional Least Squares (ULS). These two methods are discussed subsequently.

Irrespective of the estimation scheme chosen (CLS or ULS), the estimation problem is highly non-linear in the parameters. Hence non-linear optimization methods, as described in section 2.4, must be applied. As this is a computationally very demanding task, mixed SARMA models are typically estimated using approximations (See e.g. Cadzow & Ogino (1981) or Zhang & Cheng (1991)). In section 4.6 however the exact CLS estimates are used to model visual texture.

Conditional Least Squares (CLS)

When calculating the one-step-ahead prediction in 4.22 recursively, one is faced with an initial-value problem for the prediction errors. One way of dealing with this problem is to fix the initial residual values to zero. This method is sometimes referred to as Conditional Least Squares (see e.g. Abraham & Ledolter (1983)), since the estimate of $\hat{\underline{\Theta}}$ is conditional w.r.t. these initial values.

The CLS estimate is now obtained as

$$\hat{\underline{\Theta}}_{CLS} = arg\ min_{\Theta} \left\{ S_y(\Theta) = \sum_{s \in \Omega_I} \hat{\epsilon}^2(\Theta) \right\} \tag{4.27}$$

It is obvious, that the effect of fixing the initial values of the residuals to zero, is merely transient and for large images the effect will be diluted. For models close to non-stationarity, however, the initial values may have a large influence.

$r_i \in N$	θ_i	mean of $\hat{\theta}_i$	Sdev. of $\hat{\theta}_i$	$\left[\overline{\theta}_i - \frac{2\sigma_{\theta_i}}{10} ; \overline{\theta}_i + \frac{2\sigma_{\theta_i}}{10}\right]$
$AR(1,1)$	-0.300	-0.320	0.080	$[-0.336; -0.304]$
$AR(1,0)$	-0.250	-0.175	0.138	$[-0.203; -0.147]$
$AR(0,1)$	0.100	0.028	0.119	$[0.004; 0.052]$
$MA(0,1)$	0.500	0.425	0.122	$[0.401; 0.449]$
$MA(1,0)$	-0.300	-0.222	0.152	$[-0.253; -0.192]$

Figure 4.5. *The table summarizes the experiment described in section 4.3.2 for 100* 16×16 *simulated textures obeying the model specified in the first two columns. The third- and fourth column describes the empirical mean and standard deviation of the estimated parameters. The fifth column is the empirical 95%-confidence interval for the mean of the estimated parameters.*

To illustrate this, two simple experiments are carried out. In each experiment 100 samples from a mixed QP-SARMA(R(1),E(1)) model are generated (the parameters of the model are shown in the summaries in figure 4.5 and figure 4.6). The sample sizes in the two experiments conducted are 16×16 and 128×128 respectively. The estimates based on the small samples are clearly more biased than the estimates based on the the larger samples. This is at least partly due to the dilution of the transient effect from the chosen initial values of the prediction errors.

Unconditional Least Squares (ULS)

Instead of fixing all initial values of the residuals to zero, an estimate of these initial values may be obtained using a *back forecasting* procedure. This approach is used in time series analysis for short series (see e.g. Box & Jenkins (1976)). In this section a back forecasting scheme for causal SARMA models is

$r_i \in N$	θ_i	mean of $\hat{\theta}_i$	Sdev. of $\hat{\theta}_i$	$\left[\overline{\theta}_i - \frac{2\sigma_{\theta_i}}{10} ; \overline{\theta}_i + \frac{2\sigma_{\theta_i}}{10}\right]$
$AR(1,1)$	-0.300	-0.299	0.009	$[-0.300(4); -0.297]$
$AR(1,0)$	-0.250	-0.250	0.021	$[-0.255; -0.246]$
$AR(0,1)$	0.100	0.097	0.023	$[0.092; 0.101]$
$MA(0,1)$	0.500	0.501	0.018	$[0.497; 0.505]$
$MA(1,0)$	-0.300	-0.297	0.022	$[-0.301; -0.293]$

Figure 4.6. *The table summarizes the experiment described in section 4.3.2 for 100* 128 × 128 *simulated textures obeying the model specified in the first two columns. The third- and fourth column describes the empirical mean and standard deviation of the estimated parameters. The fifth column is the empirical 95%-confidence interval for the mean of the estimated parameters.*

proposed, and it is shown that a significant improvement over the CLS method may be obtained, using this method.

Suppose that a random field is generated by the stationary *forward* model

$$y_s + \sum_{r \in N_{AR}} \phi_r y_{s-r} = \epsilon_s + \sum_{r \in N_{MA}} \theta_r \epsilon_{s-r} \tag{4.28}$$

This field could equally well have been generated by the *backwards* model

$$y_s + \sum_{r \in N_{AR}} \phi_r y_{s+r} = e_s + \sum_{r \in N_{MA}} \theta_r e_{s+r} \tag{4.29}$$

which is used to generate the backward forecasts. The back forecasts needed for a QP model are outlined in figure 4.7. As it may be necessary to repeat the back forecasting procedure more than once, the forward model is used to predict values beyond the right border of the image whereas the backward

model is used to predict beyond the left side. It is not obvious, however, how one should predict values beyond the borders of the image. The problem is that predictions beyond the image require *observations* which are not known. This problem is illustrated in figure 4.8 where the "missing" observations are shown. The result of the first step of the back forecasting procedure is also shown for a 16×16 image. The simplest approach to this problem is to replace the "missing" observations with the expectation (i.e. zero).

When the back forecasts are generated, the forward model is used to generate the prediction errors. It is important to note that the noise process of the backwards model is different from the noise process of the forwards model, and cannot be used as prediction error estimates. The second step of the back forecasting procedure is shown in figure 4.9. The first and the second step outlined here constitute the first iteration in the back forecasting procedure. It may be necessary to continue the iterations until the sum of squared predictions errors (the objective functional) stops changing. Practical experiments show that the number of iterations needed, often grows significantly when the intermediate models obtained by the optimization procedure approach non-stationarity.

In figure 4.8 and figure 4.9 it is seen that the back forecasts die out rather quickly. The closer the model is to non-stationarity, the slower the back forecasts die out.

The experiment that were performed for the conditional estimation scheme is repeated for the unconditional estimation scheme in order to assess the influence of back forecasting. The results are summarized in figure 4.10, and indicate that a significant improvement of the estimates may be obtained in this way. When an empirical 95% confidence interval is considered for the mean of the CLS estimates, summarized in figure 4.5, *none* of the intervals contain the "true" value. When using the back forecasting scheme, the confidence

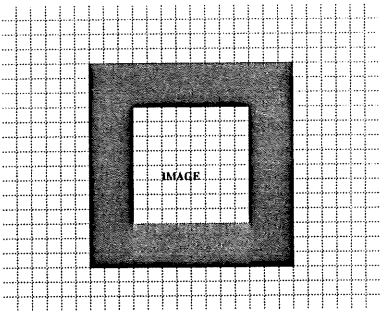

Figure 4.7. *The hatched area marks the back forecasts needed to estimate the initial values for the ULS estimator.*

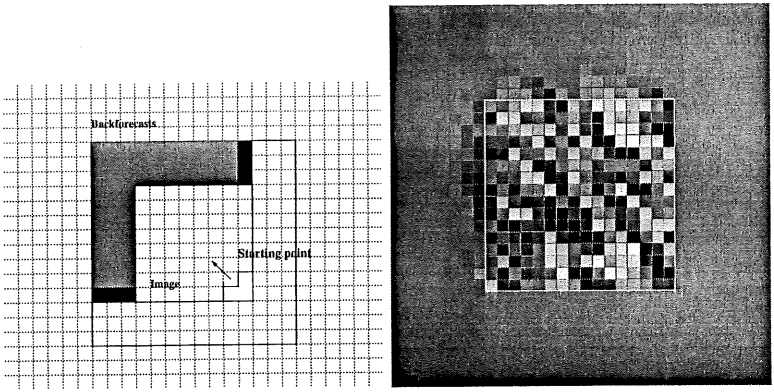

Figure 4.8. ***Left:*** *How the back forecasts are generated with the backwards model. The darkest areas represent unknown observations needed to predict beyond the image. These are set to the estimated mean of the image.* ***Right:*** *The result of performing the first back forecasting step.*

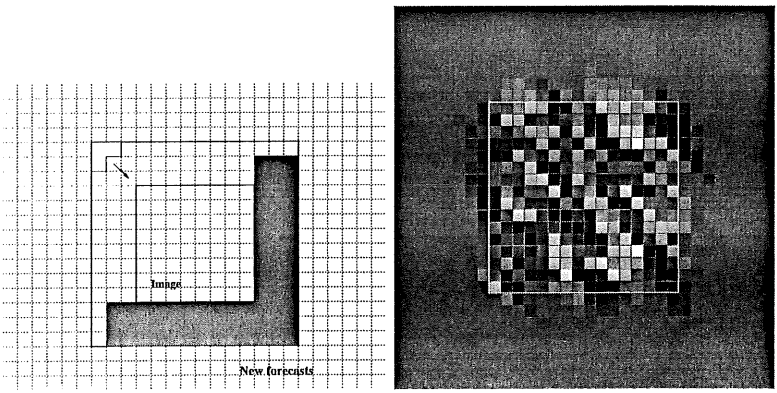

Figure 4.9. ***Left:*** *How the forwards model is used to determine the prediction errors and predict beyond the right side of the image.* ***Right:*** *The result of performing the second step of the back forecasting procedure.*

intervals for *three out of five* parameters contain the "true" parameter (for AR(1,0), AR(0,1) and MA(1,0)).

There are, however, serious problems that must be dealt with if back forecasting is to be applied in general:

- During the non-linear optimization a non-stationary model may be reached as an intermediate result. Hence the prediction errors and the back forecasts may grow without bounds.

- It seems that some of the estimated parameters of the model are slightly biased. This is illustrated in figure 4.11 where a histogram is shown for the estimated parameters. The reasons for this is not yet understood, but may be due to the "missing" observations upon which the estimates are conditioned. Although biased the results are still an improvement

$r_i \in N$	θ_i	mean of $\hat{\theta}_i$	Sdev. of $\hat{\theta}_i$	$[\overline{\theta}_i - \frac{2\sigma_{\theta_i}}{10}\ ; \overline{\theta}_i + \frac{2\sigma_{\theta_i}}{10}]$
$AR(1,1)$	-0.300	-0.339	0.095	$[-0.358; -0.320]$
$AR(1,0)$	-0.250	-0.248	0.153	$[-0.279; -0.217]$
$AR(0,1)$	0.100	0.132	0.163	$[0.067; 0.133]$
$MA(0,1)$	0.500	0.571	0.157	$[-0.602; -0.540]$
$MA(1,0)$	-0.300	-0.324	0.168	$[-0.358; -0.290]$

Figure 4.10. *The table summarizes the experiment described in section 4.3.2 for 100* 16×16 *simulated textures obeying the model specified in the first two columns. The back forecasting algorithm was used to obtain the estimates. The third and fourth column describe the empirical mean and standard deviation of the estimated parameters. The fifth column is the empirical 95%-confidence interval for the mean of the estimated parameters.*

when compared to the unconditional estimation where the bias is much larger.

This problem is not pursued further in this thesis but remain a part of ongoing research.

Practical remarks

Irrespective of the estimation scheme chosen for a given SARMA model the estimation problem must be solved using non-linear optimization. Non-linear optimization is never an easy problem to tackle and it may be relevant to make a few practical remarks.

It is extremely important to choose the initial value of the parameters prudently. Apart from avoiding many needless iterations there is a real risk that

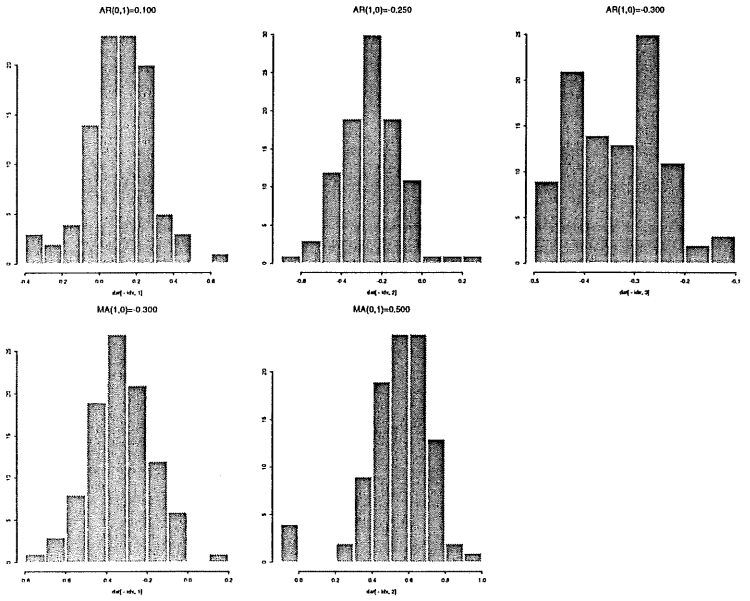

Figure 4.11. *The figure shows histogram plots of the parameters in the experiment described in section 4.3.2 for 100* 16×16 *simulated textures. The parameters were estimated using the back forecasting algorithm. Some of the parameters are clearly biased.*

the optimization never converges. A good initial guess seems to be the LS estimates for the autoregressive parameters and 0.1 for all the moving average parameters [1].

As mentioned several times, the models obtained for visual textures lies "on the edge of non-stationarity". This means that intermediate parameter values, obtained during the optimization, may very well render the model non-stationary. This, in turn, may give infinitely high values of the objective functional (this, of course, is particularly the case for large images). The only way to avoid this situation is to choose the step-length sufficiently small. It must, however, not be chosen too small as this will prevent the optimization from converging.

In such cases it becomes very important to obtain a valid estimate of the gradient. In some situations the FDA and BDA gradients (see section 2.4) are inadequate and the global optimum is never found. In such cases the computational load of the CDA gradient must be accepted. The CDA gradient is not extremely sensitive to the choice of δ.

Finally it should be mentioned that it is not necessary to require the norm of the gradient to be very small. Such a requirement is very difficult to meet in practice, especially for large models, and the estimates obtained seems to be perfectly valid for less restrictive convergence criteria.

4.3.3 Determining the order of the model

Any model-based approach to texture analysis consists of two stages. The first stage is to identify an appropriate *order* of the model. The second stage

[1] For some reason this seems to give fast convergence and, in fact, this value is used by many commercial programs as e.g. SAS or BMDP as initial guess for MA parameters in time series models.

is to estimate the parameters of the fitted model. Finding a suitable model for a texture is typically an iterative procedure. When a model is identified and estimated, an analysis of the residuals will often point to inadequacies in the chosen model. Thus the iteration continues until an appropriate model is found.

Model identification in 1-D is often carried out by considering the AutoCorrelation Function (ACF) or the Partial AutoCorrelation Function (PACF). In 2-D it is also suggested to consider the ACF and the Multiple Partial Correlation (MPC), also referred to as reflection coefficient, (See e.g. Marzetta (1980) or Kartikeyan & Sarkar (1991)). This may be a good approach when the samples are generated by a very simple underlying process. Typically, however, the underlying process is of a mixed SARMA nature and the Autocorrelation as well as the MPC will "die out" in a way which is not clearly recognizable and hence not well suited for model identification. Recently it has become popular, in e.g. time-series analysis, to use the Inverse AutoCorrelation Function (IACF) as a tool for model identification (see e.g. Chatfield (1979)). The estimation of the IACF typically involves the estimation of very high-order autoregressive models. It is obvious that such an approach often would prove useless when dealing with 2-D data as the number of parameters would grow - roughly - with the square of the model order. Apart from numerical problems, the variance of the estimates could be very high.

When we cannot formulate a model based on knowledge about the underlying phenomenon, we are often reduced to fit a "sufficiently" large model and then, successively try to reduce the dimensionality of the model. In classical regression analysis there exist many different methods which could all be referred to as "step-wise" regression. None of these methods are uniformly best, and the resulting models obtained from such procedures should be used with some care.

In this section a new backwards elimination method for SARMA models is presented. The algorithm based on the T-test statistic of the estimated parameters and the BIC of the estimated model.

One way of testing the significance of a given parameter is to consider the test statistic

$$T = \frac{\hat{\theta}_i}{\hat{\sigma}_{\theta_i}} \in t(M \cdot N - p) \tag{4.30}$$

which holds under the assumptions stated in theorem 2 (again p is the total number of parameters in the model). An estimate of $\hat{\sigma}_{\theta_i}$ is obtained through the approximation of the inverse Hessian matrix as described in section 2.4.

Alternatively we can use the test statistic

$$F = T^2 = \left(\frac{\hat{\theta}_i}{\hat{\sigma}_{\theta_i}}\right)^2 \in F(1, (M \cdot N - p)) \tag{4.31}$$

In the attempt to find formalized methods for determining the "optimal" order of the model, a number of information criteria have been devised. Probably, the best known are Akaike's Information Criterion (AIC) and the Bayes Information Criterion (BIC) given below:

$$AIC = M \cdot N \; ln(\sigma_\epsilon^2) + 2 \cdot (\underline{card} N_{AR} + \underline{card} N_{MA}) \tag{4.32}$$

$$BIC = M \cdot N \; ln(\sigma_\epsilon^2) + (\underline{card} N_{AR} + \underline{card} N_{MA}) \cdot log(M \cdot N) \tag{4.33}$$

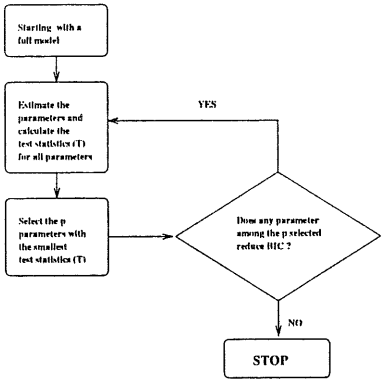

Figure 4.12. *The backwards elimination algorithm*

The AIC is not consistent, and for large values of $M \cdot N$ it may overestimate the number of parameters needed, whereas BIC is *weakly* consistent. A comparison carried out in section 4.5 clearly illustrates the drawbacks of AIC as a tool for model identification.

When the parameters of a model are estimated, we also have an estimate of the variance of the estimated parameters, according to theorem 1 or theorem 2. This is used to form a test statistic as in equation 4.30 or equation 4.31. Here the former is used.

Now the p least significant parameters are chosen according to the test statistic calculated for each estimated parameter. In order of growing significance, it is checked whether the elimination of the parameter reduces BIC. If so, the parameter is eliminated.

Parameters of the model is eliminated successively as long as the BIC of the resulting model is decreasing. The iteration stops when the BIC cannot be reduced further. A flow diagram for this algorithm is outlined in figure 4.12

Note that it is not enough just to continue selecting the least significant parameter and try to eliminate it. Even if the BIC can be reduced further, it may not be the least significant parameter that actually reduces BIC, but another parameter (usually) *among* the least significant. This is discussed below, and may be explained by correlation or collinearity between parameters.

Experimental results

In order to evaluate the merits of the backwards elimination method proposed, an experiment was carried out using synthetic data. An $NSHP(\oplus,+)$ SAR model of order $E(2)$ was used to simulate the texture shown in figure 4.13. The parameters of the model are given in figure 4.13 as well. A larger model, of order $E(5)$, is chosen as initial model for the backwards elimination procedure. The program that implements the algorithm (*prunesarma* - see appendix A) creates a logfile in which the parameters are printed out as they are eliminated. If the program attempts to eliminate a parameter and fails, this is reported as well. Below a print of the log-file is shown:

```
BIC(full model): 4.335825e+02
BIC(excluding (1,2)): 4.227682e+02
BIC(excluding (3,3)): 4.119576e+02
BIC(excluding (5,0)): 3.863926e+02
BIC(excluding (4,-1)): 3.756077e+02
BIC(excluding (3,-1)): 3.649063e+02
BIC(excluding (0,3)): 3.542166e+02
BIC(excluding (1,-4)): 3.435906e+02
BIC(excluding (1,-3)): 3.328217e+02
BIC(excluding (2,-4)): 3.222069e+02
BIC(excluding (4,-2)): 3.115970e+02
```

```
BIC(excluding (0,4)): 3.010331e+02
BIC(excluding (2,3)): 2.906688e+02
BIC(excluding (3,-3)): 2.803427e+02
BIC(excluding (2,-1)): 2.700083e+02
BIC(excluding (1,-2)): 2.596820e+02
BIC(excluding (4,0)): 2.494419e+02
BIC(excluding (3,0)): 2.389053e+02
BIC(excluding (4,3)): 2.287821e+02
BIC(excluding (4,2)): 2.181713e+02
BIC(excluding (2,2)): 2.083853e+02
BIC(excluding (3,2)): 1.980086e+02
BIC(excluding (1,4)): 1.879948e+02
BIC(excluding (4,-3)): 1.782922e+02
BIC(excluding (4,1)): 1.513117e+02
BIC(excluding (3,1)): 1.418792e+02
BIC(excluding (2,1)): 1.317907e+02
Failed to exclude (0,5), T=-1.484606e+00, BIC=1.391740e+02
BIC(excluding (1,3)): 1.249031e+02
Failed to exclude (0,5), T=-1.512721e+00, BIC=1.322505e+02
BIC(excluding (3,-2)): 1.187118e+02
Failed to exclude (0,5), T=-1.519363e+00, BIC=1.261164e+02
BIC(excluding (2,-2)): 1.121599e+02
BIC(excluding (2,-3)): 1.032278e+02
Failed to exclude (0,5), T=-1.497010e+00, BIC=1.098380e+02
BIC(excluding (3,-4)): 8.935875e+01
Failed to exclude (0,5), T=-1.714582e+00, BIC=9.643734e+01
BIC(excluding (3,4)): 6.553765e+01
BIC(excluding (2,4)): 5.555428e+01
BIC(excluding (0,5)): 3.300119e+01
Failed to exclude (1,-1), T=-2.363970e+01, BIC=5.777166e+02
```

```
Failed to exclude (0,2), T=4.118072e+01, BIC=1.753213e+03
Failed to exclude (1,1), T=5.416555e+01, BIC=2.048027e+03
Failed to exclude (2,0), T=6.876313e+01, BIC=3.208324e+03
Failed to exclude (1,0), T=-7.344821e+01, BIC=7.822854e+03
```

It should be noted that the algorithm attempts to eliminate the parameter $(0,5)$ several times before it succeeds. In fact Tjøstheim (1981) reported that the BIC criterion tends to predict models "on the edge" - i.e. models with parameters of the type $(0,p)$ or $(p,0)$. This may be explained by strong collinearity between parameters of the model. The correlation matrix for the parameters, just prior to the first attempt to eliminate $(0,5)$, is visualized in figure 4.14. The dark fields correspond to large negative correlations and, in this case, the correlation between the parameters $(0,5)$ and $(0,1)$ (corresponding to entry $(1,3)$ in the correlation image) is -0.7986. This strong correlation may, of course, be due to severe over parametrization but one could also suspect a phenomenon such as this, to origin from the sampling geometry.

As it is seen from the program output, other parameters with larger test statistics, (not lying on "the edge") reduces the BIC and these are eliminated first. Thus finally the parameter $(0,5)$ can be eliminated and, in fact, the correct model is obtained. This example shows that it is very important not to stop the elimination procedure when the least significant parameter cannot be eliminated.

Further discussions of the merits of the backwards elimination algorithm is postponed to section 4.5.

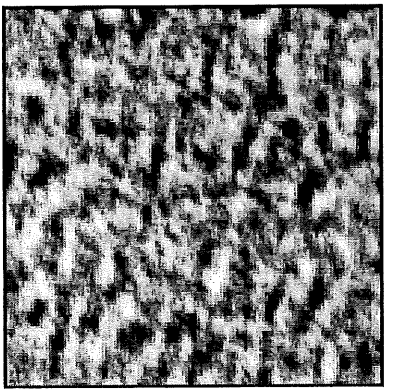

Figure 4.13. *The texture is simulated using a* $NSHP(\oplus,+)$ *SAR model of order* $E(2)$. *The parameters are as follows:* $AR(0,1) = -0.7818$, $AR(0,2) = 0.1914$, $AR(1,-1) = -0.1084$, $AR(1,0) = -0.8338$, $AR(1,1) = 0.3155$ *and* $AR(2,0) = 0.2876$. *The driving noise is i.i.d. Gaussian of zero mean and unit variance.*

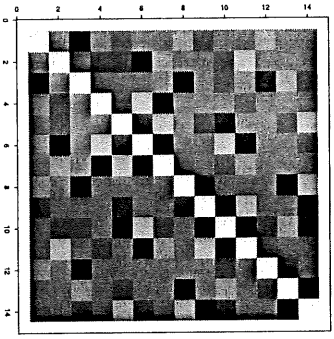

Figure 4.14. *The figure serves to visualize the correlation matrix of the SAR-parameters just prior to the first attempt to eliminate parameter no. 3 (corresponding to parameter* $(0,5)$*) The dark fields correspond to large negative correlations. The correlation between parameter no.3 (i.e.* $(0,5)$*) and parameter no. 1 (i.e.* $(0,1)$*) is -0.7986.*

4.4 The image material

A group of 15 Brodatz textures, from Brodatz (1966), were selected on the basis that they should be fine grained and stochastic. This means that we are not considering deterministic textures like e.g. brick walls here. Semi-deterministic textures like e.g. textiles will be considered in chapter 6.

The textures were scanned from the paper with an 8-bit, 300 dpi scanner. The output from the scanner is a 2400×1800 image, which is then reduced by two steps in a Gaussian pyramid (Burt (1981)). The approximately Gaussian operator is a separable, symmetric filter with values

$$0.05, 0.25, 0.40, 0.25, 0.05$$

The result is a 600×450 floating point image, where almost no pixels have identical values. To make sure that all textures have identical first-order properties a Gaussian histogram match is performed [1]. This is done by sorting all pixels while the image is in floating point format. Thereby a perfect histogram match is obtained (Carstensen (1992)). Prior to the histogram match, the textures are corrected for background variations by subtracting a 25×25 median filtered version of each texture from itself.

According to Carstensen (1992), the performance of textural features based on a Gaussian match of the texture is superior to features based on histogram equalization, for stochastic textures.

The names of the selected Brodatz textures are given in figure 4.15 and the Gaussian matched textures are shown in figure 4.16

[1] In fact a 100% correct classification of all Brodatz textures can be obtained using the first-order statistics *alone*, according to Carstensen (1992)

Fieldstone (D2)	Pressed cork (D4)	Grass lawn (D9)
Bark of tree (D12)	Calf leather (D24)	Beach sand (D29)
Water (D38)	Handmade paper (D57)	Woodgrain (D68)
Woodgrain (D69)	Woodgrain (D70)	Woodgrain (D71)
Pigskin (D92)	Fur (D93)	Ice crystals (D100)

Figure 4.15. *Names of the 15 Brodatz textures shown in figure 4.16*

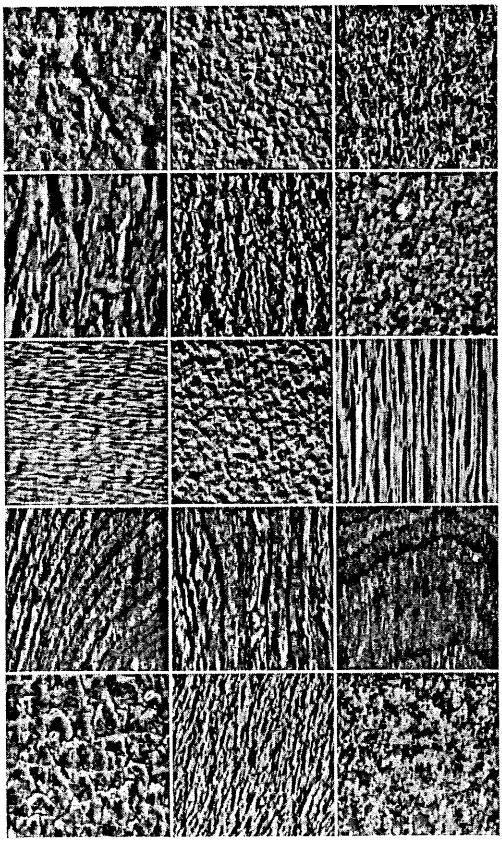

Figure 4.16. *The 15 chosen Brodatz textures after Gaussian match.*

4.5 Modelling textures with SAR models

In this section we consider QP and NSHP SAR models for the textures described in section 4.4. We evaluate AIC and BIC as criteria for model selection, and subsequently use the backwards elimination method, presented in section 4.3.3, to reduce the order of the initially selected model. It is demonstrated that the performance of the NSHP, in the BIC sense, is superior the the QP models.

4.5.1 Selecting an initial SAR model

The SAR models chosen initially were $QP(+,+)$ and $NSHP(\oplus,+)$ models of orders:

$$E(1), E(1.5), E(2), \cdots, E(8.5), E(9)$$

The parameters of each model are estimated, using the LS estimator, for each texture. Having estimated the parameters and the residual variance, the AIC and BIC is calculated for each model. In figure 4.17 - 4.20 we have plotted $log(AIC)$ and $log(BIC)$ against the number of NSHP parameters for each texture. The curves marked with "o"-legend are $log(AIC)$ and "+"-legend are $log(BIC)$. As it can be seen the AIC generally has its local minimum (if any) for larger models than the BIC, which is in perfect agreement with common experience (see e.g. Tjøstheim (1981)).

The order of the model that minimizes BIC is given for each texture in figure 4.21 - figure 4.24. The number of parameters and the corresponding BIC is given as well, in the top row of each entry.

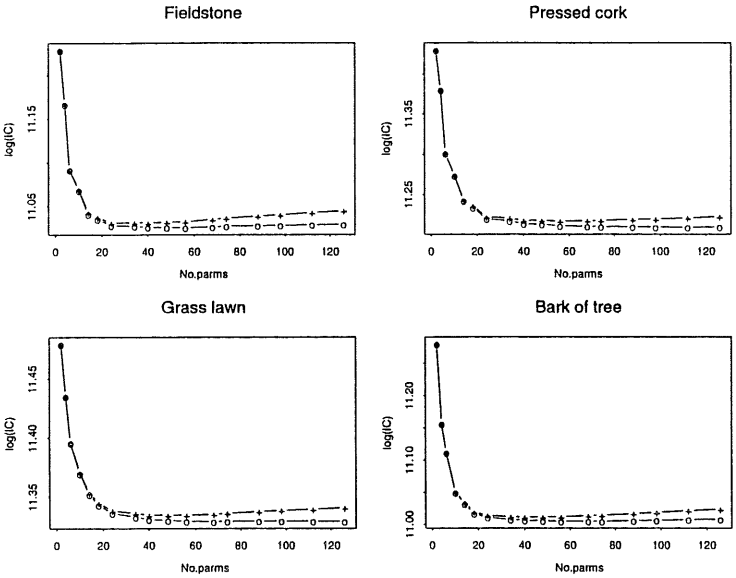

Figure 4.17. *log(AIC) and log(BIC), for the NSHP models, plotted against the number of parameters, marked with legend "o" and "+" respectively*

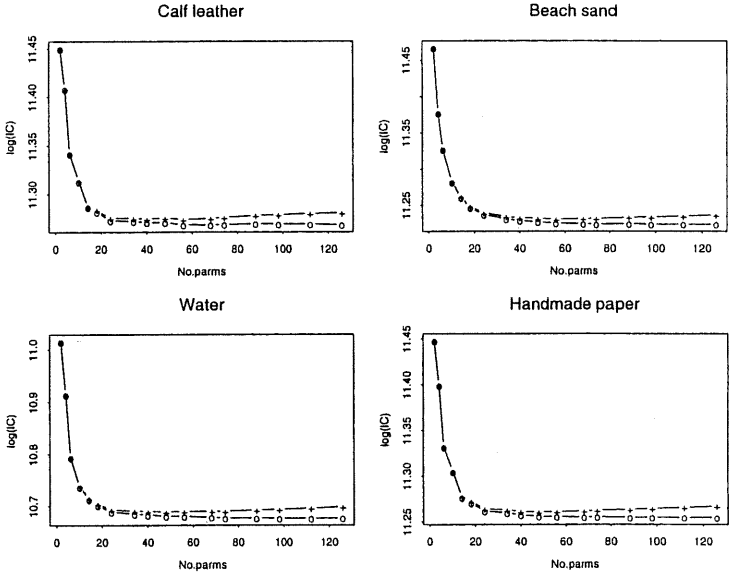

Figure 4.18. *log(AIC) and log(BIC), for the NSHP models, plotted against the number of parameters, marked with legend "o" and "+" respectively*

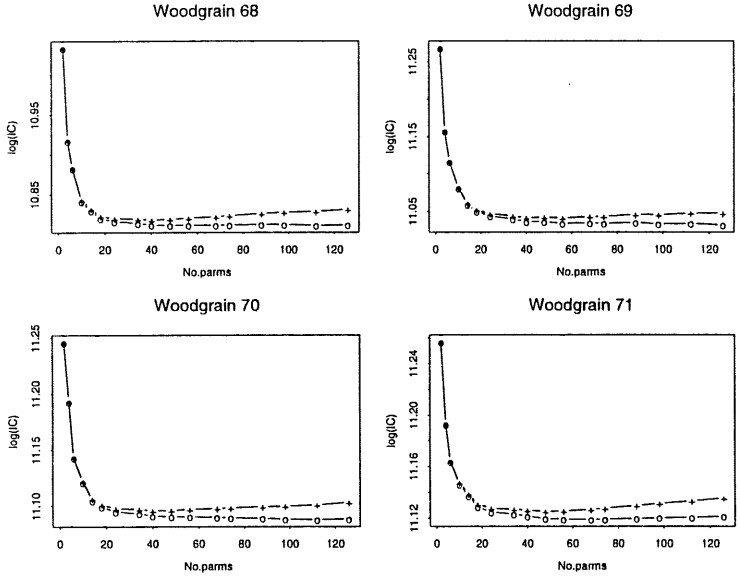

Figure 4.19. *log(AIC) and log(BIC), for the NSHP models, plotted against the number of parameters, marked with legend "o" and "+" respectively*

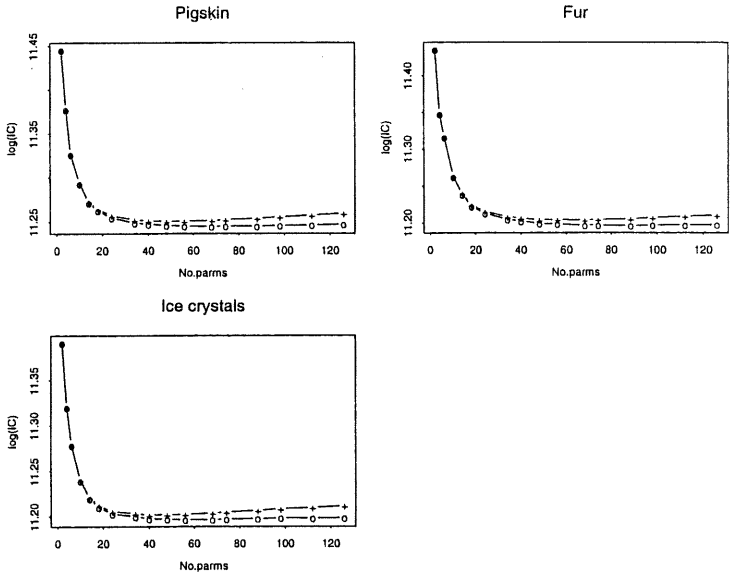

Figure 4.20. *log(AIC) and log(BIC), for the NSHP models, plotted against the number of parameters, marked with legend "o" and "+" respectively*

4.5.2 Reducing the order of a model

In section 4.3.3 we presented a backwards elimination algorithm and demonstrated the performance for synthetic data. In this section the "optimal" model (in the BIC sense), for each of the 15 Brodatz textures described above, is found using the backwards elimination algorithm.

In figure 4.21 - figure 4.24 the number of parameters of the optimal model, following the backwards elimination procedure, is given for the NSHP and QP models respectively in the lower row of each entry. The initial model used is the one pointed out with the BIC as described above in section 4.5.1. For the NSHP models the backwards elimination procedure reduced the initially chosen models by about 37% on average. The QP models were reduced by about 22.5% on average.

For the selected set of textures the performance of the NSHP models are uniformly better than the QP models in the BIC sense. Even if the initial QP model was better than the initial NSHP model, the optimal NSHP model was better than the corresponding optimal QP model. This agrees well with what one would have expected. The NSHP models are richer in the sense that it is possible to model the correlation structure simultaneously in directions that cannot be modelled (simultaneously) by any QP model.

4.5.3 Evaluation by simulation

For each texture, the estimated (BIC) optimal model is simulated. To facilitate the assessment of the visual properties of the models compared to the original texture, the simulated texture is embedded in the original. The simulated texture is placed in the lower half of the image as shown in 4.25, and the lower

Texture	Optimal model	Number of parameters	ln(BIC)
Fieldstone	E(4)	24	11.0310
(D2)		19	11.0290
Pressed Cork	E(6)	56	11.2164
(D4)		35	11.2148
Grass lawn	E(6)	56	11.3344
(D9)		32	11.3319
Bark of tree	E(5)	40	11.0116
(D12)		27	11.0089
Calf leather	E(6)	56	11.2740
(D24)		31	11.2707
Beach sand	E(6)	56	11.2310
(D29)		33	11.2278
Water	E(5.5)	48	10.6891
(D38)		27	10.6804
Handmade paper	E(5.5)	48	11.2609
(D57)		29	11.2587

Figure 4.21. *The table summarizes results obtained for the* $NSHP(\oplus, +)$ *considered. The second column state the radius of the the initial model that minimizes the BIC (i.e. before the subsequent optimization). The third column contain the corresponding number of parameters in the initial model in the top row of each entry. The number of parameters is given in the bottom row of each entry in the third column. The fourth column is giving the corresponding ln(BIC).*

Texture	Optimal model	Number of parameters	ln(BIC)
Woodgrain	E(5)	40	10.8181
(D68)		23	10.8149
Woodgrain	E(6)	56	11.0411
(D69)		32	11.0365
Woodgrain	E(5)	40	11.0959
(D70)		27	11.0939
Woodgrain	E(5.5)	48	11.1253
(D71)		30	11.1228
Pigskin	E(5.5)	48	11.2508
(D92)		32	11.2490
Fur	E(6.5)	68	11.2039
(D93)		47	11.2011
Ice crystals	E(5)	40	11.2017
(D100)		25	11.1999

Figure 4.22. *The table summarizes results obtained for the $NSHP(\oplus,+)$ considered. The second column state the radius of the the initial model that minimizes the BIC (i.e. before the subsequent optimization). The third column contain the corresponding number of parameters in the initial model in the top row of each entry. The number of parameters is given in the bottom row of each entry in the third column. The fourth column is giving the corresponding ln(BIC).*

Texture	Optimal model	Number of parameters	ln(BIC)
Fieldstone	E(5.5)	29	11.0331
(D2)		22	11.0321
Pressed Cork	E(6)	34	11.2202
(D4)		25	11.2192
Grass lawn	E(6)	34	11.3336
(D9)		27	11.3330
Bark of tree	E(5)	34	11.0414
(D12)		23	11.0399
Calf leather	E(6)	34	11.2743
(D24)		23	11.2732
Beach sand	E(6)	34	11.2424
(D29)		28	11.2419
Water	E(5.5)	29	10.7025
(D38)		23	10.7014
Handmade paper	E(5)	25	11.2716
(D57)		20	11.2709

Figure 4.23. *The table summarizes results obtained for the $QP(+,+)$ considered. The second column state the radius of the the initial model that minimizes the BIC (i.e. before the subsequent optimization). The third column contain the corresponding number of parameters in the initial model in the top row of each entry. The number of parameters is given in the bottom row of each entry in the third column. The fourth column is giving the corresponding ln(BIC).*

Texture	Optimal model	Number of parameters	ln(BIC)
Woodgrain	E(5)	25	10.8191
(D68)		20	10.8184
Woodgrain	E(6)	34	11.0602
(D69)		26	11.0593
Woodgrain	E(5)	25	11.0963
(D70)		21	11.0960
Woodgrain	E(6)	34	11.1286
(D71)		24	11.1276
Pigskin	E(6)	34	11.2588
(D92)		31	11.2586
Fur	E(6.5)	40	11.2502
(D93)		33	11.2492
Ice crystals	E(5,5)	29	11.2163
(D100)		21	11.2155

Figure 4.24. *The table summarizes results obtained for the $QP(+,+)$ considered. The second column state the radius of the the initial model that minimizes the BIC (i.e. before the subsequent optimization). The third column contain the corresponding number of parameters in the initial model in the top row of each entry. The number of parameters is given in the bottom row of each entry in the third column. The fourth column is giving the corresponding ln(BIC).*

boundary pixel values of the original image are used as initial values for the simulation, so as to make the transition from original- to simulated texture continuous. The simulation results are shown in figure 4.26-figure 4.40.

The models obtained for Fieldstone, Pressed cork, Water, Handmade paper, Woodgrain (D68) and Ice crystals are visually quite satisfactory.

The models obtained for Grass lawn, Bark of tree, Calf leather, Beach sand Woodgrain (D70) and Fur are acceptable.

For the remaining textures: Woodgrain (D69 and D71) and Pig skin , we cannot say that the obtained models are adequate.

Some of the results can, as we shall see, be improved by using mixed SARMA models. In some cases the improvement in visual quality is significant. The point here is, however, that SARMA models can only model second order properties of the textures and some textures are not adequately described in terms of their second order properties. Therefore it is not surprising that e.g. Woodgrain (D69 or D70) are poorly modelled by autocovariances. Although the inadequacies of the model partly can be attributed the large scale, it is obvious that second-order interactions cannot model such 'curved' structures.

An interesting point that becomes appreciable, when simulating realizations from the estimated texture models, is the fact that practically all of the models lie very close to non-stationarity. The non-stationarity is due to very strong *positive* correlations in the image. Indeed this seems to be a salient feature of all textures that we would normally consider interesting. Several of the textures considered here could only be simulated because the 'real' texture was used as initial values (Of course any fixed initial values would do. The important thing is that they are fixed). Had this not been the case, the output

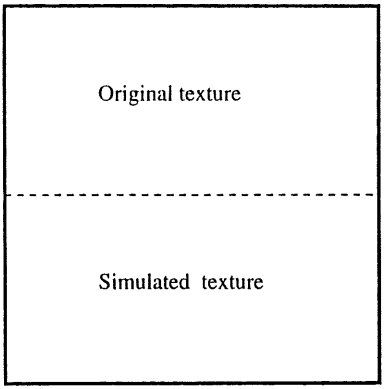

Figure 4.25. *The simulated textures are embedded in the lower half part of the original texture.*

of the model would have grown without bounds. This point is discussed further in section 4.6.

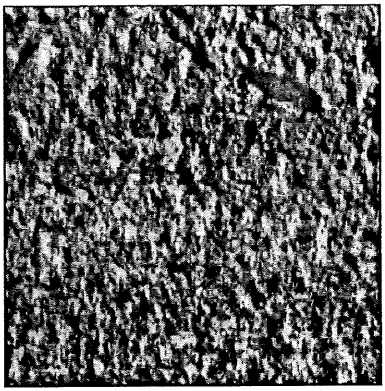

Figure 4.26. *The lower half of the image is simulated Fieldstone (D002) texture.*

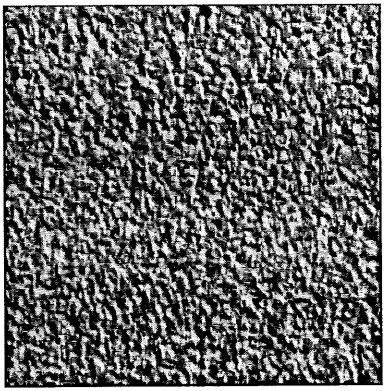

Figure 4.27. *The lower half of the image is simulated Pressed cork (D004) texture.*

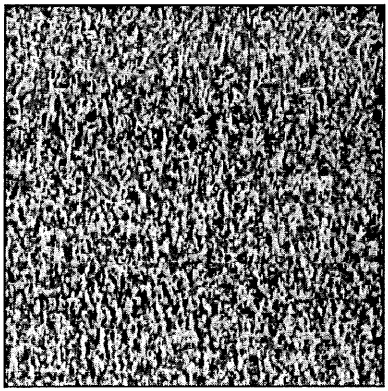

Figure 4.28. *The lower half of the image is simulated Grass lawn (D009) texture.*

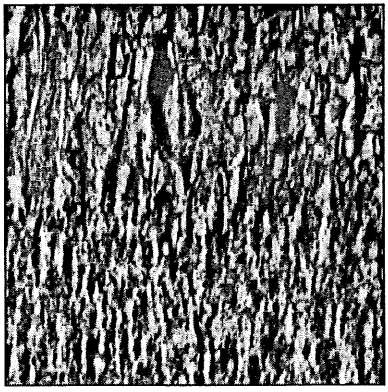

Figure 4.29. *The lower half of the image is simulated Bark of tree (D012) texture.*

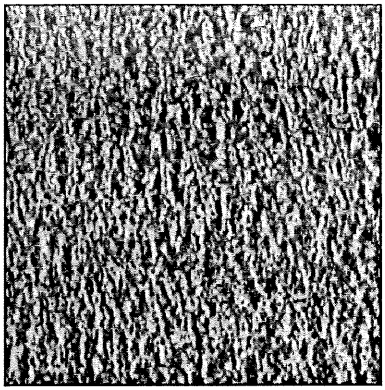

Figure 4.30. *The lower half of the image is simulated Calf leather (D024) texture.*

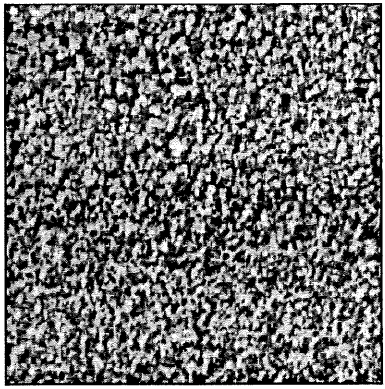

Figure 4.31. *The lower half of the image is simulated Beach sand (D029) texture.*

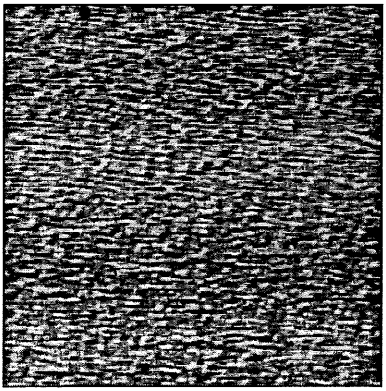

Figure 4.32. *The lower half of the image is simulated water (D038) texture.*

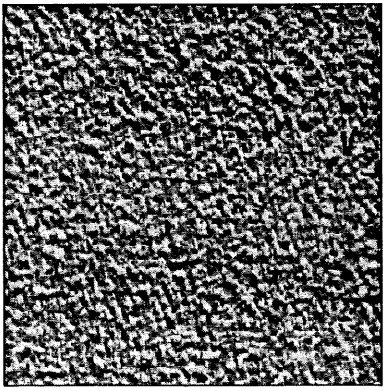

Figure 4.33. *The lower half of the image is simulated Handmade paper (D057) texture.*

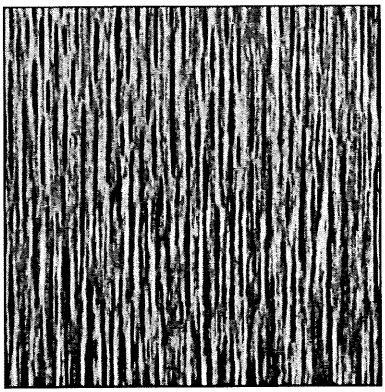

Figure 4.34. *The lower half of the image is simulated Woodgrain (D068) texture.*

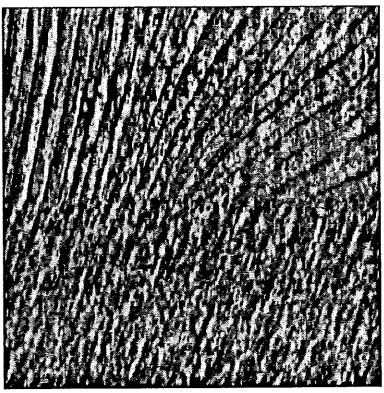

Figure 4.35. *The lower half of the image is simulated Woodgrain (D069) texture.*

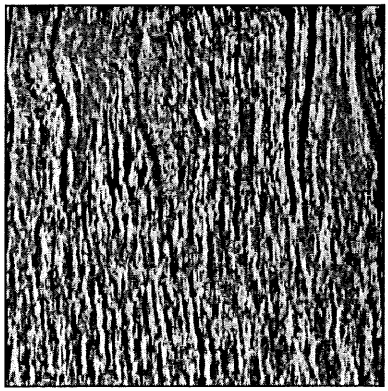

Figure 4.36. *The lower half of the image is simulated Woodgrain (D070) texture.*

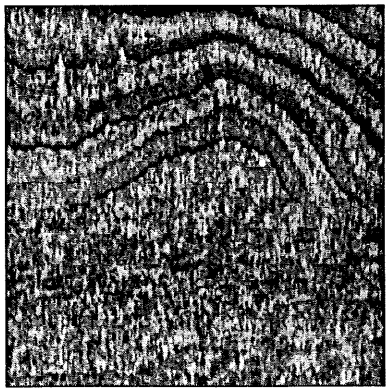

Figure 4.37. *The lower half of the image is simulated Woodgrain (D071) texture.*

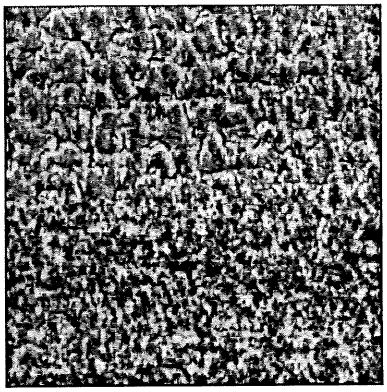

Figure 4.38. *The lower half of the image is simulated Pig skin (D092) texture.*

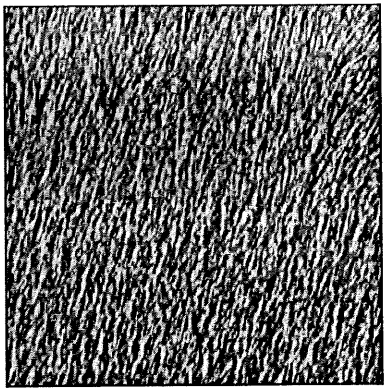

Figure 4.39. *The lower half of the image is simulated Fur (D093) texture.*

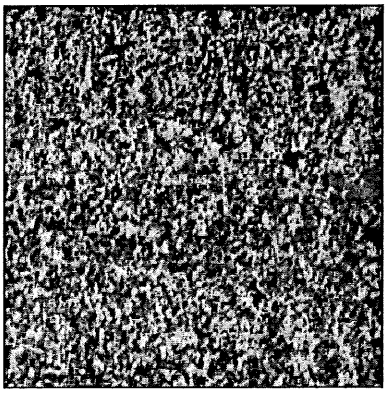

Figure 4.40. *The lower half of the image is simulated Ice crystals (D100) texture.*

4.6 Modelling textures with the SARMA model

In this section the textures considered in the previous sections are modelled using mixed SARMA models. Fitting mixed SARMA models to a given texture is a very demanding task from a computational point of view. As the number of parameters typically needed to fit the texture is large, the nonlinear optimization becomes a very time consuming part of the modelling. It is demonstrated, however, that a simple approach may be taken that effectively reduces the computational effort. It is also demonstrated that the effort is rewarded by parsimonious models with such attractive properties as low BIC and improved visual agreement with the modelled texture (when compared to pure SAR models).

4.6.1 Selecting an initial SARMA model

In section 4.5.1 an initial model was found by searching among SAR models of increasing order and select the model for which the BIC was minimized. A similar search among SARMA models of different order is not feasible as the number of possibilities are much larger. Furthermore, the computation required to estimate each of the models makes this approach even less attractive.

The work, however, can be reduced significantly by initially fitting an optimal pure SAR model as outlined in the previous sections. The model thus found is typically extended by only a very few SMA parameters. This approach is taken here, and the optimal NSHP-SAR models found in section 4.5.2 are all extended by a NSHP-SMA part, in this case of order $R(1)$. These models are estimated and the BIC thus found are shown in figure 4.41, where they are compared to the corresponding optimal SAR model. It is seen that the BIC even decreases in several cases, although the model has more parameters. In

section 4.7 it is shown that the inclusion of SMA parameters may allow for the elimination of several SAR parameters. This in turn means a further reduction of BIC.

The visual impression of the simulated SARMA textures is sometimes a great improvememnt over the results presented in figure 4.26 - figure 4.40 (an example is given in section 4.7). Having said this, it should also be remembered that many of the fitted SAR models described in section 4.5.3 were not stationary, and a valid realization of the texture could only be obtained due to the fact that the initial values were fixed during the simulation. In the case of mixed SARMA models, the inclusion of SMA-parameters tends to stabilize the model. This interesting effect is illustrated in the figure 4.42, where the SARMA models fitted to 'pressed cork' and to 'bark of tree' are simulated using the relaxation method described in section 4.2. The maximum (absolute) deviation between two consecutive iterations, denoted e_{max}, is plotted against the iteration number. It is seen that e_{max} of the unstable system, constituted by the pure SAR model, grows to infinity. The SARMA model has exactly the same autoregressive order, but the system is stabilized by the presence of the moving average part.

Even when the pure SAR model is stable, the mixed SARMA model is seen to have a faster convergence in figure 4.43.

Modelled Texure	ln(BIC) of SAR	ln(BIC) of SARMA
Fieldstone (D2)	11.0290	11.0247
Pressed cork (D4)	11.2148	11.2140
Grass lawn (D9)	11.3319	11.3319
Bark of tree (D12)	11.0089	11.0071
Calf leather (D24)	11.2707	11.2707
Beach sand (D29)	11.2278	11.2271
Water (D38)	10.6804	10.6855
Handmade paper (D57)	11.2587	11.2589
Woodgrain (D68)	10.8149	10.8154
Woodgrain (D69)	11.0365	11.0369
Woodgrain (D70)	11.0939	11.0937
Woodgrain (D71)	11.1228	11.1231
Pigskin (D92)	11.2490	11.2484
Fur (D93)	11.2011	11.2007
Ice crystals (D100)	11.1999	11.1994

Figure 4.41. *The table summarizes the results obtained for mixed SARMA models. The autoregressive order is chosen in accordance with the optimal pure SAR order. This model is extended with an NSHP-SMA part of order* $R(1)$. *The* $ln(BIC)$ *of the estimated models are presented in the rightmost column. The* $ln(BIC)$ *of the pure SAR model is presented in the middle column.*

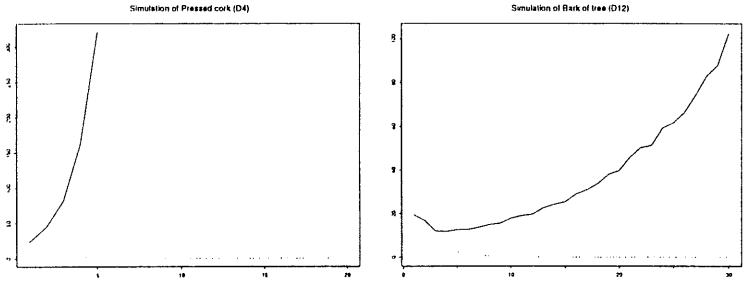

Figure 4.42. *The maximum deviation between two iterations in the simulation, denoted e_{max}, is plotted against the iteration number. It is seen that the pure SAR models are unstable. The models are stabilized using only three SMA parameters.*

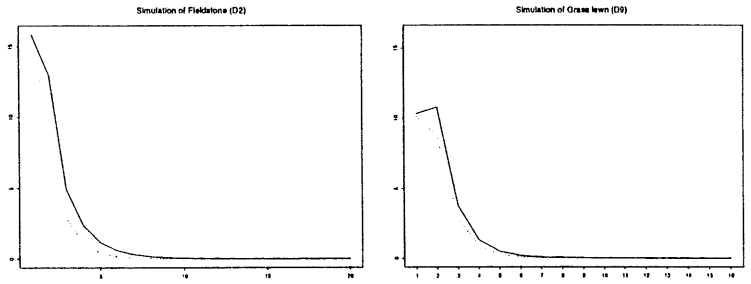

Figure 4.43. *The maximum deviation between two iterations in the simulation, denoted e_{max}, is plotted against the iteration number. It is seen that the convergence of the pure SAR models is slower than the convergence of the mixed SARMA models.*

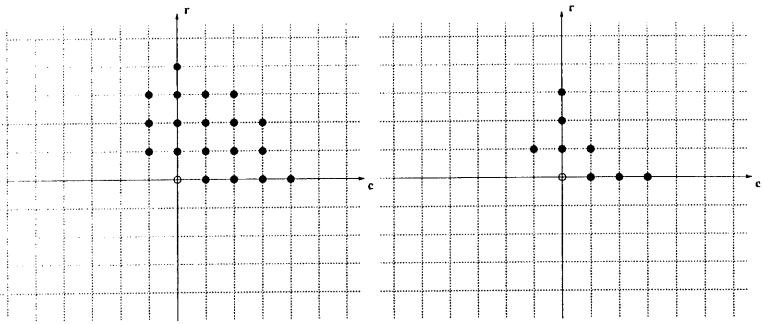

Figure 4.44. ***Left:*** *Shows the autoregressive parameters in the optimal SAR model. This autoregressive order is used in the initial SARMA model and* ***Right:*** *shows the autoregressive parameters in the BIC optimal SARMA model. Note that 11 autoregressive parameters have been eliminated.*

4.7 Reducing the order of the model

When a SAR model is extended with SMA parameters, it will typically be possible to reduce the order of autoregression further.

The optimal SAR model found in section 4.5.2 for the *fieldstone* texture is shown in figure 4.44. In the previous section this model were extended with an NSHP-SMA part of order $R(1)$ and the BIC were reduced from $6.16344E{+}04$ to $6.13727E+04$. Using the backwards elimination procedure 11 autoregressive parameters and one moving average parameter ($MA(1,-1)$) are eliminated (see figure 4.44) and BIC is reduced further to $6.12530E+04$. The texture is now represented by only 11 parameters and no visual quality is lost. In fact, the visual quality of this model is significantly better than the visual quality of the pure SAR model which contain almost twice as many parameters. This is shown in figure 4.46.

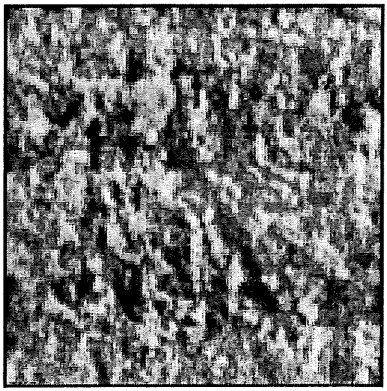

Figure 4.45. *The image shown is the original fieldstone texture (D002) from the Brodatz album.*

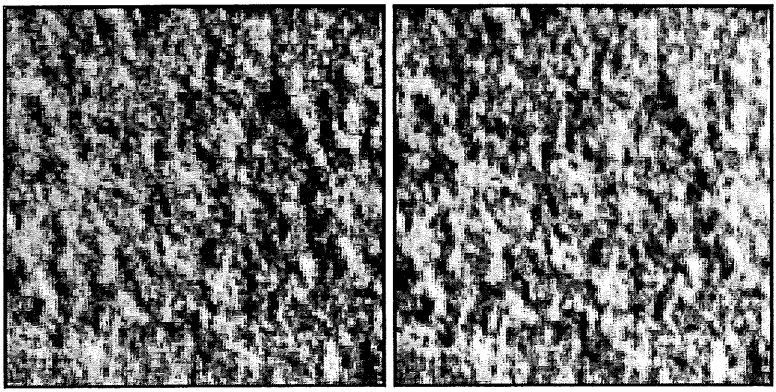

Figure 4.46. *The* ***Left*** *texture is simulated using the initial SAR model containing 19 parameters. The* ***Right*** *texture is simulated using the optimal mixed SARMA model containing 11 parameters. The same noise sequence were used in the simulation. The simulated texture should be compared to the original texture in figure 4.45*

4.8 Conclusion

The results of this chapter are summarized in the following conclusions:

- Many visual textures can be modelled using SARMA models and a very good visual agreement between the modelled texture and simulated textures may be obtained. A good (visual) agreement will, however, only be found if the texture is adequately described in terms of it's first- and second-order properties.
- Most visual textures seems to lie on "the edge of non-stationarity".
- An non-stationary SAR model can often be stabilized by extending it with a few SMA parameters.
- Mixed SARMA models are, in general, more parsimonious than SAR models and the BIC for the corresponding SARMA model is lower.
- For small images of a given texture the estimates may be improved using the proposed back forecasting scheme.
- The visual quality of textures generated using mixed SARMA models are just as good or, in some cases, better than textures generated by pure SAR models.
- Using criteria as BIC and simulation (the latter has not been presented here), the NSHP models perform uniformly better than the QP models.
- The Backwards elimination method presented here is well suited for automatic identification of appropriate SARMA models for a given texture.

Software has been developed (in compliance with the *HIPS* standard for specification, identification (backwards elimination), estimation (CLS and ULS) and simulation (relaxation and frequency domain) of SARMA models.

Chapter 5

Noncausal SAR models

In chapter 4 causal (or unilateral) SARMA models were considered for texture modelling. When considering spatial data, causality is not in general a natural constraint, and there has been a concomitant interest in investigating the properties of bilateral models. Abandoning causality, however, introduces a number of difficulties. In particular the identification and estimation of models becomes more complicated as Least Squares is not a consistent estimator for bilateral models. Due to the complexity of the likelihood function, the focus has mainly been on pure SAR models, which are also the subject of this chapter.

Noncausal (or bilateral) [1] Simultaneous AutoRegressions, sometimes referred to as NCSAR, were originally considered by Whittle (1954). The inconsistency of the LS estimates was addressed in this paper, and a large sample approach based on spectral methods was presented. First this method will only be of use for very simple models (Kashyap & Chellappa (1983)) and secondly many practical situations involve small samples for which the discounting of

[1] The terms noncausal and bilateral are used interchangeably

edge effects may not be negligible (Ord (1975)). Kashyap & Chellappa (1983) consider the likelihood function for a toroidal lattice bilateral SAR model. The torus assumption leads to a greatly simplified likelihood function and an estimator based on a Taylor-series expansion of the likelihood function was presented in this paper. It was claimed that this approximative estimator in general yields estimates close to the Maximum Likelihood estimates.

In this chapter it is established that the approximative estimator suggested by Kashyap & Chellappa (1983) does not in general give satisfactory estimates. The properties of the estimator deteriorate significantly when the model is close to non-stationarity where, as discussed in chapter 4, most visual textures are found. It is shown that phase transition phenomena occur when on the border of non-stationarity. The exact Maximum Likelihood estimator for toroidal lattice SAR models is considered and experimental results show that the estimates obtained in this way are quite satisfactory. Finally a new approximative estimator is suggested.

5.1 The SAR model

There are two non-equivalent ways (Besag (1974)) of specifying the structure of the underlying interaction among the given observations. This leads to the specification of conditional and simultaneous models respectively. The conditional models often referred to as Conditional Models (CM) or Gaussian Markov Random Field (GMRF) models are considered by e.g. Besag (1974), Besag & Moran (1975), Besag (1977), Chellappa (1985) and many others. Considering the simultaneous models, there are Simultaneous AutoRegression (SAR), Simultaneous Moving Average (SMA) and the Simultaneous AutoRegressive Moving Average (SARMA) models. The class of SAR models is a subset of the conditional models in the sense that for each SAR model there is

with identical spectral density. The converse is not always true (Kashyap & Chellappa (1983)). In spite of this, SAR models have received considerable attention for several reasons. As already mentioned the class of SAR models can be extended to include SMA parameters, and SARMA models are not a subset of CM models. It may also be argued that SAR models are in general more parsimonious than CM models (see e.g. Kashyap & Chellappa (1983), provided that both models describe the second-order properties of the observed texture adequately with a finite set of parameters and that the texture is adequately described by its second order properties. Therefore it is of interest to study the simultaneous models in their own right.

Consider a wide sense stationary, zero mean bilateral Simultaneous Autoregression

$$y_s + \sum_{r \in N} \phi_r y_{r-s} = \epsilon_s \tag{5.1}$$

where $r = (r_1, r_2) \in N$ and N is the neighbor set, $s = (s_1, s_2) \in \Omega$ where Ω is an $M \times M$ image, ϵ_s is i.i.d. $N(0, \sigma_\epsilon^2)$.

Contrary to the Gaussian Markov Random Fields (GRMF), the bilateral model does not have to be symmetrical, but symmetrical opposite neighbors must be associated with identical parameters i.e. $\phi_r = \phi_{-r}$. This is necessary to ensure the identifiability of the parameters (Besag (1974)).

If one considers the bilateral model, the stationarity condition is greatly weakened; all that is required is that no poles of the transfer function fall on the unit circle (Whittle (1954)):

$$\{1 + \sum_{(m,n)\in N} \phi_{m,n} z_r^{-m} z_s^{-n} \neq 0\} \tag{5.2}$$

for all $|z_r| = 1$ and $|z_s| = 1$

In order to simplify the notation, the set of neighbors is divided into two disjoint sets, N^*, which is completely symmetrical and N_{as} which is completely asymmetric set N_{as}. Hence $N = N^* \cup N_{as}$. If an element r belongs to N^*, then $-r$ also belongs to N^*. If an element $r \in N_{as}$, then $-r \notin N$. The symmetrical set N^* is written in terms of a new set N_s such that $N^* = \{r : r \in N_s \; or \; -r \in N_s\}$. N_s is (arbitrarily) chosen as

$$N_s = NSHP(\oplus, +) \cap N^* \tag{5.3}$$

Assume an $n = M \times M$ image vector $\underline{y}$ which is lexicographically ordered row by row. Given the model (5.1), the noise variates are obtained as a transformation of the observations:

$$\underline{\epsilon} = B(\underline{\theta}) \cdot \underline{y} \Leftrightarrow \underline{y} = B(\underline{\theta})^{-1} \cdot \underline{\epsilon} \tag{5.4}$$

where $\underline{\theta}$ denotes the vector of parameters

$$\underline{\theta} = \underline{col}\,[\phi_r, \quad \text{if } r \in N_s \cup N_{as}] \tag{5.5}$$

Theorem 3 *A bilateral SAR model driven by i.i.d. Gaussian noise of variance σ^2 and zero mean (white noise) has the simultaneous density function*

$$f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}^n} \underline{det} B(\underline{\theta}) exp(-\frac{1}{2\sigma^2} \underline{\epsilon}^T \cdot \underline{\epsilon}) \tag{5.6}$$

and the logarithmic density function is

$$ln f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = -\frac{n}{2} ln(2\pi\sigma^2) + ln(\underline{det} B(\underline{\theta})) - \frac{1}{2\sigma^2} \underline{\epsilon}^T \cdot \underline{\epsilon} \tag{5.7}$$

Proof:

Assuming that the underlying innovation process is Gaussian, $\underline{\epsilon} \in N(\underline{0}, \sigma^2 \cdot I)$, the simultaneous density of the observed image vector is also Gaussian:

$$\underline{y} \in N(\underline{0}, \sigma^2 \cdot B(\theta)^{-1}(B(\theta)^{-1})^T) \tag{5.8}$$

and $f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2)$ is the density function of this multivariate Gaussian distribution.

5.1.1 Second order properties

When SAR models are used for texture description, only the second order properties (i.e. autocovariance function or spectral density) are modelled. Given a specific model, the corresponding spectral density is defined in a simple way in terms of the parameters. The spatial frequencies are defined on an $M \times M$ grid such that $f(s) \in \left[-\frac{1}{2}; \frac{1}{2}\right[\times \left[-\frac{1}{2}; \frac{1}{2}\right[$ for each gridpoint $s = (r, c) \in [0; M-1] \times [0; M-1]$. Thus $f(s) = \left(\frac{r}{M} - \frac{1}{2}, \frac{c}{M} - \frac{1}{2}\right)^T$, and for a given spatial frequency the spectral density is defined as

$$S_y(f(s)) = \frac{\sigma_\epsilon^2}{1 + 2\underline{\theta}^T \underline{C}_{f(s)} + \underline{\theta}^T Q_{f(s)} \underline{\theta}} \tag{5.9}$$

where

$$\begin{aligned} \underline{C}_{f(s)} &= \underline{col} \begin{bmatrix} 2\cos(2\pi f(s)^T r) & \text{if } r \in N_s \\ \cos(2\pi f(s)^T r) & \text{if } r \in N_{as} \end{bmatrix} \\ \underline{S}_{f(s)} &= \underline{col} \begin{bmatrix} 0 & \text{if } r \in N_s \\ \sin(2\pi f(s)^T r) & \text{if } r \in N_{as} \end{bmatrix} \end{aligned} \tag{5.10}$$

and $Q_{f(s)} = \underline{C}_{f(s)} \underline{C}_{f(s)}^T + \underline{S}_{f(s)} \underline{S}_{f(s)}^T$

The autocovariance function is obtained as the inverse Fourier transform of the spectral density.

5.2 Simulation experiments

The simulation methods described section 4.2 are also applicable in the bilateral case. In this section, simulation by relaxation is applied. The purpose of this experiment is to illustrate how different choices of parameters, for the same order of model, influence the visual impression of the generated texture. Different realizations of the textures obtained in this way are used in subsequent sections to evaluate the performance of different estimators.

A simple first order model (Nearest Neighbor model) is considered

$$y_{r,c} + \alpha (y_{r+1,c} + y_{r-1,c}) + \beta (y_{r,c+1} + y_{r,c-1}) = \epsilon_{r,c} \tag{5.11}$$

It can be shown that this model is stationary provided $\mid \alpha \mid + \mid \beta \mid < \frac{1}{2}$. The line of proof is to consider the frequency response function of this model. The admissible parameter space is obtained as the restrictions imposed on the parameters by the requirement that no poles are allowed in the frequency response function.

Textures are simulated along the line $\alpha = \beta$, where the textures are isotropic. The parameters are moved from $(\alpha, \beta) = (0, 0)$ (in this case the output is simply the noise generated by the random generator) towards in-stationarity in both directions. Then the textures are simulated along the line $\alpha = -\beta$, where the textures are anisotropic, and again the parameters are moved towards non-stationarity. The isotropy and anisotropy lines are shown in figure 5.1 The same noise sequence is used to generate realizations of all textures, in order to make direct comparison easier.

In figure 5.2 the parameters α and β, used to generate the texture in the upper leftmost image, are both negative. As the parameters decrease towards zero along the isotropy line, and then become positive, the textures change from coarse to fine. When the parameters, being positive, approach non-stationarity the texture has, as shown in figure 5.4, a striking resemblance to the checkerboard texture obtained when simulating binary Markov Random Fields at the supercritical limit as described by e.g. Carstensen (1988) or Carstensen (1992).

Letting the parameters move along the anisotropy line, the textures shown in figure 5.3 are generated. Again there is a resemblance to the deterministic patterns obtained when simulating in the Ising model at the supercritical limit. Two textures simulated in the proximity of non-stationarity are shown (enlarged) in figure 5.5 and figure 5.6.

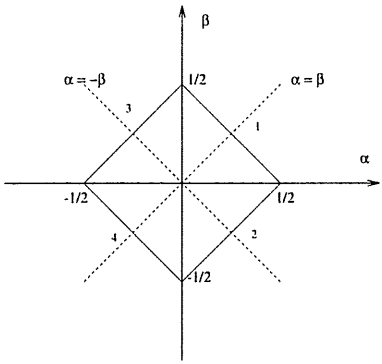

Figure 5.1. *The figure is illustrating the isotropy line $\alpha = \beta$ and the anisotropy line $\alpha = -\beta$ along which first order bilateral SAR models are simulated.*

When the parameters approach non-stationarity the convergence of the iterative simulation algorithm, using relaxation, is very slow. Close to non-stationarity the gain of the system increases towards infinity and thus the output of the system grows witout bounds (to plus or minus infinity). This phenomenon causes the texture to look almost binary. A further study of phase transitions in continuous models will not be pursued here, but is certainly a topic for further research.

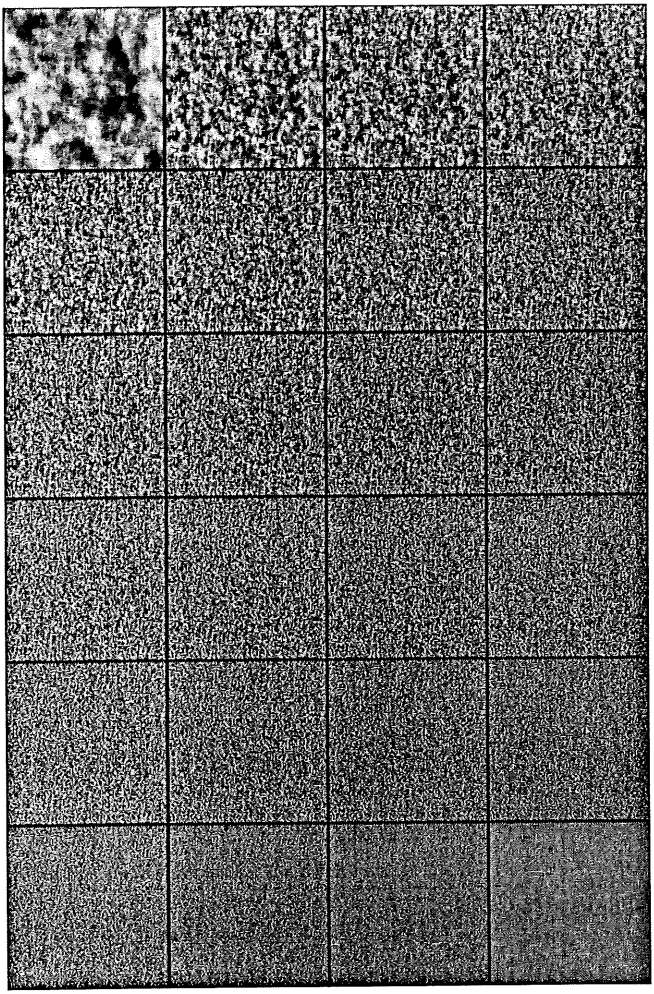

Figure 5.2. *The figure is showing textures simulated along the isotropy line $\alpha = \beta$ using a first order bilateral SAR model. As the parameters go from negative to positive the visual quality of the textures goes from coarse to fine.*

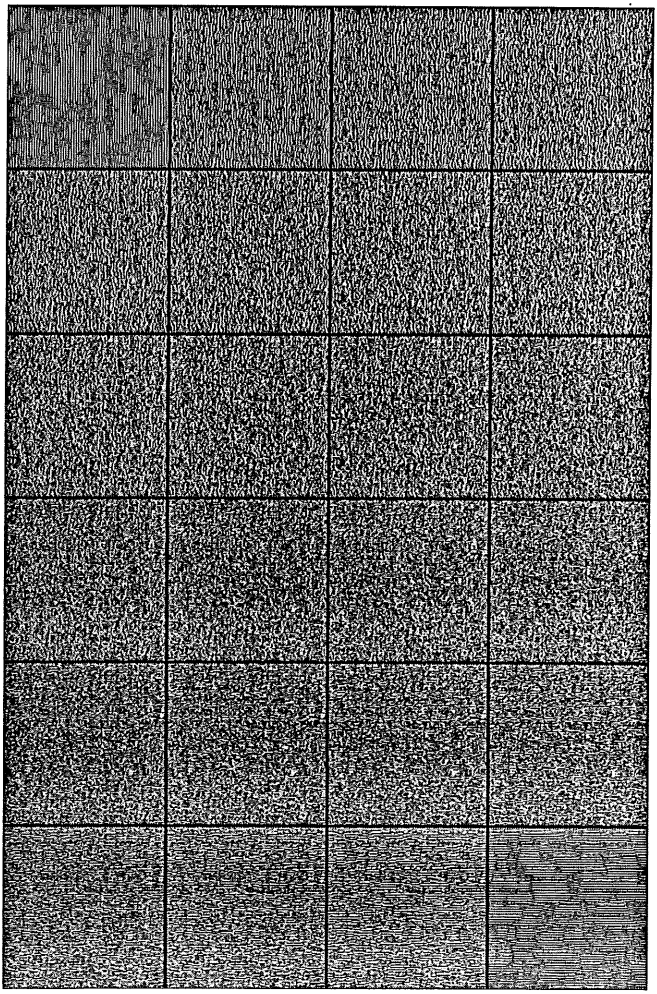

Figure 5.3. *The figure shows textures simulated along the anisotropy line* $\alpha = -\beta$ *using a first order bilateral SAR model.*

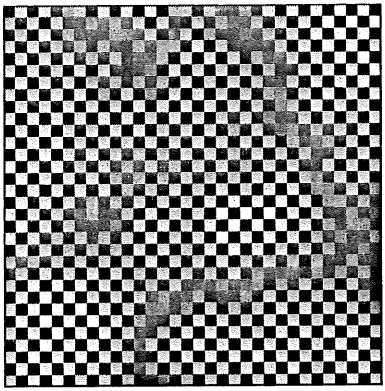

Figure 5.4. *The figure shows a small part of an isotropic texture simulated with positive parameters very close to non-stationarity. The checkerboard effect is very pronounced.*

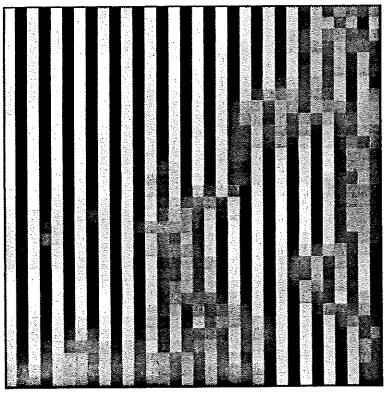

Figure 5.5. *The figure shows a small part of an anisotropic texture simulated with positive parameters very close to non-stationarity. The vertical striping effect is very pronounced.*

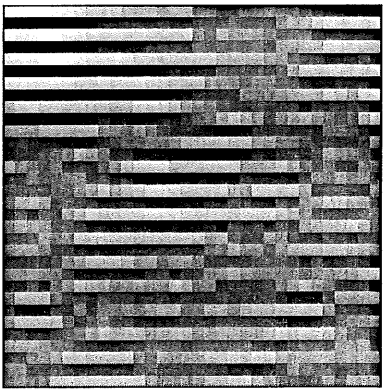

Figure 5.6. *The figure shows a small part of an anisotropic texture simulated with positive parameters very close to non-stationarity. The horizontal striping effect is very pronounced.*

5.3 Least Squares Estimation

As pointed out by Whittle (1954), the Least Squares estimator is not consistent.

Whereas the LS estimator can be used for causal models, it is a non-consistent estimator for non-causal models. This is can be seen by considering the likelihood function

$$f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}^n} det B(\underline{\theta}) exp(-\frac{1}{2\sigma^2}\underline{\epsilon}^T \cdot \underline{\epsilon}) \tag{5.12}$$

where $B(\underline{\theta})$ is the matrix that transforms the observations into noise variates. In unilateral models, the determinant of this matrix is unity and thus the likelihood is maximized by simply minimizing the sum of squared prediction errors (or residuals). For bilateral models, however, the determinant of the matrix is not unity, but on the contrary strongly dependent upon the parameters. Therefore the LS estimator will lead to nonsensical results. A more detailed exposition of the inconsistency may be found in Whittle (1954) or Ord (1975). The intention of remaining part of this section is to show how the inconsistency of the LS estimator affects the obtained estimates.

The Least Squares (LS) estimator is obtained as

$$\hat{\underline{\theta}} = -\left(\sum_{s\in\Omega_I} \underline{z}(s)\underline{z}(s)^T\right)^{-1} \left(\sum_{s\in\Omega_I} y(s)\underline{z}(s)\right) \tag{5.13}$$

where

$$\underline{z}(s) = \underline{col} \begin{bmatrix} y(s+r) & \text{if } r \in N_{as} \\ y(s+r) + y(s-r) & \text{if } r \in N_s \end{bmatrix} \tag{5.14}$$

This is easily verified as the LS estimator is minimizing the sum of squared residuals given as:

$$\underline{\epsilon}^T \underline{\epsilon} = \sum_{s \in \Omega_I} \left(y(s) + \underline{\theta}^T \underline{z}(s) \right)^2 \tag{5.15}$$

By setting the partial derivatives to zero, and solving the equations w.r.t. the parameters the above estimator is found.

Note that instead of summing over Ω, we sum over the *inner* points Ω_I - i.e. the site $s \in \Omega_I$ if all the sites in $\underline{z}(s)$ are contained in Ω.

5.3.1 Experimental results

This experiment serves to show in what way the inconsistency of the LS estimator affects the estimates. In section 5.2, textures obeying a first order bilateral SAR model were simulated along the isotropy and anisotropy lines. In the experiment conducted here, 100 realizations of each texture were simulated and for each simulated texture the parameters were estimated. The mean and standard deviation of the parameters α and β were calculated and used to obtain the (empirical) 95% -confidence limits of the mean for each set of "true" (i.e. simulation) parameters. Plots of the experiment are shown in figure 5.7 and figure 5.8. The vertical lines along the "s"-shaped curve are the empirical 95% -confidence limits obtained from 100 estimates of the parameters. Ideally the estimates should have been grouped around the dashed line. As the figures clearly show this is not the case for the LS estimates which are

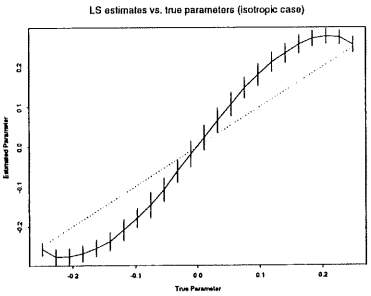

Figure 5.7. *The figure shows the systematic bias of the LS estimates for textures simulated along the isotropy line. The vertical lines along the curve are outlining the empirical 95% confidence intervals for the mean of the estimated parameters.*

obviously extremely biased. In fact, the bias seems to be deterministic. The investigation into the possibly deterministic structure of the bias, using LS estimates, has not been pursued further in this thesis.

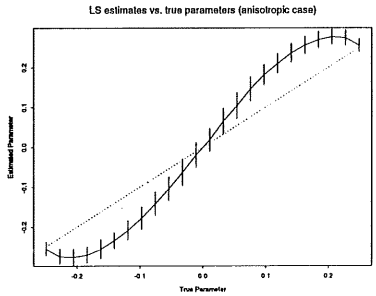

Figure 5.8. *The figure shows the systematic bias of the LS estimates for textures simulated along the anisotropy line. The vertical lines along the curve are empirical 95% confidence intervals for the mean of the estimated parameters.*

5.4 Estimation in toroidal models

As demonstrated in section 5.3.1, the LS estimator will, in general, give useless results for bilateral models. Therefore, one is forced to consider the (log-)likelihood function

$$ln f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = -\frac{n}{2} ln(2\pi\sigma^2) + ln(\underline{det} B(\underline{\theta})) - \frac{1}{2\sigma^2} \underline{\epsilon}^T \cdot \underline{\epsilon} \tag{5.16}$$

In general, for an $M \times M$ image the dimensions of the $B(\underline{\theta})$ matrix will be $M^2 \times M^2$ which, even for moderatly sized images, will be a prohibitive size for optimization of the likelihood function on any existing computer.

If, instead, a toroidal lattice SAR model is considered, the matrix $B(\underline{\theta})$ becomes block circulant with circulant blocks. The toroidal assumption is merely an assumption made for dealing with boundary effects as illustrated in figure 2.2. This type of model is considered by e.g. Kashyap & Chellappa (1983), where it is shown that an analytical expression exists for the determinant of a matrix with such a circulant structure.

Theorem 4 *For a toroidal lattice SAR model, the determinant of the transformation matrix $B(\underline{\theta})$, has a simple analytical form:*

$$ln\ \underline{det} B(\underline{\theta}) = \sum_{s \in \Omega} ln(1 + \underline{\theta}^T \underline{\psi}_s) \tag{5.17}$$

where $\underline{\psi}_s$ is defined as:

$$\underline{\psi}_s = \underline{C}_s + i \cdot \underline{S}_s \tag{5.18}$$

and the vectors $\underline{C}_s$ *and* $\underline{S}_s$ *are defined as:*

$$\underline{C}_s = \underline{col} \begin{bmatrix} 2cos(\frac{2\pi}{M} s^T r) & if\, r \in N_s \\ cos(\frac{2\pi}{M} s^T r) & if\, r \in N_{as} \end{bmatrix}$$
$$\underline{S}_s = \underline{col} \begin{bmatrix} 0 & if\, r \in N_s \\ sin(\frac{2\pi}{M} s^T r) & if\, r \in N_{as} \end{bmatrix} \tag{5.19}$$

Proof: See e.g. Kashyap (1980).

Note that $(1+\theta^T \underline{\psi}_s)$ is generally complex valued. Only when strictly symmetrical models are considered this expression is real valued.

This result can be used to obtain a computationally more tractable expression for the (log-)likelihood function

Theorem 5 *Assuming a toroidal lattice SAR model where the driving noise is i.i.d. Gaussian, the (log-)likelihood function can be written as*

$$ln f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = \frac{1}{2} \sum_{s \in \Omega} ln \left(1 + 2\underline{\theta}^T \underline{C}_s + \underline{\theta}^T Q_s \underline{\theta}\right) - \frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \underline{\epsilon}^T \cdot \underline{\epsilon} \tag{5.20}$$

where $Q_s = \underline{C}_s \underline{C}_s^T + \underline{S}_s \underline{S}_s^T$

Proof: By using theorem 4 we obtain

$$ln(1 + \underline{\theta}^T \cdot \underline{\psi}_s) = ln(1 + \underline{\theta}^T \cdot (\underline{C}_s + i \cdot \underline{S}_s))$$

$$= \frac{1}{2} ln(1 + 2\underline{\theta}^T \underline{C}_s + \underline{\theta}^T (\underline{C}_s \underline{C}_s^T + \underline{S}_s \underline{S}_s^T)\underline{\theta})$$
$$+ i \cdot tg^{-1} \left(\frac{\underline{\theta}^T \underline{S}_s}{1 + \underline{\theta}^T \underline{C}_s} \right) \quad (5.21)$$

Observing the fact that the determinant of a real matrix is real, we get that

$$\sum_{s \in \Omega} ln(1 + \underline{\theta}^T \underline{\psi}_s) = \frac{1}{2} \sum_{s \in \Omega} ln \left(1 + 2\underline{\theta}^T \underline{C}_s + \underline{\theta}^T Q_s \underline{\theta} \right) \quad (5.22)$$

This expression is inserted into equation 5.7 and we obtain the expression in equation 5.20.

It should be noted that in this form, each term in the sum that constitutes the determinant is similar to the form of the denominator in the parametric spectrum, presented in equation 5.9.

5.4.1 An Iterative Estimation Scheme

Although the likelihood function given in theorem 5 is a significant simplification, it is still a computationally expensive, and by no means simple task, to obtain the (torus) Maximum Likelihood estimates through a non-linear optimization of the (log-)likelihood function. This problem has motivated a search for more tractable solutions. Kashyap & Chellappa (1983) suggested a Taylor series expansion $ln(1 + x) \approx x - \frac{1}{2}x^2$ of the first term in (5.20). This leads to the formulation of an iterative estimator which is linear in each step. The result is presented in:

Theorem 6 *The estimates $\overline{\underline{\theta}}$ and $\overline{\sigma}^2$ maximizing $J(\underline{\theta}, \sigma^2)$ are obtained through iteration as limits of $\underline{\theta}_t$, σ_t^2 defined by*

$$\underline{\theta}_t = \left(R - \frac{1}{\sigma_t^2}S\right)^{-1}\left(\underline{V} - \frac{1}{\sigma_t^2}\underline{U}\right), \quad t = 0, 1, \cdots \tag{5.23}$$

and

$$\sigma_t^2 = \frac{1}{M^2}\sum_{\Omega}\left(y(s) - \underline{\theta}_t^T \underline{z}(s)\right)^2 \tag{5.24}$$

where

$$\begin{aligned} \underline{V} &= \sum_{s\in\Omega} \underline{C}_s \\ R &= \sum_{s\in\Omega} (\underline{S}_s \underline{S}_s^T - \underline{C}_s \underline{C}_s^T) \\ S &= \sum_{\Omega} \underline{z}(s)\underline{z}^T(s) \\ \underline{U} &= \sum_{\Omega} \underline{z}(s) y(s) \end{aligned} \tag{5.25}$$

and

$$J(\underline{\theta}, \sigma^2) = \underline{V}^T\underline{\theta} + \frac{1}{2}\underline{\theta}^T R \underline{\theta} - \frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\underline{\epsilon}^T \cdot \underline{\epsilon} \tag{5.26}$$

is a modified *likelihood function.*

The initial estimates are obtained as the LS estimates.

Proof:

As the derivation of this result is of importance in the subsequent, the proof is presented here in some detail. It may also be found (less detailed) in Kashyap & Chellappa (1983)

$$ln(1 + 2\underline{\theta C_s} + \underline{\theta}^T Q_s \underline{\theta}) \approx 2\underline{\theta}^T \underline{C_s} + \underline{\theta}^T \underline{C_s} \underline{C_s}^T \underline{\theta} + \underline{\theta}^T \underline{S_s} \underline{S_s}^T \underline{\theta} - \frac{1}{2}\left(2\underline{\theta}^T \underline{C_s} + \underline{\theta}^T \underline{C_s} \underline{C_s}^T \underline{\theta} + \underline{\theta}^T \underline{S_s} \underline{S_s}^T \underline{\theta}\right)^2 \quad (5.27)$$

Expanding on this, and only taking first and second order terms of $\underline{\theta}$ (thus skipping third- and fourth order terms of $\underline{\theta}$), we obtain

$$ln(1 - 2\underline{\theta C_s} + \underline{\theta}^T Q_s \underline{\theta}) \approx 2\underline{\theta}^T \underline{C_s} + \underline{\theta}^T \underline{C_s} \underline{C_s}^T \underline{\theta} + \underline{\theta}^T \underline{S_s} \underline{S_s}^T \underline{\theta} \quad (5.28)$$

The corresponding approximation of (5.20) denoted as $J(\underline{\theta}, \sigma^2)$ can be written as

$$J(\underline{\theta}, \sigma^2) = \underline{V}^T \underline{\theta} + \frac{1}{2}\underline{\theta}^T R \underline{\theta} - \frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \underline{\epsilon}^T \cdot \underline{\epsilon} \quad (5.29)$$

where

$$\underline{V} = \sum_{s \in \Omega} \underline{C_s}$$
$$R = \sum_{s \in \Omega} (\underline{S_s} \underline{S_s}^T - \underline{C_s} \underline{C_s}^T)$$

The estimator is obtained by setting the derivatives (w.r.t. the parameters) of this expression to zero and solve the equation w.r.t. the parameters.

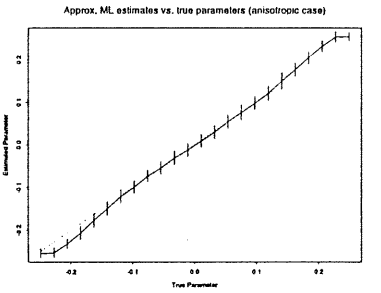

Figure 5.9. *The figure shows the systematic bias of the approximative ML-estimates for textures simulated along the anisotropy line. The vertical lines along the curve outline the empirical 95% confidence intervals for the mean of the estimated parameters.*

Experimental results

To assess the quality of the approximative ML estimator suggested by Kashyap & Chellappa (1983), the experiment described in section 5.3.1 were carried out using the approximative estimator. As shown in figure 5.9, this estimator is indeed an improvement over the LS estimator, but when the parameters lie in the proximity of non-stationarity the estimates are strongly biased. The reasons for this shortcoming of the estimator is obvious. As the parameters approach non-stationarity, the first-order Taylor-expansion of the determinant becomes a bad approximation.

5.4.2 ML estimates for toroidal SAR models

Textures considered in real applications are, as we have seen, often close to non-stationarity. As shown in the previous section, the properties of the approximative estimator suggested by Kashyap & Chellappa (1983), deteriorate in the proximity of non-stationarity. Hence we are forced to consider the exact ML estimates for toroidal SAR lattice models through non-linear optimization of the (log-)likelihood function described in theorem 5. (or, in fact, the negative (log-)likelihood, as standard optimization procedures typically minimize the objective functional).

The (log-)likelihood, considered below, is a complicated non-linear function of the parameters $\underline{\theta}$ and σ^2.

$$ln\ f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = \frac{1}{2}\sum_{s\in\Omega} ln\left(1 + 2\underline{\theta}^T\underline{C}_s + \underline{\theta}^T Q_s \underline{\theta}\right) - \frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\underline{\epsilon}^T \cdot \underline{\epsilon} \quad (5.30)$$

The optimization of non-linear functions will, in general, require knowledge of the gradient of the objective functional (in this case the (log-)likelihood function) w.r.t. the parameters involved. In many practical situations, such a gradient can only be provided through numerical approximations as e.g. Forward Difference Approximation or Central Difference Approximation. As the dimension of the problem is increased, the number of function evaluations in each step is increased (at least) with the number of extra parameters. In cases where the optimization problem is very sensitive to the accuracy of the gradient estimate, a Central Difference Approximation is typically required. In this case the objective functional must be evaluated twice for each component of the gradient. As the evaluation of the objective functional is a computationally demanding task, the evaluation of the analytical gradient, given below, is preferred.

$$\nabla_{\underline{\theta}} ln\ f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = \sum_{s \in \Omega} \frac{\underline{C}_s + Q_s \underline{\theta}}{1 + 2\underline{\theta}^T \underline{C}_s + \underline{\theta}^T Q_s \underline{\theta}} - (\underline{U} + S\underline{\theta}) / \sigma^2 \tag{5.31}$$

The evaluation of the (log-)likelihood function is, in each step of the iteration, dependent upon the noise variance σ^2, which is not known. This problem is circumvented by estimating the noise variance in each step prior to the calculation of a new search direction

$$\hat{\sigma}_t^2 = \sum_{s \in \Omega_I} \left(y(s) + \hat{\underline{\theta}}_t^T \underline{z}(s) \right)^2 \tag{5.32}$$

where $t = 0, 1, \cdots$.

The noise variance σ^2 should not be included in the gradient expression and optimized along with the parameters of the model, as this may indeed lead to serious (numerical) scaling problems.

The non-linear optimization is carried out using a Quasi-Newton method using a factored update. The algorithm used is implemented in the subroutine library *IMSL*. For further details the reader is referred to section 2.4 and, for a more comprehensive treatment of the topic, Dennis & Schnabel (1983).

Experimental results

This section will illustrate some aspects that must be considered when attempting to obtain valid estimates by applying non-linear optimization techniques to the (log-)likelihood function.

Consider, again, the simple first-order bilateral SAR model

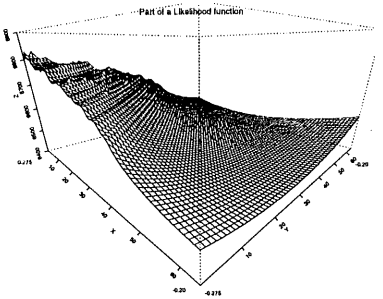

Figure 5.10. *A part of the likelihood function for a first-order bilateral SAR model given a realization of the process with the parameters* $\alpha = \beta = -0.22735$. *The likelihood is plotted in the interval* $(\alpha, \beta) \in [-0.275, -0.20] \times [-0.275, -0.20]$. *Note that the likelihood function is extremely rippled in the inadmissible part of the parameter space.*

$$y_{r,c} + \alpha\left(y_{r+1,c} + y_{r-1,c}\right) + \beta\left(y_{r,c+1} + y_{r,c-1}\right) = \epsilon_{(r,c)} \tag{5.33}$$

When attempting to obtain the ML estimates through the use of a general optimization method initial values of the parameters must be provided for the optimizer. Intuitively one may think that the LS-estimates could provide such initial values. Using these will, however, often yield nonsensical results. As shown in figure 5.7 (or figure 5.8), the choice of the LS estimates will give a non-stationary initial model. Figure 5.10 shows the likelihood function in a part of the non-stationary region where the likelihood function is extremely rippled. Clearly, the optimizer is likely to be "caught" in one of these "valleys" thus yielding a sub-optimal solution. This demonstrates that the ML-estimation is extremely dependent upon a prudent choice of initial estimates.

The approximate ML estimates turn out to be a much better initial values for the parameters. They are sufficiently close to the "true" parameters to give a fast convergence when optimizing the exact likelihood.

The experiment outlined in section 5.3.1 is carried out using the torus-ML estimator with the approximative ML estimates as initial values. The results are presented in figure 5.11. It it seen that the estimated parameters, generally, are in good agreement with the "true" parameters used to simulate the textures. Although the results are a significant improvement over the approximative ML estimates, it is clear that the empirical variances of the estimates are excessively high for the parameters $\alpha = \beta = -0.22735$ which are close to non-stationarity. Also, the estimates are slightly biased. It turns out that this, in fact, is due to numerical aspects. The optimizer is sometimes "caught" in a local minimum in the inadmissible part of the parameter space (cf. figure 5.10) and thus the global optimum is not found. This is seen by considering the histogram of the estimated parameters shown in figure 5.12. Most of the estimates are clustered around the "true" parameter. The rest (21% of the estimates) lies in the inadmissible part of the parameter space. If the initial values are chosen differently in these cases, the optimizer *in each case* find the global optimum. A check for stationarity is built into the program (this can be done for this simple model). If the optimizer finds a nonadmissible solution, the optimization is repeated with initial values all equal to zero. The histogram of the estimated parameters are shown in figure 5.12. Here, all the estimates are seen to be clustered around the "true" parameter. The experiment is summarized in figure 5.13 and the all the estimated parameters are seen to be in good agreement with the "true" parameters. Furthermore, the variance of the parameters is reduced.

This demonstrates the fact that optimization of strongly non-linear functions is never easy, and special attention must always be paid to the specific optimization problem at hand.

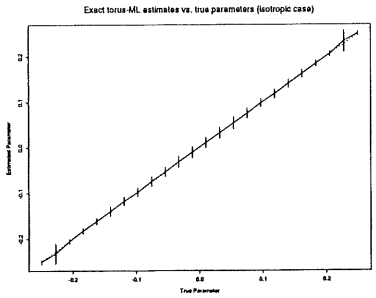

Figure 5.11. *The figure shows that reasonably good estimates may be obtained using the torus-ML estimator with the approximate ML estimates as initial values. The vertical lines along the curve outline the empirical 95% confidence intervals for the mean of the estimated parameters. Although the true parameter is contained in all the confidence intervals the variance is excessively high close to instationarity, and the estimates are slightly biased.*

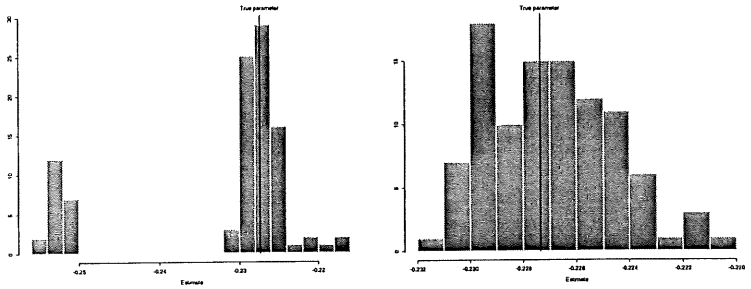

Figure 5.12. *The histogram to the* ***Left*** *shows the estimates when the approximative ML estimates are used as initial values. It is seen that 21 (out of 100) estimates are inadmissible. If alternative initial values are provided in such cases, and the optimization is restarted with these, the global optimum is found as shown in the* ***Right*** *histogram.*

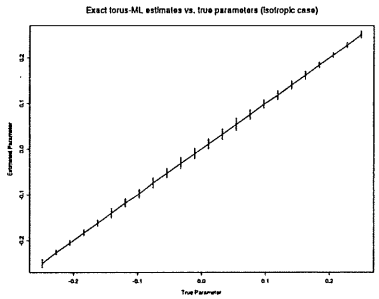

Figure 5.13. *The figure shows that good estimates may be obtained using the torus-ML estimator with the approximate ML estimates as initial values combined with a stationarity check. If an non-stationary model is obtained the optimizer is restarted with new initial values. The vertical lines along the curve are outline the empirical 95% confidence intervals for the mean of the estimated parameters.*

5.5 Other results

In this section, a new spatial spatial-frequency based estimator is presented. The investigation of the properties of this estimator is part of ongoing research but the results, using this approach does indeed give promising results.

In section 5.1.1 the spectral density of a given (bilateral) SAR model was considered.

For a given spatial frequency, $f(s) = \frac{2\pi}{M} \cdot s$ where $s = (r, c) \in [0; M-1] \times [0; M-1]$ the spectral density was defined as

$$S_y(f(s)) = \frac{\sigma_\epsilon^2}{1 + 2\underline{\theta}^T \underline{C}_{f(s)} + \underline{\theta}^T Q_{f(s)} \underline{\theta}} \tag{5.34}$$

where

$$\begin{aligned} \underline{C}_{f(s)} &= \underline{col} \begin{bmatrix} 2cos(2\pi f(s)^T r) & \text{if } r \in N_s \\ cos(2\pi f(s)^T r) & \text{if } r \in N_{as} \end{bmatrix} \\ \underline{S}_{f(s)} &= \underline{col} \begin{bmatrix} 0 & \text{if } r \in N_s \\ sin(2\pi f(s)^T r) & \text{if } r \in N_{as} \end{bmatrix} \end{aligned} \tag{5.35}$$

and $Q_s = \underline{C}_s \underline{C}_s^T + \underline{S}_s \underline{S}_s^T$

In section 5.4.2 the optimization of the exact (toroidal) likelihood function was considered, employing the analytical gradient of the likelihood function

$$\nabla_{\underline{\theta}} ln\ f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = \sum_{s \in \Omega} \frac{\underline{C}_s + Q_s \underline{\theta}}{1 + 2\underline{\theta}^T \underline{C}_s + \underline{\theta}^T Q_s \underline{\theta}} - (\underline{U} + S\underline{\theta}) / \sigma^2 \tag{5.36}$$

Again,

$$S = \sum_{s \in \Omega} \underline{z}(s)\underline{z}(s)^T \tag{5.37}$$

$$\underline{U} = \sum_{s \in \Omega} y(s)\underline{z}(s) \tag{5.38}$$

are estimates of the spatial autocovariance structure.

It is noted that the denominator part of the terms constituting the derivative of the determinant of $B(\underline{\theta})$ is identical to the denominator of the parametric spectrum.

It may be argued, that if an appropriate model is chosen, the parametric spectrum is a good estimate of the true spectrum of the observed process. Although SAR models would generally be fitted in the spatial domain, they could alternatively be fitted using a frequency domain approach. As noted by Priestly (1981), the estimation of an ARMA model (for time-series), using this approach, may be viewed as just another way of "smoothing" the periodogram. Thus, whereas the window technique smoothes the periodogram by "local averaging", the SAR method smoothes it "globally" by fitting a rational function over the complete frequency range.

By fitting the model in the spatial and spatial-frequency domain simultaneously, an iterative but in each step linear estimator may be obtained. Assume an appropriate non-parametric spectral estimate, $P(f(s))$, is obtained (by e.g. using a direct Fourier approach). Noting that the theoretical (or parametric) spectrum of the assumed model and the denominator part of the determinant are identical, the non-parametric spectral estimate $P(f(s))$ may, in a given grid-point, be viewed as an approximation of the corresponding denominator term. Thus we obtain:

$$\underline{\nabla}_{\underline{\theta}} ln\ f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) \approx \sum_{s \in \Omega} P^*(f(s))\,(Q_s\theta + \underline{C}_s) - \frac{1}{\sigma^2}\,(S\theta + \underline{U}) \tag{5.39}$$

where $P^*(f(s)) = P(f(s))/\sigma_y^2$ is the normalized spectral density.

By defining

$$W = \sum_{s \in \Omega} P^*(f(s)) Q_s \tag{5.40}$$

$$\underline{v} = \sum_{s \in \Omega} P^*(f(s)) \underline{C}_s \tag{5.41}$$

and solving the equation $\underline{\nabla}_{\underline{\theta}} ln\ f_{\underline{y}}(\underline{y} \mid \underline{\theta}, \sigma^2) = 0$ w.r.t. the parameters, we obtain the iterative estimator

$$\left(W - \frac{1}{\sigma_t^2} S\right) \underline{\theta}_t = -\left(\underline{v} - \frac{1}{\sigma_t^2} \underline{u}\right) \tag{5.42}$$

$$\sigma_t^2 = \frac{1}{M^2} \sum_{s \in \Omega} \left(y(s) + \underline{\theta}^T \underline{z}(s)\right)^2 \tag{5.43}$$

$$t = 0, 1, \cdots \tag{5.44}$$

The iteration is continued until a given stop criterion is met (e.g. $|| \underline{\theta}_t - \underline{\theta}_{t-1} ||_2 < \epsilon$).

Although there is an apparent similarity between the estimator presented here and the estimator presented by Kashyap & Chellappa (1983), these estimators are very different in nature.

5.6 Conclusion

Experiments were carried out using the inconsistent LS-estimator. It was shown that the inconsistency of this estimator lead to seriously biased estimates.

Also the approximative ML estimator suggested by Kashyap & Chellappa (1983) were considered, and it was shown that this estimator performs well for moderate values of the parameters far away from non-stationarity (i.e. very fine-grained textures).

Unfortunately most interesting textures lie in the proximity of nonstationarity and, as demonstrated in this chapter, the quality of the approximative ML-estimator deteriorates significantly as the textures considered approaches non-stationarity.

As a consequence of this the exact likelihood function for the toroidal lattice SAR model were considered. This estimator will, in general, provide good estimates. Optimization of non-linear functions is, however, not an easy task.

An alternative estimator based on a spatial spatial-frequency domain is presented. This estimator is still the topic of on-going research. Preliminary results suggest that this may, in fact, be a good solution to the estimation problems associated with bilateral SAR models.

Software is developed (in compliance with the *HIPS* standard) for specification, estimation and simulation of bilateral SAR models. The estimation methods implemented cover LS-, approximative ML- and exact torus ML- estimation. The programs are described in appendix A.

Chapter 6

Spectrum Estimation

...the tragic accident that killed H. R. Seiwell and his family destroyed the only man who could usefully look at a plot of the autocorrelation against lag...

Tukey, 1966.

The power spectrum is, indeed, one of the most powerful tools available for data analysis and is widely applied in the natural sciences. A number of efficient techniques exist for power spectrum analysis of time series or 1-D signal processing, but not many techniques are available for 2-D data. Suffice it to say that 2-D spectrum estimation continues to be an area primarily for research.

The power spectrum, $P(\underline{f})$ of a random field, $\{y(s)\}$ is the two dimensional Fourier transform of the autocovariance function $\gamma(k,l) = E\left(y(r,c)y(r+k,c+l)\right)$:

$$P(\underline{f}) = \sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty}\gamma(k,l)exp\left(-i(f_r k + f_c l)\right) \tag{6.1}$$

The problem is to obtain an estimate of $P(\underline{f})$ from a finite sample of the process $\{y(s)\}$ where the sample usually consists of values $y(s)$ on a finite lattice $\Omega = \{s : s = (r, c), 0 \leq r, c \leq M - 1\}$.

We assume that the random process, $\{y(s)\}$, is wide sense stationary (spatially homogeneous) and, for the sake of simplicity, has zero mean. Since we have only one sample realization of $\{y(s)\}$, we also assume the ergodicity property which implies that the ensemble averages can be replaced by the spatial averages.

The various methods of power spectrum estimation are normally divided into two main categories: (1) Classical nonparametric and (2) modern parametric methods. The classical methods are mostly straightforward extensions of 1-D methods. The parametric methods assume that the considered data are appropriately described by an interaction model. Once an appropriate model is fitted, the spectrum is a function of the model parameters. Hence the problem of spectrum estimation becomes one of fitting an appropriate model to the given data. One can fit several types of spatial models. In this thesis we have focused on causal and non-causal simultaneous models in chapter 4 and chapter 5 respectively, and we shall use these as the basis for parametric spectral estimators which are the main topic of this chapter.

In the next chapter the parametric spectral estimates are used for the extraction of features used for texture classification. In this chapter, parametric methods are used estimate harmonics in white noise. This is not only an academic exercise as the ability to estimate harmonics at sub-pixel precision is of paramount importance in many technical problems as e.g. the estimation of molecular structure in Scanning Tunnelling Microscopy imagery (Jorgensen (1991)).

6.1 Classical spectral estimation

The classical, or conventional, methods are typically Fourier based and are all simple extensions of their 1-D counterparts, exploiting the separability of the 2-D Fourier transform. The introduction of the Fast Fourier Transform has helped to make these methods, sometimes referred to as direct methods, quite popular as they are computationally very attractive. The direct method can be applied in one of several ways. A few typical examples are given below.

In spite of the conspicuous shortcomings of the periodogram this is one of the most commonly applied estimators

$$I(f_r, f_c) = \frac{1}{M^2}\left|\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} y(m,n)\exp\left(-i2\pi(\frac{f_r m}{M}+\frac{f_c n}{M})\right)\right|^2 \qquad (6.2)$$

It is well known that this estimate is an inconsistent estimate of the power spectrum. For example when the samples from a Gaussian white noise process are used, the periodogram estimate has a variance that is equal to the variance of the white noise process. The variance can be reduced in a number of different ways. One way is to use a data window $w(m,n)$

$$P(f_r, f_c) = \frac{1}{U}\frac{1}{M^2}\left|\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} w(m,n)y(m,n)\exp\left(-i2\pi(\frac{f_r m}{M}+\frac{f_c n}{M})\right)\right|^2 \qquad (6.3)$$

where $U = \frac{1}{M^2}\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} w(m,n)^2$ is a normalizing constant.

Using this approach, the periodogram is convolved with the transform of the window. By choosing a suitable window (see section 3.4) small sidelobes, compared to the mainlobe of the transformed window, may be obtained. This,

in effect, will smooth the periodogram. As mentioned in section 3.4 care must be taken in the window carpentry to ensure that the estimate is positive. A sufficient (but not necessary) condition is that the transformed window is positive.

Another approach to variance reduction is to segment data. Specifically Bartlett (1953) suggests to segment data into non-overlapping blocks. The spectrum is estimated in each block, using the periodogram estimator, and the final estimate is obtained by averaging the periodogram from each block. The variance is reduced by a factor equal to the number of blocks. This, of course, is at the expense of frequency resolution. Welch (1967) suggested that, in addition to the averaged periodogram, each block should be windowed using different window schemes in order to remove the sidelobe structure.

Since the data values near the edges of each block, due to the windowing scheme, have very little influence compared to the data values near the center of the block, the edge values can be reused to increase their contribution in the final estimate. This can be accomplished by allowing a certain overlap of blocks. This method is often referred to as Welch's method and is generally considered the best among the classical methods.

6.1.1 Example

One of the most common experiments in the context of spectrum estimation is to examine harmonic series embedded in white noise, so as to investigate resolution- and detection- properties of a given spectral estimator. Tjøstheim (1981) considered models of the form

$$H(r,c) = \epsilon(r,c) + A\cos(\alpha_r r + \alpha_c c) + B\cos(\beta_r r + \beta_c c) \qquad (6.4)$$

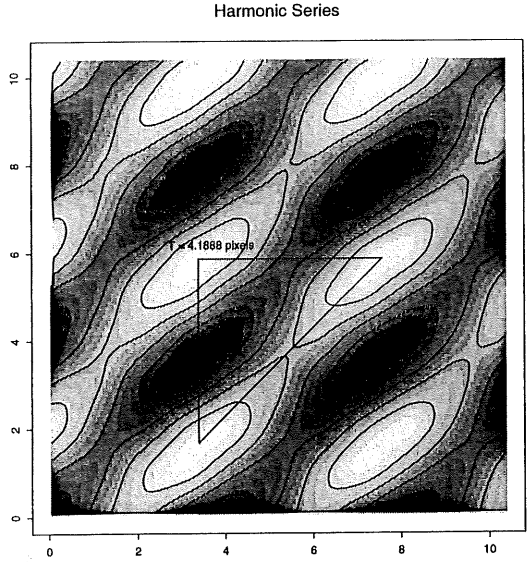

Figure 6.1. *The figure shows, enlarged, a part of the harmonic series described by equation 6.4 The length (in pixels) of the periods shown.*

where $\{\epsilon(r,c)\}$ is white Gaussian noise of variance σ_ϵ^2. The primary interest is in the detection of low amplitude cosines and the parameters are selected as $A = 0.4, B = 0.9, \alpha_r = 0.0, \alpha_c = 1.5, \beta_r = 1.5, \beta_c = 1.5$. The pure harmonic is shown, enlarged, in figure 6.1. In all the cases considered in this chapter noise variance σ_ϵ^2 is unity. In figure 6.2 the power spectrum is estimated using the direct method with different window schemes.

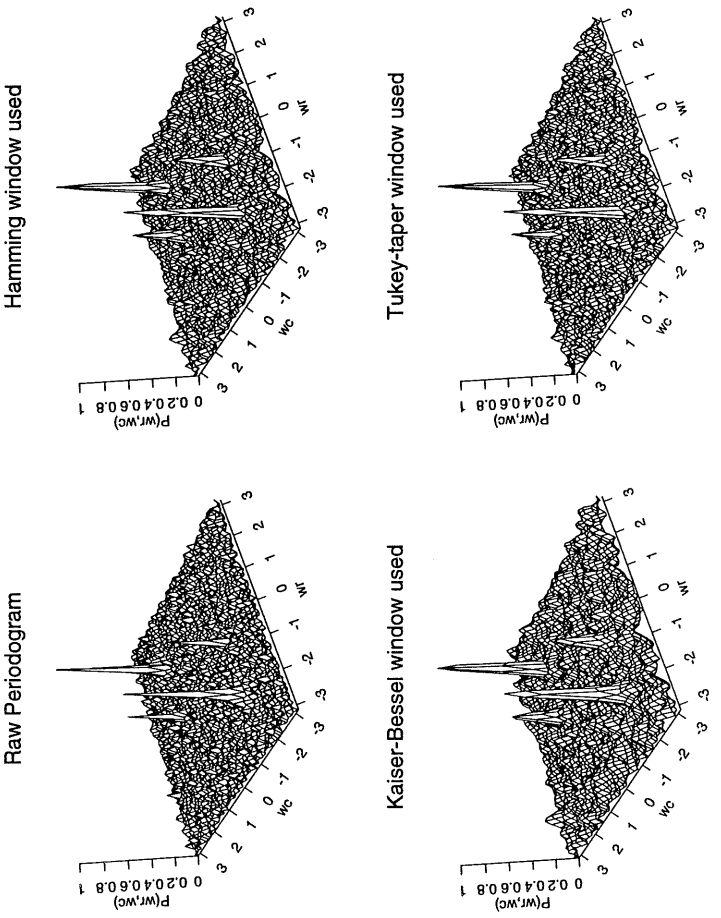

Figure 6.2. *The power spectrum of the harmonic series described by equation 6.4, superimposed on white noise of unit variance, is estimated using the direct (Fourier based) methods with different window schemes.*

6.2 Parametric methods

The main problem in relation to the classical spectral estimators is the variance-resolution tradeoff. This, and the fact that the amount of data is often limited, has motivated the search for better methods. In time-series analysis and 1-D signal processing the parametric methods have been used with considerable success. Various extensions of these methods have been proposed. Burg (1975) proposed an iterative estimator for obtaining the 2-D Maximum Entropy (ME) power spectrum. Experimental results of this method has never been reported (according to Lim (1990)). As successful iterative algorithm were later proposed by Lim & Malik (1981). Also Gaussian Markov Random Fields (GMRF) have been considered by e.g. Sharma & Chellappa (1985) where it is shown that the ME spectral estimate may be obtained using the GMRF models. In this section SARMA spectral estimation is considered using causal mixed SARMA models and noncausal SAR models.

The (non-normalized) spectral density function of y_s is given as the square modulus of the frequency response function multiplied by the noise variance:

$$\begin{aligned} P(f) &= \sigma_\epsilon^2 \frac{\left|1+\sum_{r\in N_{MA}} \theta_r exp(-2\pi i \cdot r^T \cdot f)\right|^2}{\left|1+\sum_{r\in N_{AR}} \phi_r exp(-2\pi i \cdot r^T \cdot f)\right|^2} \\ &= \sigma_\epsilon^2 \frac{\left(1+\underline{\theta}^T \underline{C}(N_{MA})_f\right)^2 + \left(\underline{\theta}^T \underline{S}(N_{MA})_f\right)^2}{\left(1+\underline{\phi}^T \underline{C}(N_{AR})_f\right)^2 + \left(\underline{\phi}^T \underline{S}(N_{AR})_f\right)^2} \end{aligned} \tag{6.5}$$

where the spatial frequencies $(f_r, f_c) = f \in \Omega_f$, and

$$\Omega_f = \{f = (f_r, f_c) = (-\frac{1}{2} + \frac{r}{M}, -\frac{1}{2} + \frac{c}{N}) : (r,c) \in \Omega\} \tag{6.6}$$

The vector notation is defined in

$$\begin{aligned}
\underline{\phi} &= \underline{col}\,[\phi_r,\ \ r \in N_{AR}] \\
\underline{\theta} &= \underline{col}\,[\theta_r,\ \ r \in N_{MA}] \\
\underline{C}(N.)_f &= \underline{col}\,[cos\,(\omega(f,r))\,,\ \ r \in N.] \\
\underline{S}(N.)_f &= \underline{col}\,[sin\,(\omega(f,r))\,,\ \ r \in N.] \\
\omega(f,r) &= 2\pi(r_r \cdot f_f + r_c \cdot f_c)
\end{aligned} \tag{6.7}$$

Using this approach, the problem of spectral estimation becomes one of fitting an appropriate model to the observed data. Strictly speaking the type of models considered here do not cover data of the type described in equation 6.4 since the associated spectral density has singularities located at the spatial frequencies of the cosines, and hence the continuity requirement is not met. This difficulty may not be absolutely critical since, from an estimation point of view, it may still be possible to approximate singular models with models having continuous spectral densities (Tjøstheim (1981)). After all this is what is done in the time series case.

Different SARMA spectral estimators have been considered for resolving, and detecting, harmonic series in white noise. Cadzow & Ogino (1981) considered quarter-plane SARMA models. The autoregressive part of the model is estimated using the least squares estimator. The spectral density of the moving average part is obtained through the residuals. It is noted that the overall resolution capability of this estimator is predominantly influenced by the autoregressive coefficient selection. Severe directional bias and peak splitting is observed for this type of estimator. This is shown in figure 6.4 and figure 6.5 where two quarter-plane SAR models were used to estimates the power spectrum of the data generated by the model 6.4. Peak splitting is not visible in figure 6.5 due to the viewing angle (which is chosen identical for all spectral

plots). While the directional bias can be reduced a little by increasing the order of the model, peak splitting remains a problem.

Chien (1981) considers a "parallel-resistor" average of the two quarter-plane models:

$$P_{AV}(f_r, f_c) = \frac{2}{\frac{1}{P_{QP(+,+)}(f_r,f_c)} + \frac{1}{P_{QP(-,+)}(f_r,f_c)}} \tag{6.8}$$

The idea is that peaks must coincide in the two spectral estimates for a peak to appear in the resulting averaged spectral estimates. In figure 6.6 the spectrum is estimated using the two quarter-plane models $QP(+,+)$ and $QP(-,+)$ both of order $R(5)$ (the spectra are shown in figure 6.4 and figure 6.5). This estimate is extremely smooth and peak splitting is removed. This heuristic type of estimator is generally considered one of the best 2-D parametric spectral estimators.

Comparatively, the use of Non-Symmetrical Half-Plane (NSHP) SAR models has received much less attention. According to Kay (1988) the 'signals in noise 'spectral estimators based on the NSHP perform poorly. The reason for this is not known. The experience of this author is, however, that the NSHP spectral estimator performs reasonably well in the experiments carried out here. In figure 6.7 an NSHP model of order R(5) is used to model the signal. As in the previous the estimate of the weak cosine is comparatively poor. This is alleviated by increasing the order of the model. In figure 6.8 a similar model of order R(9) is used. It is seen that the directional bias of the model has decreased and the detection of the weak cosine has improved significantly.

The use of mixed SARMA models, where the estimates are obtained e.g. through non-linear Conditional Least Squares (CLS), has generally been limited by the computational complexity of the estimation. The usefulness of

the mixed SARMA may not be clear in the context of 'signals in noise' spectral estimators where the primary interest is to approximate the poles of the frequency response. In other applications, where valleys in the spectral density of the process are important, the moving average parts are important in order to obtain parsimonious models. In figure 6.9 the spectrum of a mixed NSHP-SARMA(R(5),R(1)) is shown. The parameters are obtained as the CLS estimates. This spectral estimate hardly differs from the pure autoregressive spectral estimate in figure 6.7.

The models considered above are all causal. As pointed out in chapter 5 this constraint is not natural for spatial data. Sharma & Chellappa (1986) used non-causal SAR models for 2-D spectrum estimation for harmonics in white noise. In chapter 5 the approximate iterative estimator proposed by Kashyap & Chellappa (1983) (a toroidal model is assumed) was investigated. The spectral estimate obtained using this estimator is shown in figure 6.10. A non-causal SAR model of order R(5) was used. The presence of the weak cosine is just barely visible. The estimate is improved using the Maximum Likelihood estimator and the resulting spectrum is shown in figure 6.11. The use of this type of estimator is restricted by the fact that the non-linear optimization of the involved likelihood function is a very demanding task. If the length of the period to be identified is long, the order of the model should be of a corresponding order. Hence the order of the optimization problem may be prohibitive for this type of spectral estimators. These problems might be alleviated by considering the signal at different scales or by introducing 'seasonal' models (as in time-series analysis (Box & Jenkins (1976)). This is, however, a topic for future research.

6.2.1 Experiment

In some applications, it is important to be able to estimate the periods (or spatial frequencies) at sub-pixel accuracy. In these cases it is convenient that the parametric spectrum is determined as an analytical function of the parameters. Hence it is possible to use non-linear optimization to determine the exact location of a given peak. The accuracy of the location is, of course, closely connected to the appropriateness of the model used. Different parametric spectral estimators have been used to estimate the spatial frequencies of the harmonic series described in equation 6.4. A quasi-Newton method was applied to determine the exact location of the peaks and the results are listed in figure 6.3. The best overall performance was obtained using the averaged quarter-plane model. By selecting larger models it is possible to increase the performance of all the spectral estimators at the expense of increased computing time.

Model	Spatial frequency $(f_{\beta_r}, f_{\beta_c})$	Spatial frequency $(f_{\alpha_r}, f_{\alpha_c})$
True frequencies	(0.2387,0.2387)	(0.0000,0.2387)
QP(+,+) SAR(R(5))	(0.2432,0.2324)	(0.0086,0.2353)
QP(-,+) SAR(R(5))	(0.2355,0.2353)	(-0.0003,0.2301)
"Parallel" Averaged QP(+,+) and QP(-,+) SAR(R(5))	(0.2382,0.2382)	(0.0058,0.2371)
NSHP(< + >,+) SAR(R(5))	(0.2389,0.2326)	(0.0057,0.2353)
NSHP(< + >, +) SAR(R(9))	(0.2384,0.23441	(-0.0008,0.2331)
Approximate Non-causal SAR(R(5))	(0.2337,0.2374)	(0.0095,0.2386)
Exact ML Non-causal SAR(R(5))	(0.2234,0.2367)	(0.0099,0.2377)

Figure 6.3. *The exact peaklocations are found for the different spectral estimates, using non-linear optimization. The true frequencies are given in the first row - i.e.* $f_{\alpha_c} = f_{\beta_r} = f_{\beta_c} = \frac{\omega}{2\pi} = \frac{1.5}{2\pi} = 0.2387$ *and* $f_{\alpha_r} = 0.0$.

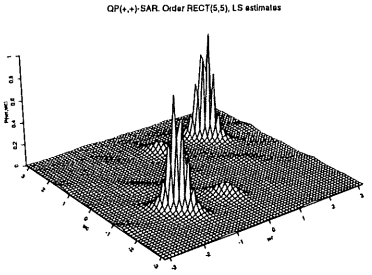

Figure 6.4. *The power spectrum estimate using a $QP(+,+)$ of order R(5). The peak splitting and the directional bias are very pronounced.*

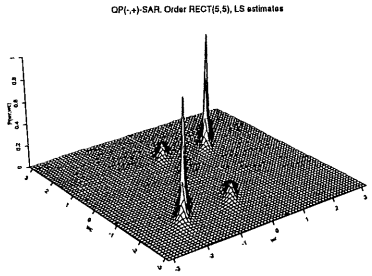

Figure 6.5. *The power spectrum estimate using a $QP(-,+)$ of order R(5). The peak splitting and the directional bias are very pronounced although this cannot be seen due to the viewing angle.*

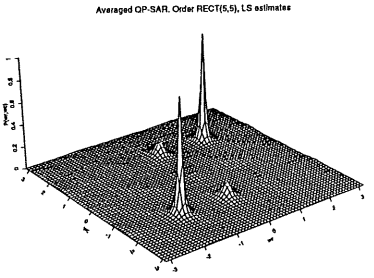

Figure 6.6. *The power spectrum is estimated using the "parallel resistor" average estimator based on the $QP(+,+)$ and $QP(-,+)$ models of order $R(5)$.*

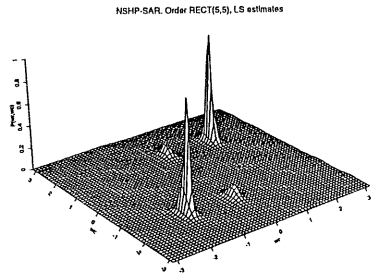

Figure 6.7. *The power spectrum is estimated using an NSHP-SAR model of order R(5). The weak cosine is not very clear and the directional bias is conspicuous.*

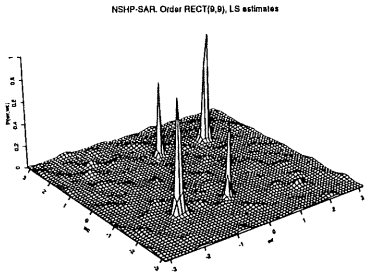

Figure 6.8. *The power spectrum is estimated using an NSHP-SAR model of order R(9). Both cosines are estimated satisfactory and the directional bias is vastly reduced.*

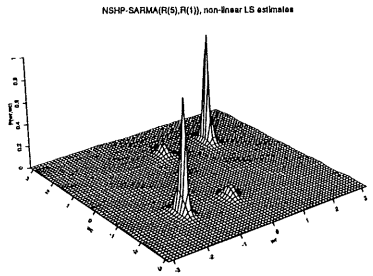

Figure 6.9. *The power spectrum is estimated using an NSHP-SARMA(R(5),R(1)) model The estimate is predominantly influenced by the autoregressive part which can be seen by comparing to the pure autoregressive estimate of the same order.*

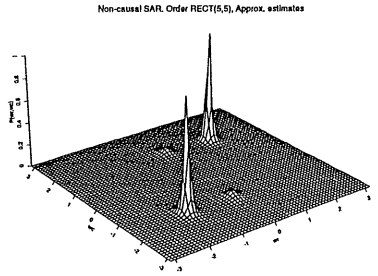

Figure 6.10. *The power spectrum is estimated using a non-causal SAR model of order R(5) and the approximative estimator proposed by Kashyap & Chellappa (1983).*

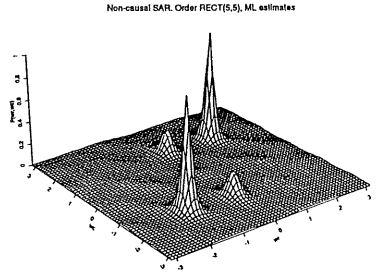

Figure 6.11. *The power spectrum is estimated using a non-causal SAR model of order R(5) and the exact ML estimator (still assuming a toroidal model) is used. The estimate of the weak cosine is improved.*

6.3 Conclusion

A number of spectral estimators have been used to estimate harmonic series embedded in Gaussian white noise. In particular we have studied the performance of parametric spectral estimators based on the simultaneous models (SAR and mixed SARMA). It is concluded that the performance of the parametric spectral estimators is very good. In particular the averaged quarter-plane spectral estimator is singled out with a very good performance. This type of spectral estimator is potentially useful for estimation of frequencies at sub-pixel precision.

Software is developed (in compliance with the *HIPS* standard) for all of the spectral estimators considered.

Chapter 7

Spectral Texture Statistics

Measures of textural information [1] must describe the information in data which is not expressed by means of individual pixel values but rather as a regularity in the 'local' variations in intensity. Such measures are of great importance in many areas of application. In medicine, the diagnosis of e.g. breast cancer from X-ray images (mammograms) or brain tumors from Magnetic Resonance (MR) images must take textural information into consideration. Remote sensing and industrial quality inspection are also areas where textural information is often the only way to quantify important properties of a considered process or geological area.

Visual textures often differs in coarseness or in (the degree of) directionality (anisotropy). As it is well known, the radial distribution of the power spectrum is sensitive to texture coarseness and the angular distribution is sensitive to directionality. In many applications the images are also corrupted by random noise. Typically such noise tends to alter the local spatial variation of intensity

[1] The term *measures of textural information* is used interchangeably with the terms *texture statistics* and *textural features*.

dramatically while (often) having a relatively uniform representation in spatial frequency. Thus, it has been popular to extract textural features from an estimate of the power spectrum for use in e.g. classification.

An early study of texture measures, by Weszka et al. (1976), (intersections of) rings and wedges were used as features for terrain classification in samples obtained from LANDSAT imagery. Features extracted from the raw periodogram were compared to features extracted from Gray Level Cooccurrence Matrices, Gray Level Differences and Gray Level Runlengths. It was concluded that Fourier features are not as effective as the other features considered. A similar conclusion were reached later in a similar comparison by Conners & Harlow (1980).

It is not the purpose of this chapter to discard these conclusions (which may or may not be correct) and prove spectral features superior to other texture statistics which are abundantly presented in the literature. It is felt, however, that spectral features are not always extracted and used appropriately. This fact was also pointed out by e.g. Wilson & Spann (1988). It is shown that spectral features do reflect important textural properties in an intuitively appealing way and that such information can be extracted in better ways than by calculating rings and wedges on the raw periodogram.

Kashyap, Chellappa, & Khotanzad (1982) were using the estimated parameters of non-causal SAR models (and the normalized residual variance) as textural features assuming that textures, in general, are represented by relatively few parameters. Such features are obviously not rotation invariant and Kashyap & Khotanzad (1986) introduced Circular Symmetric AutoRegressive (CSAR) random field models and showed that this type of models offer rotation invariance. This type of models were also used for texture classification by Mao & Jain (1992). The CSAR models, however, do not preserve the generative property that allows for a visual evaluation evaluation of the fitted model.

Both properties, i.e. rotation invariance and the generative property, may be preserved if textural properties are extracted from the parametric spectrum.

Different spectral texture statistics are discussed and it is demonstrated how these are affected by the choice of the spectral estimator. The effect of cellulase enzymatic treatment on textiles has been investigated using spectral texture statistics.

7.1 Introductory remarks

Several textures may be selected as a basis for the study of feature performance. Sets of Brodatz textures (Brodatz (1966)) are often chosen as such a basis although they represent only a small fraction of the textures in the universe. Since we cannot study all the textures in the universe it makes good sense to study feature performance on selected groups of textures that, in some sense, represent the type of textural properties we wish to quantify or classify. Of course caution must be used when attempting to extrapolate the results.

In the following a distinction is made between stochastic textures and semi-deterministic texture. The latter are textures containing an (almost) deterministic periodicity. A textile, as shown in figure 7.1, is an example of such a semideterministic texture, where the weaves form a regular pattern of almost deterministic character. The textile shown here is an example of the type considered in section 7.5, where the effect of cellulase enzymatic treatment on textiles has been investigated using spectral texture statistics.

We also wish to study feature performance with respect to stochastic textures. For this purpose the 12 Brodatz textures, listed below, have been selected.

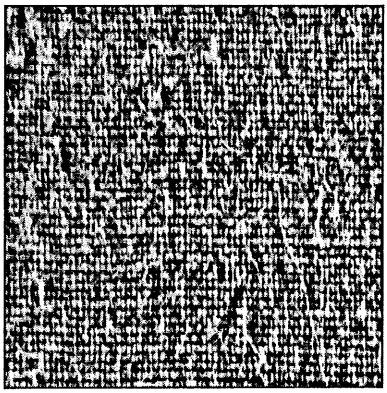

Figure 7.1. *The figure is showing a piece of (worn) cloth. The periodicity of the weaves is not completely deterministic, but very close to. Thus the term semidterministic is used to describe this type of textures.*

The label assigned to each texture is used e.g. when presenting scatter plots etc.

Brodatz texture	Label	Brodatz texture	Label
D002	1	D038	7
D004	2	D057	8
D009	3	D068	9
D012	4	D092	10
D024	5	D093	11
D029	6	D100	12

Each image was partitioned into 63 disjoint 64×64 subimages.

7.2 Rings and Wedges

Assume that an estimate, $P(u,v)$, of the power spectrum is obtained from a square $M \times M$ image at discrete spatial frequencies $(u,v) = (r/M - 1/2, c/M - 1/2)$, $(r,c) \in [0..M-1] \times [0..M-1]$

The ring-features are defined as

$$RF(r_1,r_2) = \frac{1}{\underline{card}R(r_1,r_2)} \sum_{(u,v)\in R(r_1,r_2)} P(u,v)/P_{TOT} \tag{7.1}$$

where $R(r_1,r_2) = \left\{(u,v) \mid r_1 \leq \sqrt{u^2+v^2} < r_2\right\}$, and

$$P_{TOT} = \sum_{(u,v)\neq(0,0)} P(u,v) \tag{7.2}$$

is the total power of the spectrum (with the DC value excluded).

Similarly the wedge-features are defined as

$$WF(a_1,a_2) = \frac{1}{\underline{card}W(a_1,a_2)} \sum_{(u,v)\in W(a_1,a_2)} P(u,v)/P_{TOT} \tag{7.3}$$

where $W(a_1,a_2) = \{(u,v) \mid a_1 \leq tan^{-1}(v/u) < a_2\}$

An example of the regions over which such features are sampled is given in figure 7.2. It should be obvious how features defined as intersections between rings and wedges are sampled.

Features calculated in this way were the basis for a systematic comparison of feature performance carried out by Weszka et al. (1976) where the features extracted from the raw periodogram were compared to features extracted from Gray Level Cooccurrence Matrices, Gray Level Differences and Gray Level Runlengths. The conclusion of this paper was that the performance of the Fourier features are somewhat inferior to the other features considered. This conclusion were later backed by Conners & Harlow (1980). These conclusions are often quoted in the literature. Although this may be a valid conclusion, it is important to point out that a number of factors may have deteriorated the performance of the Fourier features considered by Weszka et al. (1976):

- As the data apparently were not tapered, spectral leakage may have disturbed both wedge and ring features. From the terrain samples presented in Weszka et al. (1976) it is clear that the periodic extension of the individual samples will rarely be continuous at the borders. Thus spectral leakage will occur if the data is not tapered. It is remarkable that the performance of the wedge features are poorer than the performance of the ring features. The study presented later in this chapter point at wedge features as being one of the top performers. See also Wilson & Spann (1988).

- It is well known that there is a pronounced six-line banding in LANDSAT images (Knut Conradsen & Nilsson (1982)). This banding may have introduced peaks (or lines) in the estimated power spectrum and thus degraded the features further.

- The selection of feature-pairs seemed rather ad hoc. A systematic parametric approach to this using e.g. a Jeffries-Matusitas distance measure (as f.ex. Ersboll (1989)) or a non-parametric approach using e.g. Classification And Regression Trees (CART) (Breiman, Friedman, Olshen,

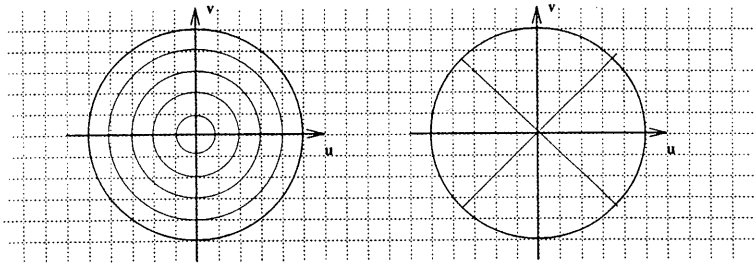

Figure 7.2. *The figure is illustrating the how ring-shaped samples and wedge-shaped samples are extracted from the power spectrum. The average intensity are calculated for each extracted area and used as a feature. Average intensity may also be calculated in intersections of rings and wedges and used as features.*

& Stone (1984)) might have given a better selection of features. In fact rings and wedges alone may not be the best choice of features at all.

- The samples considered were of size 64×64. This offers a relatively low resolution. As the periodogram estimator of the power spectrum is non-consistent and there were no attempt to smooth the periodogram, the variability of the textural features may have been large. A parametric spectral estimate might have helped this problem.

7.3 Other spectral features

Many features other than rings and wedges may, of course, be considered. In this section a number of features are listed, grouped and discussed. It has been chosen to divide features into three groups: global, harmonic and phase features. Harmonic features are only considered for semideterministic textures with periodic components. Global features may be considered for all types of textures, but typically features categorized in this way will not in general be a good choice when considering semideterministic textures.

Harmonic Features (HF) :	For a semideterministic texture, containing strong and well defined periodicities, spectral peaks are identifying the harmonics of the periodicity. The shape of the peaks, as well as their locations are described with harmonic features.
Global Features (GF) :	These features are describing the shape and orientation of the powerspectrum globally.
Phase Features (PF) :	Features measured on the phase spectrum.

7.3.1 Global features

The estimated power spectrum is approximated by a Gaussian centered at $(u, v) = (0, 0)$ where the dispersion

$$\Sigma = \begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{pmatrix} \tag{7.4}$$

assumed to be positive definite, is estimated using:

$$\hat{\sigma}_u^2 = \sum_u \sum_v u^2 p(u,v)$$

$$\hat{\sigma}_v^2 = \sum_u \sum_v v^2 p(u,v)$$

$$\hat{\sigma}_{uv} = \sum_u \sum_v uv p(u,v)$$

Here $p(u,v)$ is the normalized (estimated) power spectrum

$$p(u,v) = \frac{P(u,v)}{\sum_{(u,v)\neq(0,0)} P(u,v)} \tag{7.5}$$

It is obvious that one may consider weighted estimates (e.g. powers of inverse distance from $(u,v) = (0,0)$) of this dispersion matrix. This approach has not been pursued here.

Given the positive definite symmetrical matrix in equation 7.4 the eigenvalues are:

$$\lambda_1 = \left(\sigma_u^2 + \sigma_v^2 + \sqrt{(\sigma_u^2 - \sigma_v^2)^2 + 4\sigma_{uv}^2}\right)/2 \tag{7.6}$$

$$\lambda_2 = \left(\sigma_u^2 + \sigma_v^2 - \sqrt{(\sigma_u^2 - \sigma_v^2)^2 + 4\sigma_{uv}^2}\right)/2 \tag{7.7}$$

and the corresponding (normalized) eigenvectors

$$\underline{P}_i^T = \left(\frac{\sigma_{uv}}{\sqrt{\sigma_{uv}^2 + (\lambda_i - \sigma_u^2)^2}}; \frac{\lambda_i - \sigma_u^2}{\sqrt{\sigma_{uv}^2 + (\lambda_i - \sigma_u^2)^2}}\right) \tag{7.8}$$

Based on this approximation, many features may be extracted. Below is listed a number of such features of which some were proposed by Liu & Jernigan (1990). Other features based on this approximation are presented here as well.

1. **Correlation:**

$$GF_1 = \rho = \sigma_{uv}/\sigma_u\sigma_v \tag{7.9}$$

This feature was *not* considered by Liu & Jernigan (1990) but turns out to be an important feature for classifiaction.

2. **Major axis:**

$$GF_2 = \lambda_1 \tag{7.10}$$

Liu & Jernigan (1990)

3. **Minor axis:**

$$GF_3 = \lambda_2 \tag{7.11}$$

Liu & Jernigan (1990)

4. **Minor to major axis:**

$$GF_4 = \lambda_2/\lambda_1 \tag{7.12}$$

Liu & Jernigan (1990)

5. **Principal component direction:**

$$GF_5 = cos^{-1}\left(\frac{\sigma_{uv}}{\sqrt{\sigma_{uv}^2 + (\lambda_i - \sigma_u^2)^2}}\right) \tag{7.13}$$

Liu & Jernigan (1990)

6. **Anisotropy:**

It was proposed by Liu & Jernigan (1990) to quantify the anisotropy using the measure below:

$$\mathcal{A} = \frac{|\sigma_u^2 - \sigma_v^2|}{\sqrt{(\sigma_u^2 - \sigma_v^2)^2 - 4\sigma_{uv}^2}} \tag{7.14}$$

This feature is closely related to the fraction σ_v^2/σ_u^2 (see figure 7.3) which was not considered as a feature by Liu & Jernigan (1990). The latter feature is much simpler. When feature selection is made, using CART, this simple feature is always chosen instead of the more complicated. For this reason we propose:

$$GF_6 = \sigma_v^2/\sigma_u^2 \tag{7.15}$$

that turns out to be an important feature, at least for classifying stochastic textures.

Moment of inertia:

$$GF_7 = \sum_u \sum_v \left(u^2 + v^2\right) p(u,v) \tag{7.16}$$

Liu & Jernigan (1990)

7. **Moment of entropy:**

$$GF_8 = \sum_u \sum_v p(u,v) log(p(u,v)) \tag{7.17}$$

Liu & Jernigan (1990)

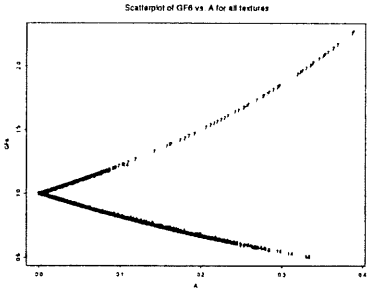

Figure 7.3. *The feature GF_6 is plotted versus the anisotropy feature proposed by Liu & Jernigan (1990). The deterministic nature of the relation between these two features is clearly demonstrated.*

7.3.2 Harmonic features

Harmonic features are used to describe the (relative) location of the harmonics as well as the size and shape of the associated peaks. Such features are typically used to discriminate between *different* semideterministic textures. The features listed below are all proposed by Liu & Jernigan (1990).

1. **Energy in major peak:**

$$HF_1 = p(u_1, v_1) \tag{7.18}$$

where (u_1, v_1) are the frequency coordinates of the maximum peak of the power spectrum.

2. **Laplacian of major peak:**

$$HF_2 = \nabla^2 P(u_1, v_1)$$

$$= P(u_1+1, v_1) + P(u_1-1, v_1 + P(u_1, v_1+1) \qquad (7.19)$$
$$P(u_1, v_1-1) - 4P(u_1, v_1)$$

It is well known that the Laplacian estimated in this way is extremely sensitive to noise. This problem can be dealt with in a number of ways. One way is to use the polynomial approximation presented by Grunkin (1993) and consider the Laplacian of a second-order polynomial fitted to the peak:

$$HF_2^* = a_1 + a_2 \qquad (7.20)$$

where

$$a_1 = \frac{45\, s_1(l) - 15\, l\ (l+1)\, s_6(l)}{l\,(2\,l+3)\,(2\,l-1)\,(l+1)}$$
$$a_2 = \frac{45\, s_2(l) - 15\, l\ (l+1)\, s_6(l)}{l\,(2\,l+3)\,(2\,l-1)\,(l+1)}$$

and

$$s_1(l) = \frac{1}{k^2} \sum_{(u,v)\neq(0,0)} u^2\, p(u,v)$$
$$s_2(l) = \frac{1}{k^2} \sum_{(u,v)\neq(0,0)} v^2\, p(u,v)$$
$$s_6(l) = \frac{1}{k^2} \sum_{(u,v)\neq(0,0)} p(u,v)$$

The polynomial is estimated over an $k \times k$ neighborhood where $k = 2 \cdot l + 1$.

3. **Laplacian of secondary peak:**

$$HF_3 = \nabla^2 P(u_2, v_2) \qquad (7.21)$$

where (u_2, v_2) are the frequency coordinates of the second largest peak in $P(u, v)$. See comment above.

4. **Spread of major peak:**

$$HF_4 = \underline{card}\left\{(u, v) \in N \mid P(u, v) \geq \frac{1}{2} kP(u_1, v_1)\right\} \tag{7.22}$$

where N is the four or eight nearest neighbors of (u_1, v_1).

5. **Squared major peak frequency:**

$$HF_5 = u_1^2 + v_1^2 \tag{7.23}$$

6. **Relative orientation of major and secondary peaks:**

$$HF_6 = \left|tan^{-1} v_1/u_1 - tan^{-1} v_2/u_2\right| \tag{7.24}$$

7. **Squared distance between major and secondary peak:**

$$HF_7 = (u_1 - u_2)^2 - (v_1 - v_2)^2 \tag{7.25}$$

7.3.3 Phase features

It is well known that the Fourier phase information of an image in many respects is of greater importance than the amplitude information. Eklundh (1979) considers rings and wedges extracted from the Laplacian of the magnitude spectrum as features for classification of the data that was earlier considered by Weszka et al. (1976). It is concluded, however, that the features extracted from the phase in this way do not carry much textural information. Phase features based on the Laplacian of major- and secondary peak phase

are proposed by Liu & Jernigan (1990). These features are used to classify textures from the Brodatz album (Brodatz (1966)) but apparently they do not convey any information relevant to texture classification.

7.4 Spectral feature performance

In this section, the performance of spectral features is studied for the selected group of stochastic features. In Carstensen (1992) the performance of cooccurrence features was evaluated on sets of semideterministic (i.e. periodic) and stochastic textures. Classification and Regression Trees (CART) were used to classify the texture samples. It was concluded that:

- It is generally easy to discriminate textures if no histogram match is performed since most Brodatz textures can be discriminated on their first-order statistics alone.
- Features based on a Gaussian match performed better than features based on histogram equalization and the decision trees tends to have more nodes if the features are based on histogram equalization.
- It is easier to classify semideterministic textures than stochastic textures.

Thus the textures considered here are histogram matched, using a Gaussian match.

There are two reasons for choosing only stochastic textures: 1) It is difficult to discriminate between stochastic features. Thus it is easier to measure differences in performance. 2) Stochastic models are not well suited to model deterministic structures. Since we wish to compare features extracted from

parametric spectral estimates with those extracted from non-parametric spectral estimates, it makes better sense to choose textures that are all modelled reasonably well with such stochastic models. In chapter 4 all the textures considered here are modelled satisfactorally with SARMA models.

In this section the global features, as well as the ring- and wedge- features, are used to classify samples of the textures listed in section 7.1. The purpose is threefold:

1. It is of interest to see how well 12 textures, that in many way look similar, can be discriminated using spectral texture statistics.
2. By using the CART classification approach we obtain an indication of the relative importance of the considered features.
3. Finally we shall study how the performance of the spectral texture statistic are influenced by the choice of spectral estimator.

7.4.1 CART classification

Classification and Regression Trees (CART) is a (relatively) new and powerful non-parametric alternative to classical methods in classification and regression. A binary decision tree is constructed and a classification is made by running down the tree and choosing the class corresponding to the terminal node. The CART program from California Statistical Software, Inc. were used. The reader is referred to Breiman et al. (1984) for a detailed exposition on the subject.

The features considered here are

- Ring features $RF_1, \cdots, RF_5$ obtained by extracting five rings.

- Wedge features $WF_1, \cdots, WF_5$ obtained by extracting five wedges in the interval $[0; \pi]$
- Global features $GF_1, \cdots, GF_8$ and the feature A.

By running the CART classification program we obtain a Cross Validation Classification Matrix indicating the estimated (mis-)classification rates. Along with this a total crossvalidated mis-classification probability is calculated. The results for features extracted from the *raw periodogram* are presented in figure 7.4. In figure 7.5 the relative importance of the features is presented.

This experiment is carried out using different types of spectral estimators:

1. The raw periodogram.
2. The logarithmic transform of the raw periodogram: $(1 + p(u, v))$.
3. The smoothed periodogram using a 3×3 median filter.
4. The parametric spectrum using a $QP(+,+)$ SAR model of order $E(6.5)$.
5. The weighted parametric spectrum using $QP(+,+)$ and $Q(+,-)$ SAR models of order $E(6.5)$
6. The parametric spectrum using a $NSHP(\oplus,+)$ SAR model of order $E(6.5)$.
7. The parametric spectrum using mixed SARMA models. The autoregressive order is the (BIC-) optimal $NSHP$ model found in section 4.5.2, extended with an $NSHP(\oplus,+) - SMA$ part of order $R(1)$.

The results presented in figure 7.6 indicate that:

- The overall classification performance of spectral texture statistics is good when stochastic textures are considered.
- Wedge features are very important when non-parametric spectral estimates are used.
- The discriminatory power of the extracted features is not enhanced by a logarithmic transform of the power spectrum. The classification tree becomes slightly simpler.
- Smoothing of the periodogram does not seem to enhance the discriminatory power of the extracted features. It should be noted, however, that the classification tree is slightly simpler.
- The use of parametric spectra will generally improve the performance of the extracted features.
- The anisotropy measure GF_6 seems to be a generally important feature when parametric spectral estimates are basis for the feature extraction.

7.4.2 A comparative study

In Carstensen (1992) the performance of features extracted from Gray Level Cooccurrence Matrices (GLCM) were evaluated with respect to their ability to classify stochastic textures. A set of five textures were selected: D004, D009, D029, D032 and D057. The crossvalidated misclassification probability were 18.3% , and the classification tree contained 13 terminal nodes.

The same set of textures has been selected to make a direct comparison of feature performance possible. The features used in section 7.4.1 are all used. The features are extracted from the weighted Quarter-Plane SAR spectral estimate as this turned out to be the top-performer in section 7.4.1. The resulting

	1	2	3	4	5	6	7	8	9	10	11	12
1	.89	.00	.03	.00	.00	.00	.00	.00	.00	.00	.00	.00
2	.00	1.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
3	.02	.00	.83	.02	.03	.02	.00	.00	.00	.02	.00	.10
4	.00	.00	.00	.92	.00	.00	.00	.00	.02	.03	.00	.02
5	.03	.00	.02	.02	.94	.00	.00	.00	.00	.00	.00	.00
6	.00	.00	.03	.00	.00	.89	.02	.00	.00	.05	.00	.19
7	.00	.00	.00	.00	.00	.00	.95	.00	.00	.00	.02	.00
8	.00	.00	.00	.00	.00	.00	.02	1.00	.00	.00	.00	.00
9	.03	.00	.00	.00	.02	.00	.00	.00	.98	.00	.00	.00
10	.03	.00	.02	.05	.02	.03	.00	.00	.00	.90	.00	.02
11	.00	.00	.00	.00	.00	.00	.02	.00	.00	.00	.98	.00
12	.00	.00	.08	.00	.00	.06	.00	.00	.00	.00	.00	.68

Figure 7.4. *Cross Validation Classification Matrix generated by CART for features extracted from the raw periodogram. Rows: Predicted class. Columns: True class. Total misclassification probability: 8.59%.*

	Relative Importance	Feature
Variable 20	100.	WF_3
Variable 19	93.	WF_2
Variable 10	93.	GF_6
Variable 18	85.	WF_1
Variable 22	84.	WF_5
Variable 2	81.	GF_1
Variable 21	63.	WF_4
Variable 6	63.	GF_5
Variable 13	60.	RF_1
Variable 15	56.	RF_3
Variable 5	54.	GF_4

Figure 7.5. *Relative variable importance generated by CART for features extracted from the raw periodogram.*

Spectral estimator	Best features	% Misclassif.	Terminal nodes
Raw Periodogram	WF_3, WF_2, GF_6	8.59	27
Logarithmic Periodogram	WF_5, WF_3, WF_1	8.99	24
3×3 median smoothed	WF_3, WF_2, GF_6	8.86	25
$QP(+,+)$ E(6.5) SAR	GF_6, GF_5, WF_5	9.79	13
Weighted E(6.5) $QP(+,+)$ and $QP(-,+)$	GF_6, WF_5, GF_8	7.40	18
$NSHP(\oplus,+)$ $E(6.5)$	GF_6, GF_1, WF_2	7.50	37
Mixed $SARMA$	GF_6, WF_2, GF_1	8.50	26

Figure 7.6. *The first column listes the spectral estimator used for the classification experiment, the second column lists the three most important features found by CART, the third column lists the crossvalidated misclassification probability and the fourth column lists the number of terminal nodes in the classification tree.*

CART-classification tree is shown in figure 7.7. The cross validated misclassification probability is reduced to 2.2% and the decision tree is extremely simple containing only *six* terminal nodes. Again the most important feature is the anisotropy measure GF_6. Also the correlation feature GF_1 is important. The scatter plot in figure 7.8 shows that there is much discriminatory information in these two features alone. The texture labels used in the scatterplot corresponds to the labels used in figure 7.7.

This illustrates that spectral texture statistics measured on a parametric spectral estimate are sensitive to subtle differences in spectral density. This example *does not* prove that spectral texture statistics are superior to cooccurrence features. It show, however, that care must be taken when features are selected for a given purpose - in this case classification.

It is more than likely that cooccurrence features extracted, using different displacement vectors, would yield results that would match the results presented here. It should be noted, however, that the choice of such displacement vectors may not be simple. The spectral features considered seems to convey much textural information for all the stochastic textures considered. Furthermore the features has intuitively appealing interpretations which is indeed a valuable property.

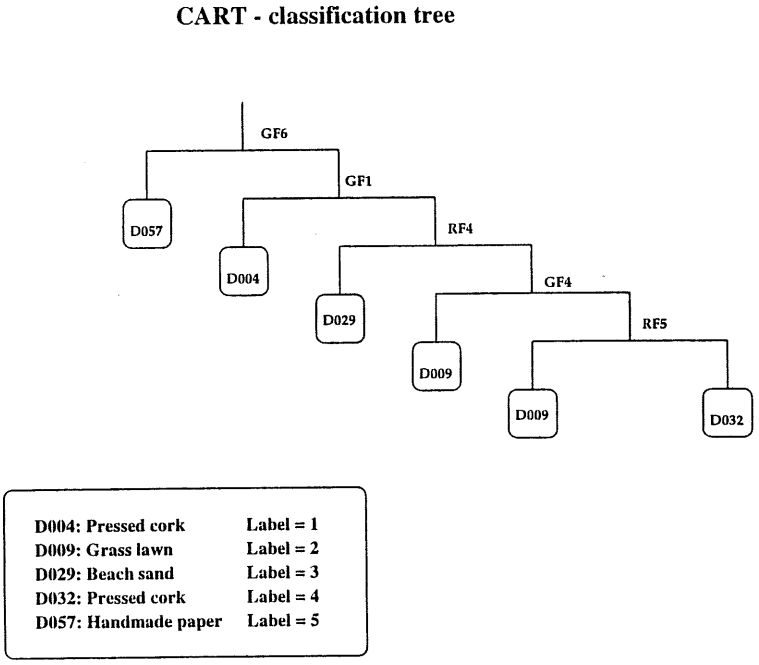

Figure 7.7. *The figure shows the CART-classification tree. The cross validated misclassification probability is 2.2% .The tree is very simple with the most important feature being the measure of anisotropy* GF_6.

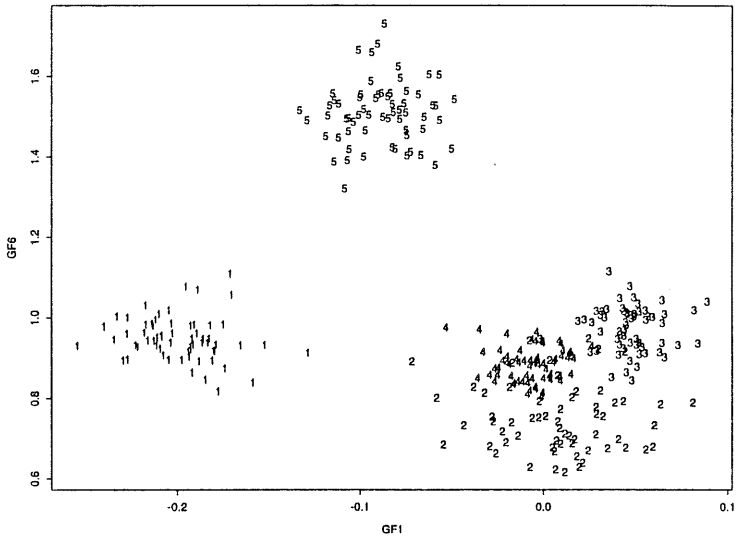

Figure 7.8. *In the scatterplot the extracted GF_1 and GF_6 features are plotted against each other for every texture sample. It is seen that much discriminatory information is offered by these two features alone.*

7.5 Measurement of enzymatic treatment effect on textile

The effect of cellulase enzymatic treatment on textiles has been investigated using standard texture algorithms. An extensive study in both the Fourier domain and the spatial domain has revealed the nature of the changes and resulted in one single feature that measures these changes in a fast and robust way. This work was previously published in Carstensen et al. (1991). In this section the focus will be on the Fourier based approach to the problem.

7.5.1 Background

This project started when the R&D group in the detergent enzyme division of Novo Nordisk (a world-leading manufacturer of detergent enzymes) expressed the wish to quantify the effects of enzymatic treatment of textiles using digital image analysis. Until now this quantification has been done qualitatively using microscopic inspection and quantitatively using panel tests and light measurements (Hunter coordinates). There was a need for a new objective, robust, fast and relatively inexpensive method.

7.5.2 Image acquisition

The image acquisition is carried out as follows. The textile is placed in homogeneous and plentiful lighting. A camera is positioned such that it's optical axis is perpendicular to the textile plane and the rectangular visual area covers as much of the textile as possible without including non-textile areas. The size of the textiles in this study is 15x10cm.

We used an RGB high-resolution slow-scan camera. The camera output is digitized by a framegrabber that generates frames of 978 by 768 pixels in the red, green and blue band. These frames are cut to 969 by 711 to eliminate acquisition artifacts. Subsequently we will only show results derived from the green band since the textiles used in this experiment are black and gray and thus contains very little or no color information.

This study regards the enzymatic treatment effect for a single type of cellulase. We want to assess the effect at different pH values and for different doses. To assess the day-to-day variation the textiles were washed on different days for each pH-level. Thus we have three factors that we want to investigate.

- pH: 3 levels, 1 2 3
- dose: 8 levels, 0 10 25 40 50 75 100 200
- day(pH): 3 levels, 1 2 3 (for pH values 7.0,8.0 and 9.0)

We have two repetitions for each combination, thus we end up with 144 images. In figure 7.9 we see 8 textiles representing the 8 doses for pH 1, day 1 and repetition 1.

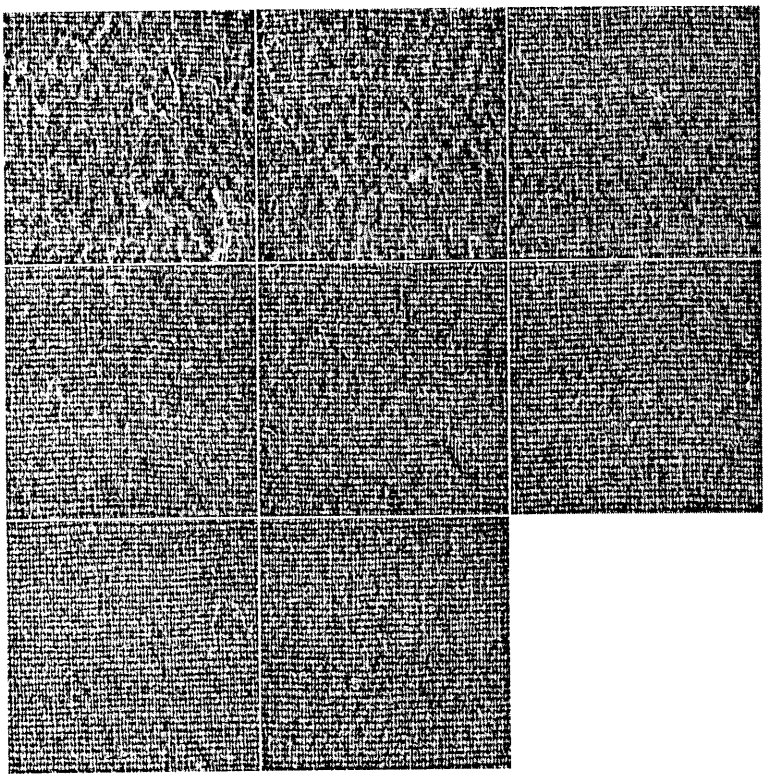

Figure 7.9. *8 textiles representing the 8 doses for pH 1, day 1.*

7.5.3 Description of visual properties

The object of the digital image analysis is to compute one feature that quantifies a given visual property from the image array. In this case the visual property is the human perception of wear. The feature has to correlate well with panel tests. For cellulase enzymatic treatment with known effects this means that the feature has to show improvement as a function of dose and show best results for pH values close to the pH with highest enzyme activity (between 7.0 and 8.0 in our case).

Obviously many different features can be computed from the image. A simple feature is the average intensity, *lightness*. This has a strong resemblance to what is measured by the Hunter coordinates. Probably this lightness feature also has a strong influence on a panel test. Figure 7.10 shows the average intensity as a function of dose for pH level 1. We see that lightness only has discriminative capability for small doses. In the context of image analysis lightness is a non-robust feature in the sense that it depends heavily on lighting conditions and camera sensitivity.

Another aspect of enzymatic effect on the textiles estimated by the panel test is the *distinctness* of the regular textile pattern. This distinctness should increase as a result of the cellulase enzymatic treatment. The regular pattern in the investigated textiles resembles a rectangular grid structure. The well defined period of this grid makes it appropriate to look at the textiles in the Fourier domain. This is done in the next section.

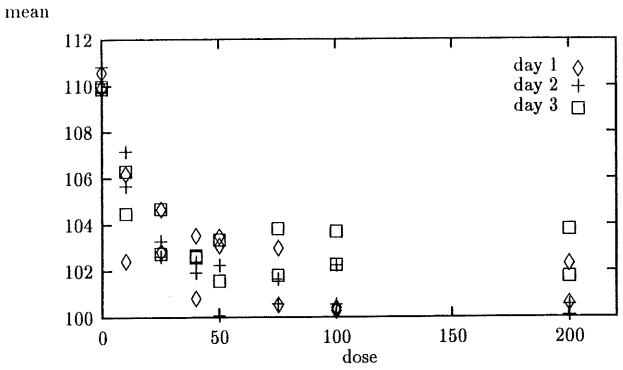

Figure 7.10. Average intensity as a function of dose for pH level 1.

7.5.4 Analysis in the Fourier domain

The classical way of obtaining an estimate of the power spectrum is the periodogram, $I(u,v)$ as given in equation 3.26. The log-power spectrum is given as $log(1.0 + I(u,v))$.

The periodogram is a non-consistent estimate of the power spectrum. *Bartlett's method* is one way to deal with this. The image is split up in a number of non-overlapping subimages. The periodogram is calculated in each subimage, followed by an averaging over the subspectra.

Figure 7.11 shows the full resolution power spectra of the textiles in figure 7.9. The concentric circles are isolines for the spatial frequency. Several high-intensity spots in the power spectrum is showing the periodicity of the weaves. The spots of lower intensity in the high-frequency areas are higher harmonics. We see that the intensity in the low-frequency areas (near the center of the power spectrum) is fading for higher doses of enzyme. To illustrate this effect

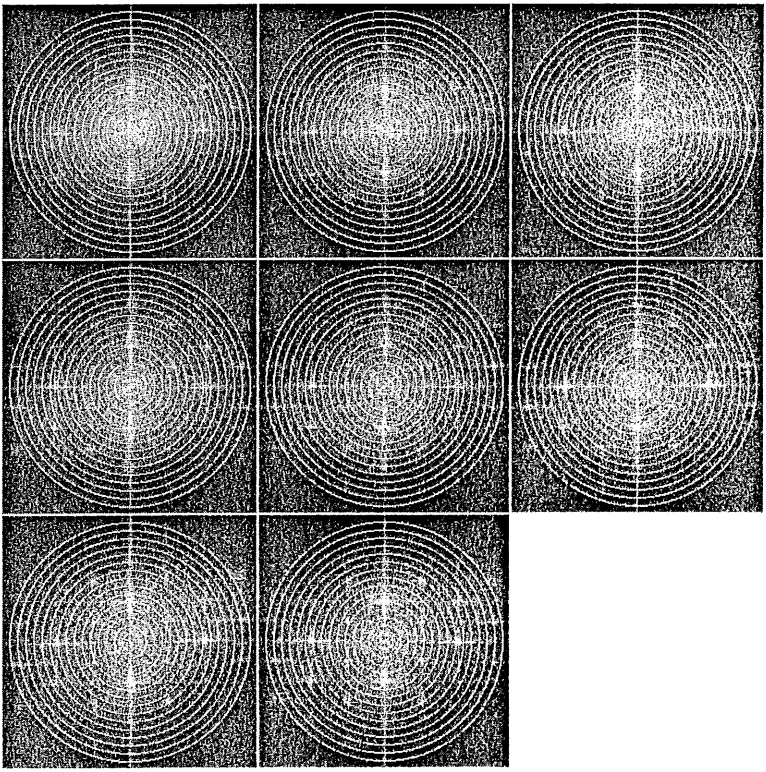

Figure 7.11. *Power spectra of the textiles in figure 7.9.*

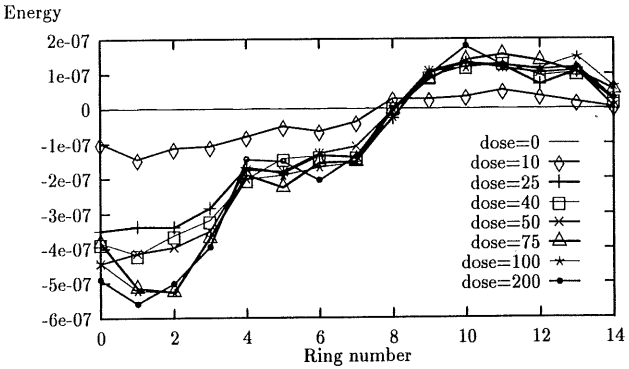

Figure 7.12. Average of power spectra rings relative to dose 0 for the spectra in figure 7.11.

we computed the average of the power spectrum in the rings between the concentric circles and plotted it versus the radius of the rings. These averages are computed for each of the power spectra in figure 7.11, and the average corresponding to dose 0 subtracted from the averages of each of the other doses. The plot is shown in figure 7.12, and it is obvious that the averages in the low-frequency areas are decreasing for higher doses. We also note that all the curves has approximately the same intersection at a frequency corresponding to the frequency of the weaves. Thus having established that the power spectrum actually contains relevant information about the textile wear, we will try to quantify this in a single Fourier feature.

7.5.5 Spectral texture features.

Texture features derived in the spatial frequency domain have been investigated e.g. in Weszka et al. (1976) and Liu & Jernigan (1990). The features tested in the present context are listed below.

1. Rings
2. Wedges
3. Inertia
4. Entropy
5. Anisotropy

All features has been computed on both the full resolution power spectrum and the power spectrum estimated using Bartletts's method. The Bartlett spectral estimate performed significantly better than the full resolution power spectrum.

The features were computed on both the power spectrum and the log-power spectrum. It turned out that the features calculated on the log-power spectrum performed significantly better than the power spectral features. Furthermore we found that inertia and entropy features performed better than the other features. The inertia feature performed generally a little better than the entropy feature, and it seems to be a more natural way summarize the phenomena observed in figure 7.12.

The inertia feature I and log-power inertia LI is computed as

$$I = \sum_{(u,v)} (u^2 + v^2) \mid P(u,v) \mid^2$$

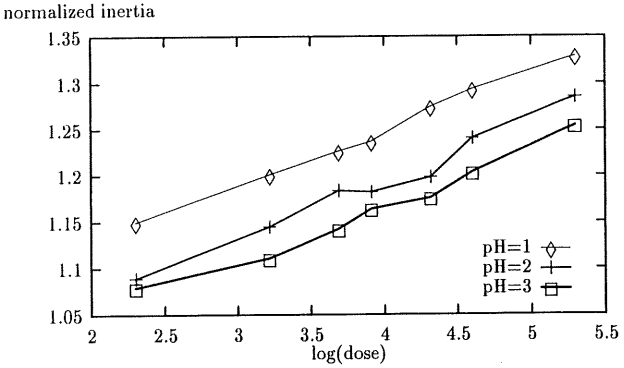

Figure 7.13. *Normalized log-power inertia versus log(dose). We see that the measure reflects the expected ranking.*

$$LI = \sum_{(u,v)} (u^2 + v^2) \log(1 + \mid P(u,v) \mid^2)$$

where we are summing over all frequencies of the Bartlett spectral estimate. The normalized inertia is the inertia divided by the inertia of the corresponding textile with $dose = 0$. In figure 7.13 we show the normalized log-power inertia vs. log(dose) for all three values of pH. Thus the measure is averaged over days and repetitions. It can be seen that there is a clear distinction between the performance of the enzymes at the three pH values. In addition there seems to be an approximately linear relation between the inertia and log(dose).

Discussion of results

The spectral approach has provided us with a useful feature and a lot of insight regarding the nature of this problem. The use of the FFT algorithm however introduces some, somewhat technical, limitations regarding the size of the

image and computational speed. It is also less flexible in removing textile irregularities from the analysis. Therefore a fast and robust method have been implemented in the spatial domain for the final system, which is in operation at NOVO today. See Carstensen et al. (1991).

7.6 Conclusion

In this chapter, different spectral features have been investigated. The features were extracted from different estimates of the powerspectrum, and it was shown that spectral features do convey much textural information in an intuitively appealing way. It was shown that features extracted from parametric spectral estimates performed very well. In particular the averaged quarter-plane SAR model was singled out with a very good performance. Two new features (the correlation and anisotropy features), introduced in this chapter, turned out to be among the top performers for classification purposes. Finally it was shown that features extracted from the Bartlett estimate of the powerspectrum was well suited to measure the effect of cellulase treatment on textiles.

Chapter 8

Conclusion

The analysis of texture confronts us with two important questions: *what* is texture and *how* do we describe texture ? In particular, many attempts have been made to answer the last question. This thesis has contributed to these attempts be going through selected theory, case studies and practical applications.

8.1 Summary

Fourier methods have been reviewed and the importance of windowing was illustrated. Also the problems in relation to classical spectral estimation was discussed.

Causal Simultaneous AutoRegressive Moving Average (SARMA) models were considered. These models are generative which means that textures may be simulated from a fitted model. In this way the subjective, visual quality of the fitted model may be evaluated. Different textural features may be extracted

from the fitted model. In this thesis the parametric spectrum is considered as the basis for feature extraction. The identification of an appropriate model is an important step when modelling textures. An automatic identification algorithm, based on Bayes Information Criterion (BIC) and statistical test theory, were introduced. The method is basically a backwards elimination method. It was demonstrated that this method is well suited to identify models correctly. Pure SAR models, with different regions of support (i.e. QP and NSHP) were used to model a selected database of stochastic textures. It was demonstrated, through estimation and simulation, that good models could be obtained for the selected textures. In general, the best performance was obtained using the NSHP models. It is an interesting fact, that all of the visual textures considered here lies on the 'edge' of non-stationarity. This seems to be a salient feature of most natural textures.

The estimation of mixed SARMA models is a non-linear optimization problem. The performance of the Conditional Least Squares (CLS) estimator were studied and it was demonstrated that the estimates, thus obtained, are likely to be biased for small images with large positive correlations. This is due to the conditioning on initial values for the prediction errors. An Unconditional Least Squares estimator were proposed for SARMA models, employing iterated back forecasting. It was demonstrated that significant improvements in the estimates could be obtained using this method.

Using the backwards elimination method, previously introduced, it was demonstrated that the mixed SARMA models are significantly more parsimonious than the pure SAR models. Furthermore, the visual quality of the mixed SARMA models is a significant improvement over pure SAR models - at least in some cases. It was demonstrated that, whereas the texture models obtained, using pure SAR models, was often non-stationary, the mixed SARMA models, estimated for the same texture, was stationary.

Causality is not, in general, a natural property for spatial data. Therefore it is of interest to study non-causal models. It was demonstrated that the Least Squares estimator gives useless results. Kashyap & Chellappa (1983) introduced an approximative ML estimator for toroidal SAR models. Using simulation and estimation, it was demonstrated that this estimator is also severely biased when the texture considered is in the proximity of non-stationarity. Therefore the exact ML estimator was implemented employing a quasi-Newton method. Although this method yields very good results, it is obviously a computationally very demanding task to obtain the ML estimates through non-linear optimization of the likelihood function. A new approximative ML estimator is proposed. Simulation studies shows that phase-transition phenomena, not unlike the phenomena encountered for binary Markov random fields, occur for non-causal SAR models.

The parametric spectral estimators based on the (causal and non-causal) SARMA models were used to estimate harmonics superimposed on white noise. It was demonstrated that good results may be obtained using this type of models. In particular the averaged quarter-plane models yielded very good results.

Finally, spectral textural features were extracted from the estimated power spectrum, using parametric as well as non-parametric methods. Two new features were introduced. All features were used to classify a selected database of stochastic features using the CART classifier. This experiment demonstrated that very good classification results may be obtained using spectral texture features. Furthermore, the results are improved when parametric spectral estimator are used. Features extracted from the Fourier power spectrum was used to measure the effect of enzymatic treatment of textiles. It was demonstrated that good features could be constructed and that much insight into the nature of the problem is gained by considering spectral properties.

An extensive collection of software has been developed during the course of this work. The software developments, directly related to the problems considered in this thesis, are listed in appendix A.

Appendix A

Developed software

This chapter serves as partial documentation for the software developed in relation to the problems considered in this thesis. The software developed comply with the *HIPS* standard and are all written in standard *C*.

A.1 Fourier methods

A.1.1 peakfind

The *peakfind* program is designed used to find peaks in the power spectrum.

Use: peakfind [-k kernelsize] [-r reject] [-R min. peakdistance from DC] [-n number of peaks to be detected] [-H] [-U] < input > output

The algorithm used to accomplish this is described in Grunkin (1993) and is based on a polynomial approximation in a small window. The window size is specified using the -*k* switch. When a candidate peak is located it is tested,

using an F-test, whether it is a peak or not. The conservativeness of this test is selected using the *-r* switch. Typically, it is not desirable to locate the DC value, or peaks close to the DC value. The minimum distance (from the DC value) in which peaks should be located is specified using the *-R* switch. Often it is of interest to locate only the 'most important' peaks - i.e. the first order harmonics. This is done automatically as the algorithm 'ranks' the identified peaks. The number of peaks to be identified is selected using the *-n* switch. If the *-H* option is set, only half of the powerspectrum is searched for peaks (which makes sense as the powerspectrum is symmetric). The *-U* option makes the program print a help message. The output image is binary where located peaks are marked with a white pixel. A list of the located peak is printed to *stderr*.

A.1.2 window

The *window* program is used to design the (square) windows described in section 3.4.

Use: window [-N size] [-T truncation point] [-a alfa]
[-sep] [-w window-id] [-U] > output

The size of the window is chosen using the *-N* switch. A square window of the specified size is create. The truncation point of the window is selected using the *-T* switch. By default the truncation point is equal to the window size. By setting the *-sep* option a separable window is designed. By default the window have circular support. The type of window is selected using the *-w* switch:

w=0: Truncation window.

w=1: Hamming window.

w=2: Kaiser-Bessel window.

w=3: Fejer-Bartlett (triangle) window.

w=4: Tukey-taper window.

w=5: Hanning window.

If a Kaiser-Bessel window is chosen, the *-a* switch is used to control the trade-off between mainlobe width and sidelobe height (see section 3.4). If the Tukey-taper window is chosen, the *-a* switch is used to control the amount of data to be tapered. The *-U* option makes the program print a help message. The output of the program is a float image which may be used multiplicatively to the data.

A.2 Causal SARMA models

Some of the programs described in this section is also applicable for non-causal SAR models.

A.2.1 mkmodel

This program is used to specify models of a given order in terms of an area of support and a given geometrical shape - i.e. a (rotated) rectangle or a circle. The notation used in the high-level specification of models comply with the notation presented in chapter 2 (for the support) and chapter 4 (for the order). A low-level specification may also be given in terms of the specific pixel sites and associated parameter values. In each case the specification is translated into a format which is read by the library of programs described subsequently.

At first sight this may seem a rather cumbersome way of dealing with models. In practical work, however, it gives the user the freedom to experiment with models, excluding parameters etc. A model may be estimated and subsequently used for parametric spectral estimation, simulation, filtering and so on.

Below the syntax of the specification language is described using a simple BFN grammar. The terminal symbols are all printed in capital **boldface**. The implementation of this small and simple grammar are carried out using *YACC* and *LEX* which are found as standard software on most standard *UNIX* platforms.

modspec	→	speclist .
speclist	→	speclist spec ;
	|	spec ;
spec	→	**ID** = def
def	→	highlevel
	|	lowlevel
highlevel	→	shape **AND** support
lowlevel	→	lowlevel **N** (**INT** , **INT** , **REAL**)
	|	**N** (**INT** , **INT** , **REAL**)
shape	→	**RECT** (**REAL** , **REAL** , **REAL**)
	|	**ELLIP** (**REAL** , **REAL** , **REAL**)
support	→	nshpexp
	|	qpexp
	|	**NC**
nshpexp	→	**NSHP** (pmexp , pmexp)
qpexp	→	**QP** (pmexp , pmexp)
pmexp	→	< + >
	|	< − >
	|	+
	|	-

The input to the program is an *ASCII* file with one or more model specifications complying with the defined grammar. The output is a number of pixel sites, constituting the neighbor set of the model, and associated parametervalues.

Example

An ASCII file, "myfile", is edited and contain the specification of two quarter-plane models:

```
qp1 = ELLIP(3.0,3.0,0.0) AND QP(+,+);
qp2 = ELLIP(3.0,3.0,0.0) AND QP(-,+);
.
```

These two models are translated using the command

```
mkmodel < myfile
```

and two new files, qp1 and qp2, are constructed. These files are used as command line parameters to other programs. The specified models are quarter plane models with parameters confined to a circle of radius 3.0.

A.2.2 showmod

This program is used to inspect previously specified models e.g. when the parameters have been estimated. Furthermore the program is transforming any given model, f.ex. a high-level specification, into a low-level specification which may subsequently be manipulated and translated again using the **mkmodel** program. The program is also used to inspect the 'shape' of a given model. If f.ex. the **prunesar** program has been used, the **showmod** program may be used to see how the remaining parameters are distributed.

Use: showmod [-l] -m mod
showmod [-s] -m mod

Example

A model is specified, estimated and inspected using the following sequence of commands:

```
mkmodel < myfile
sarmaest -ar qp1 < texture.hips
showmod -m qp1
```

The output of showmod is written to *stderr*, and may be piped to a file. Then the model may be manipulated in different ways. Some parameters may be deleted, inserted or altered and the file is translated again using **mkmodel**.

A.2.3 gssarmasim

This program is used to simulate realizations of interaction models specified using the **mkmodel** program. This is of interest e.g. when a model is identified and estimated for a given texture. The method implemented here is the Gauss-Seidel method described in section 4.2.

Use: gssarmasim [-ar armod] [-ma mamod]
-size N [-it N_{it}] [-stop delta] [-sdev epsilon] [-newseed] > output

The program simulates mixed SARMA models and thus either an AR part, an MA part or both must be specified. The size, N, of the simulated square image must be specified. Either the number of iterations or a stop criterion *delta* must be provided. If both are specified the program terminates when one of these to criteria are met. The texture is simulated on basis of a white noise process generated by a call to an *IMSL* random generator. The standard deviation of the noise process defaults to one (1.0) but can be specified using the *-sdev* switch. If different realizations are wanted, the *-newseed* option is used. By default the same realization is realized every time the program is called with the same set of parameters.

Example

A model is specified and used to model a sample of a given texture. In order to assess the visual quality of the fitted model a sample is simulated. This is done with the sequence of commands given below:

```
mkmodel < mymodel
sarmaest -ar armodel -ma mamodel -grad 0.001 -delta 0.0001 < mytexture
gssarmasim -size 128 -it 20 -stop 0.000001 -ar armodel -ma mamodel > sample
```

A.2.4 foursim

The function of this program is the same as the **gssarmasim** program and it may be called with the same parameters. The only difference is the method employed for simulation. In this program the frequency based method, described in section 4.2 is used. The frequency response of the specified model is calculated and multiplied with the Fourier transform of the generated noise sequence. The output is the inverse Fourier transform of the product.

A.2.5 sarmaest

This program is used to estimate previously specified models on the basis of a given input image. The program is capable of estimating pure SAR or SMA models and mixed SARMA models. The pure SAR models are estimated using the Least Squares normal equations (See section 4.3.1. The SMA and mixed SARMA models are estimated using conditional non-linear Least Squares estimates employing a quasi-Newton method (implemented in *IMSL*)

(See section 4.3.2). A logfile (*sarmaest.log*) is generated containing initial values of the parameters and the parameters obtained as a result of the estimation. The variance of the parameters are given as well as the test statistic for the hypothesis that the parameter is zero against all alternatives. If SMA or mixed SARMA models are estimated the number of function calls and gradient evaluations of the objective functional is given. The residual variance and the AIC and BIC is calculated. The parameters of the input model is updated to the value of the estimates.

Use: sarmaest [-ar armod] [-ma mamod]
-delta delta [-grad grad] [-cda | -bda | -fda] < input

The *-delta* switch must be used to specify the $\delta > 0$ used in the gradient approximation chosen with the *-cda*, *-bda* or *-fda* switch. The *-cda* (Central Difference Approximation) is chosen by default. A stopcriterion may be provided using the *-grad* switch. When the 2-norm of the gradient is less than specified, the iteration stops.

A.2.6 backforec

The *backforc* program is implementing the back forecasting algoritm proposed in section 4.3.2.

Use: backforec [-h horizon] [-ar armod] [-ma mamod]
-delta delta [-grad grad] [-cda | -bda | -fda] < input

The program is used exactly as *sarmaest*. The only difference is that the *-h* switch may be used to control the forecasting horizon. The default value is set to 10.

A.2.7 prunesarma

The *prunesarma* program is implementing the backwards elimination algorithm described in section 4.3.3. and is used to identify an appropriate model order.

Use: prunesarma [-ar armod] [-ma mamod] [-p searclength] < input

The program takes the model, specified using the *-ar* and *-ma* switches, and eliminates the parameters iteratively. The program terminates when none of the parameters in the searchlist can be eliminated without increasing the BIC. The length of the searchlist is specified using the *-p* switch. The program produces a log-file *prunesarma.log* that allows the user to see which parameters were eliminated and in which order. If the program attempted to eliminate a parameter, and failed, this is also reported in the log-file. In the end of the log-file the outline of the full model is printed. The eliminated parameters are marked with a '-', the remaining parameters with '+' and the origin with 'o'. On termination, the models are updated with the remaining parameters and the corresponding estimates.

A.2.8 sarmaspec

The *sarmaspec* program is used to obtain the parametric spectrum for an estimated model.

Use: sarmaspec [-ar armod] [-ma mamod] [-size imsize] [-start 0.0 | -0.5] > output

The model is specified using the *-ar* and *-ma* switches. The resolution of the spectrum is set, using the *-size* switch. By default the DC value is located at the center of the image. By using *-start* -0.5 the DC value may be placed

in the upper left corner (as the standard transform). The output image is in float format.

A.2.9 saraver

The *saraver* program is used to obtain the averaged QP parametric spectrum described i chapter 6.

Use: saraver [-ar1 armod1] [-ar2 armod2] [-size imsize] [-start 0.0 | -0.5] > output

The program is used as the *sarmaspec* program. The difference is that both models, specified using the *-ar1* and *-ar2* switches, is assumed to be autoregressive models, previously estimated using the *sarmaest* program. There is, in fact, no restrictions as to the type of autoregressive models given as input. Thus an averaged NSHP spectrum may also be obtained.

A.2.10 sarmafilter

The *sarmafilter* program is used to obtain the residuals of a given model.

Use: sarmafilter [-ar armod] [-ma mamod] < input > output

When the model, specified using the *-ar* and *-ma* switches, is identified and estimated, the residuals may be obtained when the program is run with the input image used to estimate the parameters. The output image is in float format.

A.2.11 exactpeak

The *exactpeak* program is used to find local maxima of parametric spectral at sub-pixel precision. This is done using a quasi-Newton method for non-linear optimization. The present version of the program can only be used for pure SAR models. The program can also be used for the averaged QP SAR model.

Use: exactpeak -ar1 armod1 [-ar2 armod2] -fr f_r -fc f_c -delta δ

A model is specified using the *-ar1* switch. If an additional model is specified using the *-ar2* switch it is assumed that the models should be averaged as described in e.g. chapter 6. There may be several local maxima in a parametric spectrum. Therefore starting values must be provided to the program to make sure that the desired maximum is found. Starting values may be found using e.g. the *peakfind* program. A CDA gradient approximation is used. For this approximation δ must be specified. The results of the optimization is written to *stdout*.

A.3 Non-causal SAR models

The models, used as input to these programs, are specified using the *mkmodel* program described previously. Similarly the *showmod* program may be used to inspect the models.

A.3.1 ncsarest

The *ncsarest* implements a non-linear optimization of the likelihood function for toroidal SAR models thus providing the exact ML estimates. The method is described in chapter 5.

Use: ncsarest -ar armod -init -ls -delta δ -grad stop-crit < input

The model is specified using the *-ar* switch. If other initial values than the default values (all equal to 0.1) should be used, the *-init* option is used. When this flag is set, the parameter values of the input model is used. Often, a suitable alternative to the default values are the approximative ML estimates obtained using the *chellappa* program. The program terminates when the 2-norm of the gradient is less than the stopcriterion specified using the *-grad* switch. If only the LS estimates are desired the *-ls* option is used.

A.3.2 chellappa

The *chellappa* program is used to obtain the approximative ML estimates. The method is described in chapter 5.

Use: chellappa -ar armod -stop stop crit. -ls -trace < input

The model is specified using the *-ar* switch. The LS values are used as initial values. If only the LS estimates are desired the *-ls* option is used. The program terminates when the 2-norm of the change of parameters is less than the stop criterion specified using the *-stop* switch. The -trace option is used to see how the parameters are changing, iteration by iteration.

A.4 Spectral texture features

A.4.1 feweri

This program is used to calculate the average power in rings and wedges of a given estimate of the power spectrum as described in chapter 7.

Use: feweri -nrings number of rings -nwedges number of wedges -header < input

The input image is assumed to be an estimate of a power spectrum in float format. The number of rings and wedges to be calculated is specified using the *-nrings* and *-nwedges* switches. The *-header* option makes the program print out which rings and wedges are being calculated.

A.4.2 gaussfit

The *gaussfit* program is used to calculate the global features described in chapter 7.

Use: gaussfit < input

The input image is assumed to be an estimate of a power spectrum in float format. The program prints the following features to *stdout*:

GF1: The correlation feature.

GF2: First principal component λ_1.

GF3: Second principal component λ_2.

GF4: The fraction $\frac{\lambda_2}{\lambda_2}$.

GF5: Principal direction.

A: The Liu & Jernigan anisotropy feature.

GF6: Alternative anisotropy feature.

GF7: Moment of inertia.

GF8: Entropy.

References

Abraham, B. & Ledolter, J. (1983). *Statistical methods for forecasting.* J. Wiley & Sons, New York. 445 pp.

Bartlett, M. S. (1953). *An Introduction to Stochastic Processes with Special Reference to Methods and Applications.* Cambridge University Press.

Besag, J. & Moran, P. (1975). On the estimation and testing of spatial interaction in gaussian lattice processes. *Biometrika*, **62**(3), 555–562.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, **36**, 192–236.

Besag, J. (1977). Efficiency of pseudolikelihood estimation for simple gaussian fields. *Biometrika*, **64**(3), 616–618.

Box, G. E. P. & Jenkins, G. M. (1976). *Time Series Analysis - Forecasting and control.* Holden-Day, San Francisco. 575 pp.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees.* Wadsworth & Brooks/Cole advanced books & software, Monterey, California. 358 pp.

Brodatz, P. (1966). *Textures - A Photographic Album for Artists and Designers.* Dover, New York.

Burg, J. P. (1975). *Maximum Entropy Spectral Analysis.* Ph.d dissertation, Stanford University.

Burt, P. (1981). Fast filter transforms for image processing. *Computer Graphics and Image Processing*, **16**, 20–51.

Cadzow, J. A. & Ogino, K. (1981). Two dimensional spectral estimation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **29**(3), 396–401.

Carstensen, J., Grunkin, M., & Conradsen, K. (1991). Measurement of enzymatic treatment effect on textile using digital image analysis..

Carstensen, J. (1988). Stochastic texture models. Master's thesis, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, Lyngby. In Danish.

Carstensen, J. (1992). *Description and Simulation of Visual Texture.* IMSOR ph.d thesis # 59, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, Lyngby. 234 pp.

Chatfield, C. (1979). Inverse autocorrelations. *Journal of the Royal Statistical Society, Series A*, **142**, 363–377.

Chellappa, R. (1985). Two-dimensional discrete gaussian Markov random field models for image processing. In Kanal, L. & Rosenfeld, A. (Eds.), *Progress in Pattern Recognition 2*, pp. 79–112. North-Holland, Amsterdam.

Chien, H. (1981). *Two-Dimensional Spectral Esitimation from Autoregressive Models with Various Areas of Support.* Ph.d dissertation, University of Rhode Island.

Conners, R. & Harlow, C. (1980). A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**(3), 204–222.

Cooley, J. W. & Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Math. Computation*, **19**, 297-301.

Dennis, J. E. & Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, New Jersey. 378 pp.

Dudgeon, D. E. & Mersereau, R. M. (1984). *Multidimensional Digital Signal Processing*. Prentice-Hall. 400pp.

Eklundh, J. O. (1979). Note, on the use of fourier phase features for texture discrimination. *Computer Vision, Graphics and Image Processing*, **9**, 199-201.

Ersboll, B. (1989). *Transformations and classifications of remotely sensed data*. Ph.D. thesis, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, Lyngby. 297 pp.

Grunkin, M. (1993). Restoration of bandlimited images corrupted by periodic noise. *In submission.*

Haralick, R. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, **67**(5), 786-804.

Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, **66**(1), 51-83.

Heideman, M. (1984). Gauss and the history of the fast fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **1**, 14-21.

Huang, T. S. (1972). Two-dimensional windows. *IEEE Trans. Audio and Electroacoustics*, **20**(1), 88-90.

Jain, A. K. (1981). Advances in mathematical models for image processing. *Proceedings of the IEEE*, **69**, 502-528.

Jorgensen, J. F. (1991). Restoration of stm-images and lattice measurements..

Kaj Madsen, P. H. & Hansen, P. C. (1991). *Robust c Subroutines for Non-Linear Optimization.* Institute for Numerical Analysis. 61 pp.

Kartikeyan, B. & Sarkar, A. (1991). An identification approach for 2d autoregressive models in describing textures.. *Computer Vision, Graphics and Image Processing*, **53**(2), 121–131.

Kashyap, R. & Chellappa, R. (1983). Estimation and choice of neighbors in spatial interaction models of images. *IEEE Transactions on Information Theory*, **29**(1), 60–72.

Kashyap, R. L. & Eom, K.-B. (1988). Robust image modeling techniques with an image restoration application. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **36**(8), 1313–1325.

Kashyap, R. L. & Khotanzad, A. (1986). A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(4), 472–481.

Kashyap, R., Chellappa, R., & Khotanzad, A. (1982). Texture classification using features derived from random field models. *Pattern Recognition Letters*, **1**(1), 43–50.

Kashyap, R. (1980). Univariate and multivariate random field models for images. *Computer Graphics and Image Processing*, **12**, 257–270.

Kashyap, R. L. (1981). Analysis and synthesis of image patterns by spatial interaction models. In Kanal, L. N. & Rosenfeld, A. (Eds.), *Progress in Pattern Recognition*, pp. 149–186. North-Holland, Amsterdam.

Kashyap, R. L. (1984). Characterization and estimation of two-dimensional arma models. *IEEE Transactions on Information Theory*, **30**(5), 736–745.

Kay, S. M. (1988). *Modern Spectral Estimation.* Prentice-Hall Signal Processing Series, Englewood Cliffs, New Jersey. 543 pp.

Knut Conradsen, Jan Gunulf, O. H. & Nilsson, G. (1982). The application of remote sensing in mineran exploration..

Lim, J. S. & Malik, N. A. (1981). A new algorithm for two-dimensional maximum entropy power spectrum estimation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **29**, 401–413.

Lim, J. S. (1990). *Two-Dimensional Signal and Image Processing*, Vol. 1. Prentice-Hall International Editions. 694 pp.

Liu, S. & Jernigan, M. (1990). Texture analysis and discrimination in additive noise. *Computer Vision, Graphics and Image Processing*, **49**, 52–67.

Madsen, H. (1989). *Time Series Analysis (in dansih).* Institute of Mathematical Statistics and Operations Research. 345 pp.

Mao, J. & Jain, A. (1992). Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, **25**(2), 173–188.

Marzetta, T. L. (1980). Two-dimensional linear prediction: autocorrelation arrays, minimum-phase prediction error filters, and reflection coefficient arrays. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **28**, 725–733.

O'Conner, B. T. & Huang, T. S. (1981). Stability of general two-dimensional filters. In Huang, T. S. (Ed.), *Two-Dimensional Digital Signal Processing I*, pp. 85–154. Springer Verlag, Berlin.

Oppenheim, A. V. & Schafer, R. W. (1975). *Digital Signal Processing.* Prentice-Hall. 585 pp.

Ord, K. (1975). Estimation methods for models of spatial interaction. *Journal of the American Statistical Association*, **70**(349), 120–126.

Papoulis, A. (1962). *The Fourier Integral and its Applications.* McGraw-Hill Book Company. 318 pp.

Priestly, M. B. (1981). *Spectral Analysis and Time Series.* Academic Press, London. 890 pp.

Rao, A. (1990). *A Taxonomy for Texture Description and Identification.* Springer-Verlag. 197 pp.

Ripley, B. (1987). *Stochastic simulation.* J. Wiley & Sons. 237 pp.

Ripley, B. (1990). Thoughts on pseudorandom numbers. *J. Comout. Appl. Math.*, 165–172.

Sharma, G. & Chellappa, R. (1985). A model-based approach for estimation of two-dimensional maximum entropy power spectra. *IEEE Transactions on Information Theory*, **31**(1), 90–99.

Sharma, G. & Chellappa, R. (1986). Two-dimensional spectrum estimation using noncausal autoregressive models. *IEEE Transactions on Information Theory*, **32**(2), 268–275.

Tjøstheim, D. (1981). Autoregressive modeling and spectral analysis of array data in the plane. *IEEE Transactions on Geoscience and Remote Sensing*, **19**(1), 15–24.

Tomita, F. & Tsuji, S. (1990). *Computer Analysis of Visual Texture.* Kluwer Academic Publishers. 173 pp.

Tou, J. T. (1980). Pictorial feature extraction and recognition via image modelling. *Computer Graphics and Image Processing*, **12**, 376–406.

V. Novikov, V. B. & Valteris, S. (1985). Generation of markov random fields with given parameters (russian). *Statisticeskije Problemy Upravlenija*, **70**, 96–101.

van Gool, L., Dewaele, P., & Oosterlinck, A. (1985). Texture analysis anno 1983. *Computer Vision, Graphics and Image Processing*, **29**, 336–357.

Welch, P. D. (1967). The use of fft for the estimation of power spectra: a method based on time averages over short modified periodograms. *IEEE Trans. Audio and Electroacoustics*, **15**, 70–73.

Weszka, J., Dyer, C., & Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics*, **6**(4), 269–285.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, **41**, 434–449.

Wilson, R. & Spann, M. (1988). Finite prolate spheroidal sequencies and their applications ii: image feature description and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**, 193–203.

Woods, J. W. (1972). Two dimensional discrete markovian fieds. *IEEE Transactions on Information Theory*, **18**, 232–240.

Zhang, X.-D. & Cheng, J. (1991). High resolution two-dimensional arma spectral estimation. *IEEE Transactions on Signal Processing*, **36**(3), 765–770.

Zou, L. & Liu, B. (1988). On resolving two-dimensional sinusoids in white noise using different spectral estimates. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **36**(8), 1338–1350.

Index

Ph. D. theses from IMSOR

1. **Sigvaldason, Helgi.** (1963). *Beslutningsproblemer ved et hydro-termisk elforsyningssystem.* 92 pp.

2. **Nygaard, Jørgen.** (1966). *Behandling af et dimensionerings problem i telefonien* 157 pp.

3. **Krarup, Jakob.** (1967). *Fixed-cost and other network flow problems as related to plant location and to the design of transportation and computer systems.* 159 pp.

4. **Hansen, Niels Herman.** (1967). *Problemer ved forudsigelse af lydhastighed i danske farvande. Analyse af et stokastisk system. Del 1: Tekst. Del 2: Figurer og tabeller.* 104 pp. + 95 pp.

5. **Larsen, Mogens E.** (1968). *Statistisk analyse af elementære kybernetiske systemer.* 210 pp.

6. **Punhani, Amrit Lal.** (1968). *Decision problems in connection with atomic power plants.* 133 pp.

7. **Clausen, Svend.** (1969). *Kybernetik, systemer og modeller.* 205 pp.

8. **Vidal, R.V. Valqui.** (1970.) *Operations research in production planning. Interconnections between production and demand. Volume 1-2.* 321 pp.

9. **Bilde, Ole.** (1970). *Nonlinear and discrete programming in transportation, location and road design. Volumes 1-2.* 291 pp.

10. **Rasmusen, Hans Jørgen.** (1972). *En decentraliseret planlægningsmodel.* 185 pp.

11. **Dyrberg, Christian.** (1973). *Tilbudsgivning i en entreprenør virksomhed.* 158 pp.

12. **Madsen, Oli B.G.** (1973). *Dekomposition og matematisk programmering.* 271 pp.

13. **Dahlgaard, Peter.** (1973). *Statistical aspects of tide prediction. Volume 1. Volume 2: Figures and tables.* 202 pp. + 170 pp.

14. **Spliid, Henrik.** (1973). *En statistisk model for stormflodsvarsling.* 205 pp.

15. **Pinochet, Mario.** (1973). *Operations research in strategic transportation planning. The decision process in a multiharbour system.* 374 pp.

16. **Christensen, Torben.** (1973). *Om semi-markov processer. Udvidelser og anvendelser inden for den sociale sektor.* 239 pp.

17. **Jacobsen, Søren Kruse.** (1973). *Om lokaliseringsproblemer, modeller og løsninger.* 355 pp.

18. **Marqvardsen, Hans.** (1973). *Skemalægning ved numerisk simulation.* 222 pp.

19. **Mortensen, Jens Hald.** (1974). *Interregionale godstransporter. Teoridannelser og modeller.* 223 pp.

20. **Severin, Juan Melo.** (1974). *Introduction to operations research in systems synthesis. A chemical process design synthesis application.* 249 pp.

21. **Spliid, Iben & Uffe Bundgaard-Jørgensen.** (1974). *Skitse til en procedure for kom-*
22. *munalplanlægning.* 544 pp.

23. **Mosgaard, Christian.** (1975). *International planning in disaster situations.* 187 pp.

24. **Holm, Jan.** (1975). *En optimeringsmodel for kollektiv trafik.* 246 pp.

25. **Jesson, Pall.** (1975). *Stokastisk programmering. Del 1: Modeller. Del 2: Metodologiske overvejelser og anvendelser.* 333 pp.

26. **Iversen, Villy Bæk.** (1976). *On the accuracy in measurements of time intervals and traffic intensities with application to teletraffic and simulation.* 202 pp.

27. **Drud Arne.** (1976). *Methods for control of complex dynamic systems. Illustrated by econometric models.* 209 pp.

28. **Togsverd, Tom.** (1976). *Koordinering af kommunernes ressourceforbrug.* 295 pp.

29. **Jensen, Olav Holst.** (1976). *Om planlægning af kollektiv trafik. Operationsanalytiske modeller og løsningsmetoder.* 321 pp.

30. **Beyer, Jan E.** (1976). *Ecosystems. An operational research approach.* 315 pp.

31. **Bille, Thomas Bastholm.** (1977). *Vurdering af Egnsudviklingsprojekter. Samspil mellem benifit-cost analyse og den politiske vurdering i en tid under forandring.* 260 pp.

32. **Holst, Erik.** (1979). *En statistisk undersøgelse af tabletserier.* 316 pp.

33. **Aagaard-Svendsen, Rolf.** (1979). *Econometric methods and Kalman filtering.* 300 pp.

34. **Hansen, Steen.** (1979). *Project control by quantitative methods.* 230 pp.

35. **Scheufens, Ernst Edvard.** (1980). *Statistisk analyse og kontrol af tidsafhængige vandkvalitetsdata.* 152 pp.

36. **Lyngvig, Jytte.** (1981). *Samfundsøkonimisk planægning.* 252 pp.

37. **Troelsgård, Birgitte.** (1981). *Statistisk bestemmelse af modeller for rumlufttemperatur.* 213 pp.

38. **Raft, Ole.** (1981). *Delivery planning by modular algorithms.* 220 pp.

39. **Jensen, Sigrid M.** (1981). *Analyse af interregionale togrejser. + Figurer og appendices.* 212 pp. + 174 pp.

40. **Ravn, Hans.** (1982). *Technology and underdevelopment. The case of Mexico.* 376 pp.

41. **Hansen, Sten.** (1983). *Phase-type distributions in queueing theory.* 209 pp.

42. **Ferreira, Jose A.S.** (1984). *Optimal control of discrete-time systems with applications.* 252 pp.

43. **Behrens, Jens Christian.** (1985). *Mathematical modelling of aquatic ecosystems applied to biological waste water treatment + Appendix 1-2.* 32 pp. + 389 pp. + 180 pp.

44. **Poulsen, Niels Kjølstad.** (1985). *Robust self tuning controllers.* 240 pp.

45. **Madsen, Henrik.** (1985). *Statistically determined dynamic models for climate processes. Part 1-2.* 428 pp.

46. **Sørensen, Bo.** (1986). *Interactive distribution planning.* 253 pp.

47. **Lethan, Helge B.** (1986). *Løsning af store kombimatoriske problemer.* 173 pp

48. **Boelskifte, Søren.** (1988). *Dispersion and current measurements. An investigation based on time series analysis and turbulence models.* Risø-M-2566. 154 pp.

49. **Nielsen, Bo Friis.** (1988). *Modelling of multiple access systems with phase type distributions.* 253 pp.

50. **Christensen, John M.** (1988). *Project planning and analysis. Methods for assessment of rural energy projects in deveoping countries.* Risø-M-2706. 158 pp.

51. **Olsen, Klaus Juel.** (1988). *Texture analysis of ultrasound images of livers.* 162 pp.

52. **Holst, Helle.** (1988). *Statistisk behandling af nærinfrarøde refleksionsmålinger.* 309 pp. + app.

53. **Knudsen, Torben.** (1989). *Start/stop strategier for vind-diesel systemer.* 275 pp.

54. **Ersbøll, Bjarne Kjær.** (1989). *Transformations and classifications of remotely sensed data. Theory and geological cases.* 297 pp.

55. **Kragh, Anders Laage.** (1990). *Kø-netværksmodeller til analyse af FMS anlæg.* 205 pp.

56. **Hansen, Christian Kornerup.** (1990). *Statistical methods in the analysis of repairable systems reliability.* 56 pp.

57. **Parkum, Jens Ejnar.** (1992). *Recursive identification of time-varying systems.* 206 pp.

58. **Bilbo, Carl M.** (1992). *Statistical analysis of multivariate degradation models.* 167 pp.

59. **Carstensen, Jens Michael.** (1992). *Description and simulation of visual texture.* 234 pp.

60. **Halse, Karsten.** (1992). *Modeling and solving complex vehicle routing problems.* 372 pp.

61. **Hendricks, Elbert.** (1992). *Identification and estimation of nonlinear systems using physical modelling.* 273 pp.

62. **Windfeld, Kristian.** (1992). *Application of computer intensive data analysis methods to the analysis of digital images and spatial data.* 190 pp.

63. **Iwersen, Jørgen.** (1992). *Statistial control charts : Performance of Shewhart and Cusum charts.* 326 pp.

64. **Olsson, Carsten Kruse.** (1993). *Image processing methods in materials science.* 274 pp.

65. **Sejling, Ken.** (1993). *Modelling and prediction of load in heating systems.* 283 pp.

66. **Søgaard, Henning T.** (1993). *Stochastic systems with embedded parameter variations - applications to district heating.* 280 pp.

67. **Grunkin, Michael.** (1993). *On the analysis of image data using simultaneous interaction models.* 223 pp.