5-1-1990

A Stochastic Modeling Approach to Region-and Edge-Based Image Segmentation

Aly A. Farag
Purdue University

Edward J. Delp
Purdue University

Farag, Aly A. and Delp, Edward J., "A Stochastic Modeling Approach to Region-and Edge-Based Image Segmentation" (1990). *Department of Electrical and Computer Engineering Technical Reports.* Paper 721.
https://docs.lib.purdue.edu/ecetr/721

A Stochastic Modeling Approach to Region-and Edge-Based Image Segmentation

Aly A. Farag
Edward J. Delp

TR-EE 90-34
May 1990

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

A STOCHASTIC MODELING APPROACH TO REGION- AND EDGE-BASED IMAGE SEGMENTATION

Aly A. Farag

Edward J. Delp

Purdue University
School of Electrical Engineering
West Lafayette, Indiana 47907

TR-EE-90-34

May 1990

This is left blank

ACKNOWLEDGMENTS

The authors would like to express their appreciation for the help provided by Robert Stevenson of Purdue University and Linda Schwartz of the University of Minnesota, Duluth.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

NOTATIONS

S Set of sites in a graph (indexing set for the nodes on a graph).
$S = \{s \in S: 0 \leq s \leq N-1\}$ for a general graph.
$S = \{(x,y): 0 \leq x \leq M-1, 0 \leq y \leq N-1\}$ for an M × N lattice.
$S = \{(x,y): 0 \leq x, y \leq N-1\}$ for an N × N square lattice.

(x,y) Pixel's row and column position in a discrete image.

η Neighbor system, $\eta = \{\eta_s, s \in S\}$.

η_s Neighbor set for site (pixel location) s, $\eta_s \subset S$..

C Set of Cliques.
$C \subset S$ is formed from sites $s \in \eta_s$

Ω Sample (configuration) space.

Ξ State space. $\Xi = \{\xi: \xi \in [0, q-1]\}$
ξ a possible gray level value and q is the number of gray levels, e.g. q = 64, 128, 256 for typical images.

G_s A random variable associated with the gray level distribution at site (pixel) s.

Σ σ-algebra (Borel field).

P Probability measure.

G Random field representing the observed image process,
$G = \{G_s, s \in S\}$.

G^h Random field representing the unobserved high level

	region (labeling) process.
$\mathbf{G}^l$	Random field representing the observed low level (pixel) process.
ω	A sample image (configuration). $\omega = \{g_s, s \in S\}$.
$U(\cdot)$	Energy function.
$V_C(\cdot)$	Clique potentials.
$\mathbf{g}$	Observed image or image realization (i.e., $\mathbf{g} = \omega$).
$\underset{\sim}{\mathbf{g}}$	A lexicographic (row-concatenation) representation for the image.
$\mathbf{g}^h$	A realization (sample) of the unobserved scene (labeling) process.
$\mathbf{g}^l$	A realization (sample) of the observed region process (pixel process).
$\mathbf{f}$	A function (deterministic) describing the original (unobserved) image.
$\mathbf{h}$	Edge enhancement filter.
$\mathbf{h}_G$	Gaussian kernel.
$F(\cdot)$	The Fourier transform of $\mathbf{f}(\cdot)$.
$G(\cdot)$	The Fourier transform of $g(\cdot)$.
$H(\cdot)$	The Fourier transform of $\mathbf{h}(\cdot)$.
∇G	The Gradient of the Gaussian operator.
$\nabla^2 G$	The Laplacian of the Gaussian operator.

ABSTRACT

The purpose of image segmentation is to isolate objects in a scene from the background. This is a very important step in any computer vision system since various tasks, such as shape analysis and object recognition, require accurate image segmentation. Image segmentation can also produce tremendous data reduction. Edge-based and region-based segmentation have been examined and two new algorithms based on recent results in random field theory have been developed.

The edge-based segmentation algorithm uses the pixel gray level intensity information to allocate object boundaries in two stages: edge enhancement, followed by edge linking. Edge enhancement is accomplished by maximum energy filters used in one-dimensional bandlimited signal analysis. The issue of optimum filter spatial support is analyzed for ideal edge models. Edge linking is performed by quantitative sequential search using the Stack algorithm. Two probabilistic search metrics are introduced and their optimality is proven and demonstrated on test as well as real scenes. Compared to other methods, this algorithm is shown to produce more accurate allocation of object boundaries.

Region-based segmentation was modeled as a MAP estimation problem in which the actual (unknown) objects were estimated from the observed (known) image by a recursive classification algorithms. The observed image was modeled by an Autoregressive (AR) model whose parameters were estimated locally, and a Gibbs-Markov random field (GMRF) model was used to model the unknown scene. A computational study was conducted on images having various types of texture images. The issues of parameter estimation, neighborhood selection, and model orders were examined. It is concluded that the MAP approach for region segmentation generally works well on images having a large content of microtextures which can be properly modeled by both AR and GMRF models. On these texture images, second order AR and GMRF models were shown to be adequate.

CHAPTER 1
INTRODUCTION

1.1. Background

This research examines existing model-based image segmentation techniques and develops new techniques based on recent advances in random field theory. The focus of this study is the two-dimensional images that result from digitization of two-dimensional projections of three-dimensional scenes. Image segmentation is used to isolate objects in the discrete image from the background. The isolated objects should describe accurately the physical three-dimensional objects in the scene. Since the output of the segmentation process is used by other components of the computer vision system for interpretation and recognition, the accuracy of image segmentation affects overall computer vision system performance [BaB82].

The extensive literature on image segmentation reflects the continuing need for more accurate image segmentation techniques. In Rosenfeld's widely acknowledged annual survey (e.g. [Ros89]), a separate section is always assigned to papers dealing with edge detection and/or image segmentation. For the last twenty years, nearly fifteen percent of the annual literature on image processing presented in this survey has consistently centered on problems of edge detection and/or image segmentation. In this chapter, a few (among the many) image segmentation techniques are discussed briefly, and Section 1.4 presents the problem statement.

Haralick and Shapiro [HaS85] classified image segmentation techniques as: Measurement space guided spatial clustering, single linkage region growing schemes, hybrid linkage region growing schemes, centroid linkage region growing schemes, spatial clustering schemes, and split and merge schemes. In another classification Nevatia, 1986 [Nev86] divides image segmentation techniques into two major approaches: Edge based and region based. In edge-based methods, the local discontinuities are enhanced first and then linked to form complete boundaries. In region-based methods, regions of similar homogeneous properties are found. These regions are then used to provide object boundaries. We will adopt the classification of [Nev86] in this thesis.

1.2. Edge-based Image Segmentation

Edge detection is an image segmentation technique based on the detection of discontinuities in the pixel gray levels. An edge can be defined as a difference in the image characteristics within a local region. A line can be defined as a pair of edges of finite width with common characteristics in the region between them. A region may be considered as a finite area bounded by a closed region [GoW87]. Edges are important because most of the information in a picture lies on the boundaries between regions. Biological visual systems appear to make use of edge detection, but not of thresholding [RoK82]. The edge map (resulting from the edge detection system) constitutes the input to a number of important schemes, e.g. shape analysis [AnD88] and motion estimation [ChD88] as well as many other problems in computer vision. Edges can be classified into several types (e.g. step edges, linear edges, roof edges, texture edges, etc.). No single technique can detect all types of edges accurately, as is reflected in the wealth of edge detection literature.

Edge detection can be categorized into parallel and sequential methods [Dav75]. In parallel methods, edge points are found based on the gray level of those points and some sets of its neighbors. It is not necessary in these techniques to decide first if other sets of points lie on an edge. In sequential methods, the result at any point is dependent upon the result of the edge operator at previously examined points.

1.2.1. Parallel Edge Detection

A parallel strategy for edge detection consists of two stages: Edge element enhancement followed by edge element linking. We will, briefly, examine a number of techniques which have been widely used for edge enhancement. Figure 1.1 shows a block-diagram for a parallel edge detection

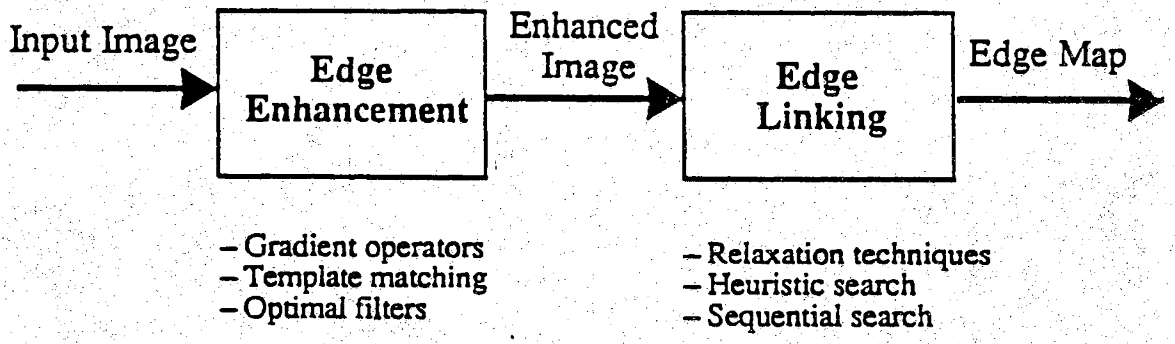


Figure 1.1. A block diagram of parallel edge detection system.

system.

1.2.1.1. Classical Gradient Operators

Classical gradient (and template matching) operators seek to isolate pixels that satisfy some heuristic observations about the edge. The basic assumption is that edges in an image can be enhanced by the gradient operation which can be approximated digitally by a variety of methods (e.g. [Rob65], [DuH73], [Kir71], and [Pre70]). A template (mask or window) is an array designed to detect some invariant regional properties [GoW87]. The template matching approach assumes ideal intensity step edges. A series of templates that produces a maximum response at the center of the edge is then formulated (e.g. Hueckel [Hue71], Fri and Chen [FrC77]). These operators are a set of masks representing discrete approximations to ideal edges of various orientations. For a certain set of templates (e.g. the Sobel template), eight masks are convolved with the image (in parallel). The maximum of the magnitudes of all responses is taken to be the edge pixel strength in the enhanced edge map. The edge orientation, at pixel location, is taken to be the compass orientation corresponding to that maximum.

Abdou and Pratt [AbP79] examined the performance of these classic edge detection operators. It is generally known that the performance of gradient operators, as well as template matching operators, are highly affected by noise. There is no systematic method for selecting the size of the operator or the template. The enhanced edges are discontinuous and irregular and spatial localization of edges cannot be accurately achieved.

Other operators have been introduced in the literature, e.g. the Nevatia and Babu operator [NeB80] and the moment-based operator of Reeves, Akey and Mitchell [ReA83]. Nalwa and Binford [NaB86] fit a series of one-dimensional surfaces to data on small windows of the image and choose the least square error surface description. The concern is to detect short edge elements (edgels), not individual edge pixels. Nalwa and Binford [NaB86] claim that their operator is superior to other operators in terms of edge localization and resolution. The surface fitting approach is the heart of many other techniques (e.g. the facet model [Har81], the Hueckel operator [Hue71], and the Prewitt operator [Pre70]). A number of studies examined the behavior of edge operators on test images at different signal-to-noise ratios as well as on actual images (e.g. Abdou and Pratt [AbP79], Haralick [Har84], Grimson and Hildreth [GrH85], Haralick [Har85], and Lyvers and Mitchell [LyM88]).

All of these methods can only enhance (emphasize) the edge information; the edge contours then are obtained from the enhanced edges by a variety of techniques (e.g. [Pra78] and [RoK82]). Thresholding is the most commonly used technique in almost all gradient-type edge operators reported (e.g. [AbP79], [Can83]). Relevant issues with threshold methods are selection of the threshold, and the discontinuities (incomplete edges) in the resulting edge map. A threshold can be selected from the Receiver Operating Characteristic (ROC) of the edge operator. The ROC for a certain operator is a plot of the detection probability versus the probability of false alarm using the threshold as a parameter [AbP79]. The value of the threshold providing optimum detection probability is used. As Eichel and Delp [EiD85b] commented, the problem of incomplete edges (after thresholding) is a result of the classical trade-off between the probability of detection and the probability of false alarm. Thus, if a high threshold value is chosen, broken contours will result. If, on the other hand, a lower value of the threshold is selected, then many false directions and multiple responses on strong edges occur.

1.2.1.2. Optimal Filtering Method

Edges that are associated with sharp transitions have high spatial frequencies. Thus an edge can be enhanced or detected by high-pass filtering. The filter is usually selected to satisfy some optimality criteria. Schanmugam et al. [ShD79] derived an optimal filter to enhance step edges. The filter can be implemented in the frequency domain and is based on the properties of the prolate spheroidal wave functions and their usage to represent band-limited signals. A similar filter was proposed by Modestino and Fries [MoF77]. Marr and Hildreth [MaH80] introduced an edge detection operator which uses the Laplacian of the Gaussian kernel ($\nabla^2 G$). This operator provides an optimization to the spatial and the frequency localization of the edges. The $\nabla^2 G$ operator is also orientation independent (isotropic). The performance of this operator has been critically evaluated by Berzins [Ber84] and Clark [Cla89].

Canny [Can83] introduced an optimum filter that provides two desired characteristics when used to enhance step edges: Good detection (i.e., high probability of detection and low probability of assigning wrong edge points as edges), and good edge localization (i.e., detected edge points should be as close to the center of the true edge as possible). A very efficient approximation to the Canny filter (for step edges) is the derivative of the Gaussian kernel. The Gaussian kernel was shown to provide almost eighty percent of the optimal filter response to step edges, and an immunity to multiple responses of about ninety percent of the optimal filter [Can86]. Intensity edges are allocated at the points of maximum gradient magnitude. The major problem with this filter, which is similar to the $\nabla^2 G$, lies in the selection of the filter spatial support.

The rationale behind the design of the majority of edge enhancement operators is that the edge information lies on local maxima of the image intensity function. Differentiation is, of course, one of the well known optimization techniques used to obtain the location of local maxima. The problem with numerical data is that the function is defined only at specific locations in the domain of support. Since differentiation does not depend on the data continuously, edge enhancement (based on differentiation) is an ill-posed problem in the sense of Hadamard [BeP88]. As a consequence, numerical differentiation may not provide a unique solution (edge definition), and it may produce an amplification of high frequency noise. To reduce the noise effect and to fill in wherever data are missing or not reliable, regularization techniques (filters) are used (e.g. [KaS88], [BeP88]). Hence, the purpose of the regularization filters in edge enhancement is to convert the ill-posed numerical differentiation problem into a well-posed problem.

The optimum filters we mentioned previously actually can be separated into two operators: A smoothing operator (filter) followed by a differentiation operator. The smoothing filter serves as a regularization filter. The Gaussian filter is only one possible choice from the class of regularization filters. The regularization issue will be examined in Chapter 4.

1.2.2. Sequential Edge Detection

A number of sequential edge detection techniques have been developed including heuristic search and dynamic programming (e.g. [Mar72], [Mar76]), and statistical techniques ([NaJ77], [NaM78], [Coo79], [CoE80], [ElS81], [ElC82], [BaE81], [EiD84], [EiD85a], [EiD85b], and [EiD88]). Sequential techniques can be applied to the original image, (e.g. [Mar72], [Mar76], [ElS81]) or to enhanced image (e.g. [AsM78], [Coo79], [EiD84]). Techniques applied on processed images often provide much better results. This is to be expected, since preprocessing is performed to emphasize the desired features and removes much of the noise in the original picture.

Martelli [Mar72] presented edge detection as a problem of finding an optimal path in a weighted graph. In his model, the properties of the edge were embedded in the structure of the graph. Each path in the graph corresponded to a possible edge in the picture and its cost was low if it corresponded to an edge. Graph searching techniques developed in artificial intelligence (e.g. the A^* algorithm) were used for the search. His technique, however, did not provide a quantitative measure for performance under noise effect. The metric used to guide the search was ad hoc. Martelli [Mar76] showed that sequential graph search provides substantial improvement in computing time as compared to dynamic programming, and that execution time depends upon the signal-to-noise ratio (SNR).

Ashkar and Modestino [AsM78] developed a sequential search technique to link edges enhanced by an optimal filter. The metric they developed to guide the search was formed of three components: A measure of local information obtained from the edge detector applied in preprocessing, a measure of curvature of typical contours to be extracted that are expected to be relatively smooth, and a measure proportional to the degree of similarity between the extracted outline and a nominal or prototype outline. The technique is problem related and cannot be compared with other techniques.

Basseville et al. [BaE81] introduced a sequential edge detection algorithm based on changes in gray levels in vertical lines of the image. Local edge detection (line-by-line) is obtained by a recursive filter which follows the slow variations of the mean gray level in the homogeneous areas. The filter also serves as a sequential detector of the jumps in the mean.

Cooper and co-workers (e.g. [Coo79], [CoE80], and [ElC82]) described a suboptimal boundary finding algorithm based on a maximum likelihood formulation introduced by Cooper [Coo79]. The original formulation [Coo79] was intended to detect blobs in enhanced images (the magnitude and phase of a gradient operator were needed). Only results of the algorithm on elementary and test images were reported.

Recently, Eichel and Delp (e.g. [EiD84], [EiD85a], [EiD85b], and [EiD88]) developed a new algorithm known as sequential edge linking (SEL). The SEL algorithm sequentially links edge features emphasized by an edge enhancement technique. The algorithm has been shown to work well on a number of test as well as real world scenes. The effect of SNR relative to the algorithm's performance has been analyzed. Briefly, the input to the SEL algorithm is an edge magnitude and angle map, and the linking process starts from a root node chosen to be of highest edge magnitude. Subsequent nodes are selected by the Stack algorithm based upon the value of the path metric. A second-order Markov model was used for the path process and the model parameters were estimated a priori using an ensemble of test as well as real scenes. Only the magnitude map enters into the metric evaluation. The angle map determines the initial search direction of the root node and provides the sign of the search at subsequent nodes. The search is terminated if the path loops on itself, if the best path intersects a previously detected path, if the best path reaches the boundary of the image, or if there is a stack overflow [EiD85b].

Sequential edge linking depends upon several elements including the model used, the search algorithm, the edge hypothesis definition, etc. The main problems with sequential edge detection techniques are first, that heuristic search techniques (e.g. [Mar72], [Mar76]) are ad hoc and problem dependent. Second, model-based techniques (e.g. [BaE81], [EiD85a]) provide good results provided that a suitable model was selected. In general, there is no easy way to determine which model should be chosen for a given problem. Third, these models depend upon the results from the preprocessing step(s) and error can propagate throughout if this step was performed incorrectly. Fourth, in general sequential techniques are computationally more expensive than parallel techniques. Fifth, essentially all model-based techniques require some thresholding on which to base the existence of edge elements. Hence the problems associated with threshold selection play a major role in the accuracy of the techniques. Major advantages of model-based sequential techniques, on the other hand, include that they enable quantitative evaluation of the algorithm performance under different SNR, and that a wide range of images can be analyzed.

1.3. Region-based Image Segmentation

In region-based segmentation, we find regions in the image that share some properties (such as color or intensity) [Nev86]. Among region-based methods are thresholding, region growing, split and merge, clustering, and texture segmentation. Image segmentation by thresholding is the simplest technique and involves the basic assumption that the objects and the background in the digital image have distinct gray level distributions. If this assumption holds, the gray level histogram contains two distinct peaks and a threshold can be easily obtained. Segmentation is performed by assigning regions having gray levels below the threshold to the background, and assigning those regions having gray levels above the threshold to the objects, or vice versa. However, due to inaccuracies in the image acquisition system (shadows, irregular lighting, dust on the camera lens, etc.) the above assumption is not often valid. A number of methods for threshold selection to handle these inaccuracies have been proposed; yet this technique is only suitable for simple images and when the imaging system is controllable. A number of surveys for threshold selection methods exist in the literature (e.g. [Wes78], and [FaD86]).

In the region growing method, we start with small regions having uniform (or nearly uniform) properties. Neighboring regions are then merged, based upon their relative properties. The process is continued until no new regions are found. Prior information about the scene might be used in deciding the merging criterion. Zucker [Zuc76a] surveys region growing methods. The major problem with this method is that it is not always easy to decide upon a merging criterion. In addition, the method usually provides a fragmented segmentation.

In split and merge methods, both region splitting and merging are possible. A criterion is set for merging and for splitting, and each region is examined relative to these criterion. Prior information can also be used in these methods. Examination of various split and merge methods may be found in the survey paper of Haralick and Shapiro [HaS85].

In clustering, the pixels are clustered in a feature space based on their properties; the clusters are not required to be spatially contiguous [Nev86]. These methods are effective when the number of clusters is known a priori [CoA79].

Finally, there exists a large number of texture based segmentation methods. This has been an active area of research and has drawn considerable attention (e.g. [Har86], [KaE89], [DeE87], [GeG86], [GeG87], [BoL88], and [LaD89]). Of particular interest to us are the statistical based techniques. These methods will be examined in Chapter 3.

1.4. Problem Statement

Several problems remain unresolved in image segmentation. Among these are:

1. Determining the selection criteria for the optimal edge enhancement filter spatial support.

2. Different techniques can be used to emphasize various types of edges in images of real scenes (e.g. step edges, linear edges, roof edges, etc.) that can be emphasized by different techniques. A better edge map (in terms of the number of edges and edge localization) should result if features corresponding to various edges were used in the segmentation algorithm. Ways to combine these features in the segmentation algorithm need to be developed.

3. The SEL algorithm provides a quantitative method for the linking of enhanced edge elements. The algorithm is based on an ARMA model for the path process and the discrete step isotropic model (D-SIM model) for the image observations. The adequacy of these models for general paths and image data has not yet been justified. The need exists, therefore, for a path metric based on general path and image models. The need also exists for an optimal set of rules that describes the prior knowledge about the scene which

should be included in the path metric.

4. Recent region-based segmentation techniques based on random field models have shown promising results in textured and noisy image segmentation, for example. A number of problems are inherent in these techniques, however. The problem of optimal model selection and parameter estimation is very wide open. Also, the computational burden required to execute such techniques must be reduced in order for them to be of any practical use.

5. Quantitative methods for the combination of edge-based segmentation techniques and region-based techniques in the segmentation algorithm need to be developed.

These problems constitute the basis of this thesis research. We will be concerned mainly with the first four problems, and they will be examined in light of recent progress in random field theory.

1.5. Thesis Outline

In Chapter 2, mathematical preliminaries of stochastic image models are introduced. This chapter provides the necessary background for the chapters that follow. In Chapter 3, a new algorithm for texture segmentation based on composite random field formulation is introduced. In Chapter 4, optimum edge enhancement filters are studied, the issue of filter spatial support is addressed, and a measure for the filter spatial support is developed. Chapter 5, examines edge linking by sequential search. Sequential search algorithms are treated and a new path metric is introduced. Various implementation strategies are also considered. Finally, Chapter 6 provides conclusions and suggestions for future research.

The main contributions of this thesis to image segmentation are the following:

1. Development of a recursive algorithm for region segmentation based upon composite random field models. The algorithm is very easy to implement and is adequate for any image size and for any number of gray levels.

2. Presenting a unified theory for edge enhancement filters and examining the issue of the spatial support of the edge enhancement filters using ideal edge models.

3. Extending the SEL algorithm to other random field models and developing techniques for parameter estimation from the actual data.

4. Introducing a new optimum path metric which is very convenient for edge linking by sequential search.

CHAPTER 2
RANDOM FIELD IMAGE MODELS: MATHEMATICAL PRELIMINARIES

2.1. Introduction

The use of random field image models in image segmentation (e.g. [GeG84], [Bes86], [GeG86], [Gra87], [DeE87], [KaE88], [LaD89], and [BoL89]) has been the subject of much recent interest. The problem of image segmentation is, usually, cast as a maximum a posteriori (MAP) estimation problem. The computational burden can now be handled using either stochastic optimization techniques, such as simulated annealing (SA), (e.g. [KiG83], [Cer85], and [GeG84]), or by a deterministic technique due to Besag [Bes86] known as iterated conditional modes (ICM).

In this chapter we examine various random field image models applicable to the region segmentation problem. The applicability of random field models to image synthesis can be attributed to the following: First, they are flexible in accommodating various probability distributions, neighbor sets, etc. In other words, a general description is possible for random field models independent of the particular probability distribution of the field variates and, independent of the neighbor system chosen. Secondly, it is possible to evaluate, quantitatively, the fitting of various random field models to the same image. Finally, a mathematical foundation forms the basis for these types of models (e.g. Whittle [Whi54], Bartlett [Bar75], Besag [Bes74], Woods [Woo72], Griffeath [Gri76], Deguchi and Morishita [DeM78], Kashyap [Kas80], [Kas81], Geman and Geman [GeG84]), which makes the analysis very tractable.

We will be concerned mainly with random field models defined on a finite support rectangular lattice. This family of models has also been used extensively in various digital image processing applications: in image coding (e.g. [DeK79], [Jai81]), texture images (e.g. [Har79], [KaC83], [CrJ83], [KaK86]), image restoration (e.g. [Woo81], [ChK82], [GeG84]), and boundary detection (e.g. [Gem87], [KaE88]).

Let the index set $S = \{s \in S: 0 \leq s \leq Q-1\}$ be a set of Q−sites, and $\eta = \{\eta_s, s \in S\}$ be a neighbor system on S. The pair (S, η) is a graph. The set S can be the set of pixel sites in a discrete image. In this case, S is a lattice representing the support of the image, that is,

$$S = \{(x,y): 0 \leq x \leq M-1, 0 \leq y \leq N-1\}$$

for an M × N image.

Definition 2.1: A neighbor system η on the set of sites S is a collection $\eta = \{\eta_s: s \in S\}$ of finite subsets of S such that:
(i) $s \in \eta_s$, and
(ii) $s \in \eta_r$ if and only if $r \in \eta_s$, $r, s \in S$.
The sites $r \in \eta_s$ are called the neighbors of $s \in S$. □

Definition 2.2: A clique C over the graph (S, η) is a subset of S which either consists of a single site or multiple sites, where every site is a neighbor of every other site in set η. □

Figure 2.1 is an illustration of possible elements of the neighbor set η and their cliques C on an M × N lattice S. More specific examples will be given when we consider Markov and simultaneous random fields on a lattice.

Let the observed image process be G which is, in general, a family of random variables indexed by the sites in the set S, that is,

$$G = \{G_s, s \in S\},$$

where G_s is a random variable associated with the gray level distribution at site $s \in S$. The state space on which G is defined is the set of all possible values that the random variable G_s might take. We will denote the state space by Ξ and, for an integer state space of size q, Ξ is expressed as follows:

$$\Xi = \{\chi: \chi \in [0, q-1]\}.$$

For a discrete-valued image, q represents the number of possible gray levels (e.g. 256). The configuration space (the set of all possible realizations, or the set of all possible discrete images) will be denoted by Ω. Ω will also be referred to as the sample space. The size of the configuration space Ω is Ξ^S.

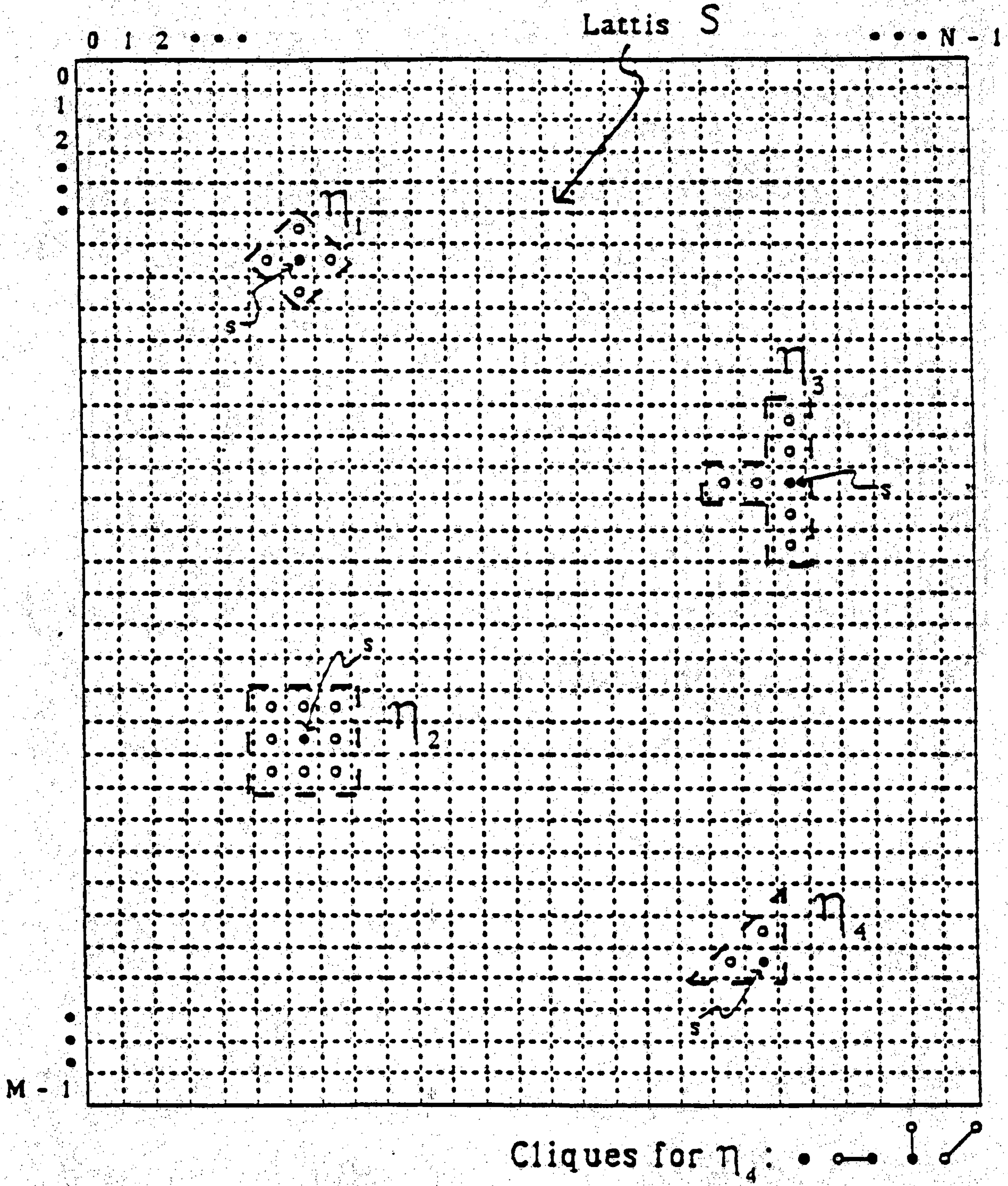


Figure 2.1. An illustration of the neighbor set and cliques over a triangular lattice.

Obviously, Ω is very large for even the modest size image; e.g. on a square lattice of size 64 × 64 and 16 gray levels, the set Ω contains $16^{64 \times 64}$ possible configurations (image realizations). Formally, we will write Ξ and Ω as follows:

$$\Xi = \{\chi: \chi \text{ is countable}\}, \ \chi \text{ can be } 0,$$

and

$$\Omega = \{\omega = (G_0, G_1, \ldots, G_{Q-1}): G_i \in \Xi, i \in [0, Q-1]\}.$$

The configuration $G = \{G_0 = g_0, G_1 = g_1, \cdots, G_{Q-1} = g_{Q-1}\}$ will be written as $\{G = \omega\}$.

Let Σ be the smallest σ-algebra with respect to which all the random variables G_s, $s \in S$ are measurable. We are now ready to define what is meant by a random field.

Definition 2.3: A random field $G = \{G_s: s \in S\}$ is given by $(\Omega, \Sigma, P, \{G_s\})$, where P is a probability measure on (Ω, Σ) such that:

$$P(G_s = g_s; s \in S) > 0$$

for all finite (non-empty) $S_1 \subset S$ and arbitrary $g_s \in \Xi$. □

Definition 2.4: A random field G is said to be finite if the set of sites S is finite. □

In a certain configuration (realization) $\omega \in \Omega$, the random field G is a deterministic function $g(\cdot)$ of the sites $\{s \in S\}$. This function might represent the gray level pixel values of an image. The deterministic function g(S) will be represented in either of two forms: As a matrix,

$$\mathbf{g} = g(S) = \begin{bmatrix} g(0,0) & g(0,1) & \ldots & g(0,N-1) \\ g(1,0) & g(1,1) & \ldots & g(1,N-1) \\ \ldots & \ldots & \ldots & \ldots \\ g(M-1,0) & g(M-1,1) & \ldots & g(M-1,N-1) \end{bmatrix}, \tag{2.1}$$

or as a unit vector (lexicographic form),

$$\underline{g} = \left[g(0,0) \ldots g(0,N-1)\ g(1,0) \cdots g(1,N-1) \ \cdots \ g(M-1,0) \cdots g(M-1,N-1) \right]^t, \quad (2.2)$$

where t denotes matrix (vector) transposition.

Among the random field models which have found most use in image processing are the class of Markov-Gibbs random field models and the class of simultaneous random field models. In this chapter we examine in detail various issues pertaining to these two classes of random field models. In particular, we examine the issues of model selection, neighbor set selection, and model coefficients (parameter) estimation.

2.2. Markov Random Fields (MRF)

Consider a graph (S, η) of Q sites, and a random field $G = \{G_s, s \in S\}$ defined on S. We provide a definition for the neighbors of a site $s \in S$ in terms of conditional probability.

Definition 2.5: A site $r(\neq s)$ is said to be a neighbor of site s $(s,r \in S)$ if and only if the conditional probability $P(G_s | G_0, G_1, \ldots, G_{s-1}, G_{s+1}, \ldots, G_{Q-1})$ is dependent upon the variable G_r [Bes74]. □

This definition implies that any system of Q sites, each with specified neighbors, clearly generates a class of valid random fields. Any member of this class is called a Markov random field [Bes74] which is formally defined as follows:

Definition 2.6: The random field G is a Markov random field (MRF) with respect to a neighbor system η if

$$P(G = \omega) > 0 \text{ for all } \omega \in \Omega, \text{ and} \quad (2.3)$$

$$P(G_s = g_s | G_r = g_r, r \neq s) = P(G_s = g_s | G_r = g_r, r \in \eta), \quad (2.4)$$

for all $s \in S$ and $\{G_s, s \in S\} \in \Omega$. □

The neighbor system η is restricted to be symmetric for a MRF.

The most general description of G is in terms of the joint probability: $P(G) = P(G_0, G_1, \ldots, G_{Q-1})$. Another description of G is in terms of the conditional probability (2.4). It should be pointed out that (2.3) defines a

Markov random field if η includes all the sites in S.

Definition 2.7: The local characteristics for a random field **G** is defined in terms of the following conditional density:

$$p(G_s|G_r, r \neq s) = \frac{p(\omega)}{\sum\limits_{g_s \in \Xi} p(\omega)} \qquad s \in S,\ \omega \in \Omega. \quad \square \qquad (2.5)$$

A very important fact with any finite random field is that the local characteristics completely define the field (see [Gri76] pp. 430-431). The local characteristics is, however, very difficult to obtain. An equivalent yet much simpler representation is possible using the so-called potentials. This representation comes from the equivalence of the finite Markov and the finite Gibbs distributions ([Spi71], [Gri76]) or via the Hammersely-Clifford Theorem [Bes74]. We will present the Gibbs random field later, but first we present a form of the Hammersely-Clifford Theorem.

Theorem 2.1 [Bes74]: Let $G = \omega = 0$ denote the all-zero observation. Assume that $P(0) > 0$. Let $U(G) = \mathit{ln}\ \{P(G)/P(0)\}$. Assume that the state space Ξ is finite. Then, the most general form U(G) may take in order to give a valid probability structure to the process G is

$$\begin{aligned} U(G) = & \sum_{0 \leq i \leq Q-1} g_i V_i(g_i) + \sum\sum_{0 \leq i < j < Q-1} g_i g_j V_{i,j}(g_i, g_j) + \\ & \sum\sum_{0 \leq i < j < k < Q-1} (g_i g_j g_k) V_{i,j,k}(g_i, g_j, g_k) + \ldots + \\ & (g_0 g_1 g_2 \cdots g_{Q-1}) V_{0,1,2,\ldots,Q-1}(g_0, g_1, \cdots, g_{Q-1}), \end{aligned} \qquad (2.6)$$

where for any $0 \leq i < j < \ldots < s < Q-1$, the function $V_{i,j,\ldots,s}$ may be non-null if and only if the sites $i, j, \cdots, s \in S$ form a clique. □

Equation (2.6) can be rewritten in the following form:

$$U(\omega) = \sum_{c \in C} V_c(\omega), \qquad (2.7)$$

where C is the set of cliques for η and $V_c(\omega)$ is a function on Ω which depends only on those values $G_s \in \omega$ for which $s \in C$. The family $\{V_c, c \in C\}$ corresponds to the potential functions in the terminology of the Gibbs random field ([Spi71], [Gri76]). The function $U(\omega)$ in (2.7) is known as the energy function [GeG84].

2.2.1. MRF on a General Graph: Auto-models

Consider a graph (S, η) having a set of Q-sites. A class of MRF models called auto-models is defined as follows:

Definition 2.8: The Markov random field G is said to follow an auto-model if the energy function in (2.7) is such that:

$$U(G) = \sum_{0 \le i \le Q-1} g_i V_i(g_i) + \sum\sum_{0 \le i < j \le Q-1} \beta_{ij} g_i g_j, \qquad (2.8)$$

where the parameters $\beta_{ij} = 0$ unless i and j are not neighbors [Bes74]. □
The above definition implies that the probability structure of the system is dependent upon contributions from the cliques of no more than two sites. We now provide two examples from the family of auto-models.

Example 2.1. Auto-binary MRF

Here the state space is binary, that is, $\Xi = \{\chi : \chi \in [0, 1]\}$ and the energy function has the following form:

$$U(G) = \sum_{0 \le i \le Q-1} \alpha_i g_i + \sum\sum_{0 \le i < j \le Q-1} \beta_{ij} g_i g_j, \qquad (2.9)$$

where α_i and β_{ij} are parameters [Bes74]. The conditional probability density function of the random variable G_i, given the values of the variables at the neighbor sites, is given by

$$p_{G_i|G_j}(g_i|g_j, j \in \eta) = \frac{\exp(g_i(\alpha_i + \sum_{0 \le i < j < Q-1} \beta_{ij} g_j))}{1 + \exp(\alpha_i + \sum_{0 \le i < j < Q-1} \beta_{ij} g_j)}. \qquad (2.10)$$

Example 2.2. Auto-normal MRF

In this class of Markov random fields, the random variables $\{G_i, i \in S\}$ are jointly Gaussian. Let G_i, $i \in S$ be Gaussian with mean μ_i and variance σ^2. The conditional and joint density functions for the field G are easily defined as follows: Over the neighborhood η, given the values of the random variables G_j, $j \in \eta$, the random variable G_i is also Gaussian with variance σ^2, but with a mean value μ given by

$$\mu = E\{G_i | G_j, j \in \eta\} = \mu_i + \sum_{0 \leq i < j \leq Q-1} \beta_{ij}(g_i - u_j). \qquad (2.11)$$

Hence, the random variable G_i has a Gaussian distribution with mean μ_i and variance σ^2, and when conditioned on the random variables in its neighborhoods, it has a Gaussian distribution with mean μ as defined in (2.11) and a variance σ^2.

From the knowledge of the probability density of the random variable G_i, $i \in S$ (the marginal probability density), that is

$$p_{G_i}(g_i, i \in S) = \frac{1}{\sigma\sqrt{2\pi}} \exp(g_i - \mu_i)^2 / 2\sigma^2, \qquad (2.12)$$

we can easily write the joint probability density for the random field **G** (by the chain rule) in the following form:

$$p_G\big(g = g_0, g_1 \cdots g_{Q-1}\big) = \frac{1}{(2\pi\sigma^2)^{-Q/2}} |\mathbf{B}|^{1/2} \cdot \exp\Big(-\frac{1}{2\sigma^2}[g - \bar{g}]^t \mathbf{B}[g - \bar{g}]\Big), \qquad (2.13)$$

where

$$\bar{g} = [\mu_0\, \mu_1 \cdots \mu_{Q-1}]^t \qquad (2.14)$$

is an $Q \times 1$ vector and **B** is an $Q \times Q$ symmetric matrix with elements b_{ij} given by:

$$b_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\beta_{ij} & \text{if } i \neq j \end{cases} \qquad (2.15)$$

Similarly, we can define other auto-models, e.g. auto-binomial, auto-Poisson, etc. (see Besag [Bes74]). This class of models will be proved useful later on as we examine sequential search techniques for edge linking.

2.2.2. MRF on a Lattice

We now restrict the graph (S, η) to be such that the sites $(x,y) \in$ S and the neighbor system η are defined on a lattice. Without loss of generality, we will consider the lattice S to be square. That is,

$$S = \{s = (x,y): 0 \le x, y \le N-1\}.$$

We will study the conditional probability form in (2.4) for the random field $G = \{G_{x,y}, (x,y) \in S\}$. The specific form of the neighbor η specifies the order of the Markov random field. The convention adopted for the set η is shown in Figure 2.2 and Figure 2.3 shows the neighbor arrangements of the set η for up to the seventh-order MRF. The forms of η, C, and $U(G)$ for the first- and second-order Markov random fields are given below.

Example 2.3. First-order MRF

$$\eta = \{(x-1,y), (x+1,y), (x,y-1), (x,y+1)\}. \tag{2.16}$$

$$C = \{[(x,y)], [(x,y), (x,y+1)], [(x,y), (x+1,y)]\}. \tag{2.17}$$

$$U(G) = \sum g_{x,y} V_{x,y}(g_{x,y}) + \sum g_{x,y} g_{x+1,y} V_{1,x,y}(g_{x,y}, g_{x+1,y}) + \sum g_{x,y} g_{x+1,y} V_{2,x,y}(g_{x,y}, g_{x,y+1}), \tag{2.18}$$

where $V_{x,y}$, $V_{1,x,y}$, $V_{2,x,y}$ are arbitrary sets of functions subject to the summability of $U(G)$.

The ranges of the summations in (2.18) are such that each clique involving at least one site internal to the system contributes a single term to the representation [Bes74]. Let the set $\{g, t, t', u, u'\}$ denote the partial realization $\{G_{x,y}, G_{x-1,y}, G_{x+1,y}, G_{x,y-1}\ G_{x,y+1}\}$, that is, let $G_{x,y} = g$, $G_{x-1,y} = t$, $G_{x+1,y} = t'$, $G_{x,y-1} = u'$, and $G_{x,y+1} = u'$. The conditional probability density structure at site (x,y) is given by

$$p(g|t, t', u, u') = \frac{\exp\left(w_{xy}(g;t,t',u,u')\right)}{\sum\limits_{(x,y) \in S} \exp\left(w_{xy}(z;t,t',u,u')\right)}, \tag{2.19}$$

where

$$w_{xy} = g\left(V_{xy}(g) + tV_{1,x-1,y}(g,t) + t'V_{1,x,y}(g,t') + uV_{2,x,y-1}(g,u) + u'V_{2,x,y}(g,u')\right) \tag{2.20}$$

and the summation extends over all possible values z at $(x,y) \in S$.

The functions $V(\cdot)$, $V_1(\cdot)$, $V_2(\cdot)$ are selected to give (2.19) an appropriate distributional form for a particular situation. For example, in a homogeneous random field, the functions $V(\cdot)$ are independent of (x,y), and

○	$(x,y-1)$	○
$(x,y-1)$	(x,y)	$(x,y+1)$
○	$(x+1,y)$	○

Figure 2.2. Convention for η.

		7	6	7		
	5	4	3	4	5	
7	4	2	1	2	4	7
6	3	1	●	1	3	6
7	4	2	1	2	4	7
	5	4	3	4	5	
		7	6	7		

Figure 2.3. Hierarchy of random field models (up to seventh-order).
(Numbers indicate the order of the model with respect to the pixel ●)

in an isotropic random field, $V_1(\cdot) = V_2(\cdot)$. We will study two cases of binary and Gaussian observations in more detail in the next subsection.

Example 2.4. Second-order MRF

The neighbor system and the cliques are as follows:

$$\eta = \{(x-1,y),\ (x+1,y),\ (x,y-1),\ (x,y+1),\ (x-1,y-1),$$
$$(x-1,y-1),\ (x+1,y-1),\ ,(x-1,y-1),\ (x+1,y+1)\}. \tag{2.21.a}$$

$$C = \{[(x,y)],\ [(x,y),\ (x,y+1)],\ [(x,y),\ (x+1,y)],$$
$$[(x,y),\ (x+1,y+1)],\ [(x,y),\ (x+1,y-1)],$$
$$[(x,y),(x+1,y),(x,y+1)],\ [(x,y),\ (x+1,y),\ (x+1,y+1)],$$
$$[(x,y),\ (x+1,y),\ (x+1,y-1)],\ [(x,y),\ (x,y+1),\ (x+1,y+1)],$$
$$[(x,y),(x+1,y),\ (x,y+1),\ (x+1,y+1)]\}. \tag{2.21.b}$$

The function $U(\cdot)$ and the conditional probability $p(g_s | g_r,\ s = (x,y) \in S,\ r = (k,l) \in \eta)$ can be written easily as in the first-order Markov by adding terms corresponding to the additional cliques.

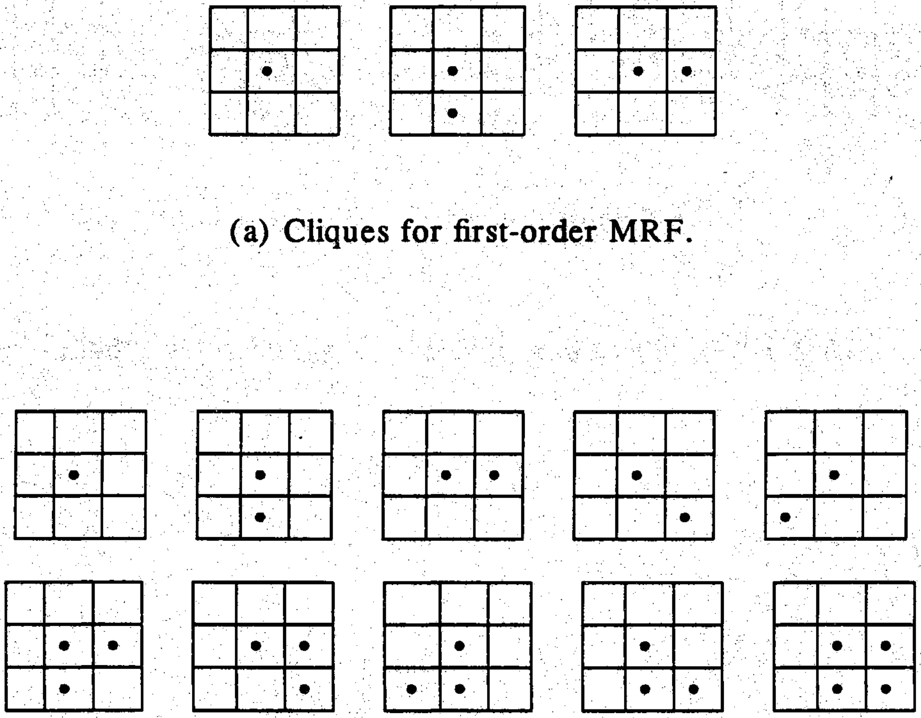
Figure 2.4 is a graphical representation for the neighbor sets and the cliques for the first- and second-order Markov random fields. Higher-order Markov can be treated similarly. We now shift to more specific Markov random field models which we will use in our work.

2.2.2.1. Binary Markov Random Field

Here the state space, as in the binary auto-models, is assumed to be formed of two possible values, 0 or 1, that is, $\Xi = \{\chi : \chi \in [0,1]\}$. The equation for the energy function in (2.6) is greatly simplified. Rewriting (2.18) for the binary state space we get for the first-order MRF

$$U(G) = \alpha \sum_{x,y \in S} g_{x,y} + \beta_1 \sum_{x,y \in S} g_{x,y} g_{x+1,y} + \beta_2 \sum_{x,y \in S} g_{x,y} g_{x,y+1}, \tag{2.22}$$

where α, β_1, β_2 are arbitrary parameters. The conditional probability density function in (2.19) becomes



(a) Cliques for first-order MRF.

(b) Cliques for second-order MRF.

Figure 2.4. Cliques for the first- and second-order MRF.

$$p(g|t,t',u,u') = \frac{\exp(g(\alpha + \beta_1(t+t') + \beta_2(u+u'))}{1 + \exp(\alpha + \beta_1(t+t') + \beta_2(u+u'))}. \qquad (2.23)$$

Similar formulas for higher-order binary MRF are easily obtained [Bes74].

2.2.2.2. Gaussian Markov Random Field

If the random variables $G_{x,y}$ defined on $(x,y) \in S$ have a Gaussian distribution, the field G is called a Gauss-Markov random field. The conditional and the joint probability density can be written as in the auto-normal case, that is, if the random variables $G_{x,y} \sim N(\mu_{xy},\sigma^2)$, then given the value of the random variable $G_{k,l}$, $(k,l) \in \eta$, the random variable $G_{x,y}$ is Gaussian with mean μ and variance σ^2, that is,

$$E\{G_{x,y}|G_{k,l} = g_{k,l},\ (k,l) \in \eta\} = \mu. \qquad (2.24)$$

The quantity μ for the first- and second-order Gauss-Markov random field is defined as follows:

$$\mu_1 = \alpha + \beta_1(g_{x-1,y} + g_{x+1,y}) + \beta_2(g_{x,y-1} + g_{x,y+1}), \qquad (2.25a)$$

and

$$\mu_2 = \alpha + \beta_1(g_{x-1,y} + g_{x+1,y}) + \beta_2(g_{x,y-1} + g_{x,y+1}) + \gamma_1(g_{x-1,y-1} + g_{x+1,y+1}) + \gamma_2(g_{x-1,y+1} + g_{x+1,y-1}), \qquad (2.25b)$$

where α, β's, and γ's are parameters.

Another representation for the Gauss-Markov random field which provides some insight about the interaction of the variables at a given site with those of neighboring sites on a lattice can be written as follows ([Woo72], [KaC83]): Let the Gauss-Markov random field G be defined on a square lattice S and a symmetric neighbor system η. The Markov property (2.4) was defined such that, for two sites s, r $\in$ S,

$$P(G_s = g_s|G_r = g_r,\ r \neq s) = P(G_s = g_s|G_r = g_r,\ r \in \eta). \qquad (2.26)$$

This definition implies that the conditional mean of the random variable G_s must be of the form (2.25a) or (2.25b), that is,

$$E\{G_s | G_r,\ s \neq\ r\} = \sum_{r \in \eta} \theta_r G_{(s+r)}, \tag{2.27}$$

where $E\{\cdot\}$ denotes the expected value. By the orthogonality principle, the difference

$$E_s = \{G_s - \sum_{r \in \eta} \theta_r G_{(s+r)}\} \tag{2.28}$$

is orthogonal to $\{G_r$, for all $r \neq\ s\}$, that is,

$$E\{E_s G_r\} = \begin{cases} 0 & \text{if } r \neq\ s \\ \nu & \text{if } r = s \end{cases}, \tag{2.29}$$

for some constant ν.

Now, assume the Gauss-Markov random field G is also zero mean (this is not a critical assumption since we can always subtract the mean value from a certain observation). The above equation implies that

$$E\{E_s | G_r,\ r \neq\ s\} = 0. \tag{2.30}$$

From (2.28) - (2.30), G_s can be written as

$$G_s = \sum_{r \in \eta_{as}} \theta_r (G_{(s+r)} + G_{(s-r)}) + E_s \tag{2.31}$$

where η_{as} is the asymmetric neighborhood of site s defined as follows [Che85]:

$$r \in \eta_{as} \rightarrow -r \notin \eta_{as}. \tag{2.32}$$

The symmetric neighborhood η is related to η_{as} by the following relation:

$$\eta = \{r: r \in \eta_{as} \cup (-r: r \in \eta_{as})\}. \tag{2.33}$$

It is easy to show that the error term E_s is not white; in fact, the correlation of the random variable E_s is as follows:

$$E\{E_s E_r\} = \begin{cases} \nu & \text{if } s = r \\ -\theta_{(s-r)}\nu & \text{if } (s-r) \in \eta \\ 0 & \text{otherwise.} \end{cases} \tag{2.34}$$

In a finite lattice S a problem arises in defining the neighbors at boundary sites. One way to handle this situation is by the so-called torus assumption ([Mor73], [KaC83]). Here the lattice S is represented by a torus, that is, the lattice is folded such that the right edge touches the left edge and the lower edge touches the upper edge. This can be described as follows: Partition the lattice S into two mutually exclusive and totally inclusive subsets S_i (Interior) and S_b (boundary). For $s \in S_i$,

$$G_s + \sum_{r \in \eta} \theta_r G_{(s+r)} = E_s, \tag{2.35}$$

and for $s \in S_b$,

$$G_s + \sum_{r \in \eta} \theta_r G_{1,(s+r)} = E_s, \tag{2.36}$$

where, for $s = (k,l)$ and $r = (x,y)$,

$$G_{1,(s+(x,y))} = \begin{cases} G_{[s+(x,y)]} & \text{for } [s+(x+y)] \in S \\ G_{[(k+x-1) \oplus N+1,\ (l+y-1) \oplus N+1]} & \text{for } [s+(x+y)] \notin S \end{cases} \cdot \tag{2.37}$$

Thus the summation in (2.37) is modulo N. Therefore, the finite lattice form Gauss-Markov random field equation can be written as follows:

$$G_s = \sum_{r \in \eta} \theta_r G_{(s \oplus r)} + E_s, \tag{2.38}$$

where $E = \{E_s;\ s \in S\}$ is a stationary Gaussian random field with a correlation structure as in (2.34). In matrix form, (2.38) can be written (for a certain realization) using the lexicographic representations (2.2) as follows:

$$H(\underset{\sim}{\theta})\underset{\sim}{g} = \underset{\sim}{e} \tag{2.39}$$

where $\underset{\sim}{g}$ and $\underset{\sim}{e}$ are $N^2 \times 1$ column vectors representing the gray level information at pixel sites and the innovation (non-white), respectively, and the matrix $H(\underset{\sim}{\theta})$ is $N^2 \times N^2$ block circulant and symmetric (see [KaC83]).

The representation in (2.39) resembles an input-output system. To ensure stationarity, it can be shown that the following condition must be satisfied [KaC83].

$$\gamma_s = (1 - 2\theta^t \phi_s) > 0 \quad \text{for all } s \in S, \tag{2.40}$$

where

$$\theta = [\theta_r,\ r \in \eta_{as}]^t, \text{ and} \tag{2.41}$$

$$\phi_s = [\cos(\frac{2\pi}{N}(s-1)^t r), r \in \eta_{as}]^t \qquad 1 = (1,1). \tag{2.42}$$

The form (2.39) enables an easier simulation of Gauss-Markov random field using the Kashyap algorithm [Kas81] instead of using the Monto Carlo methods [HaH64], for example. Also, the likelihood function is easier to write in terms of the parameters θ. This will be examined extensively when we study the issue of parameter estimation.

2.2.3. Gibbs Random Fields

The concept of Gibbs random field (GRF) has its origin from statistical mechanics, where random fields may be considered as equilibrium distributions for a variety of physical systems [Gri76]. We will examine a few properties of GRF and study the equivalence between GRF and MRF. This equivalence helps in obtaining an expression for the joint probability distribution for the MRF.

Consider a random field G defined on the graph $\{S, \eta\}$.

Definition 2.9: A Gibbs distribution relative to $\{S, \eta\}$ is a probability measure P' on Ω with the following form:

$$P'(\omega) = \frac{1}{Z} e^{-U(\omega)/T}, \tag{2.43}$$

where Z and T are constants and $U(\cdot)$ is a function [GeG84]. □

The parameter Z is called the partition function, T is called the temperature, and $U(\cdot)$ is called the energy function. The function $U(\cdot)$ has the following form:

$$U(\omega) = \sum_{c \in C} V_c(\omega), \tag{2.44}$$

where C is the set of cliques for η and $V_c(\omega)$ is a function on Ω which depends only on those values of G_s of ω for which $s \in C$. The set $\{V_c, c \in C\}$ is called the potential. The partition Z is a normalizing constant and is given by the following equation:

$$Z = \sum_{\omega} e^{-U(\omega)/T}. \tag{2.45}$$

Example 2.5. The Ising Model ([Bes74] and [GeG84])
$S = \{(x,y): 0 \leq (x,y) \leq N-1\}$, $\eta = \{(x-1,y), (x+1,y), (x,y-1), (x,y+1)\}$.
The energy for a configuration $\omega \in \Omega$ is

$$U(\omega) = \sum_{(x,y) \in S} V_{\{x,y\}}(g_{x,y}) + \sum_{(x,y) \in S} V_{\{(x,y),(x+1,y)\}}(g_{x,y}, g_{x+1,y}) + \sum_{(x,y) \in S} V_{\{(x,y),(x,y+1)\}}(g_{x,y}, g_{x,y+1}). \qquad (2.46)$$

For the binary case, that is, $\Xi = \{\chi: \chi \in [0, 1]\}$, the energy simplifies to

$$U(\omega) = \alpha \sum_{(x,y) \in S} g_{x,y} + \beta(\sum_{(x,y) \in S} g_{x,y}g_{x+1,y} + \sum_{(x,y) \in S} g_{x,y}g_{x,y+1}), \qquad (2.47)$$

for some constants α and β. It is obvious that (2.46) and (2.47) bear a great deal of similarity to (2.18) and (2.22), respectively. □

Definition 2.10: Let C be the set of cliques in S. V is called a neighbor potential if $V_A = 0$ whenever $A \notin C$. □

The Gibbs random field (GRF) is a Markov random field (MRF) if and only if the potential V is a neighbor potential [Gri76]. The equivalence between the GRF and MRF is established by the following Theorem.

Theorem 2.2 [Gri76]: Let η be a neighbor system for S. The finite Gibbs random field **G** is a MRF with respect to η if and only if $P'(\omega) = P(G = \omega)$ is a Gibbs distribution with respect to η. □

Notice that the expression of the local characteristics in terms of the potentials is the thrust of the Hammersely-Clifford Theorem we listed earlier in the study of the MRF (Theorem 2.1). The equivalence of MRF and GRF, as established by Theorem 2.2, enables the specification of the MRF by the potentials, which are much easier to specify than the local characteristics. In many cases the local characteristics is impossible to specify [GeG84]. An extensive discussion on the forms of the potential function $V(\cdot)$ and the energy $U(\cdot)$ can be found in [Spi71], [Bes74], and [Gri76]. In summary, the equivalence of the MRF and GRF on finite lattices provides us with an intuition about the functional form of the interaction between the random variables in a certain neighbor system. We now turn to the problem of parameter estimation.

2.2.4. Parameter Estimation for Finite Markov Random Fields

The two classical techniques for parameter estimation are the methods of maximum likelihood and of least squares, respectively. Neither of these methods can be applied directly in the present settings, except in the Gaussian case. The Gibbs-Markov distribution contains a complicated normalizing function which, in general, makes the likelihood function very difficult to analyze, or even to compute numerically. Even in the Gaussian case, the likelihood function is hard to compute. For these reasons, several approximate techniques for parameter estimation have been suggested in the literature for the MRF model (e.g. [Woo72], [Bes74], [Bes75], [MoB75], [Bes78], [KaC83], and [DeE87]). We will examine only a few techniques in this section.

Given a realization $G = \omega \in \Omega$, we shall assume that the conditional density function $p_{G_s|G_r}(g_s|g_r, r \in \eta)$ is fully specified in terms of a vector ψ consisting of a few unknown parameters, that is,

$$p_{G_s|G_r}(g_s|g_r, r \in \eta) = p_s(\psi). \tag{2.48}$$

The objective is to obtain a reasonable estimate of ψ from the realization $G = \omega$.

2.2.4.1. The Coding Method

The coding method [Bes74] can be applied for general graphs (S, η), that is, S need not be a rectangular lattice. The first step is to divide up the sites $s \in S$ into two groups, the one of "dependents" S_d, the other of "conditioners" S_c. In particular, let S_d denote a subset of sites $s \in S$ chosen in such a way that no two members of S_d are neighbors. Assign the color black to each site in S_d and the color white to each site in S_c. It is evident then, by the conditional probability formulations, that the set of black-site variates, given the values at the white sites, are mutually independent. Figure 2.5 shows the coding for the first- and second-order Markov. In this figure, the sites marked 1 are not neighbors; similarly the sites marked 2, 3, 4 are not neighbors in the sense of Definition 2.5 and the Markov property (2.4).

1	2	1	2	1	2	1
2	1	2	1	2	1	2
1	2	1	2	1	2	1
2	1	2	1	2	1	2
1	2	1	2	1	2	1
2	1	2	1	2	1	2
1	2	1	2	1	2	1

(a) Coding for the first-order Markov.

1	3	1	3	1	3	1
2	4	2	4	2	4	2
1	3	1	3	1	3	1
2	4	2	4	2	4	2
1	3	1	3	1	3	1
2	4	2	4	2	4	2
1	3	1	3	1	3	1

(b) Coding for the second-order Markov.

Figure 2.5. Coding for the first- and the second-order Markov random fields.

Let us define the conditional likelihood function, Γ, as follows:

$$\Gamma = \prod_{s \in S_d} p_s(\psi). \tag{2.49}$$

The log likelihood (a monotone function of its argument) is

$$ln\,\Gamma = \sum_{s \in S_d} p_s(\psi). \tag{2.50}$$

Now if the functional form of the conditional probability density $p_s(\cdot)$ is known (e.g. Gaussian, Poisson, Binomial, etc.), maximization of (2.50) can be easily carried out.

For certain η, different codings are possible and the estimates from different codings are not necessarily the same. There is no optimal way of relating the estimates from the different codings, and simple averaging might not be adequate [Bes74]. The coding method is not efficient in the sense that only a fraction of the data is used in getting the estimate. It is, however, easy to implement.

2.2.4.2. The Pseudo-Likelihood Method

To overcome the less efficient use of data in the coding technique, Besag [Bes75] proposed another alternative in terms of the so-called "Pseudo-likelihood." This technique provides an estimate $\hat{\psi}$ based upon the maximization of

$$ln\,\Gamma = \sum_{s \in S} ln\,p_s(\psi) \tag{2.51}$$

with respect to ψ.

Of course, (2.51) is not the true log-likelihood function for the sample (except in the case of complete independence). In spite of this, the pseudo-likelihood method uses the data more efficiently than the coding method. The consistency of the resulting estimate from the pseudo-likelihood method was proved by Geman and Graffigne [GeG86] (see also Graffigne [Gra87]). The technique is easy to implement for estimation in Gibbs-Markov settings and will be used in the following chapters for parameter estimation.

2.2.4.3. Estimation on Gaussian Markov Random Fields

In the Gauss-Markov random field, (2.39) an exact expression for the log-likelihood function can be written as follows ([MoB75], [KaC83]):

$$\mathit{ln}\ \mathrm{p}(\mathrm{g}|\theta,\nu) = 0.5 \sum_{\eta} \log(1 - 2\theta^t\phi_s)$$

$$- (\mathrm{N}^2/2)\log(2\pi\nu) - \frac{1}{2\nu}\underset{\sim}{\mathrm{g}}^t H(\theta)\underset{\sim}{\mathrm{g}}, \tag{2.51}$$

where θ and ϕ_s are as defined in (2.41) and (2.42), respectively. The above equation can be solved numerically to obtain the maximum likelihood estimates of the parameters θ and ν. This estimate is both efficient and consistent. It is not computationally attractive, however.

An approximate form for the MLE which is computationally simple was suggested by Woods [Woo72]. Also, the coding and the pseudo-likelihood methods of Besag ([Bes74] and [Bes75]) provide other approximations.

(i) An Approximate MLE Estimate

The approximate MLE can be written as follows [KaC83]:

$$\hat{\theta} = [\sum_{s\,\in\,S_I} q_s q_s^t]^{-1} [\sum_{s\,\in\,S_I} q_s g_s], \tag{2.52.a}$$

and

$$\hat{\nu} = \frac{1}{N^2} \sum_{s\,\in\,S_I} (g_s - \hat{\theta}^t q_s)^2, \tag{2.52.b}$$

where S_I are the interior sites and q_s is given by

$$q_s = [g_{s\oplus r},\ r \in \eta]^t. \tag{2.52.c}$$

Recall that g_s is a scalar quantity equal to the gray level value at site $s \in S$. It can be shown that $\hat{\theta}$ is asymptotically consistent [KaC83].

(ii) The Coding Technique

The maximization of the conditional likelihood function (2.49) can be performed easily for the Gaussian case. For example, in the first-order Markov $\eta = \{(x-1,y), (x+1,y), (x,y-1), (x,y+1)\}$, the coding estimate for θ can be easily shown to be

$$\hat{\theta} = [\sum_{s \in S_d} q_s q_s^t]^{-1} [\sum_{s \in S_d} q_s g_s], \tag{2.53}$$

where

$$q_s = [g_{(s+r)} + g_{(s-r)};\ r = (x,y+1),(x+1,y)]^t. \tag{2.54}$$

The dependent subset $S_d \subset S$ is formed such that:

$$S_d = (s \in S \text{ and } r \in \eta \notin S),$$

where η is the first-order Markov neighborhood of the site s in (2.16). That is, S_d is formed from S with every other site skipped.

It is easily seen from (2.51) that in the Gaussian case, pseudo-likelihood corresponds to the least squares method.

2.3. Simultaneous Random Field Models

We will study only Simultaneous models on a lattice in this section. Also, we will consider, without loss of generality, a square lattice, that is, $S = \{s = (x,y) : 0 \leq x,y \leq N-1\}$. As before, the random field **G** is described by $(\Omega, \Sigma, P, \{G_s\})$. Consider another random field $\mathbf{W} = \{\mathbf{W}_s, s \in S\}$ in which the random variables $\mathbf{W}_s$ have a distribution with zero mean and unit variance. There is no restriction on the neighbor system η, i.e. it need not be symmetric. We will study two representations from the class of simultaneous models, namely, the simultaneous autoregressive and the simultaneous autoregressive moving average models.

2.3.1. Simultaneous Autoregressive Models (SAR)

The SAR model for finite lattice is described by a difference equation as follows [Kas81]:

$$\mathbf{G}_s = \sum_{r \in \eta} \theta_r \mathbf{G}_{(s+r)} + \sqrt{\rho}\, \mathbf{W}_s, \tag{2.55}$$

where η is an arbitrary neighbor set, $\{\mathbf{W}_s\}$ is a sequence of zero mean uncorrelated random variables with unit variance, ρ is unknown parameter representing the actual noise variance, and $s = (x,y) \in S$ and $r = (k,l) \in \eta$ specify pixel locations on the image.

Using (2.2) and the toroidal assumption in (2.32), we can write the finite lattice form corresponding to (2.55) as follows:

$$G_s = \sum_{r \in \eta} \theta_r G_{(s+r)} \oplus \sqrt{\rho}\, W_s, \tag{2.56.a}$$

which can be put (for a certain realization) in the following matrix form:

$$B(\theta)\ \underset{\sim}{g} = \sqrt{\rho}\ \underset{\sim}{w}. \tag{2.56.b}$$

Again, the matrix $B(\theta)$ is block circulant involving at most N distinct blocks [Kas81].

Another form for (2.55) which is useful in studying certain properties of the SAR model is provided by the following equation:

$$A(z)\ G_s = \sqrt{\rho}\ W_s, \tag{2.57}$$

where

$$A(z) = 1 - \sum_{r \in \eta} \theta_r z^r, \tag{2.58}$$

and the factor z^r is defined such that if $r = (x, y)$, $z^r = z_1^x z_2^y$.

The corresponding spectral density function is given by

$$S_{GG}(u, v) = \frac{\rho}{A(z_1,z_2)\ A(z_1^{-1},z_2^{-1})}, \tag{2.59}$$

where $z_1 = e^{-\frac{\sqrt{-1}\,2\pi u}{N}}$, $z_2 = e^{-\frac{\sqrt{-1}\,2\pi v}{N}}$. Notice that this is one of the few instances where the two-dimensional spectral density function can be factorized. This is always the case, however, in one-dimensional spectral density. If an image can be represented accurately by a model as in (2.55), the spectral density in (2.59) becomes superior to other two-dimensional spectral estimation techniques (e.g. the maximum entropy method). This is because the form in (2.59) is fully determined by the neighbor set η and the parameter set θ_r, $r \in \eta$ and ρ. The maximum entropy method, on the other hand, requires good estimates for the autocorrelation function, and also requires that the order of the autoregressive model fitted to the data be known (e.g. [Mal82]).

2.3.2. Simultaneous ARMA Models

An ARMA model is characterized by the two polynomials A(z) and B(z) defined in (2.60) and (2.61) [Kas81].

$$A(z) = 1 - \sum_{r \in \eta_1} \theta_r z^r , \quad \theta_r = \theta_{-r}, \tag{2.60}$$

$$B(z) = 1 + \sum_{r \in \eta_2} \phi_r z^r , \quad \phi_r = \phi_{-r}, \tag{2.61}$$

where η_1 and η_2 are symmetric neighbor sets and the parameters θ and ϕ satisfy the following equation:

$$A(z) > 0 \quad \text{and} \quad B(z) > 0 \quad \text{for all } |z_1| = 1 \text{ and } |z_2| = 1. \tag{2.62}$$

The difference equation form for the ARMA model is given by

$$G_s = \sum_{r \in \eta_1} \theta_r G_{(s \oplus r)} + \sqrt{v} W_s, \tag{2.63}$$

or, equivalently,

$$A(z) G_s = \sqrt{v} W_s, \tag{2.64}$$

where the polynomials $A(\cdot)$ and $B(\cdot)$ are given by (2.60) and (2.61), respectively. The input process $\{W_s\}$ has zero mean and is correlated with G_s. Notice the difference between the AR and ARMA representations with respect to the innovation **W**. In the AR representation, **W** is uncorrelated with **G**. The innovation **W** in the ARMA representation has the following power spectral density function:

$$S_{WW}(z) = A(z)B(z). \tag{2.65}$$

The condition (2.62) on $A(\cdot)$ and $B(\cdot)$ is necessary to insure stationarity of **G** as given by (2.63). The spectral density of **G** in (2.65) is given by

$$S_{GG}(u,v) = \frac{vB(z_1,z_2)}{A(z_1,z_2)}. \tag{2.66}$$

Thus, given any spectral density which is a ratio of two positive linear combinations of sinusoids in u and v, there exists a corresponding ARMA model as in (2.62).

2.3.3. Parameter Estimation on Simultaneous Models

Given an image configuration (gray level information for the sites of the discrete image), we would like to fit a model to the image. The successful fit depends upon the proper selection of the model type and its parameters. Suppose we want to fit an SAR model to a given image. How shall we select the neighbor system η and the parameters θ? The problem of neighbor system selection is discussed elsewhere (e.g. [Bes74], [Woo72], [KaC83]). We will introduce conventions for a few possible η here and emphasize the issue of parameter estimation. We will also emphasize the SAR model on a square finite lattice.

Let's represent a site $s = (x,y) \in S$ by ● and the sites $\{r = (k,l): (k,l) \in \eta\}$ by the symbol ○.

(a) Causal neighbor set:

$$\eta = \{(x,y): (0,0) \in \eta, x \le 0, y \le 0\}.$$

Graphically, this is shown in Figure 2.6(a).

(b) Semicausal neighbor set:

$$\eta = \{(x,y): (0,0) \in \eta, x \le 0, y \text{ arbitrary or vice versa}\}.$$

Graphically, this is shown in Figure 2.6(b).

(c) Unilateral neighbor set: $\eta \subset S^+$ such that:

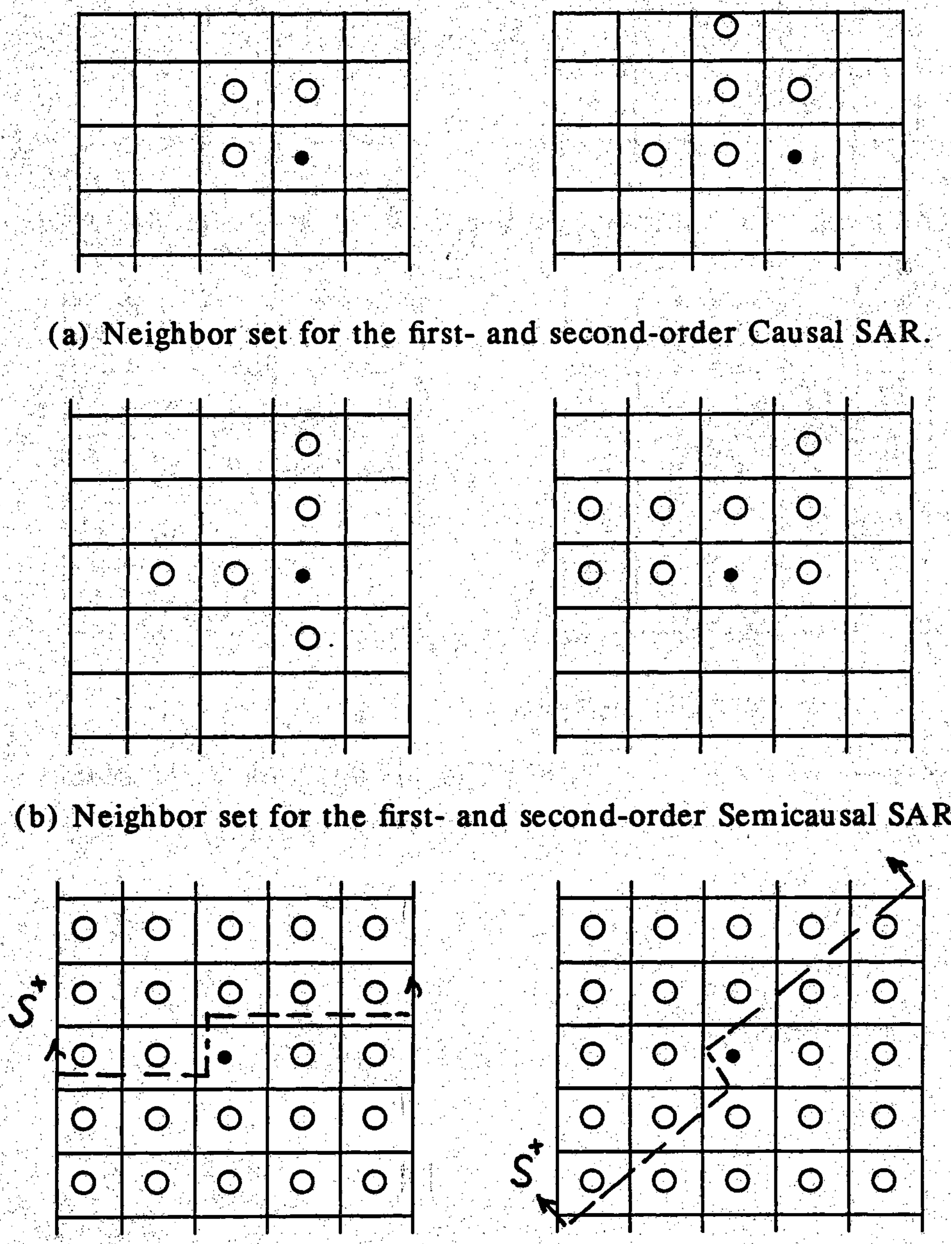
(i) $s_1 \in S^+$, $s_2 \in S+ \Rightarrow s_1 + s_2 \in S^+$,

(ii) $s \in S^+ \Rightarrow -s \notin S^+$,

(iii) $(0, 0) \notin S^+$.

Examples of S^+ are shown in Figure 2.6(c). Note that S^+ is not unique [Kas81].

The two classical techniques of maximum likelihood and least squares have also been used for parameter estimation of the simultaneous random field models (e.g. [Bes74], [Kas81], [KaC83]). The least squares estimates are known to be asymptotically consistent for SAR models with unilateral neighbor sets. It is, however, inconsistent for bilateral neighbor sets. Asymptotically efficient maximum likelihood estimates (MLE) can be



(a) Neighbor set for the first- and second-order Causal SAR.

(b) Neighbor set for the first- and second-order Semicausal SAR.

(c) Examples of the non symetric half plane S^+.

Figure 2.6. Some neighbor set definitions for the SAR models.

obtained if an appropriate choice for the distribution of the random field G is made. An approximate MLE which can be calculated iteratively has been proposed by Kashyap and Chellappa [KaC83]. Finally, robust estimation techniques have very limited use thus far in spatial interaction models. Only special cases which are direct extensions of the one-dimensional case have been reported [KaE88].

2.3.3.1. Least Squares Estimation (LS)

Consider a finite square (N × N) image obeying the toroidal SAR model in (2.55). An approximation to the least square estimation for the parameters in (2.55) is given by the following equations [KaC83]:

$$\hat{\theta} = [\sum_{s \in S} \mathbf{z}_s{}^t \, \mathbf{z}_s]^{-1} [\sum_{s \in S} \mathbf{z}_s g_s], \tag{2.67}$$

$$\hat{\rho} = \frac{1}{N^2} \sum_{s \in S} (g_s - \hat{\theta}^t \, \mathbf{z}_s)^2, \tag{2.68}$$

where $\mathbf{z}(\cdot)$ is

$$\mathbf{z}_s = [g_{(s \oplus r)}, r \in \eta]^t. \tag{2.69}$$

Equations (2.68 and 2.69) are for infinite stationary SAR models. They provide only an approximation for the finite SAR case. This estimate is in general inconsistent for nonunilateral SAR models [KaC83]. One advantage, however, of the LS estimate is that it is much easier to calculate than other methods (e.g. the maximum likelihood or robust techniques).

2.3.3.2. Maximum Likelihood Method (MLE)

Consider the SAR toroidal representation given in (2.56b). Assume the innovation W(·) is Gaussian. It is easy to show that the likelihood function *ln* p(G|θ, ρ) has the following form:

$$ln \, p(g|\theta, \rho) = ln \, (|B(\theta)|) - \frac{N^2}{2} ln \, (2\pi \rho) - \frac{1}{2\rho} \sum_{s \in S} (g_s - \theta^t \mathbf{z}_s)^2, \tag{2.70}$$

where | | stands for the determinant and $\mathbf{z}_s$ is as defined in (2.69). Since the matrix $B(\theta)$ is block circulant, the determinant can be obtained by Fourier transform method (e.g. [Hun73], [Kas80], [KaC83]). The determinant

$|B(\theta)|$, therefore, can be written in the following form:

$$|B(\theta)| = \prod_{s \in S} (1 - \theta^t \Phi_s), \qquad (2.71)$$

where

$$\Phi_s = \left[\exp(\sqrt{-1}\,\frac{2\pi}{N}\,(s-1)^t\, r),\ r \in \eta\right]^t \quad ,1 = (1,\ 1). \qquad (2.72)$$

Hence (2.70) can be simplified to

$$ln\ p(g|\theta,\ \rho) = \sum_{s \in S} ln\ (1 - \theta^t\Phi_s) - \frac{N^2}{2}\, ln\ (2\pi\rho) - \frac{1}{2\rho} \sum_{s \in S} (g_s - \theta^t z_s)^2 \qquad (2.73)$$

Now the ML estimate can be obtained by maximizing (2.73) with respect to θ and ρ.

Since the log-likelihood function is not quadratic in θ, an easy expression for the estimates is not possible. Numerical techniques, therefore, have been used to obtain the estimates (e.g. [KaC83]). Replacing the term $ln\ (1 - \theta^t\Phi_s)$ with the approximation $ln\ (1 + a) = a - a^2/2$, a simple iterative approximation for the MLE which is also asymptotically consistent is given in Theorem 2.5 [KaC83].

Theorem 2.5 [KaC83]: The estimates $\hat{\theta}$ and $\hat{\rho}$, maximizing (2.73) using the above approximation for $ln\ (1 + a)$, are obtained as the limits of θ_k, ρ_k defined by

$$\theta_{k+1} = (\mathbf{P} - \frac{1}{\rho_k}\mathbf{Y})^{-1} (\mathbf{R} - \frac{1}{\rho_k}\mathbf{T}) \qquad k = 0,\ 1,\ 2,\ \cdots, \qquad (2.74)$$

and

$$\rho_k = \frac{1}{N^2} \sum_{s \in S} (g_s - \theta^t_k z_s)^2 \qquad k = 0,\ 1,\ 2,\ \cdots, \qquad (2.75)$$

where

$$\mathbf{Y} = \sum_{s \in S} z_s z_s^t \qquad \text{m × m matrix,} \qquad (2.76)$$

$$\mathbf{T} = \sum_{s \in S} z_s g_s \qquad \text{m × 1 vector,} \qquad (2.77)$$

$$\mathbf{R} = \sum_{s \in S} c_s \qquad \text{m × 1 vector,} \qquad (2.78)$$

$$P = \sum_{s \in S} S_s {S^t}_s - C_s {C^t}_s \qquad m \times m \text{ matrix}, \tag{2.79}$$

$$C_s = [\cos\frac{2\pi}{N}((s - 1)^t r),\ r \in \eta]^t,\ \ 1 = (1,1) \quad m\times 1 \text{ vector}, \tag{2.80}$$

$$S_s = [\sin\frac{2\pi}{N}((s - 1)^t r),\ r \in \eta]^t,\ \ 1 = (1,1) \quad m\times 1 \text{ vector}, \tag{2.81}$$

and $z(\cdot)$ as in (2.69). □

The initial values, θ_0 and ρ_0 are the least square estimates in (2.67 - 2.68).

2.4. Summary

In this chapter we studied recent progress in random field theory which has been used in computer vision. It has been shown that in order to use a random field model in the analysis of an image, we need to chose a specific model (e.g. simultaneous or Gibbs-Markov), then we need to specify (estimate) the order of that model and its parameters (coefficients). Both of these issues were studied for the Gibbs-Markov and the Simultaneous random fields. We will use the results of this chapter in our algorithms for image segmentation and boundary estimation.

CHAPTER 3
REGION-BASED SEGMENTATION

3.1. Introduction

In this chapter we develop a statistical approach to image segmentation for images that can be adequately described by the random field models of Chapter 2. These images include textures of fine structure (microtextures) or of regular (periodic) placement structure (e.g. checker board textures). The observed image process G is modeled as a composite of two random processes, a high level process G^h and a low level process G^l, that is, $G = (G^h, G^l)$. Each of the three processes is a random field, and all are defined on the same lattice S. The high level process (labeling or coloring process) G^h is used to characterize the spatial clustering of pixels into regions. The type of information it reflects pertains to size, shape, orientation, and frequency of regions in the scene. It also reflects spatial continuity of regions, that is, if a pixel belongs to a certain region type then the neighboring pixels belong to the same region type with high probability. The low level process (pixel process) G^l simply describes the statistical dependence of pixel gray level values in each region.

Let the number of regions in the scene be M, the number of possible gray levels be q, and consider a square lattice $S = \{s = (x,y): x,y \in [0, N-1]\}$. We will consider a square $N \times N$ lattice only for convenience; the treatment to follow is valid for any rectangular lattice. The processes G^h and G^l are discrete parameter random fields, and their state spaces are defined as follows:

$$\Xi^h = \{\xi^h: \xi^h \in [0, q_1-1]\}. \tag{3.1}$$

$$\Xi^l = \{\xi^l: \xi^l \in [0, q_2-1]\}. \tag{3.2}$$

We will consider the most general case where the two processes are defined on the same state space, that is, we assume $q_1 = q_2$.

The high level process G^h can be described as follows: For each pixel $s \in S$, $g^h(s) = k$ denotes the fact that pixel s in the realization g^h belongs to region type $k \in [1,M]$. In other words, the realization g^h is a partition of S into M regions. Each region type can, of course, occur in more than one pixel location within the lattice S. For example, a Landsat image can consist of numerous areas of wheat all characterized by the single region type *wheat.*

The observed image g can now be described as follows: Consider a region type k. The gray level value at pixel $s \in S$ of the observed image g equals that of region type k, that is,

$$g(s) = g^{l_k}(s) \qquad \text{if } g^h(s) = k, \qquad \text{for all } k \in [1,M] \text{ and all } s \in S, \quad (3.3)$$

where $g^{l_k}(\cdot)$ denotes the gray levels of the region type k. In other words, for each configuration $G \in \Omega$, realizations g^h for the scene and $g^l = \{g^{l_k}: k \in [1,M]\}$ for the regions are specified. The realization $g = (g^h, g^l)$ is specified as follows: At each pixel s, g(s) is the value of the region specified by $g^h(s)$. Figure 3.1 illustrates the above setup. In this figure we assume that the observed image consists of a number of regions, each can be described by the random field models of Chapter 2.

In this chapter we are going to consider textured regions. That is, we assume that the scene consists of a number of textured regions. The segmentation problem can be stated as follows: Given a realization g of a textured image, determine the high level realization g^h that produced g. Since g^h is not observed, the goal is to determine an estimate $\hat{g}^h$ based on the observation g.

A possible segmentation criterion can be the maximum a posteriori estimation (MAP) by which the high level configuration that has the overall maximum probability, given the observation g, is obtained. A second possible criterion is to maximize the a posterior marginal probability at each pixel, that is, obtain a classification of each individual pixel that has maximum probability, given the observation g. The MAP setup is fundamental; it is, however, computationally extensive. Two well known studies of MAP segmentation are Geman and co-workers (e.g. [GeG86], [GeG87] and [Gra87]) and Derin and Elliott (e.g. [DeE87] and [LaD89]). In these studies, stochastic relaxation by simulated annealing (SA) and dynamic

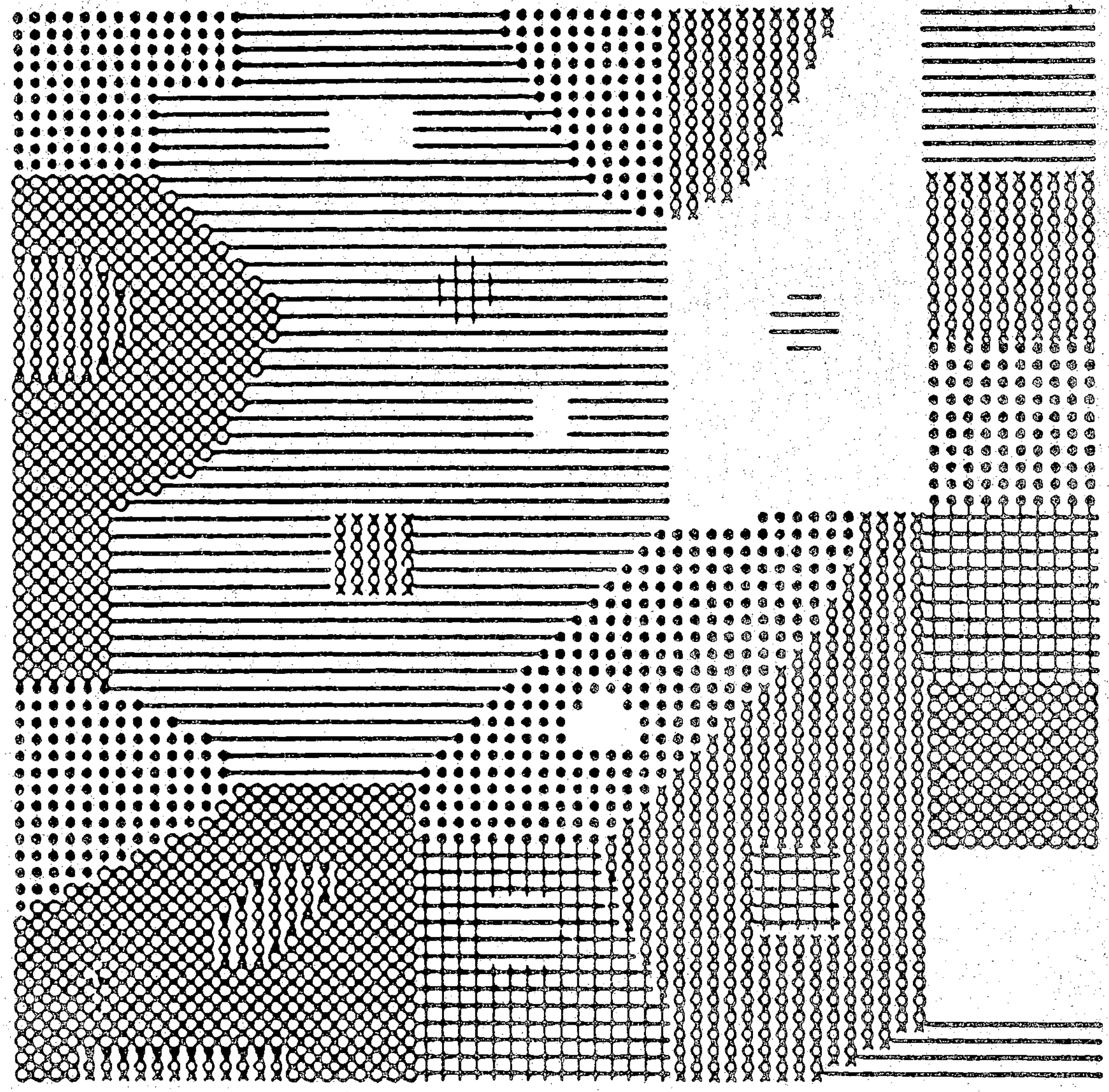


Figure 3.1. An illustration of the high level and low level processes.
The scene is assumed to be formed of a six-different regions.
The six-labels (colors) are shown on the right (adopted from [Bes86]).

programming were used to solve the MAP segmentation problem, as well as the recently proposed deterministic iterated conditional mode (ICM) optimization approach of Besag [Bes86].

The MAP segmentation involves the determination of $\hat{g}^h$ that maximizes $P(G^h = g^h | G = g)$ with respect to g^h. By Bayes's rule,

$$P(G^h = g^h | G = g) = \frac{P(G = g | G^h = g^h) P(G^h = g^h)}{P(G = g)}. \tag{3.4}$$

Since the denominator of (3.4) does not affect the optimization, the MAP segmentation can be obtained, equivalently, by maximizing the numerator of (3.4) or its natural logarithm, that is, we need to find $\hat{g}^h$ which maximizes

$$\Gamma(G, G^h) = \mathit{ln}\, P(G = g | G^h = g^h) + \mathit{ln}\, P(G^h = g^h). \tag{3.5}$$

The first term of (3.5) is the likelihood due to the texture information and the second term is due to the high level process. The optimization in (3.5) is to be done over a space of M^{N^2} possible image configutations. We will discuss optimization methods later, but first we will describe the random fields G^h and G^l.

3.2. Image Modeling

For the MAP approach to be analytically tractable, an exponential form for the quantities $P(G = g | G^h = g^h)$ and $P(G^h = g^h)$ is necessary. In the studies on image modeling by Kashyap and co-workers, the simultaneous and conditional Markov random field models have been shown to be adequate for the analysis of various types of textures (e.g. [Kas81], [KaC83], [KaK86], and [KaE89]). Various texture configurations can be easily generated using the Kashyap algorithm [Kas81], which requires only deterministic calculations (the discrete Fourier transforms). This is unlike the stochastic techniques such as Monto Carlo simulations [HaH64] or the Gibbs sampler [GeG84]. On the other hand, in the studies by Geman and co-workers (e.g. [GeG87] and [Gra87]) and Derin and Elliott [DeE87] on MAP segmentation, Gibbs-Markov random fields have been shown to be adequate for representing the high level process G^h.

In this chapter we will use a simultaneous random field model for the observed image (to obtain the quantity $P(G = g|G^h = g^h)$), and a second-order Gibbs-Markov random field for the high level process (to obtain the quantity $P(G^h = g^h)$.) These choices will be shown to be convenient for the analysis as well as for the implementation of the MAP algorithm.

3.2.1. The Region (Low Level) Process

Let the observed image g be formed of M different textures. Given the actual scene (high level process G^h), the observed image process becomes exactly the low level process, that is,

$$P(G = g|G^h = g^h) = P(G^l = g^l). \qquad (3.6)$$

We will use an SAR model (2.55) for the M textures, that is,

$$G^{l_k}{}_s = \sum_{r \in \eta^{l_k}} \theta^{l_k}{}_r G^{l_k}{}_{(s \oplus r)} + \sqrt{\rho_k}\, W_s, \text{ for all } s \in S^k \text{ and } k \in [1,M], \qquad (3.7)$$

where $G^{l_k}{}_s$ is the random variable at site (pixel location) s in the k^{th} texture, $\{\theta^{l_k}{}_r\}$ is the set of parameters for the k^{th} texture, W_s is the Gaussian innovation with zero mean and unit variance, ρ_k is the unknown parameter that represents the actual noise variance, and η^{l_k} is the neighbor system of the SAR model over region type k.

To simplify the notation, we make the following assumptions:

1. The support S^k of the region (texture) k is a torus, that is, the toroidal representation (2.35)-(2.37) is assumed to hold over each region.
2. All regions are adequately modeled by an SAR model defined on the same neighbor system η, that is,

$$\eta^{l_k} = \eta \text{ for all } k \in [1,M].$$

3. Given the high level process G^h, the probability distribution $P(G = g|G^h = g^h) = P(G^l = g^l)$ can be written in the following factorized form:

$$P(G^l = g^l) = \prod_{k=1}^{M} P(G^{l_k} = g^{l_k}). \qquad (3.8)$$

The first assumption will not, in general, hold except for the special case of large rectangular lattices, that is, if the supports S^k, $k \in [1,M]$ are rectangular with large extents. This assumption simplifies the analysis. The second assumption is not critical and can be easily relaxed especially for supervised region segmentation, that is, when the parameters of each region are known a priori by the segmentation algorithm. The third assumption is very critical to our analysis and is quite reasonable except at the boundaries between different regions.

Now by the first assumption, we can represent the SAR model (3.3) for region k as in (2.56b), that is,

$$B^{l_k}(\Theta^{l_k})\, \underset{\sim}{g}^{l_k} = \sqrt{\rho_k}\, \underset{\sim}{W}, \quad k \in [1,M], \tag{3.9}$$

where B^{l_k} is a block circulant matrix of size $N_k^2 \times N_k^2$. Inherent in the above representation is the assumption that a region k is defined on a square lattice $S^k = \{s = (x,y) : x,y \in [0, N_k - 1]\}$.

From (3.8) and the likelihood equations in (2.70)-(2.73), it is straightforward that the first term in (3.5) can be written as follows:

$$\begin{aligned} ln\, P(G = g | G^h = g^h) &= ln\, P(G^l = g^l) \\ &= \sum_{k=1}^{M} \Big\{ \sum_{s \in S^k} ln\,(1 - (\Theta^{l_k})^t \Phi_s^{l_k}) - \frac{N_k^2}{2} ln\,(2\pi \rho_k) \\ &\quad - \frac{1}{2\rho_k} \sum_{s \in S^k} g_s^{l_k} - (\Theta^{l_k})^t z_s^{l_k} \Big\}, \end{aligned} \tag{3.10}$$

where Θ^{l_k} is the parameter vector for the SAR model on region k, and $z_s^{l_k}$ and $\Phi_s^{l_k}$ are as defined in (2.69) and (2.72), respectively.

3.2.2. The Scene (High Level) Process

As we indicated before, the process G^h will be modeled by a Gibbs-Markov random field. The probability distribution is given by

$$P_{G^h}(G^h = g^h) = \frac{e^{-U(g^h)}}{Z^h}, \tag{3.11}$$

where $U(g^h)$ is the energy function and Z^h is the normalization function. (See

(2.43)-(2.45).) The model in (3.11) has been used in various studies in MAP segmentation. Geman et al. [GeG87] used a first-order Ising model (2.46) while Derin and Elliott [DeE87] used a second- order Gibbs-Markov model with pair clique potentials, that is, only parameters of the horizontal, vertical, and the two diagonal cliques were considered. A second order Gibbs-Markov model was also used by Besag [Bes86] (see also [SiC88] and [BoL88]). We will use a similar model to that in Derin and Elliott [DeE87], for its convenience to recursive calculations. We will make the following assumption concerning the high level process:

$$P(G^h = g^h) = \prod_{s \in S} P(G_s^h = g_s^h | G_r^h = g_r^h,\ r \in \eta^h), \tag{3.12}$$

where η^h is the second-order neighborhood in (2.21a).

The energy function in (3.11) will be expressed as

$$U(g^h) = \sum_{c \in C} V_c(g^h), \tag{3.13}$$

where $V_c(g^h)$ are the clique potentials defined as follows [DeE87]:

$$V_c(g^h) = \begin{cases} -\beta & \text{if } g^h{}_s = g^h{}_r \text{ and } s,r \in c \\ \beta & \text{otherwise,} \end{cases} \tag{3.14}$$

where, again, we consider only pair cliques. The parameter β is positive, so that two neighboring pixels are more likely to belong to the same class (region) than to different classes. Increasing β has the effect of increasing the size of the regions and smoothing their boundaries.

Now, after taking the natural logarithm of the two sides in (3.12) and using (3.13), we obtain

$$\mathit{ln}\, P(G^h = k) = -\sum_{c \in C} V_c(g^h) - \mathit{ln}\, Z^h. \tag{3.15}$$

3.3. The MAP Algorithm

Using (3.10) and (3.15) we can now write the MAP equation (3.5) as follows:

$$\begin{aligned}\Gamma(G,G^h) &= \mathit{ln}\, P(G = g | G^h = g^h) + \mathit{ln}\, P(G^h = g^h) \\ &= \sum_{k=1}^{M} \Big\{ \sum_{s \in S^k} \mathit{ln}\, (1 - (\Theta^{l_k})^t \Phi_s^{l_k}) - \frac{N_k^2}{2} \mathit{ln}\, (2\pi \rho_k)\end{aligned}$$

$$- \frac{1}{2\rho_k} \sum_{s \in S^k} g_s^{l_k} - (\Theta^{l_k})^t z_s^{l_k} \Big\}$$

$$- \sum_{c \in C} V_c(g^h) - ln\, Z^h. \tag{3.16}$$

In order to simplify the formulations, we will make the following noncritical assumption: Assume that the parameter ρ_k is independent of k, that is, $\rho_k = \rho$ for all the SAR models over the regions k. Hence, we can write

$$\sum_{k=1}^{M} \frac{N_k^2}{2} ln\, (2\pi \rho_k) = K_0\, ln\, 2\pi \rho, \tag{3.17}$$

where K_0 is constant, and the parameter ρ can be the average of all the values $\{\rho_k\}$ of different regions. Now we can write (3.16) in the following form:

$$\Gamma(G,G^h) = \sum_{k=1}^{M} \sum_{s \in S^k} f^k{}_s - \sum_{c \in C} V_c(g^h) - (ln\, Z^h + K_0\, ln\, 2\pi \rho), \tag{3.18}$$

where

$$f_s{}^k = \Big\{ ln\, (1 - (\Theta^{l_k})^t \Phi_s^{l_k}) - \frac{1}{2\rho} (g_s^{l_k} - (\Theta^{l_k})^t z_s^{l_k}) \Big\}. \tag{3.19}$$

A recursive formula for (3.18) can be written as follows:

$$\Gamma^{(0)}(G,G^h) = -\, ln\, (Z^h + \frac{N^2}{2}\, ln\, 2\pi \rho) \tag{3.20}$$

$$\Gamma^{(n)}(G,G^h) = \Gamma^{(n-1)}(G,G^h) - \sum_{k=1}^{M} \sum_{s \in S^{k(n)}} f_s{}^k - \sum_{c \in Cn-1,n} V_c(g^h), \tag{3.21}$$

where

$$S^{k(n)} = \{ s = (i,n) \colon g^h{}_{(i,n)} = k, \quad 1 \leq i \leq N \}. \tag{3.22}$$

$$C^{(n-1),n} = \{c \colon c \text{ is a clique with pixels only}$$

$$\text{in column } n \text{ or columns } n-1 \text{ and } n \}. \tag{3.23}$$

It is easy to show that $\Gamma^{(N)}(G,G^h) = \Gamma(G,G^h)$ as in (3.18). Hence, it is possible to calculate the MAP segmentation (3.16) recursively.

A recursive formulation that used Gibbs-Markov random field models for both the high level and the low level processes was obtained by Derin and Elliott [DeE87]. The formulation in (3.18)-(3.23) is different from that of [DeE87] in a number of respects. First, an SAR model is used for the low level process which has been shown to be adequate for the synthesis and the analysis of various natural textures. The parameters of the model can be estimated on line without too much uncertainty as in the case of the Gibbs-Markov model. Second, the order of the SAR model is a parameter in the formulation which adds to the flexibility of the algorithm for various types of textures (one should not expect that various textures in a scene would be represented by models of similar orders). Third, the Derin-Elliott algorithm is based on a number of assumptions that will not, in general, hold in practice. In particular, there is no evidence that second-order Gibbs-Markov random fields with only pair cliques would represent natural textures. Finally, the parameter estimation procedure suggested in [DeE87] can only be adequate for images with very small number of gray levels, and is not computationally feasible for large number of textures in the scene.

In order to evaluate the MAP segmentation, we need to know (estimate) all the parameters in (3.18) and (3.19). That is, we need to estimate the parameters $\{\theta^{l_k}\}_{k=1}^{M}$, ρ, the number of regions M, the clique parameter β in (3.14), and we need to calculate the local characteristics Z^h.

From (3.20) and (3.21), we can argue that the local characteristics is not actually affecting the segmentation results and can be ignored. Therefore, we will assume that the initial condition is zero, that is, $\Gamma^{(0)}(\mathbf{G},\mathbf{G}^h) = 0$. Hence, the only requirements for the segmentation algorithm, as defined by (3.20)-(3.23), are the parameters $\{\theta^{l_k}\}_{k=1}^{M}$, ρ, and M. In this thesis, we will separate the parameter estimation problem from the segmentation process. That is, the estimation will be obtained in advance (off line), and the segmentation algorithm will be provided with these estimates.

The algorithm in (3.20)-(3.23) could be carried out by a dynamic programming algorithm, for example. Since the size of the state space is too large (M^{N^2}), execution time will be enormous for any practical size image.

We have used, instead, the following procedure:

Algorithm:

Step 0: Obtain an estimate of the parameters discussed above. In our experiments we used first-, second-, and third-order SAR models.

Step 1: Assume zero initial conditions and carry out one iteration in (3.21) as follows:

> (i) From the knowledge of the SAR model order, the vectors Φ_s and $\mathbf{z}_s$ in (3.16) are evaluated over strips of data. For example, in the first- and secod-order SAR models the strips will be formed of 3-rows.
>
> (ii) The quantity $\Gamma(\cdot)$ is evaluated on 3×3 blocks of data, in each strip, for the first- and secod-order order SAR models and on 5×5 blocks of data for the third order SAR model, and so on. From Figure 2.3, we need a block size of 5×5 for up to seventh-order SAR models (in this case, the strips in (i) will be five rows.) The pixel in the center of the block is classified as to belong to region type k if the value of $\Gamma(\cdot)$ is maximum when evaluated by the parameters of region k. The procedure is continued to obtain one line of labeling which will be used as the initial condition for the following lines.
>
> (iii) Increment the strips by a new row of data and obtain a line of labeling as in (ii).

Step 2: Stop when a whole cycle is performed, that is, when the last strip is reached.

Figure 3.2 is a flowchart for the steps used in executing the algorithm. The algorithm has been used on various synthesistic textures as well as real textured scenes. We now examine its performance.

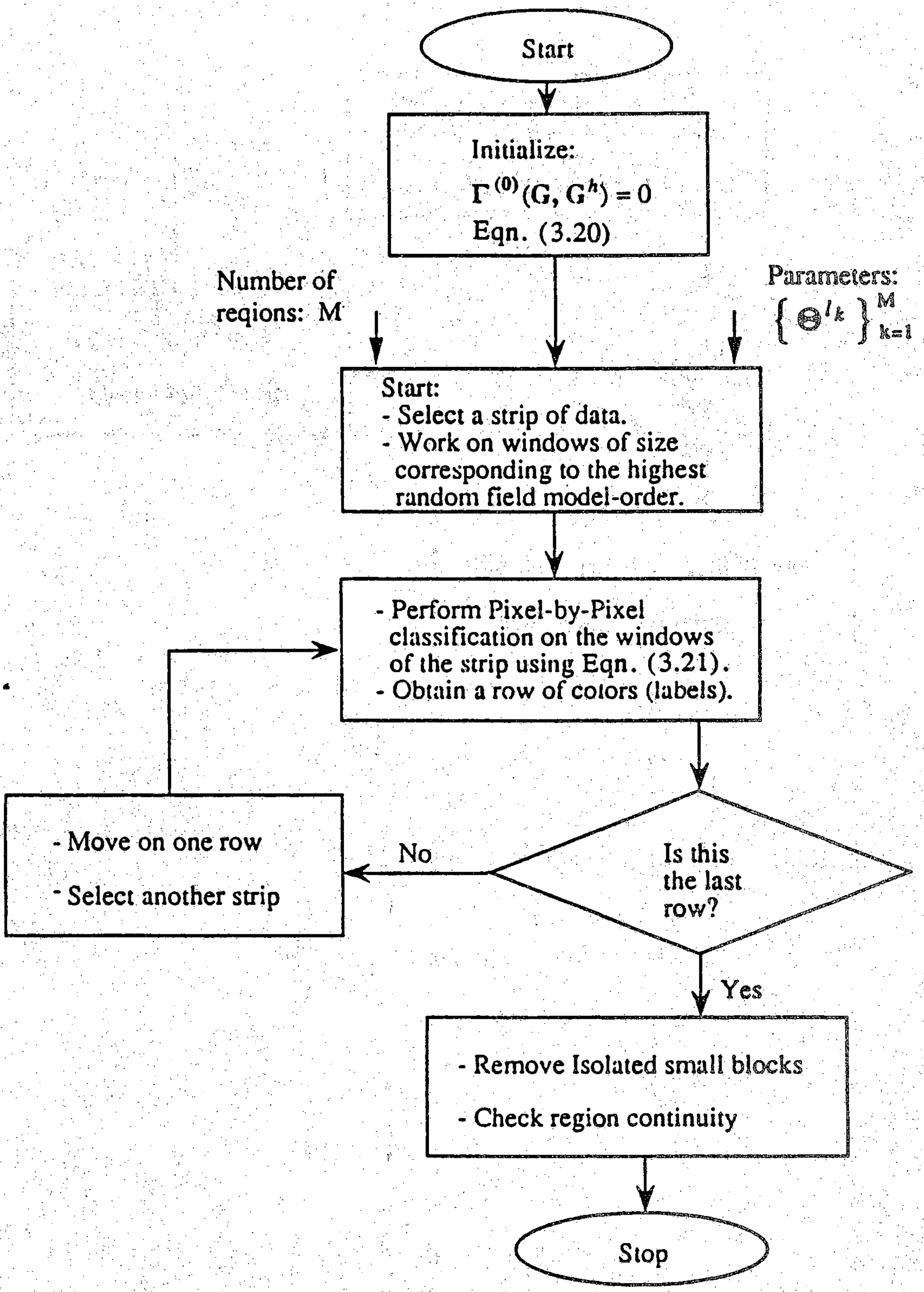

Start
Initialize:
$\Gamma^{(0)}(G, G^h) = 0$
Eqn. (3.20)
Number of reqions: M
Parameters: $\left\{ \Theta^{l_k} \right\}_{k=1}^{M}$
Start:
- Select a strip of data.
- Work on windows of size corresponding to the highest random field model-order.
- Perform Pixel-by-Pixel classification on the windows of the strip using Eqn. (3.21).
- Obtain a row of colors (labels).
Is this the last row?
No
Yes
- Move on one row
- Select another strip
- Remove Isolated small blocks
- Check region continuity
Stop

Figure 3.2. A flow chart of the MAP segmentation algorithm.

3.4. Texture Simulation

There have been a variety of approaches to the generation of synthetic-texture images [Har86]. We are mainly interested in statistical approachs based on Gibbs-Markov random field (GMRF) models and simultaneous autoregressive (SAR) models. Both models have been used in generating various textural patterns (e.g. [Kas81], [CrJ83], [KaC83], [GeG86], and [DeE87]). In this thesis, we have used the Kashyap algorithm for texture synthesis [Kas81]. The Appendix outlines the Kashyap algorithm for generating configuations from Conditional Markov models (a subclass of GMRF models) and SAR random field models.

In the segmentation algorithm we described above, we used SAR models for the low level process. We will generate few texture patterns using SAR models of different orders. The convention used for the vector $\underset{\sim}{\theta}$ for up to the third-order SAR is given by the following equation (see Figures 2.1 and 2.2):

$$\underset{\sim}{\theta} = [\theta_1 \mid \theta_2 \mid \theta_3] \tag{3.24}$$

where

$$\theta_1 = [\theta_{(x,y-1)},\ \theta_{(x+1,y)},\ \theta_{(x,y-1)},\ \theta_{(x,y+1)}\ ]^t \tag{3.25}$$

$$\theta_2 = [\theta_{(x-1,y-1)},\ \theta_{(x+1,y+1)},\ \theta_{(x-1,y+1)},\ \theta_{(x+1,y-1)}]^t \tag{3.26}$$

$$\theta_3 = [\theta_{(x-2,y)},\ \theta_{(x+2,y)}, \theta_{(x,y-2)},\ \theta_{(x,y+2)}]^t, \tag{3.27}$$

where t stands for vector transposition. In the above equations, $[\theta_1]$ is the set of coefficients for the first-order SAR, $[\theta_1 \mid \theta_2]$ is the set of coefficients for the second-order SAR, and $[\theta_1 \mid \theta_2 \mid \theta_3]$ are the third-order SAR coefficients.

Figure 3.3 shows sixteen (64 × 64) different texture configurations generated by the SAR model. The parameters used for these configurations are shown in Table 3.1. We have used up to third-order SAR models. The histograms for the configurations in Figure 3.3 are plotted in Figure 3.4. It is evident from these figures that the SAR model can in fact be used in generating configuartions corresponding to various microtextures.

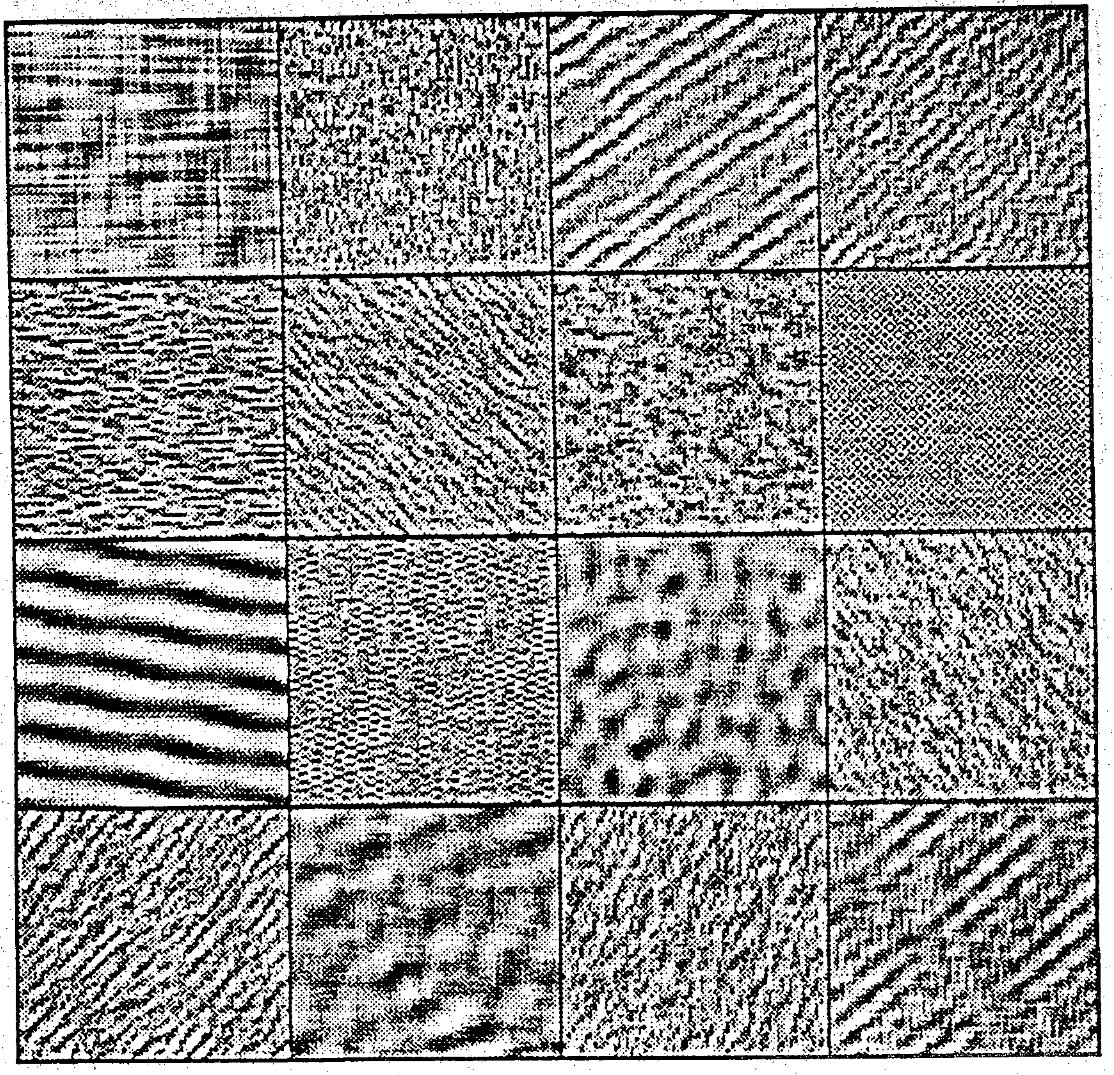


Figure 3.3. Sixteen 64 × 64 artificial textures from the SAR model.

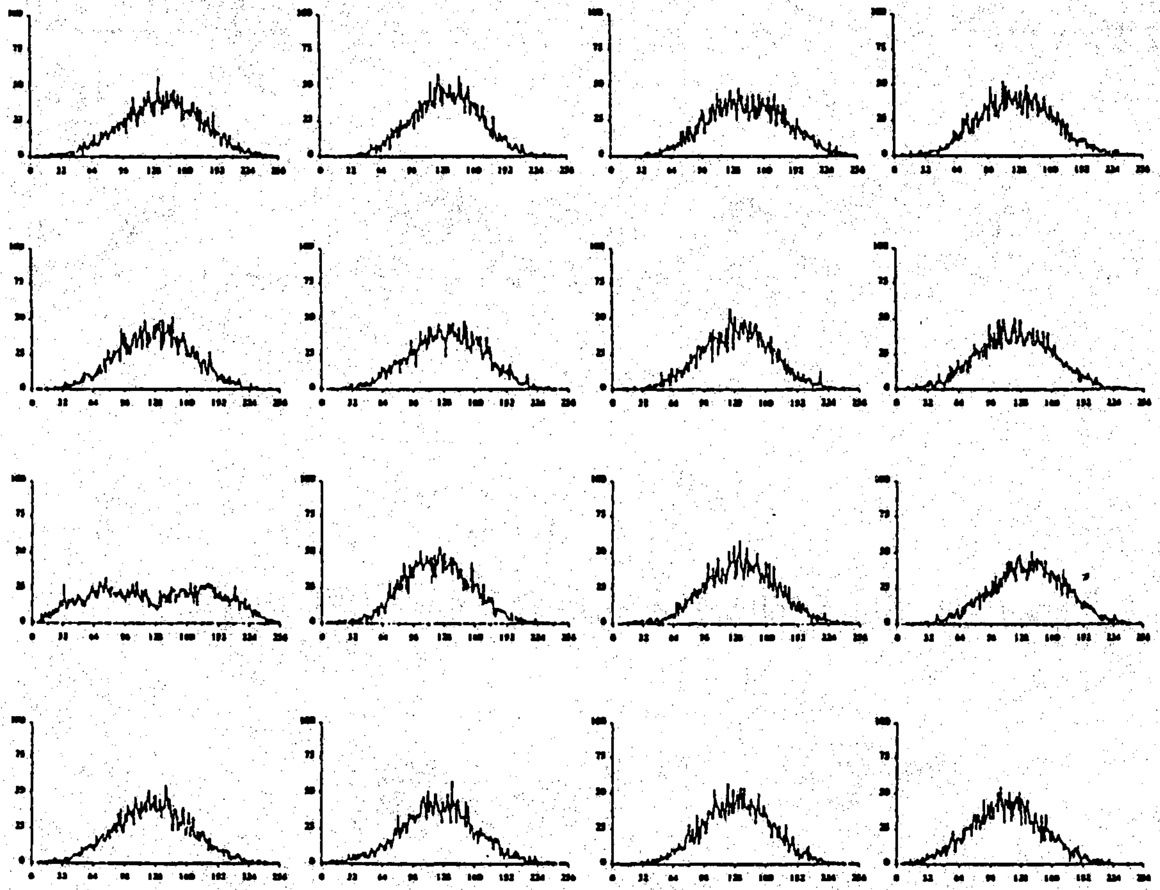


Figure 3.4. The gray level histograms for the textures in Figure 3.3.

Table 3.1. Model coefficients for Figure 3.3.

<table>
<tr><th colspan="4">First-Row</th></tr>
<tr><td>0.9704</td><td>0.18</td><td>0.18</td><td>0.18</td></tr>
<tr><td>0.0</td><td>0.18</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.9735</td><td>0.0</td><td>1.1011</td><td>0.0</td></tr>
<tr><td>0.0</td><td>-0.12</td><td>0.0</td><td>1.1011</td></tr>
<tr><td>-0.9686</td><td>0.0</td><td>-1.039</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>-1.039</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>-0.1806</td></tr>
<tr><td>0.0</td><td>0.0</td><td>-0.1806</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><th colspan="4">Second-Row</th></tr>
<tr><td>-0.12</td><td>0.0</td><td>0.18</td><td>0.0</td></tr>
<tr><td>-0.12</td><td>0.0</td><td>0.1</td><td>0.0</td></tr>
<tr><td>0.26</td><td>0.0</td><td>0.15</td><td>-0.14</td></tr>
<tr><td>0.26</td><td>0.0</td><td>0.12</td><td>-0.14</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.28</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.28</td><td>0.11</td><td>0.22</td></tr>
<tr><td>0.0</td><td>-0.14</td><td>0.0</td><td>0.23</td></tr>
<tr><td>0.0</td><td>-0.14</td><td>0.0</td><td>0.23</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><th colspan="4">Third-Row</th></tr>
<tr><td>0.5256</td><td>-0.1</td><td>0.3794</td><td>0.18</td></tr>
<tr><td>0.5256</td><td>-0.1</td><td>0.3794</td><td>-0.1</td></tr>
<tr><td>0.5081</td><td>0.2</td><td>0.1825</td><td>-0.1</td></tr>
<tr><td>0.5081</td><td>0.2</td><td>0.1825</td><td>-0.1</td></tr>
<tr><td>-0.2814</td><td>0.0</td><td>0.0</td><td>0.22</td></tr>
<tr><td>-0.2814</td><td>0.0</td><td>0.0</td><td>0.22</td></tr>
<tr><td>-0.2480</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>-0.2480</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.2</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.2</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>-0.15</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>-0.15</td><td>0.0</td><td>0.0</td></tr>
<tr><th colspan="4">Fourth-Row</th></tr>
<tr><td>0.12</td><td>1.0388</td><td>0.28</td><td>0.5357</td></tr>
<tr><td>0.12</td><td>0.0</td><td>0.28</td><td>0.5357</td></tr>
<tr><td>0.0</td><td>0.9046</td><td>-0.14</td><td>0.5246</td></tr>
<tr><td>0.0</td><td>0.0</td><td>-0.14</td><td>0.5246</td></tr>
<tr><td>-0.14</td><td>-0.7288</td><td>0.0</td><td>-0.3126</td></tr>
<tr><td>-0.14</td><td>0.0</td><td>0.0</td><td>-0.3126</td></tr>
<tr><td>0.28</td><td>0.0</td><td>0.22</td><td>-0.25</td></tr>
<tr><td>0.28</td><td>0.0</td><td>0.22</td><td>-0.25</td></tr>
<tr><td>0.0</td><td>-0.1814</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>-1088</td><td>0.0</td><td>0.0</td></tr>
<tr><td>0.0</td><td>0.0</td><td>0.0</td><td>0.0</td></tr>
</table>

3.4.1. Fitting An SAR Model to Natural Textures

Figure 3.5 shows four (256 × 256) natural textural images (cork, gems, peb, and woody). The histograms are plotted in Figure 3.6. We analyzed these four textures (64 × 64 portions) using SAR-models up to the third order. Each column in Figure 3.7 represents an original image (top) and the first-, second-, and third-order (bottom) SAR-model fit. The model coefficients for each case are shown in Table 3.2. The top most element in Table 3.2 is the parameter ρ in (2.55), and the other values are those of the parameter $\{\theta_r, r \in \eta\}$.

The criterion used to test the quality of the fit is visual similarity. From Figure 3.5, it is seen that a third-order SAR fit resembles reasonably well the first image (woody), a second-order SAR fit is reasonable of the second image (gems), a third-order SAR fit is reasonable for the third image (peb), and a first-order SAR fit is reasonable for the fourth image (cork). The parameters of the model were estimated using equations (2.74-2.81).

Although the results of the model fitting to natural textures does not, in general, produce a close replica of the actual image, an accurate segmentation results can be obtained. In the segmentation algorithm, the parameter set $\{\theta^{l_k}\}_{k=1}^{M}$ plays a key role in the accuracy of the resulting segmentation. In the study by Kashyap and Khotanzad [KaK86] on texture classifications using similar models, features based on the parameters $\{\theta^{l_k}\}_{k=1}^{M}$ where shown to be robust for texture classification. In our experiments on synthetic texture segmentation, the generated patterns (Figure 3.3) have very similar histograms (Figure 3.4), yet very accurate segmentation was obtained as we will show next. We should also point out that a similar conclusion was obtained by Geman et al. [GeG87] in their study on texture segmentation using Gibbs-Markov models for both the high level and the low level processes. In that study, second-order Gibbs-Markov models with pair cliques were used, and the MAP estimates were approximated by using simulated annealing.

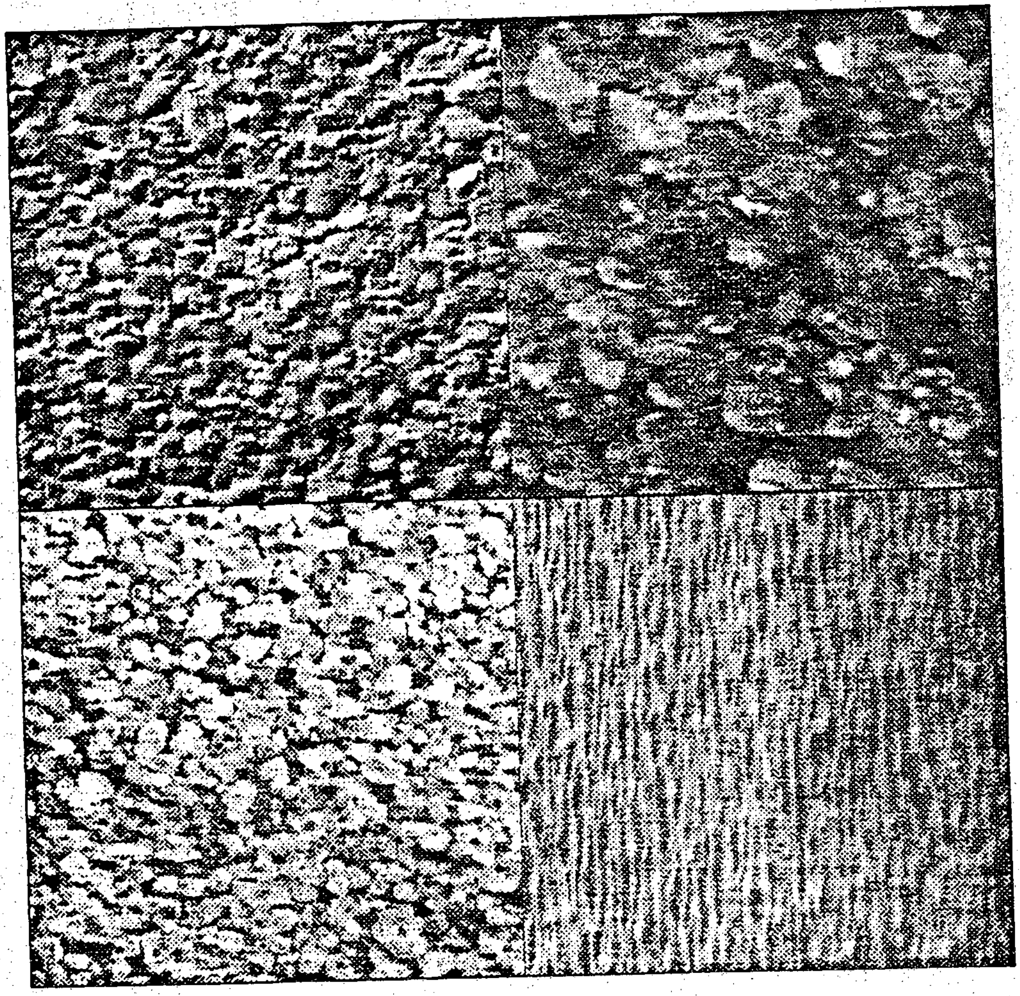


Figure 3.5. Four natural textures.
(Upper left, Cork; Upper right, Gems;
Lower left, Peb; Lower right, Woody.)

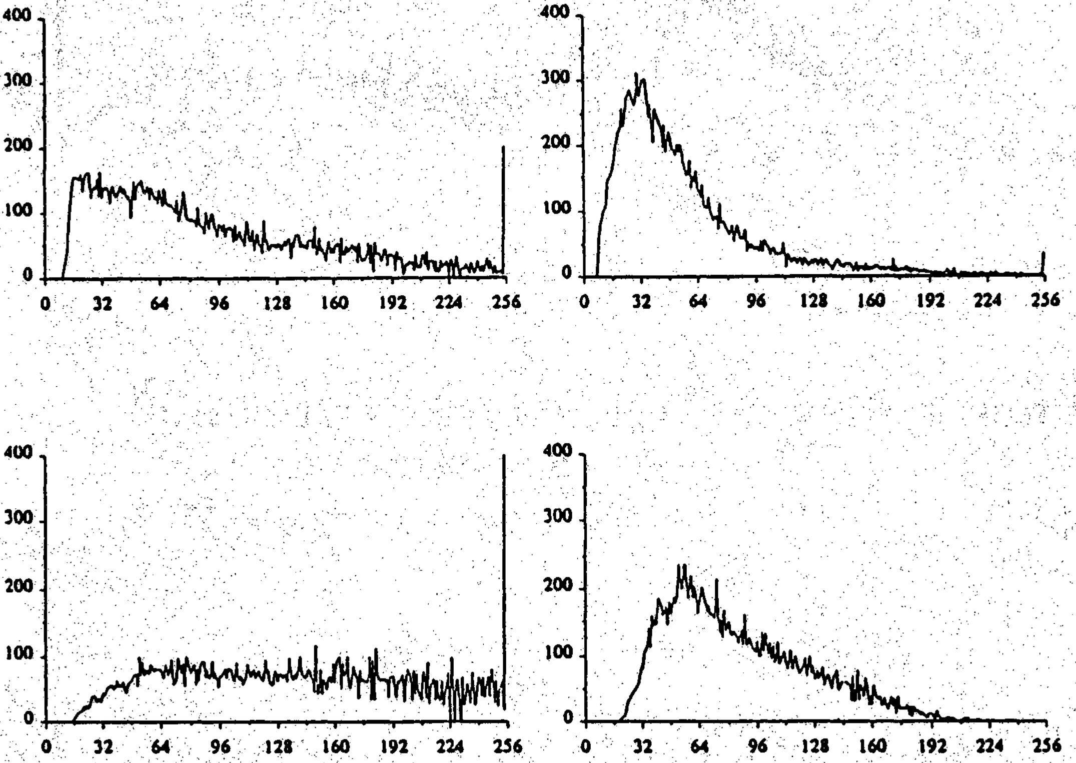


Figure 3.6. The gray level histograms for the textures in Figure 3.5.

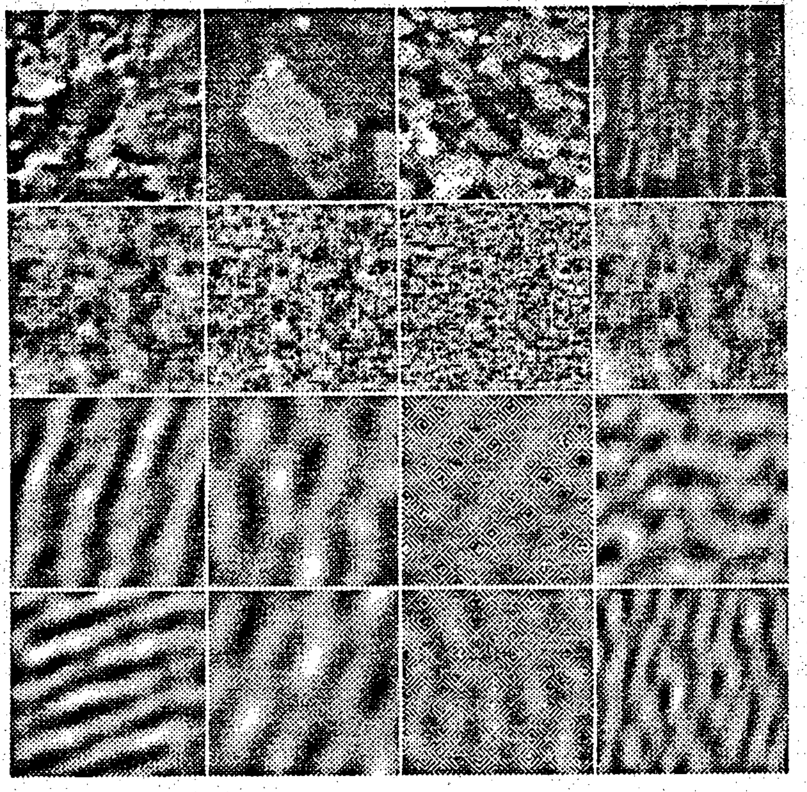


Figure 3.7. Fitting SAR models to natural textures.
(First column: Cork; Second column: Gems;
Third column, Peb; Fourth column, Woody.)

Table 3.2. Model coefficients for Figure 3.7.

First-order SAR			
116.62	487.72	2474.14	555.3286
0.1232	0.2366	0.1359	0.2479
0.1232	0.2366	0.1359	0.2479
0.3084	0.2022	0.1816	0.2395
0.3084	0.2022	0.1816	0.2395
Second-order SAR			
72.223	146.85	678.29	394.3153
-0.0739	0.0468	-0.0202	0.1277
-0.0739	0.0468	-0.0202	0.1277
0.2618	-0.0074	0.0327	0.1160
0.2618	-0.0074	0.0327	0.1160
0.1407	0.2619	0.2895	0.0784
0.1407	0.2619	0.2895	0.0784
0.1911	0.2176	0.2350	0.2210
0.1911	0.2176	0.2350	0.2210
Third-order SAR			
64.52	150.627	695.29	391.4627
-0.0376	0.0342	-0.0246	0.1354
-0.0376	0.0342	-0.0246	0.1354
0.1664	-0.0163	0.0312	0.1104
0.1664	-0.0163	0.0312	0.1104
0.1095	0.2293	0.2700	0.0824
0.1095	0.2293	0.2700	0.0824
0.1549	0.1802	0.2184	0.2237
0.1549	0.1802	0.2184	0.2237
-0.0473	0.0494	0.0373	-0.0148
-0.0473	0.0494	0.0373	-0.0148
0.1706	0.0488	0.0124	0.0041
0.1706	0.0488	0.0124	0.0041

3.5. Experimental Results

We now examine the results of the segmentation algorithm on synthetic as well as natural textures. In all of the experiments reported in this chapter, we fixed the value of the parameter β in (3.14) to be 1.25.

3.5.1. Segmentation of Synthetic Textures

The segmentation algorithm in Section 3.3 was applied to various combinations of the synthetic textures shown in Figure 3.3. Let's index these textures according to their rows and columns location. For example, the top left hand texture will denoted by T_{11} and the bottom right hand texture will be denoted by T_{44}.

Figure 3.8a is a 128 × 128 image that was constructed from the following textures: T_{12} (upper left), T_{21} (upper right), T_{23} (lower left), T_{33} (lower right) and the texture in the middle was constructed from T_{11}. From Table 3.1, the orders of the SAR models used were: second-, first-, second-, first-, and second-order, respectively. Hence, the strips used in the MAP algorithm were formed of three rows, and pixel classification was performed on 3 × 3 blocks. Figure 3.8b is the resulting coloring (classification) from the MAP algorithm. The coloring was accurate except at the boundaries between different textures. The boundary effect is most severe at boundaries between various textures and texture T_{11}.

Figure 3.9a is also a 128 × 128 image that was constructed from the following textures: T_{13} (upper left), T_{23} (upper right), T_{31} (lower left), T_{34} (lower right) and the texture in the middle was constructed from T_{33}. From Table 3.1, all textures were generated by a second-order SAR model, except T_{33} that was generated by first-order model. The strips used in the MAP algorithm were formed of three rows, and pixel classification was performed on 3 × 3 blocks. Figure 3.9b is the resulting coloring (classification) from the MAP algorithm. As in Figure 3.8, the coloring was accurate except at the boundaries between different textures. The boundary effect is most severe at boundaries between various textures and texture T_{33}.

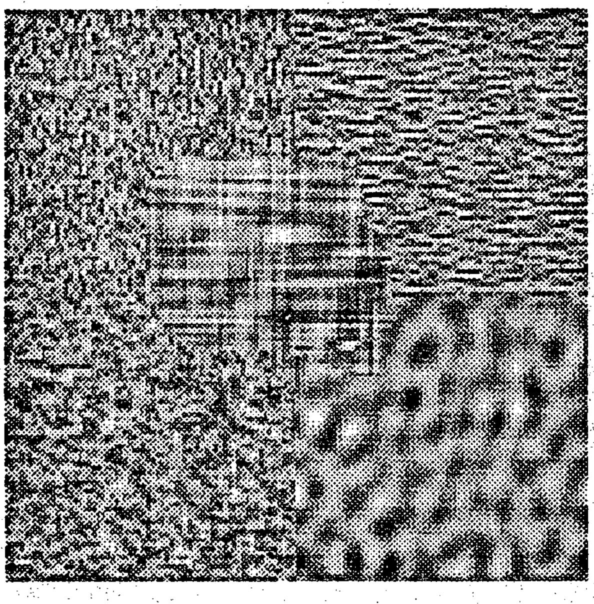
Finally, the 128 × 128 image in Figure 3.10a was constructed from the following textures: T_{33} (upper left), T_{32} (upper right), T_{31} (lower left), T_{42}

(lower right) and the texture in the middle was constructed from T_{14}. From Table 3.1, orders of the SAR model used were: second-, third-, second-, third-, and second-order, respectively. The strips used in the MAP algorithm were formed of three rows, and pixel classification was performed on 4 × 4 blocks. Figure 3.9b is the resulting coloring (classification) from the MAP algorithm. Again, the coloring was accurate except at the boundaries between different textures, and, in particular, at the boundaries of texture T_{14}.

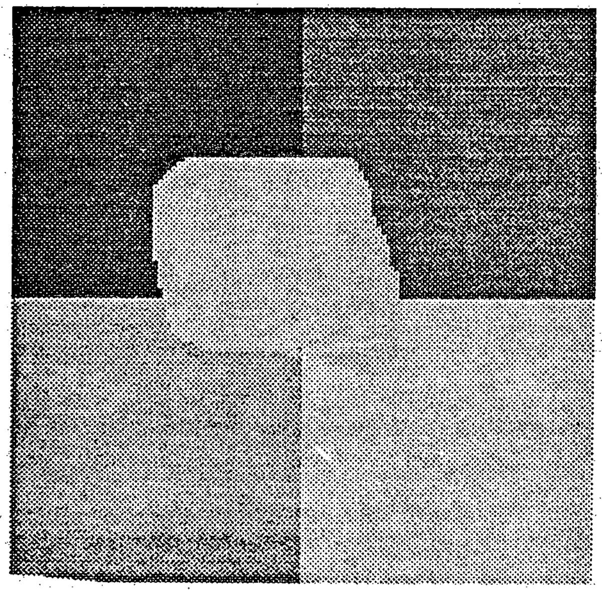
The above results clearly indicate that the MAP algorithm performs well on synthestic textures except at the nonuniform boundaries between various textures where the pixel's neighborhood is not well defined. These results indicate the importance of the parameters set $\{\theta^{l_k}\}_{k=1}^{M}$, ρ, for the segmentation algorithm. Almost no change in the classification was observed when we varied the value of the parameter β in (3.14). This might be due to the fact that the textures generated do not have macro-structure within them. That is, the textures have a large degree of microtextural content.

It is also important to notice that in spite of the fact that the histograms of all the images in Figure 3.3 are similar, yet the segmentation algorithm was able to correctly classify the pixels of each region accurately. This is indeed the most significant advantage of region-based segmentation over edge-based segmentation that will be examined in Chapters 4 and 5.

Finally, Figure 3.11a shows an image that was formed by adding (pixel by pixel) a synthestic texture (similar to T_{23}, except the size is 256 × 256) to an image that was formed from five rectangular regions with gray levels (starting from upper left to lower right) of 50, 100, 150, 200, respectively. The gray level histogram of the texture image is centered around a gray level value of 80 (very similar to the histogram in the second row and thrid column of Figure 3.4), and the addition of the two images resulted in a gray level distribution that is well within the 0-255 range with very few pixels that have a gray level outside this range. The purpose of running this experiment is test the sensitivity of the MAP algorithm to regions which have a nearly constant gray level distribution. Figure 3.11b shows the classification results. On the rectangular regions, less accurate classification was obtained especially on the triangles with higher gray levels. Within the texture itself, better

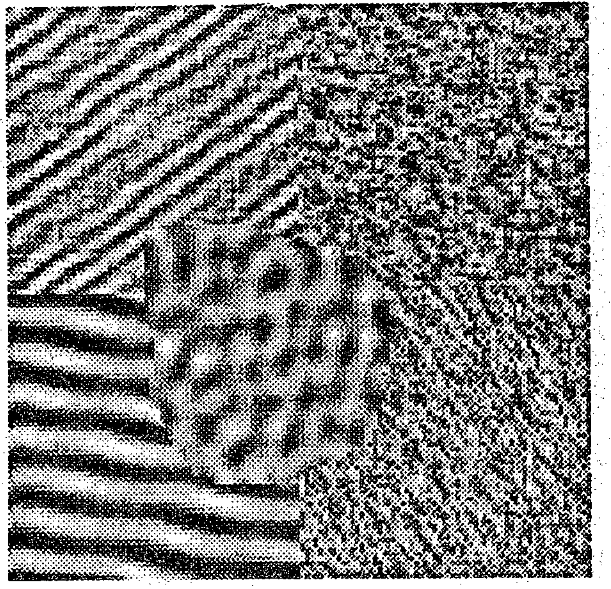


(a) Original textures.

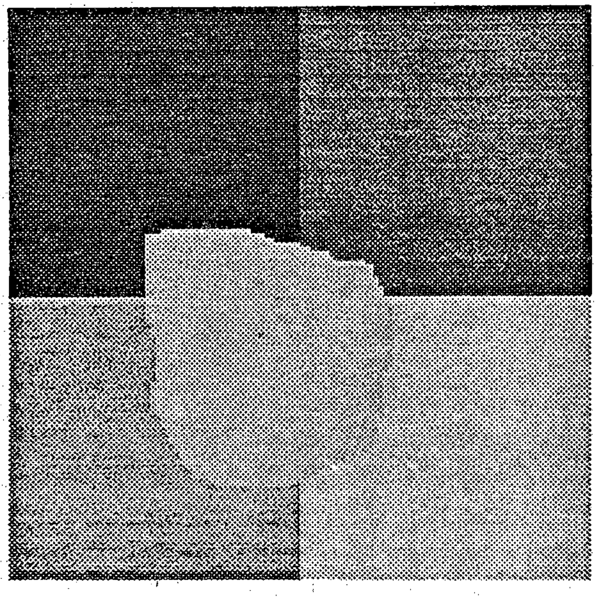


(b) MAP coloring results.

Figure 3.8. Segmentation of synthetic textures. T_{12}(upper left), T_{21}(upper right), T_{23} (lower left), T_{33}(lower right), and T_{11}(middle) in Figure 3.3.

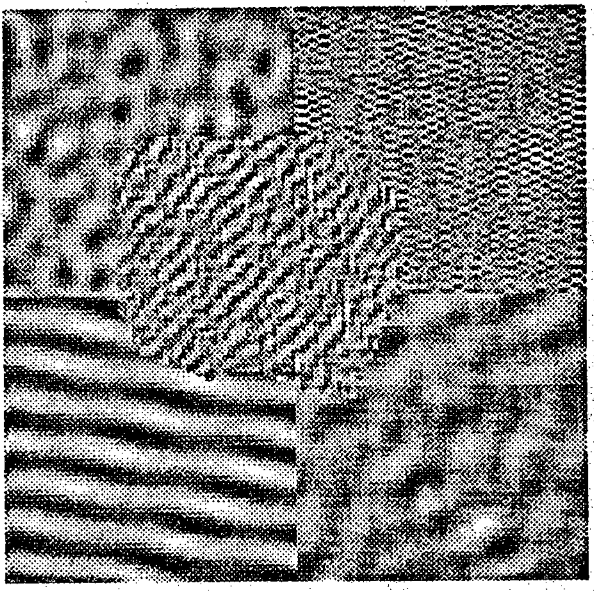


(a) Original textures.

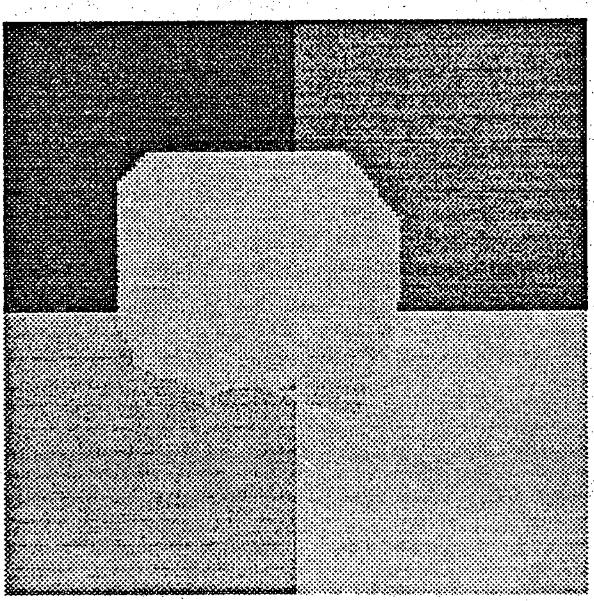


(b) MAP coloring results.

Figure 3.9. Segmentation of synthetic textures. T_{13}(upper left), T_{23}(upper right), T_{31} (lower left), T_{34}(lower right), and T_{33}(middle) in Figure 3.3.



(a) Original textures.



(b) MAP coloring results.

Figure 3.10. Segmentation of synthetic textures. T_{33}(upper left), T_{32}(upper right), T_{31} (lower left), T_{42}(lower right), and T_{14}(middle) in Figure 3.3.

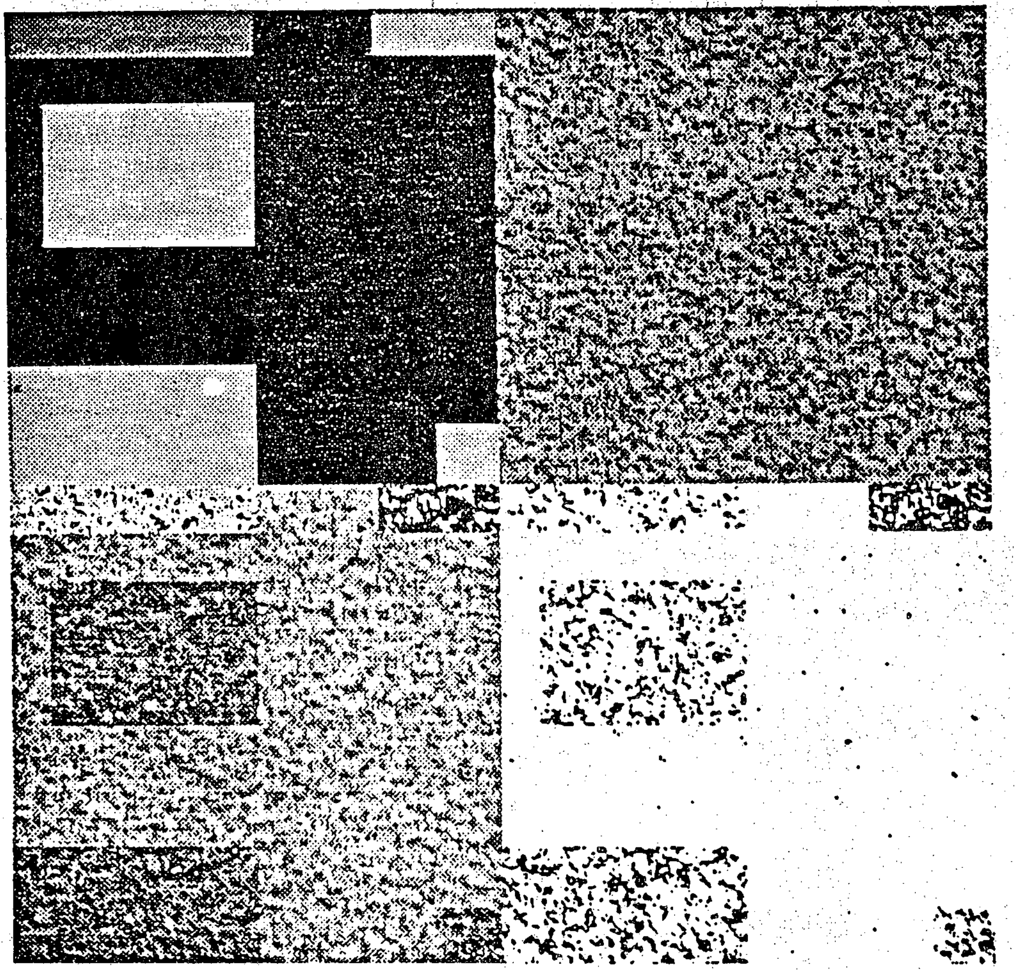
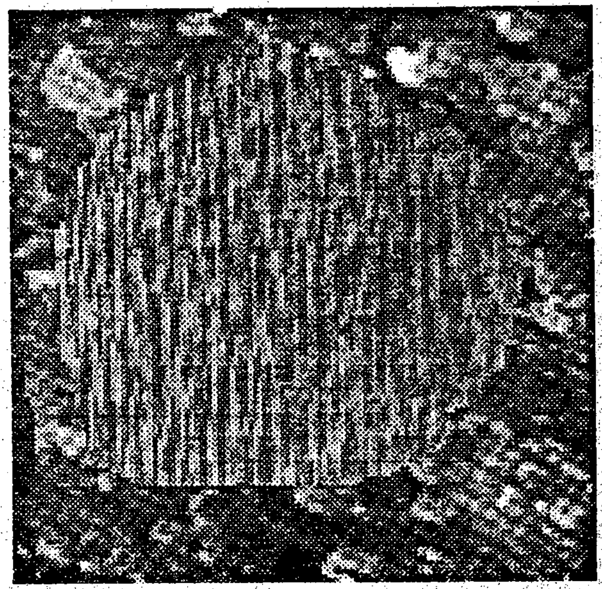
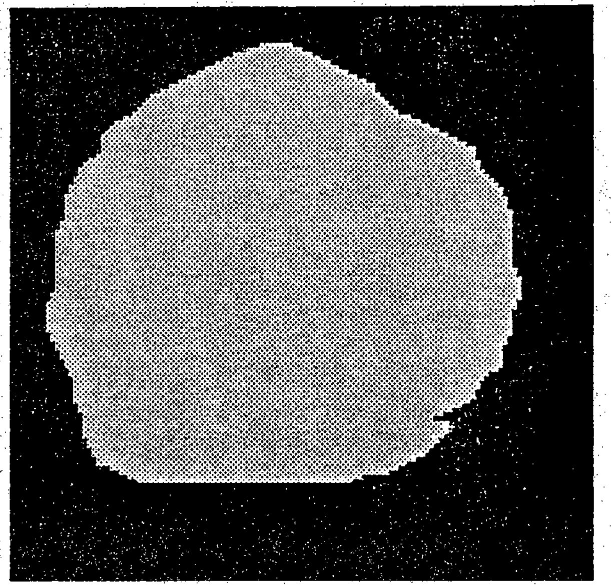


Figure 3.11. Segmentation of synthetic textures.
Texture T_{23} on top of rectangles with constant gray level.
(Upper left: original rectangles, upper right: original texture,
lower left: texture superimposed on rectangles, lower right: MAP coloring)

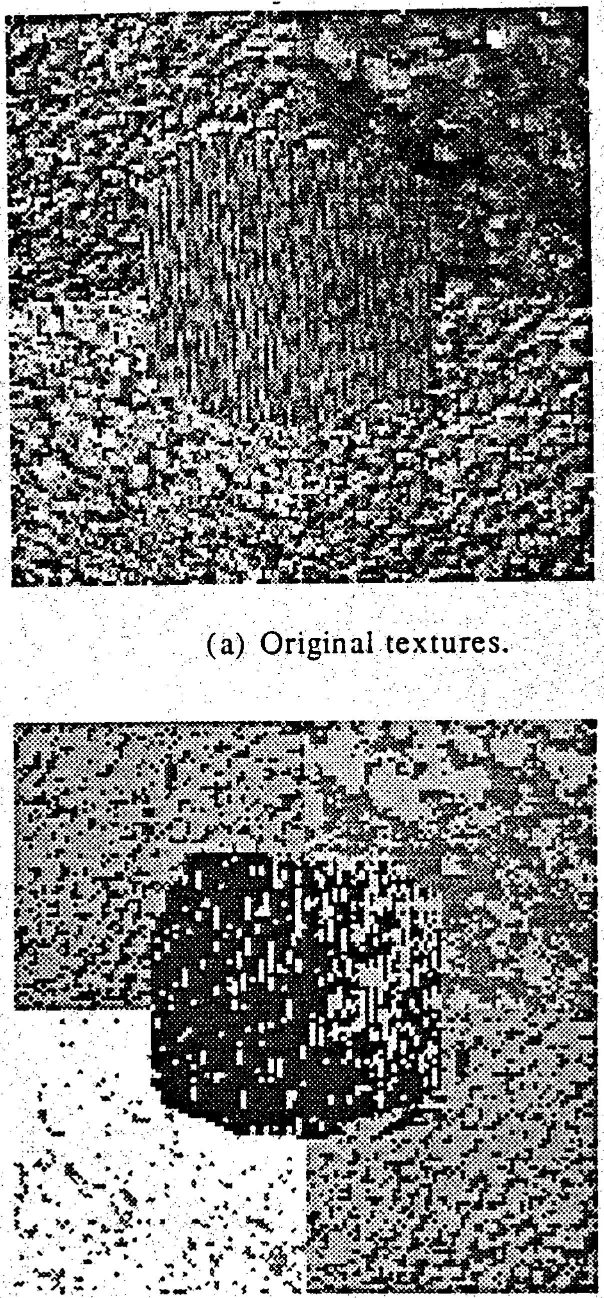


(a) Original textures.



(b) MAP coloring results.

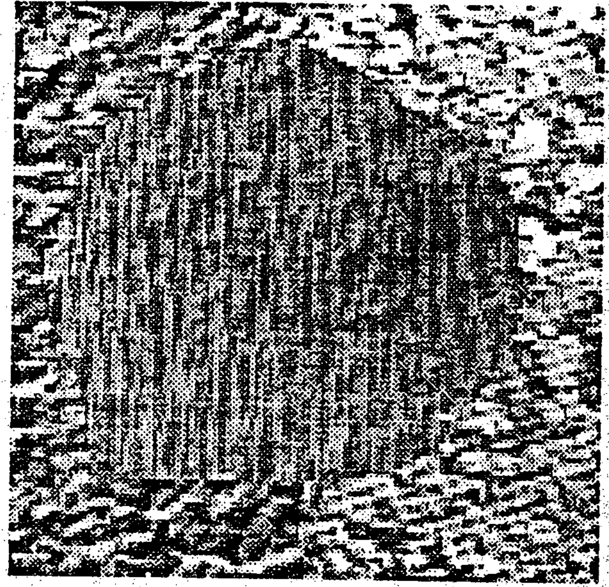
Figure 3.12. Segmentation of natural textures.
(Woody and Gems textures on Figure 3.5.)



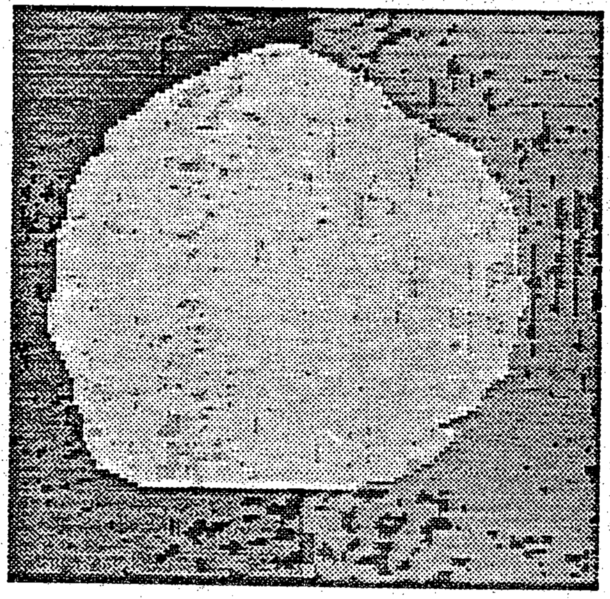
(a) Original textures.

(b) MAP Coloring results.

Figure 3.13. Segmentation of natural textures.
Cork (upper left), Gems (upper right), Peb (lower left),
Cork (lower right), Woody (middle) on Figure 3.5.



(a) Original textures.



(b) MAP Coloring results.

Figure 3.14. Segmentation of natural textures.
Cork (left), Peb (right), Woody (middle) on Figure 3.5.

classes in Figure 3.14 than in Figure 3.13. It is expected that the algorithm will perform better with fewer classes than with large number of classes.

From Figure 3.6, the histograms of the upper left hand and lower left hand textures in Figure 3.5 are quite similar. Also the histograms in the upper right hand and the lower right hand are similar. Hence our experimentation with synthestic and natural textures suggests that the MAP algorithm described in this chapter can indeed be used to classify various textures. It is to be emphasized, however, that only with images of microtextural content that the MAP algorithm performs best. Based on the results other researchers have shown (e.g. [GeG87], [DeE87], [BoL89]), the results of the MAP algorithm developed here is more accurate and much faster than was reported in [DeE87].

3.6. Summary

In this chapter we examined the region segmentation problem and introduced a new algorithm for MAP segmentation. A recursive implementation for the algorithm was outlined. Results of the algorithm on various synthestic and natural textures clearly indicate the effectiveness of the approach to texture segmentation. The simulation results in this chapter were obtained by the Kashyap algorithm presented in the Appendix.

CHAPTER 4
EDGE-BASED SEGMENTATION:
1. OPTIMAL EDGE ENHANCEMENT

4.1. Introduction

As indicated in Chapter 1, edge detection is a two-stage process: edge enhancement followed by edge linking. The edge enhancement stage specifically defines the edge content of the image. The enhanced edges are obtained following a specific protocol (e.g. optimality criterion in optimal edge enhancement filters). This protocol, in essence, provides a code for what is considered to be an edge. The second stage, edge linking, uses the edge information provided by the enhancement step to create a one-bit representation for the boundaries of objects in the image. The methods used to enhance the edges also provide a clue for linking the edges. For example, in the $\nabla^2 G$ operator the location of zero-crossings is the decoding method used to link the edges. Similarly, in the ∇G operator the points of maximum gradient provide the decoding used to link the edges.

As we pointed out in Chapter 1, no existing technique can provide accurate edge enhancement under all circumstances. Also, no technique brings about a satisfactory enhancement for all types of edges. Thus the linking stage is not at all trivial. In fact, a good edge linking technique should provide correction to the errors introduced by the enhancement step. The edge linking stage requires the knowledge of the technique used to enhance the edges (i.e. whether it is a Laplacian operator or a gradient operator) as well as the types of errors that are introduced by that technique. It is hoped that such knowledge can be used quantitatively to obtain the object boundaries accurately within a small probability of error.

The edge detection system presented here dictates the use of model-based techniques in both stages of the system. Due to the known problems with classic edge enhancement techniques (e.g. gradient operators, template matching, etc.), attention has been directed toward other methods derived from optimality constraints.

Let us define the following terms: In a certain configuration $\omega \in \Omega$, the intensity function (gray level distribution) defined on the support region of image is essentially a deterministic function. In the continuous spatial domain, the intensity function of the input image (the observed image) is denoted by $\mathbf{f}(x,y)$ and the desired output image is denoted by $\mathbf{g}(x,y)$. It is assumed that $\mathbf{g}(x,y)$ is the response of a continuous linear system (filter) whose impulse response is denoted by $\mathbf{h}(x,y)$. The corresponding discrete representation is denoted by $\mathbf{f}(m,n)$, $\mathbf{g}(m,n)$, and $\mathbf{h}(m,n)$, respectively. When the filter $\mathbf{h}(\cdot,\cdot)$ is given by the Gaussian kernel, it is distinguished by the script G. In other words, the filter is denoted by $\mathbf{h}_G(\cdot,\cdot)$. In the spatial frequency domain, the two-dimensional Fourier transforms for the above quantities are denoted by $F(\nu,\mu)$, $H(\nu,\mu)$, and $G(\nu,\mu)$, respectively. Finally, quite often results will be proved for the continuous one-dimensional case for simplicity and where two-dimensional generalization is a trivial extension. Similar notations are followed for either case. We stress that the sample functions $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are deterministic for any image configuration.

Figure 4.1 is a block diagram of the edge enhancement process, it illustrates the input-output terminology used in this chapter.

4.2. Maximum Energy Filters for Step Edges

Maximum energy filters applied to enhance step edges were first introduced by Dickey and Shanmugam [DiS77]. A detailed analysis was also given in Shanmugam et al. [ShD79]. Corrections to the original derivation were later provided by Lunscher [Lun83]. We will examine the maximum energy filter and demonstrate its relationship to the Mar-Hildreth $\nabla^2 G$ operator [MaH80] as well as to the Canny ∇G operator [Can83].

In the one-dimensional case (the two-dimensional case follow easily), this filter can be derived as follows: Let the input signal (image) be $\mathbf{f}(x)$, the output signal be $\mathbf{g}(x)$, and the filter impulse response be $\mathbf{h}(x)$. Let $F(\nu)$, $G(\nu)$, and $H(\nu)$ be the corresponding Fourier transforms. Let the edge width, the small region over which a distinct intensity change occurs with respect to surrounding parts of the image, be τ. This is also referred to as

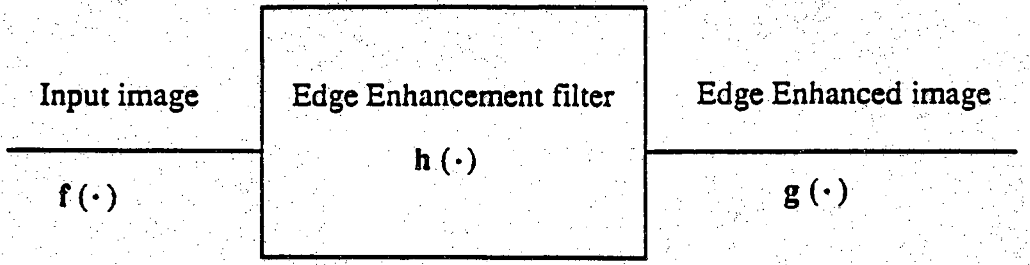


Figure 4.1. A block diagram of the edge enhancement process.

the edge resolution. We will assume that the edge is ideally centered at the origin in the spatial domain and that the important edge information is within some finite frequency range, $\nu \in [-W, W]$. Let the input image be a unit step function, that is, $f(x) = u(x)$. The optimization problem is: Choose $H(\nu)$ which maximizes

$$\gamma = \frac{\int_{-\tau/2}^{\tau/2} |g(x)|^2 \, dx}{\int_{-\infty}^{\infty} |g(x)|^2 \, dx} \tag{4.1}$$

with the constraints

$$F(\nu) = \frac{1}{j2\pi\nu} + \frac{1}{2}\delta(\nu), \tag{4.2a}$$

$$G(\nu) = H(\nu)F(\nu), \tag{4.2b}$$

$$H(\nu) = 0 \quad \text{for } |\nu| > \ W, \text{ and } \nu = 0. \tag{4.2c}$$

Equation (4.2a) is the Fourier transform of a step edge, and (4.2b) implies that the desired filter is to be linear and space-invariant. Equation (4.2c) implies that the desired $H(\cdot)$ is band limited and has a zero dc response (i.e. is less sensitive to constant or slowly varying input.) The required solution is based on the properties of prolate spheroidal functions [SlP61].

The pertinent properties of the prolate spheroidal wave functions are the following. Given any $\tau > \ 0$ and any $W > \ 0$, we can find a countably infinite set of real functions $\psi_0(x), \psi_1(x), \psi_2(x), \cdots$ and a set of real positive numbers

$$\lambda_0 > \lambda_1 > \lambda_2 > \ \dots .$$

with the following properties:

$$\int_{-\infty}^{\infty} \psi_i(x)\ \psi_j(x) dx = \begin{cases} 0 & \text{for } i \neq j \\ 1 & i = j. \end{cases} \tag{4.3}$$

In the interval $\tau/2 \leq \ x \leq \tau/2$, the $\psi_i(x)$ are orthogonal and complete in $L^2_{\tau/2}$ subspace, that is,

$$\int_{-\tau/2}^{\tau/2} \psi_i(x)\ \psi_j(x) dx = \begin{cases} 0 & \text{for } i \neq j \\ \lambda_i & i = j \end{cases} \tag{4.4}$$

and for a general x (real, or complex),

$$\lambda_n \psi_n(x) = \int_{-\tau/2}^{\tau/2} \frac{\sin W(x - s)}{\pi (x - s)} \psi_n(s) ds, \quad n = 0, 1, 2, \dots , \qquad (4.5)$$

In the above equation, the quantities ψ's and the λ's are functions of the product τW [SIP61]. This dependence can be made explicit if we write

$$\lambda_i = \lambda_i(c), \text{ and}$$

$$\psi_i(x) = \psi_i(c,x), \ i = 0, 1, 2, \dots .$$

where $2c = \tau W$. In the above equation, $\psi_i(c,x)$ is a function of x that depends on the product of the spatial and frequency resolutions of the step edge.

4.2.1. Rotation Invariant Filter

Consider the following constraint on $H(\nu)$:

$$H(\nu) = H(-\nu). \qquad (4.6)$$

This constraint restricts the desired $H(\cdot)$ to be even (i.e. rotation-invariant in the two-dimensional case). From (4.2a), $F(\nu)$ is an odd function, hence $G(\nu)$ is odd; therefore, the inverse Fourier transform $g(x)$ must be an odd function.

Since the odd-numbered prolate spheroidal wave functions are odd, the maximum energy solution can be written in the following form:

$$g(x) = \sum_{n(\text{odd})} a_n \, \psi_n(c,x), \qquad (4.7)$$

where $\psi_n(c,x)$ is the n^{th} prolate spheroidal functions of order zero [SIP61]. It can be obtained (theoretically) from the integral in (4.5) if $\lambda_n(c)$ is known. Substituting (4.7) into (4.1) and using the properties (4.3 - 4.5), γ can be written as

$$\gamma = \frac{\sum\limits_{n(\text{odd})} |a_n|^2 \lambda_n}{\sum\limits_{n(\text{odd})} |a_n|^2}. \qquad (4.8)$$

Now, since the values λ_n, $n = 1, 2, 3, \dots$ are positive and of decreasing values, γ is bounded below by zero and above by λ_1, that is,

$$0 < \gamma \leq \frac{\lambda_1 \sum_{n(odd)} |a_n|^2}{\sum_{n(odd)} |a_n|^2} = \lambda_1. \tag{4.9}$$

Hence, the maximum energy is obtained when $a_n = 0$ for $n > 1$; thus, the eigenvalue λ_1 is the maximum γ for a specified value of c. Therefore, the optimum output will have the following form:

$$g(x) = a_1\, \psi_1(c,x). \tag{4.10}$$

Now, $H(\nu) = G(\nu)/F(\nu)$ for step edges is obtained from the Fourier transform of $\psi_1(c,x)$ and (4.2a). It is easy to show that the required filter has the following transfer function:

$$H_{step}(\nu) = \begin{cases} k_1 \nu \psi_1(c, \nu\tau/2W) & \text{for } |\nu| < W \\ 0 & \text{elsewhere} \end{cases} \tag{4.11}$$

where

$$c = \frac{W\tau}{2}. \tag{4.12}$$

In (4.11), observe that the Fourier transform for the function $\psi_1(x)$ has the same form, but x has been replaced by $\nu\tau/2W$.

The function $\psi_n(c,x)$ can be approximated by the following form [SlP61]:

$$\psi_n(c,x) = \frac{\sqrt{\lambda_n(c)}}{u_n(c)} S_{on}(c, 2x/\tau) \tag{4.13}$$

where

$$[u_n(c)]^2 = \int_{-1}^{1} [S_{on}(c,x)]^2 dx, \tag{4.14}$$

and the functions $S_{on}(c,x)$ are orthogonal and is band limited to the interval $\nu = (-\,c,c)$. An efficient approximation, due to Streifer [Str68], for $S_{on}(c,x)$ of the form (4.15):

$$S_{on}(c,x) = \left(\frac{c}{\pi}\right)^{1/4} 2^{-n/2}\, (n!)^{-1/2} \cdot \exp(-\,cx^2/2)\; T_n(c^{1/2}x) \tag{4.15}$$

for $|x| < c^{-1/4}$ and $n << c$. $T_n(\cdot)$ is the *nth* Hermite polynomial defined by [Fra81]

$$T_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2}), \tag{4.16}$$

with the recursion formula

$$T_n(x) = 2xT_{n-1}(x) - 2(n-1)T_{n-2}(x). \tag{4.17}$$

For example,

$$T_0(x) = 1, \quad T_1(x) = 2x, \quad T_2(x) = 4x^2 - 1, \text{ etc.} \tag{4.18}$$

The approximation to $\psi_n(\cdot)$ becomes

$$\psi_n(c,x) \approx \frac{\sqrt{\lambda_n(c)}}{u_n(c)} \left(\frac{c}{\pi}\right)^{1/4} 2^{-n/2} (n!)^{-1/2}$$

$$\cdot \exp(-2cx^2/\tau^2)\, T_n(c^{1/2}\, 2x/\tau) \tag{4.19}$$

for $|x| < \dfrac{\tau c^{-1/4}}{2}$.

For $n = 1$, the function $\psi_1(\cdot)$ can be obtained by substituting $T_1(x) = 2x$ into (4.19) which results in

$$\psi_1(c,x) \approx kx \exp(-2cx^2/\tau^2), \tag{4.20}$$

where k is a constant. Now let $x = \nu\tau/2W$ in (4.19) and substituting into (4.11) gives the asymptotic approximation to the optimal step edge filter in the following form

$$H_{step}(\nu) = \begin{cases} k_2\, \nu^2 \exp(-c4\pi^2\nu^2)/2W^2 & \text{for } |\nu| < W \\ 0 & \text{elsewhere} \end{cases} \tag{4.21}$$

which is strictly valid on $|\nu| < Wc^{-1/4}$ [ShD79] [Lun83]. The parameter c in (4.21) provides a tuning factor for $H(\cdot)$. If c is chosen to be large, edge resolution decreases but edge visibility increases (i.e. edges grow thicker) and vice versa. For detecting blurred edges, c is chosen so that the edge resolution, τ, is greater than the width of the blurred edge. The two-dimensional form of (4.21) can be easily written as follows:

$$H_{step}(\nu,\mu) = \begin{cases} k\,(\nu^2 + \mu^2) \exp(-\,c4\pi^2(\nu^2 + \mu^2))/2W^2 & \text{for } |\nu|, |\mu| < W \\ 0 & \text{elsewhere} \end{cases} .$$

In their implementation, Shanmugam et al. thresholded the output. The results for a step edge were two peaks on either side of a zero-valued valley. Because the filter was restricted to be even, the output was odd. Thus thresholding is not justified since it ignores the all-important issue of edge localization.

The similarity of the filter in (4.21) to the Laplacian of the Gaussian filter of Marr and Hildreth [MaH80], the $\nabla^2 G$ operator, is evident. In [MaH80], edges are allocated at the zero-crossings instead of thresholding the output. The exponent of (4.21) clearly is dimensionless, thus the filter is invariant to scale transformation (i.e. isotropic).

Figure 4.2 shows the one- and two-dimensional Laplacian of the Gaussian filter.

4.2.2. A Directional Filter

Now consider the same set of constraints presented in (4.2a - 4.2c). By removing constraint (4.6) the filter will not be restricted to be even. The bandlimited filter output has the form in (4.7) and $0 \leq n \leq \infty$, that is,

$$g(x) = \sum_{n=0}^{\infty} a_n \, \psi_n(c,x), \qquad (4.22)$$

where c is still as in (4.12).

Again the values λ_i are positive and of descending order; hence γ in (4.1) is bounded above by λ_0. Thus, the maximum energy solution has the form

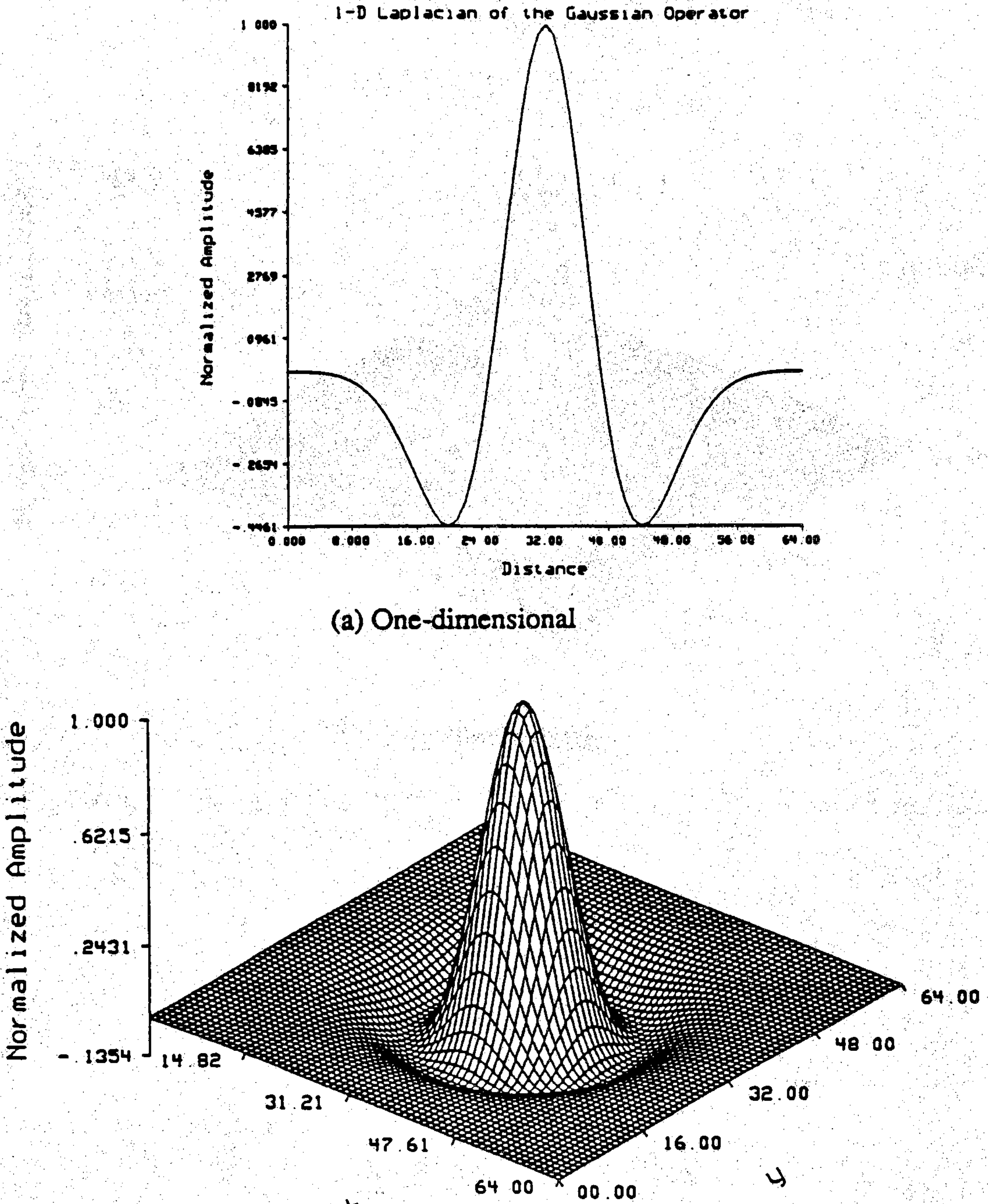
$$g(x) = a_0 \, \psi_0(c,x). \qquad (4.23)$$

Using constraints (4.2a) and (4.2b), the filter $H(\nu)$ can be obtained as the inverse Fourier transform of $G(\nu)/F(\nu)$. Using the approximations in (4.18) and (4.19), $\psi_0(c,x)$ can be written as

$$\psi_0(c,x) = k_1 \exp(-\, 2cx^2/\tau^2) \qquad (4.24)$$

where k_1 is a constant. The Fourier transform of (4.24) is

$$\mathcal{F}\{\psi_0(c,x)\} = k_1 \sqrt{\frac{\pi\tau^2}{2c}} \exp(-\, (2\pi u)^2\tau^2/8c). \qquad (4.25)$$



(a) One-dimensional

(b) Two-dimensional

Figure 4.2. One- and two-dimensional Laplacian of the Gaussian filter.

From (4.2a) and (4.24), after taking the inverse Fourier transform, we get the filter impulse response in the form

$$\mathbf{h}(x) = k\, x \exp(-\, x^2/2\sigma^2) \tag{4.26a}$$

where

$$\sigma = \sqrt{\frac{\tau}{2W}}, \tag{4.26b}$$

$$k = \frac{k_1 2c}{\tau^2}. \tag{4.26c}$$

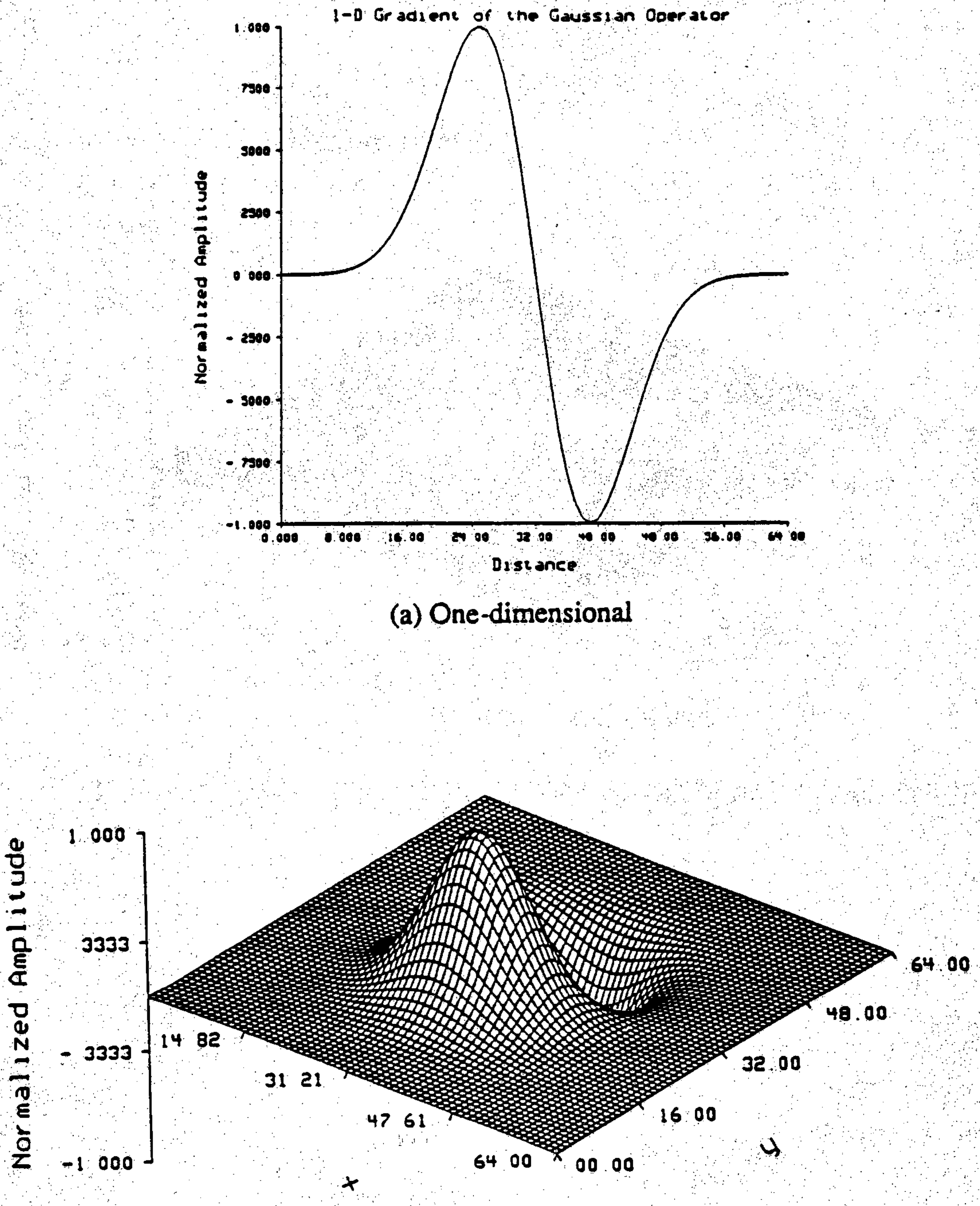
The filter in (4.26) is similar to the ∇G operator, which compares favorably to the optimal filter for step edge enhancement of Canny ([Can83], [Can86]).

The above treatment shows that the maximum energy filter embodies a number of edge operators. Hence the optimality of the $\nabla^2 G$ operator [MaH80] and the directional ∇G operator [Can83] can be explained in terms of the maximum energy criterion used in [DiS77] and [ShD79]. The relationship between these filters can be used to study the issue of filter support in the $\nabla^2 G$ and the ∇G operators.

Figure 4.3 shows the one- and two-dimensional ∇G filter. The filter is odd symmetric across the origin.

4.2.3. Regularization Filters

The rationale behind the design of the majority of edge enhancement operators is that the edge information lies on local maxima of the image intensity function. Differentiation is, of course, one of the well known optimization techniques used to obtain the location of local maxima. The problem with numerical data is that the function is defined only at specific locations in the domain of support. Since differentiation does not depend on the data continuously, edge enhancement (based on differentiation) is an ill-posed problem in the sense of Hadamard [BeP88]. As a consequence, numerical differentiation may not provide a unique solution (edge definition), and it may produce an amplification of high frequency noise. To reduce the noise effect and to fill-in wherever data are missing or not reliable,



(a) One-dimensional

(b) Two-dimensional

Figure 4.3. One- and two-dimensional Gradient of the Gaussian filter.

regularization techniques (filters) are used (e.g. [KaS88], [BeP88]). Hence, the purpose of the regularization filters in edge enhancement is to convert the ill-posed numerical differentiation problem into a well-posed problem.

The optimum filters we studied previously can be separated into two operators: a smoothing operator (filter) followed by a differentiation operator. The smoothing filter serves as a regularization filter. The Gaussian filter is only one possible choice from the class of regularization filters. We will describe, briefly, a few properties of this class of filters.

Suppose that the filter $\mathbf{h}(\cdot)$ can be parameterized by a single parameter ρ, that is, $\mathbf{h}(x) = \mathbf{h}(x,\rho)$ for one-dimensional filter, and the parameter ρ is positive (i.e. $\rho \geq 0$.) Let $H(\nu,\rho)$ be its Fourier transform. $H(\nu,\rho)$ is called a regularization filter if it satisfies the following conditions ([TiA77], [ToP86], [BaC88], [BeP88]):

(1) $H(\nu,\rho)$ is bounded for $\rho \geq 0$ and all ν.

(2) $H(\nu,\rho)$ is an even function with respect to ρ, and it is square integrable (i.e. $H(\nu,\rho) \in L_2$).

(3) $j\nu H(\nu,\rho) \in L_2$.

(4) $\lim_{\nu \to \infty} H(\nu,\rho) = 0$ for all $\rho > 0$.

(5) $\lim_{\rho \to 0} H(\nu,\rho) = 1$ for all ν and, $H(\nu,0) = 1$ for all ν.

It is easy to show that the Gaussian filter satisfies all the above properties. In this case, the parameter ρ is simply the standard deviation of the Gaussian kernel (i.e. $\rho = \sigma_f$.) The Gaussian filter is also a low pass filter with cutoff frequency which is a function of the standard deviation of the filter. The previously examined optimal filters that combine the regularization and numerical differentiation processes were based on the maximum energy output criterion. Since the edges correspond to high spatial frequencies in the image, to avoid a large degree of blur, the Gaussian filter must not induce excessive spectrum alteration.

Results of the regularization theory suggest that whenever the above criterion is met, the actual shape of the filter is not important. This might also explain why so many of the edge operators in the literature have a similar form, as we pointed out above (see also [BaC88]).

4.3. Selection of Filter Spatial Support

Edges in real world images are not necessarily located at abrupt changes in intensity (i.e., they are not in the form of ideal step functions). However, some insight into the nature of filter spatial support can be gained by using step-edge models. Two such models were used by Lunscher and Beddoes [LuB86a] in their study on the $\nabla^2 G$ operator. By modeling the edge as an ideal step edge blurred by a Gaussian kernel, they were able to derive a design criterion for the $\nabla^2 G$ operator spatial support in terms of the step-edge width, degree of blur, and the signal-to-noise ratio (SNR). We will derive a Lunscher-Beddoes type of criterion for the size of the ∇G operator (4.26) using the same edge models in [LuB86a].

4.3.1. Staircase Edges

The edge model used to represent a blurred infinite staircase edge of ascending magnitude is:

$$\mathbf{f}(x) = \mathbf{h}_G(x,\sigma_b) * \sum_{n=-\infty}^{\infty} u(x - nT), \tag{4.27}$$

where $\mathbf{h}_G(x,\sigma_b)$ is the Gaussian kernel describing the blur and is defined as

$$\mathbf{h}_G(x,\sigma_b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp(-\frac{x^2}{2\sigma_b^2}). \tag{4.28}$$

The response of the ∇G operator to the edge function in (4.27) is given by the following equation:

$$\mathbf{g}(x) = \nabla G * \mathbf{f}(x), \tag{4.29a}$$

that is,

$$\mathbf{g}(x) = \nabla \mathbf{h}_G(x,\sigma_f) * \mathbf{f}(x), \tag{4.29b}$$

$$= \mathbf{h}_G(x,\sigma_f) * \mathbf{h}_G(x,\sigma_b) * \sum_{n=-\infty}^{\infty} \delta(x - nT), \tag{4.29c}$$

where $\mathbf{h}_G(x,\sigma_f)$ is the Gaussian kernel with standard deviation σ_f used to smooth the edge before taking the derivative. In (4.29) we used the fact that the convolution and differentiation are commutative and that the derivative of the step function is the Dirac delta function,

$$\frac{d}{dx} u(x - nT) = \delta(x - nT). \tag{4.30}$$

Edges are located at the maximum directional derivative (maximum gradient in the two-dimensional case). Taking the derivative of (4.30) and setting the result to zero, it is easy to show that the edges are located at $x = - nT, n = 0, 1\ 2, , \cdots$.

Edge strength can be evaluated easily by transforming (4.29) into the frequency domain. The Fourier transform of the Gaussian pulse is given by

$$H_G(\nu) = \exp(- 2\pi^2\sigma^2\nu^2). \tag{4.31}$$

The Fourier transform of (4.29) is

$$G(\nu) = \exp\big(2\pi^2(\sigma_f^2 + \sigma_b^2)\nu^2\big) \cdot \frac{1}{T} \sum_{n = -\infty}^{\infty} \delta(\nu - \frac{n}{T}), \tag{4.32}$$

which can be written as

$$G(\nu) = \frac{1}{T} \sum_{n = -\infty}^{\infty} \exp(- 2\pi^2(\sigma_f^2 + \sigma_b^2)n^2/T^2) \cdot \delta(\nu - \frac{n}{T}). \tag{4.33}$$

To compare the results we derived for the ∇G operator with the $\nabla^2 G$ results derived in [LuB86a], we define the following parameters:

$$\alpha = \sigma_b/\sigma_f \tag{4.34}$$

$$\beta = T/\sigma_f. \tag{4.34}$$

Thus, (4.32) becomes

$$G(\nu) = \frac{1}{\beta\sigma_f} \sum_{n = -\infty}^{\infty} \exp(- 2\pi^2 n^2(1 + \alpha^2)/\beta^2) \cdot \delta(\nu - \frac{n}{T}). \tag{4.36}$$

Recall that the output g(x) results from convolution of the edge enhancement filter $h(x) = \nabla G$ and the input signal (image) f(x). Edges are allocated at the points of maximum response (i.e., the origin for the ideal step-edge model.) Hence, the edge strength can be readily obtained as follows:

$$|g(0)| = |\int_{-\infty}^{\infty} G(\nu)d\nu|. \tag{4.37}$$

This is easily evaluated to be

$$|g(0)| = \frac{1}{\beta\sigma_f} \sum_{n=-\infty}^{\infty} \exp(-2\pi^2 n^2(1+\alpha^2)/\beta^2). \tag{4.38}$$

Due to symmetry of the exponent in (4.38), a further simplification follows:

$$|g(0)| = \frac{2}{\beta\sigma_f} [0.5 + \sum_{n=1}^{\infty} \exp(-2\pi^2 n^2(1+\alpha^2)/\beta^2)]. \tag{4.39}$$

The plot of (4.39) vs β and α is shown in Figure 4.4. The following measures can be readily obtained: For small α (e.g. $\alpha < 0.2$), the 3dB value of the magnitude response is at $\beta = 0.27$. For large α, the 3dB value of the magnitude response is at $\alpha = 0.51$ for $\beta \geq 5.5$. Outside the 3dB-level region in Figure 4.4 (the plateau), the magnitude response decays linearly in dB per decade α, and at a higher rate per decade β. The edges can be separated if

$$\begin{aligned} \sigma_b &\leq 0.6\sigma_f \\ T &\geq 7.6\sigma_b. \end{aligned} \tag{4.40}$$

For negligible blur, edges are separated if $T \geq 6.7\sigma_f$. Comparing with the $\nabla^2 G$ figures [LuB86a], edges are fully resolved for reasonable blur if

$$\begin{aligned} \sigma_b &\leq 0.51\sigma_f \\ T &\geq 5.5\sigma_b, \end{aligned} \tag{4.41}$$

and for negligible blur if $T \geq 2.75\sigma_f$. Hence, the filter spatial support is smaller if the directional derivative is used as the criterion for edge localization, rather than the zero-crossings criterion. This is in accordance with what many authors have found: That the ∇G operator requires a narrower filter and thus provides better spatial localization than the Laplacian [BaC88].

4.3.2. Square Wave Edges

The edge model for periodically rising and falling intensity is given by the following relation [LuB86a]:

$$f(x) = \mathbf{h}_G(x,\sigma_b) * u(x)\, u(T-x) * \sum_{n=-\infty}^{\infty} \delta(x - 2nT) \tag{4.42}$$

where $\mathbf{h}_G(x,\sigma_b)$ is given by (4.29) and $u(\cdot)$ is the unit step function. The response of the ∇G operator is given by

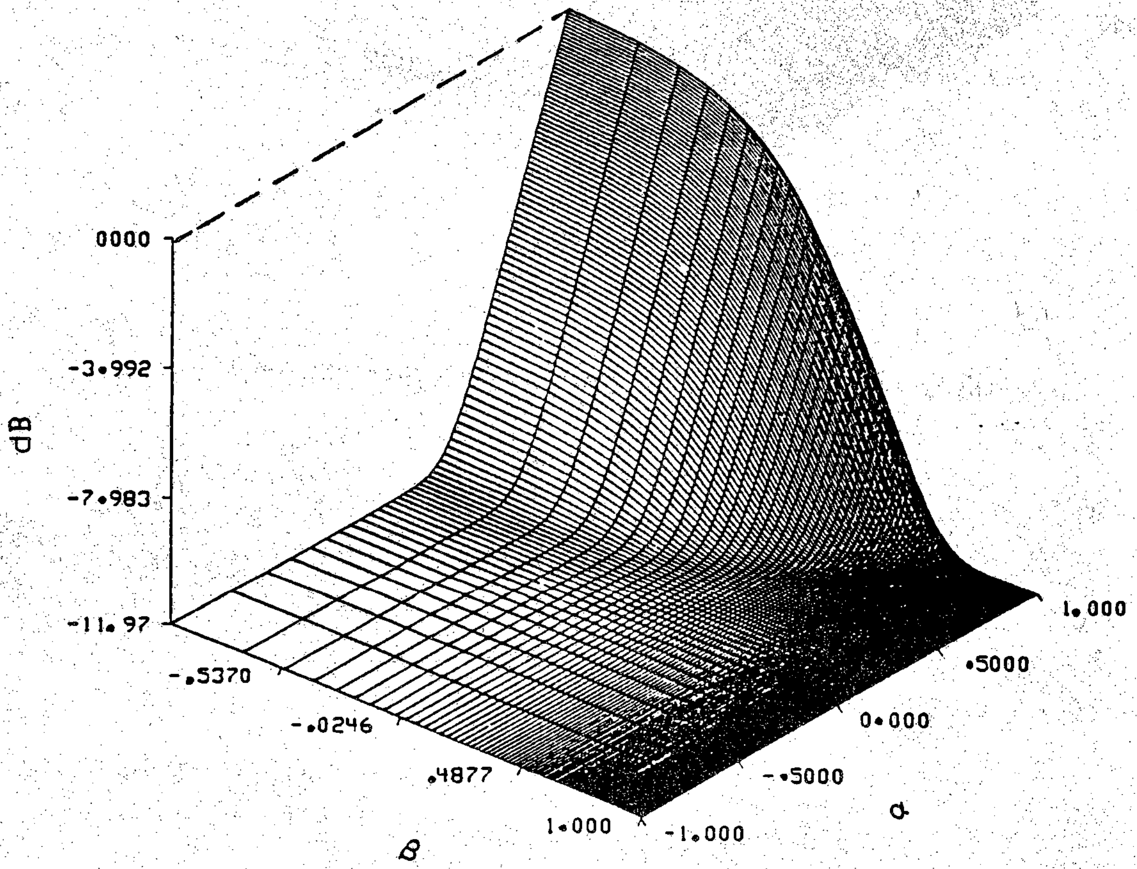


Figure 4.4. Magnitude response for the staircase edges.

$$\begin{aligned} g(x) &= \nabla h_G(x,\sigma_f) * f(x) \\ &= h_G(x,\sigma_f) * h_G(x,\sigma_b) * [\nabla u(x)u(T-x)] * \sum_{n=-\infty}^{\infty} \delta(x-2nT) \\ &= h_G(x,\sigma_f) * h_G(x,\sigma_b) \\ &\quad * [u(T-x)\delta(x) - u(x)\delta(T-x)] * \sum_{n=-\infty}^{\infty} \delta(x-2nT). \end{aligned} \tag{4.43}$$

It is easy to carry the right most convolution of (4.43) which simplifies to

$$g(x) = h_G(x,\sigma_f) * h_G(x,\sigma_b) * \left\{ \sum_{n=-\infty}^{\infty} \delta(x-2nT) - \delta(x-(2n+1)T) \right\}. \tag{4.44}$$

The Fourier transform of g(x) is

$$\begin{aligned} G(\nu) &= \exp(-2\pi^2\nu^2(\sigma_f^2+\sigma_b^2)) \\ &\quad \cdot \left\{ \frac{1}{2T} \sum_{n=-\infty}^{\infty} \delta(\nu - \frac{n}{2T}) - \frac{1}{2T} e^{-j2\pi\nu T} \sum_{n=-\infty}^{\infty} \delta(\nu - \frac{n}{2T}) \right\} \end{aligned} \tag{4.45}$$

which can be simplified to the following form:

$$\begin{aligned} G(\nu) &= \frac{1}{2T} \sum_{n=-\infty}^{\infty} (1 - e^{-j\pi n}) \\ &\quad \cdot \exp(-2\pi^2(\sigma_f^2+\sigma_b^2)n^2/4T^2)\delta(\nu - \frac{n}{2T}). \end{aligned} \tag{4.46}$$

Again let $\alpha = \sigma_b/\sigma_f$ and $\beta = T/\sigma_f$. Substituting in the above equation we obtain

$$\begin{aligned} G(\nu) &= \frac{1}{2\beta\sigma_f} \sum_{n=-\infty}^{\infty} (1 - e^{-j\pi n}) \\ &\quad \cdot \exp(-\pi^2 n^2(\alpha^2+1)/2\beta^2)\ \delta(\nu - \frac{n}{2T}). \end{aligned} \tag{4.47}$$

The edge strength is

$$\begin{aligned} |g(0)| &= \left|\int_{-\infty}^{\infty} G(\nu)d\nu\right| \\ &= \left|\frac{1}{2\beta\sigma_f} \sum_{n=-\infty}^{\infty} (1 - e^{(-j\pi n)}) \exp(-\pi^2 n^2(\alpha^2+1)/2\beta^2)\right| \end{aligned}$$

$$= \frac{1}{2\beta\sigma_f} \sum_{n=-\infty}^{\infty} (1 - \cos(\pi n)) \exp(-\pi^2 n^2(\alpha^2 + 1)/2\beta^2)$$

$$= \frac{1}{\beta\sigma_f} \sum_{n=1}^{\infty} (1 - \cos(\pi n)) \exp(-\pi^2 n^2(\alpha^2 + 1)/2\beta^2). \qquad (4.48)$$

The plot of (4.48) vs β and α is shown in Figure 4.5. The following measures can be obtained: The 3dB value of the edge strength in (4.48) for small α (e.g. $\alpha < 0.2$) is obtained at $\beta = 6.7$. For large α, the 3dB is reached first for $\alpha = 0.5$ and $\beta \geq 7.36$. Steady state 3dB is reached at $\alpha < 0.6$, $\beta \geq 7.6$. The response decays also linearly in dB per decade α, and at a much higher rate per decade β. Therefore it can be concluded that a blurred square wave image is fully resolved for

$$\sigma_b \leq 0.5\sigma_f \text{ and } T \geq 7.36\sigma_f, \qquad (4.49)$$

and for the negligible blur case, $T \geq 6.7\sigma_f$. The figures obtained for the $\nabla^2 G$ were [LuB86a]

$$\sigma_b \leq 0.51\sigma_f, \text{ and } T \geq 1.36\sigma_f, \qquad (4.50)$$

and for negligible blur, and $T \geq 1.15\sigma_f$. Again, note that the filter spatial support needed for edge localization based on the directional derivative is smaller as compared to the spatial support required when the zero-crossings criterion is used.

An analysis similar to the above can be performed if a known noise distortion (e.g. Gaussian), in addition to the blur is considered. In the $\nabla^2 G$ analysis and with Gaussian noise included (i.e. the edge model includes another Gaussian convolution), it was shown that at high SNR the measures for filter spatial support did not change significantly [LuB86a], [LuB86b]. The same conclusion can be drawn also in the case of the ∇G operator. Finally, due to the separability of the two-dimensional Gaussian kernel, the issue of filter spatial support in two-dimensions is essentially the same as in one-dimension. Practical aspects of digital implementation of the $\nabla^2 G$ operator have been examined in [LuB86b] (see also [Log77]).

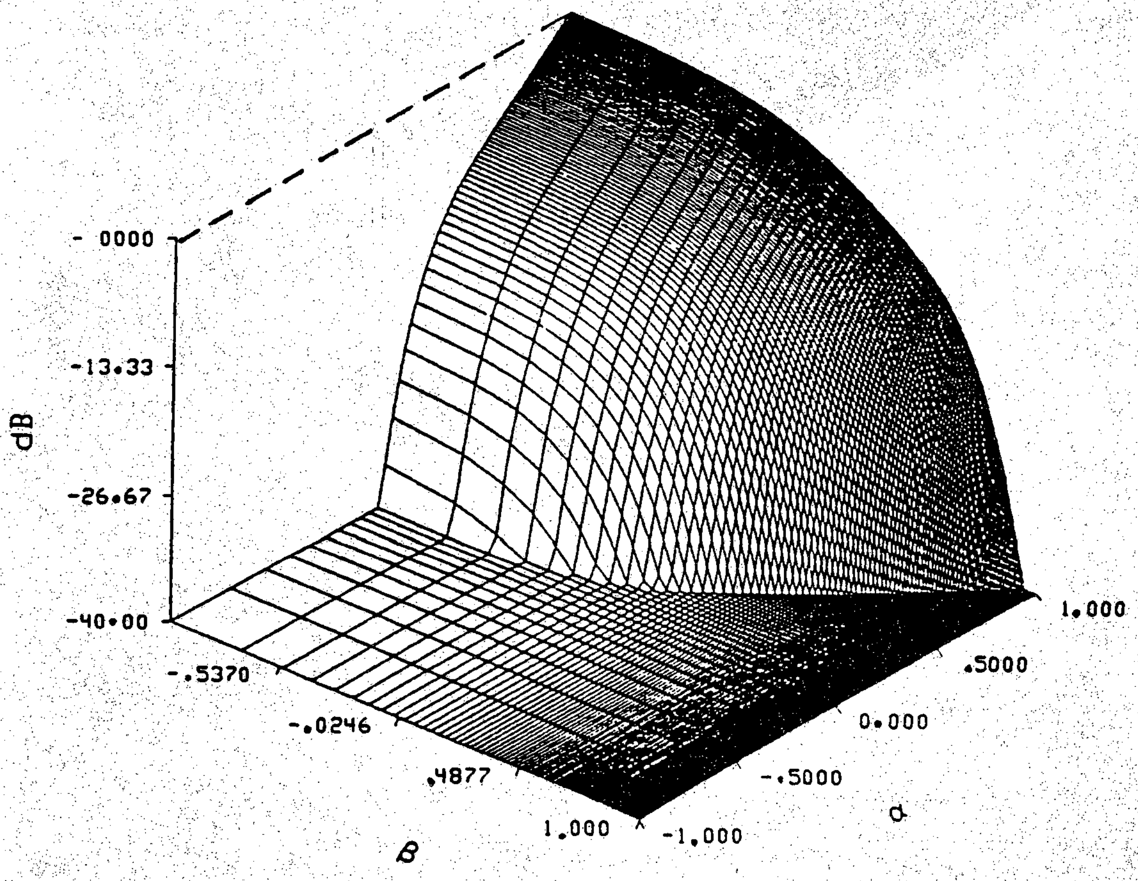


Figure 4.5. Magnitude response for the square wave edges.

4.4. Discussion and Conclusions

First, we emphasize that the filter of Shanmugam et al. [ShD79], the $\nabla^2 G$, and the ∇G operators are optimal (in the sense of the stated criterion) for the enhancement of step edges only. We have proved that following the general optimization problem in [ShD79], we can derive the equation for the $\nabla^2 G$ and the ∇G operators. These two operators are implemented in the spatial domain and have found wide applications, unlike the Shanmugams's filter which is implemented in the frequency domain. Several studies have been reported on the performance of the $\nabla^2 G$ and the ∇G operators. Canny [Can83] has shown that for ideal step edges, edge localization as obtained by the ∇G operator is superior by a factor of 1.63 to that obtained by the $\nabla^2 G$ operator. He also showed that the SNR of the ∇G operator is better than that of the $\nabla^2 G$ operator by a factor of $1.16\sigma_f$ when the input is an ideal step edge embedded in white Gaussian noise.

Since edges from zero-crossings form closed contours ending at the boundary of the image, corners will be displaced. Thus vertexes cannot be detected correctly by the $\nabla^2 G$ operator, and spurious edges are introduced ([Ber84], [BaC88]). Corners or vertexes are properly located, however, by the magnitude of the gradient. Different authors have reported different results with the $\nabla^2 G$ operator (e.g., [Har84], [GrH85], and [Har85]). The accuracy of the $\nabla^2 G$ operator for edge enhancement can be best addressed, however, using edge models. This approach was used by a few authors (e.g. [Ber84] and [LuB86a]). Ideally, the $\nabla^2 G$ operator locates edge contours accurately if intensity changes are linear [MaH80]. This linearity, however, is not satisfied in most real world images. Berzins [Ber84] studied the performance of the $\nabla^2 G$ operator on corners, curves, and nonlinearity of the intensity surface. Lunscher and Beddoes [LuB86b] studied the effects of noise and quantization on the $\nabla^2 G$ performance in the context of digital filter performance.

The results of Berzins [Ber84] show that the displacements of the edge location as detected by the $\nabla^2 G$ operator is less than σ_f provided that: (1) size of the region is large compared to σ_f; (2) the radius of curvature of the region is large compared to σ_f; (3) the distance to the nearest sharp corner is larger compared to Θ/σ_f, where Θ is the angle of the corner in radians, (4) the magnitude of the second derivative of the image intensity is small compared to that of the filter's normalized impulse response multiplied by the size of the intensity jump at the edge. His results showed that for linear intensity variations (except at the corner), the displacement of the actual edge contours is less than $\sqrt{2}\sigma_f$, and that the gradient magnitude was small when the displacement was comparable to σ_f, so that contours with large displacements would be filtered out by a minimum threshold on the slope of the Laplacian. Large displacements or the disappearance of the contour occur when the second derivative of the intensity was compared to the second derivative of the smoothing filter. Berzins also showed that nonlinear illuminations lead to spurious contours in case where the second derivative of the intensity was positive (opposite to the second derivative at the center of the smoothing kernel).

The main advantage that the Laplacian of the Gaussian has over the gradient of the Gaussian is its simplicity for implementation. The Laplacian is a directionless quantity, and the zero-crossings form closed contours hence edges are easily traced.

In summary, we have demonstrated, mathematically, the equivalence of a number of optimal edge enhancement filters. We have indicated that modern approachs for edge enhancement based on regularization theory suggest that the actual shape of the filter is not important, provided that certain criterion is met. The issue of filter spatial support has been studied. We derived the corresponding criterion of Lunscher-Beddoes [LuB86a] for the ∇G operator on ideal edges. This provides a reasonable background for the linking process to be studied in Chapter 5.

CHAPTER 5
EDGE-BASED SEGMENTATION:
2. EDGE LINKING BY SEQUENTIAL SEARCH

5.1. Introduction

While the literature is abundant on quantitative methods for edge enhancement, essentially ad hoc techniques have been used to link edges. This often results in broken edges where the object's boundaries are not well defined. The introduction of sequential search techniques to link edges has been shown to provide better object boundaries. The use of search techniques in edge detection was first introduced by Martelli [Mar72], [Mar76]. He used edge properties in the evaluation function of the A^* algorithm. His path metric, however, was problem related and ad hoc. Some improvements to this approach were introduced by Ashkar and Modestino [AsM78]. Most notable was the application of tree search techniques to the edge enhanced image instead of using the original image as in [Mar72]. Still, the path metric used to guide the search algorithm was problem related and to a great extent ad hoc.

A major step toward an analytic linking algorithm was taken by Cooper [Coo79]. He formulated edge linking as a maximum likelihood (ML) estimation problem. Realizing the computational burden needed to implement this approach, he resorted to suboptimal techniques that were implemented on simple boundaries [Coo79], [CoE80], and [ElC82]. Another significant improvement to the search technique was introduced by Eichel and Delp [EiD84], [EiD85a], [EiD85b], and [EiD88]. Their sequential edge linking (SEL) algorithm is similar to that of [AsM78], but the path metric was quantitative. The metric in the SEL algorithm is a probabilistic measure similar to that introduced by Massey [Mas72] for sequential decoding.

This chapter examines recent progress in sequential search techniques applicable to edge linking. Specifically, we extend the sequential edge linking

(SEL) algorithm of Eichel and Delp [EiD84] to more general paths and image models, we develop estimation techniques for the metric in SEL using random field theory, and we introduce a new path metric based on the linear model. We examine implementation details and use these metrics on various test as well as real world images.

To provide a rationale for the use of graph search techniques to link edges, we examine the exact ML estimation technique for object boundary allocation. We will show that only suboptimal approaches can be implemented in practice and that graph search algorithms provide an easier and relatively faster implementation for such approaches. In simple terms, a maximum likelihood formulation for the edge linking problem is stated as follows: On a random field G let us define a boundary process B. The observed image is considered a sample function of G. The ML method provides an edge map for the objects in the image by maximizing the joint probability of the image and the boundary processes, that is, we need to maximize the following expression:

$$\begin{aligned} L &= \mathit{ln}\, P(GB), \\ &= \mathit{ln}\, P(G|B) + \mathit{ln}\, P(B), \end{aligned} \tag{5.1}$$

where $\mathit{ln}(\cdot)$ is the natural logarithmic function.

To solve the system in (5.1), we need to specify the boundary process B and the a posteriori probability $P(G|B)$. Several approaches for carrying out the above optimization have been introduced in the literature (e.g., [NaM78], [Coo79], and [GeG88]). We present below one of these approaches which is based on simple models. This approach is due to Cooper [Coo79] (see also [CoE80], [ElS81], and [ElC82]).

Cooper [Coo79] considered the ML formulation for a simple situation: The original image f was assumed to have a single object with gray level f_0 and a background with gray level f_b. The difference $\Delta = f_0 - f_1$ is the contrast. The quantity $\Delta/2$ was subtracted from the original image. The observed image g on the lattice $S = \{(x,y): 0 \leq x, y \leq M-1\}$ is assumed to be formed from the original image plus additive zero-mean Gaussian noise. Hence a pixel at location $(x,y) \in S$ has the following gray level values:

$$G_{xy} = \begin{cases} \Delta/2 + \mathbf{W}_{xy} & \text{on the object} \\ -\ \Delta/2 + \mathbf{W}_{xy} & \text{on the background} \end{cases} \qquad (5.2)$$

The noise random variable has a Gaussian probability density with mean zero and variance σ, that is,

$$\mathbf{W}_{x,y} \sim N(0, \sigma^2). \qquad (5.3)$$

Now, consider a discrete time stochastic process $G = \{G_{xy},\ x,y \in [0, M-1]\}$ with configuration space Ω and state space Ξ, that is, $G(\omega): \Omega \rightarrow \Xi,\ \omega \in \Omega$ (Chapter 2). A boundary element b_i is defined to be a line segment separating two adjacent pixels. An object boundary is a closed directed sequence $\{b_i\}$ which does not intersect itself.

Cooper [Coo79] modeled the boundary process as a K^{th} order Markov chain. The state X_m of the Markov chain was defined to be the last K-boundary elements allocated before the present location on the boundary, that is,

$$X_m = \{b_n\}_{n=m-K+1}^{m} . \qquad (5.4)$$

Note that the transition from state X_m to state X_{m+1} requires the knowledge of one element b_{m+1}. The transition probability of the Markov chain is defined as follows:

$$P\big(X_m = x_m | X_n = x_n\big) = P\big(b_m, b_{m-1}, \ldots, b_{m-K+1} | b_n, b_{n-1}, \ldots, b_{n-K+1}\big). \qquad (5.5)$$

The left-hand side will be written simply as $P(x_m | x_n)$.

Now, consider a random boundary of length N with a prior probability $P_0(N)$. The probability density of the boundary $P_B(b_1, b_2, \ldots, b_N)$ can be factorized by the chain rule as follows:

$$P_B(b_1, b_2, \ldots, b_N) = P(x_N | x_{N-1}) P(x_{N-1} | x_{N-2}) \ldots P(x_2 | x_1) P(x_1) \qquad (5.6)$$

or,

$$P_B(b) = \prod_{m=2}^{N} P_X(x_m | x_{m-1}) \cdot P(x_1). \qquad (5.7)$$

The likelihood function of the boundary process is now defined as follows:

$$\begin{aligned} L_B &= \mathit{ln}\,(P_B(b)\cdot P_0(N)),\\ &= \mathit{ln}\,P_B(b) + \mathit{ln}\,P_0(N),\\ &= \mathit{ln}\,P(x_1) + \sum_{m=2}^{N} \mathit{ln}\,P_B(x_m|x_{m-1}) + \mathit{ln}\,P_0(N). \end{aligned} \tag{5.8}$$

The a posteriori distribution $P(G = g|B{=}b)$ can be expressed by the total probability rule as follows:

$$\begin{aligned} P(G = g|B{=}b) = {} & P(G|B{:}b_m \in \text{object pixels})\cdot P(\text{object}) +\\ & P(G|B{:}b_m \in \text{boundary pixels})\cdot P(\text{background}). \end{aligned} \tag{5.9}$$

Assuming that the noise random variables $\{W_{xy}\,;\,(x,y)\in S\}$ are independent and identically distributed (*iid*), two adjacent edge elements $b_m = (G_{xy} - G_{x+1,y})$, and $b_{m+1} = (G_{x+1,y} - G_{x+2,y})$ are Gaussian, however, not independent. As an approximation, Cooper [Coo79] also assumed an *iid*-distribution for all boundary elements. From these assumptions, the quantities $P(G|B{:}b_m \in \text{object pixels})$ and $P(G|B{:}b_m \in \text{boundary pixels})$ can be written as products of univariate Gaussian probabilities, that is,

$$\begin{aligned} P(G|B) = {} & k_1 \prod_{(x,y)\in\text{object}} \exp - (g_{xy} - \Delta/2)^2/2\sigma^2 +\\ & k_2 \prod_{(x,y)\in\text{background}} \exp - (g_{xy} + \Delta/2)^2/2\sigma^2, \end{aligned} \tag{5.10}$$

where k_1 and k_2 are constants.

From (5.8) and (5.10), the likelihood L in (5.1) can now be written as follows:

$$\begin{aligned} L = {} & \mathit{ln}\,P(x_1) + \mathit{ln}\,P_0(N) + \sum_{m=2}^{N} P_B(x_m|x_{m-1}) + C -\\ & \frac{1}{2\sigma^2}\sum_{(x,y)\in\text{object}} (g_{xy} - \Delta/2)^2 -\\ & \frac{1}{2\sigma^2}\sum_{(x,y)\in\text{background}} (g_{xy} + \Delta/2)^2, \end{aligned} \tag{5.11}$$

where C is a constant.

Now, in order to obtain the ML solution, we need to maximize (5.11) over all possible boundary edge sequences. This requires the estimation of the state sequence $\{X_m\}$ and the complete knowledge of the boundary process; hence, the calculations cannot be made recursively.

The above analysis shows clearly that even with the simplest image model (a binary image having one object corrupted with additive Gaussian noise), the optimal solution is hard to compute. This provides an incentive for suboptimal solutions. In the following we examine in detail how such a solution can be obtained using graph search.

5.2. The Graph Search Problem

Graph search has been an active research area in AI as well as other disciplines for many years (e.g., [Zig66], [HaN68], [Jel69], [Nil71], and [KaK88]). Before we state the general graph search problem, we will describe, briefly, some of the terminology associated with graph search. Consider a graph defined on a set of sites (nodes) S. On this graph, we define a set of line segments (arcs) $\{b_{r,s};\ r,s \in S\}$. The segment $b_{r,s}$ is an arc from node r to node s. Node r is called the predecessor (parent) of s, and s is the successor of r. A cost $C_{r,s}$ is assigned to arc $b_{r,s}$ such that $C_{r,s} \geq \delta > 0$, where δ is a small positive number. The graph is called a directed graph if, for every two nodes r, s $\in$ S, $C_{r,s} \neq C_{s,r}$. Figure 5.1 shows an example of a graph structure.

A tree is a directed graph with the following properties: Only one node (root node) has ongoing arcs, whereas all other nodes have exactly one incoming arc and any number of outgoing arcs. The root node is said to be of depth zero while the depth of any other node is defined to be the depth of its parent plus 1. A node on the tree with no successors is called a tip node. A path **p** over a graph is a sequence of directed arcs. A solution path over a graph is a path consisting of finitely many arcs that begins at the root node r_0 and ends at a goal node r. The cost C(**p**) of a path **p** on a graph is the sum of the costs of arcs that make up that path.

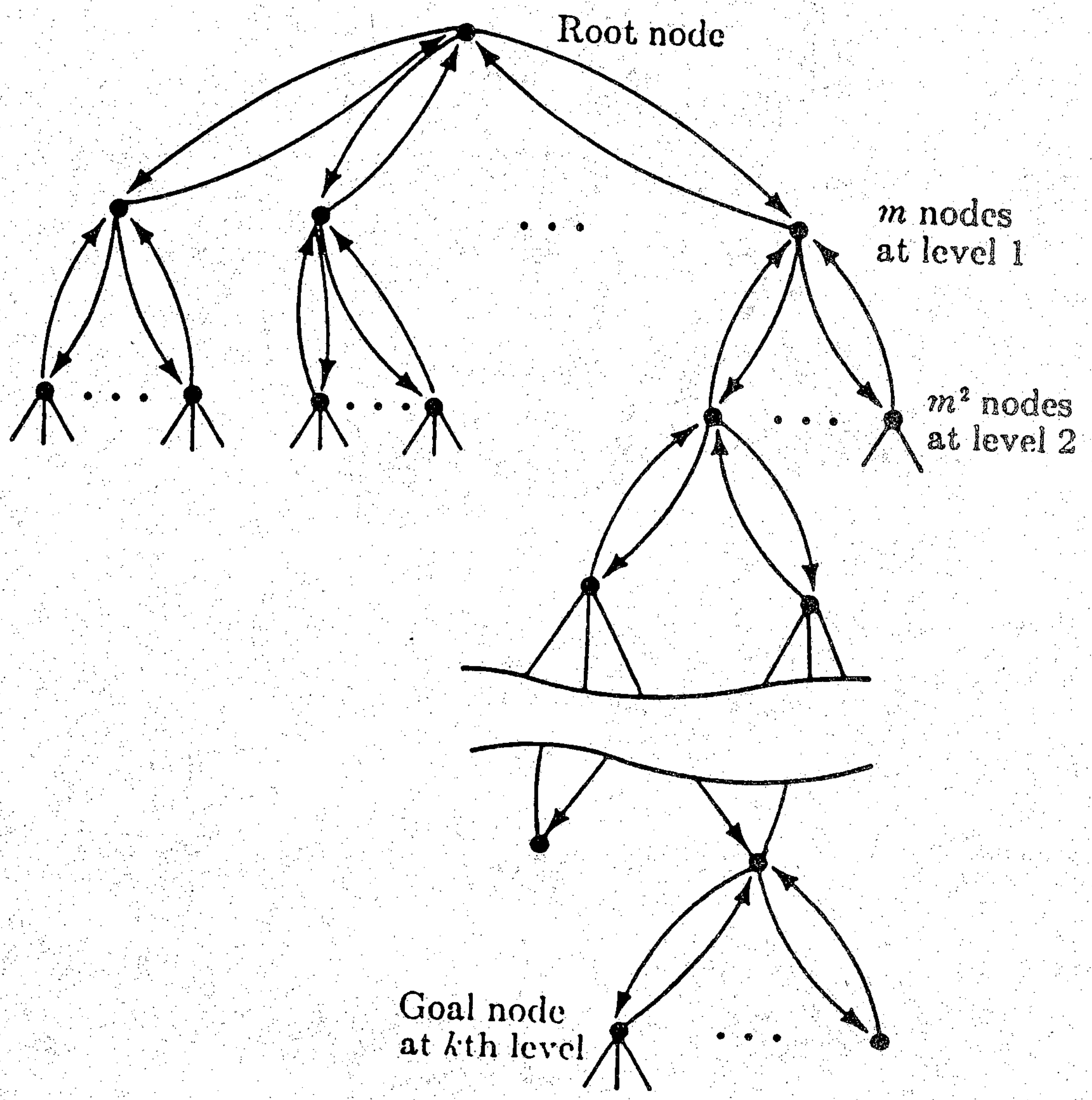


Figure 5.1. An example of a graph.

Now, given a weighted directional graph with start node s and a set of goal nodes $\{s_g\}$, the graph search problem can be stated as follows: Find a least-cost path (optimal path) from s to members of the set $\{s_g\}$.

Among the most studied algorithms for optimal path finding are the A* (e.g., [HaN68], [Nil71], and [DeP88]) and the Stack (e.g., [Zig66] and [Jel69]) algorithms, both of which execute sequential search using heuristic estimates of the costs of different paths in the graph. In the rest of this section we will discuss, briefly, these algorithms which have been used in essentially all the work on sequential edge linking in the literature (e.g., [Mar72], [ElC82], [AsM78], [EiD84]).

1. The A* Algorithm

The A* algorithm (e.g., [Nil80], [DeP88]) was developed for additive cost measures, that is, where the cost of the path is defined as the sum of the costs of its arcs. The algorithm employs an additive evaluation function,

$$\gamma(r) = \beta(r) + \alpha(r), \qquad (5.12)$$

where $\beta(r)$ is the cost of the currently evaluated path from the start node s to node r, and $\alpha(r)$ is a heuristic estimate for the cost of the path remaining between r and some goal node.

The algorithm constructs a tree of selected paths of the graph using the elementary operation of node-expansion, that is, generating all successors of a given node [DeP88]. Starting with s, the algorithm selects for expansion that leaf node of the tree which has the lowest value of $\gamma(\cdot)$, and only maintains the lowest evaluation function-path to any given node. The search halts as soon as a node selected for expansion is found to satisfy the goal conditions.

The A* algorithm finds a shortest solution path only if the heuristic function $\alpha(\cdot)$ never overestimates the actual distance to the goal. The optimality of the A* is studied elsewhere (e.g. [HaN68], [DeP88]).

The sequence of operations in the A* algorithm can be summarized as follows [DeP88]:

1. Place the start node s on a list called **OPEN** of an unexpanded node.
2. If **OPEN** is empty, exit with failure; no solution exists.

3. Remove from **OPEN** a node r at which $\gamma(r) = \beta(r) + \alpha(r)$ is minimum and place it on a list called **CLOSED** to be used for expanded nodes. Ties are broken arbitrarily, but in favor of a goal node.

4. If r is a goal node, exit successfully with the solution obtained by tracing back the pointers from r to s. The pointers are established in steps (5) and (6)

5. Expand node r, generating all its successors with pointers back to r.

6. For every successor r' of r:

   a. Calculate $\gamma(r')$.

   b. If r' was neither in **OPEN** nor in **CLOSED**, then add it to **OPEN**. Assign the newly computed $\gamma(r')$ to node r'.

   c. If r' already resided in **OPEN** or **CLOSED**, compare the newly computed $\gamma(r')$ with that previously assigned to r'. If the new value is lower, substitute it for the old (r' now points back to r instead of its predecessor). If the matching node r' resided in **CLOSED**, move it back to **OPEN**.

7. Go to step (2).

2. The Stack Algorithm

The Stack algorithm, or the ZJ algorithm ([Zig66], [Jel69]) uses a stack or table in which the values of the arc costs (path metric) for previously explored and extended paths are stored. Each table entry contains information about the node location on each path previously explored together with the value of the path metric up that node in the tree. There is one table entry for each previously explored path. The entries are ordered in terms of decreasing values of the path metrics, that is, the pointer to the top of the stack refers to the node with the largest path metric. The Stack algorithm then attempts to expand the path through the node currently identified at the top of the stack. The basic assumption here is that the path metric is monotonically increasing along the correct paths and monotonically decreasing along the wrong paths.

The Stack algorithm searches for the optimum path in the following manner:

1. Initialize: Clear the stack, and insert the entry corresponding to the root node.
2. Retrieve the entry with the largest path metric at the top of the stack. If the corresponding node is at the end of the tree, the search is completed. Otherwise, go to step (3).
3. Compute the parameter β_j for the successors of the node identified in step (2). Entries for these successor nodes are created and inserted at appropriate positions of the stack while the entry for the predecessor node is deleted.
4. Go to step (2).

The number of calculations required by the A^* and the Stack algorithms is a random variable whose distribution depends on the quality of the data. The algorithms, however, are much faster than the exhaustive search approach.

The Stack algorithm was used in edge detection by Ashkar and Modestino [AsM78] and by Eichel and Delp [EiD85a]. The A^* algorithm has been used by other researchers (e.g., [Mar72] and [ElC82]).

5.3. Edge Linking as a Graph Search

5.3.1. Problem Statement

On an image support, the graph $\{S,\eta\}$ is such that: S is a lattice, $S = \{(x,y): 0 \leq x,y \leq M-1\}$ and any internal site (node) $s = (x,y)$ has a unique set of eight neighbors (Figure 5.2a). Given a node $s = (x,y)$ on an edge path **p**, the path can be extended in eight possible directions. An edge path, therefore, is a tree in which each node has eight outgoing branches. The depth into the tree indicates the position along the path. Figure 5.3 illustrates the tree structure where the root node is at level 0. Nodes at level $k \geq 0$ correspond to points at position $(k + 1)$ along the path originating from the root node. A node $r_n = (x,y)_n$ is described by the horizontal location (row-location) and vertical location (column-location), respectively.

Definition 5.1: Over the lattice S, an edge path **p** of N nodes is defined to be a connected set of adjacent nodes (ordered set) satisfying some

$(x-1,y-1)$	$(x,y-1)$	$(x-1,y-1)$
$(x,y-1)$	(x,y)	$(x,y+1)$
$(x-1,y-1)$	$(x+1,y)$	$(x+1,y+1)$

(a) The eight neighbors for node s = (x,y).

4 $(135^\circ \le \theta < 180^\circ)$	3 $(90^\circ \le \theta < 135^\circ)$	2 $(45^\circ \le \theta < 90^\circ)$
5 $(180^\circ \le \theta < 225^\circ)$	0 (0°)	1 $(0^\circ \le \theta < 45^\circ)$
6 $(225^\circ \le \theta < 270^\circ)$	7 $(270^\circ \le \theta < 315^\circ)$	8 $(315^\circ \le \theta < 360^\circ)$

(b) The eight possible directions from node (x,y).

Figure 5.2. Neighbors of node s = (x,y) and direction quantization.

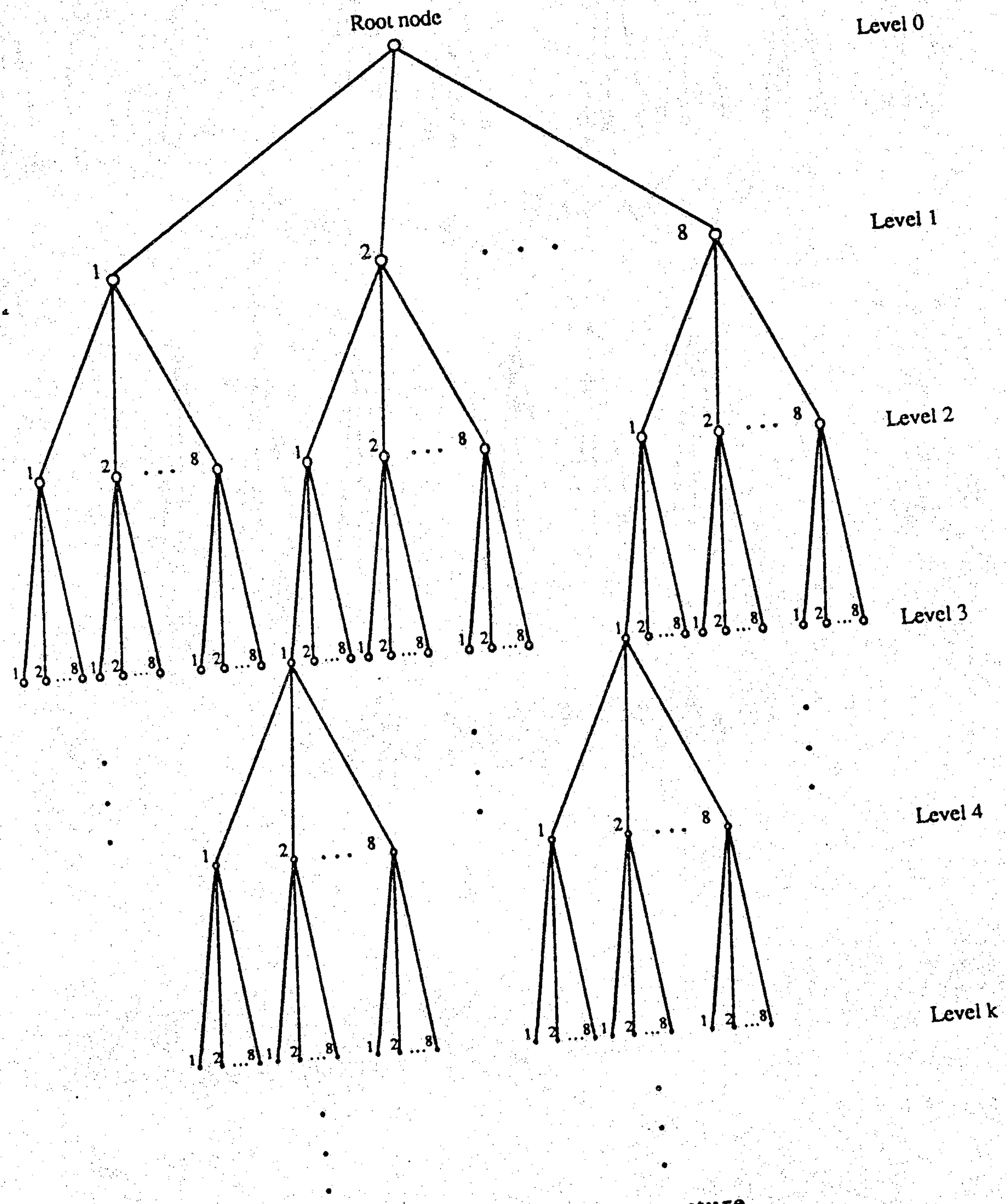


Figure 5.3. Basic tree structure.
(Each node has eight outgoing branches.)

(arbitrary) conditions. □

The path can be described in one of two ways [EiD85b]:

(i) By co-ordinates of the ordered set, that is,

$$\mathbf{p} = \{(x,y)_1,(x,y)_2 \ldots (x,y)_N\}, \text{ or} \tag{5.13}$$

(ii) By a starting node, starting direction, and a set of letters to guide the direction from the starting node, e.g.

$$\mathbf{p} = (x,y)_0 \times \mathrm{d}_0 \times (\mathrm{a}_1, \mathrm{a}_2 \ldots \mathrm{a}_N), \quad \mathrm{a}_k \in \mathrm{A}, \tag{5.14}$$

where the set A is the alphabet of possible directions. For example, in the SEL algorithm, A is restricted to have three possibilities: $\mathrm{A} = \{\mathrm{L}, \mathrm{D}, \mathrm{R}\}$ where L denotes left, D denotes direct (straight), and R denotes right.

Definition 5.2 [Rud76]: Consider two sets A and B. If there exists a one-to-one mapping of A onto B, we say that A and B are equivalent. This is represented by $\mathrm{A} \sim \mathrm{B}$ and has the following properties:

Reflexive: $\mathrm{A} \sim \mathrm{A}$.

Symmetric: If $\mathrm{A} \sim \mathrm{B}$ then $\mathrm{B} \sim \mathrm{A}$.

Transitive: If $\mathrm{A} \sim \mathrm{B}$ and $\mathrm{B} \sim \mathrm{C}$, then $\mathrm{A} \sim \mathrm{C}$.

Any relation with these three properties is called an equivalence relation. □

Let us denote the random field variables corresponding to a given path of length N (5.13) as

$$\mathbf{g} = \mathrm{g}_{\mathrm{r}_1}, \mathrm{g}_{\mathrm{r}_2}, \ldots, \mathrm{g}_{\mathrm{r}_N} = [\mathrm{g}_1, \ldots, \mathrm{g}_N]. \tag{5.15}$$

This path **p** imposes an ordering on the variables $\{\mathrm{g}_{\mathrm{r}_k}\}$, i.e.

$$\mathrm{g}_{\mathrm{r}_1} < \mathrm{g}_{\mathrm{r}_2} < \ldots < \mathrm{g}_{\mathrm{r}_N}. \tag{5.16}$$

Now consider pairs of nodes $\mathrm{r}_k, \mathrm{r}_l$ of the image lattice, and some random root node r_0 and let's denote by $<$ the binary relation distance from r_0. Therefore, $\mathrm{r}_k < \mathrm{r}_l$ if the Euclidean distance $|\mathrm{r}_k - \mathrm{r}_0| < |\mathrm{r}_l - \mathrm{r}_0|$. The relation $<$ is a partial order on the set of lattice nodes S because it is reflexive, antisymmetric, and transitive. Let's also denote the set of all paths with root node r_0 by $\mathrm{Y}_{\mathrm{r}_0}$, that is,

$$\mathrm{Y}_{\mathrm{r}_0} = \{\mathrm{p}: \mathrm{r}_k < \mathrm{r}_l \ \text{ for all } \mathrm{k} < 1\}. \tag{5.17}$$

Hence, the set of nodes $\{r_k\}$ comprising a path $\mathbf{p} \in Y_{r_0}$ is totally ordered (or, linearly ordered). Therefore, there is an isomorphism (one-to-one correspondence) between nodes of a path and the random field observations along that path which imposes a linear ordering on the set $\mathbf{g}$, that is,

$$g_1 < g_2 < \ldots < g_N \quad g \in Y, \tag{5.18}$$

where $g \in Y$ is understood to mean the corresponding $\mathbf{p} \in Y_{r_0}$ for some r_0 [EiD85a]. The set Y can be the set of paths that do not "double back on themselves", or those that are "almost straight lines", etc.

In order to use existing fast algorithms for sequential search (e.g., the A* and Stack algorithms) in edge linking, a metric to guide the search is necessary. This quantity is a measure of differentiation between various paths in the enhanced edge map and is denoted by the path metric. For accurate edge linking, the path metric should satisfy some important requirements (e.g., [CoE80], [EiD85a]). Among these requirements are the following: (1) The metric should not be biased by the path length. This means that all paths need to be compared on the basis of their probability of being on an edge path, regardless of their length. (2) The metric should have the necessary drift properties. That is, its value should be high on the correct path and decreasing otherwise. (3) The metric should be easy to calculate. For example, if the metric value at some node on the graph is related to the value of the metric at neighboring nodes, a great savings in the overall search time will result. This can be achieved if the metric equation can be written as a difference equation which enables recursive evaluation.

Two model-based metrics have been developed in the literature by Cooper and co-workers (e.g., [CoE80] and [ElC82]), and by Eichel and Delp [EiD85a]. We examine these metrics next, before introducing a new metric based on the linear model. The metrics in the pioneering work of Martelli ([Mar72], [Mar76]) and that of Ashkar and Modestino [AsM78] will not be examined further since they are problem related and to a great extent ad hoc. The idea in [AsM78] for using a prototype contour will be used, however, in a different context.

5.3.2. The Cooper Algorithm

Cooper and co-workers (e.g., [CoE80] and [ElC82]) used sequential graph search (the A^* algorithm) to implement a suboptimal boundary finding algorithm. The metric used was based on the likelihood expression in (5.11). Their algorithm can be summarized as follows: Each node in the graph represents a state which is defined by the K^{th} order Markov chain (5.4 - 5.8). A likelihood value is assigned to each node which corresponds to the maximum of the likelihoods of all paths leading to that node from a predetermined start node. According to the edge element definition [CoE80], each node has exactly three successor nodes. Figure 5.4 shows the tree structure in Cooper's algorithm and Figure 5.5 shows the node definition on the tree.

Let X_i be an arbitrary node (a state) on the tree and X_{i+1}^j, $j = 1,2,3$ are its successors. the metric used has the following form:

$$ln\ \hat{L}(X_{i+1}^j) = ln\ \hat{L}(X_1) + ln\ P_B(X_{i+1}^j|X_i) + D(X_{i+1}^j) \qquad j = 1,2,3. \quad (5.19)$$

In the above metric definition, the quantity $D(X_{i+1}^j)$ is the change in the picture data likelihood caused by adding node (state) X_{i+1}^j to the boundary sequence defined by the most likely path from the start node to node X_{i+1}^j. The quantity $P_B(X_{i+1}^j|X_i)$ is the state transition probability defined in (5.5), and $\hat{L}(X_1)$ is the likelihood of the starting node. Methods for estimating these quantities are discussed in [CoE80] and [ElC82].

A few comments are in order about the Cooper algorithm. First, while the formulation is based on the maximum likelihood, the approximations involved in the calculation of the components in (5.19) make the procedure non-standard and almost impossible to replicate. The effect of the model parameters on each of the quantities in (5.19) cannot be determined. Also, the relative importance of each component of the metric cannot be readily seen. Second, although the metric in (5.19) is recursive, still an enormous number of calculations are required for any typical size image. The results shown by the authors so far have been mainly based on simple test images. Finally, the algorithm as a whole is based on the independence of observations assumption which does not necessarily hold (Chapter 2). The algorithm, however, does show the difficulties involved in implementing an

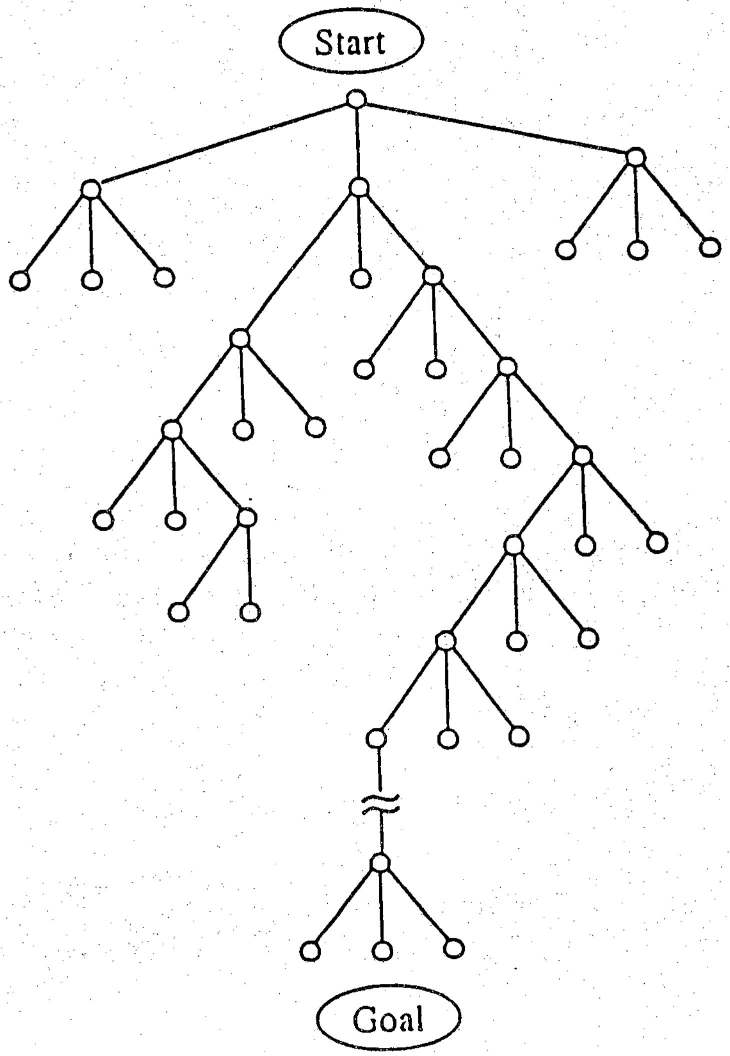


Figure 5.4. Tree structure for sequential search.
(Each node can be extended only in three directions.)

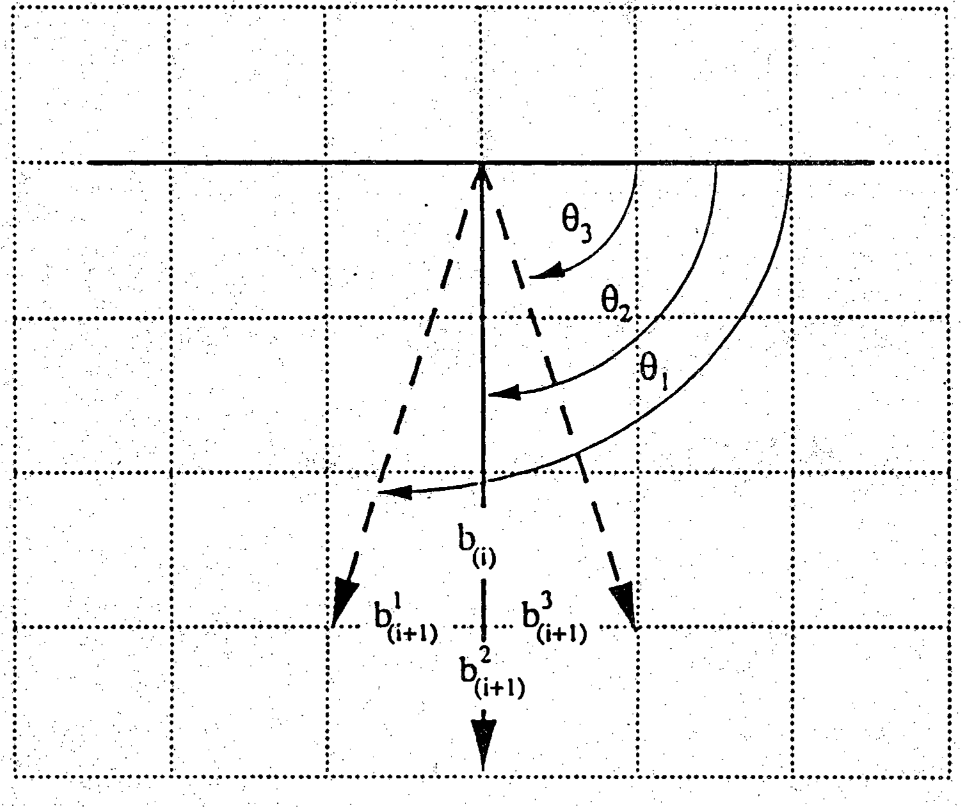


Figure 5.5. Node definition in the Cooper algorithm.

optimal approach for the boundary finding problem.

5.3.3. The SEL Algorithm

As in Cooper [Coo79], the path **p** in SEL [EiD85a] [EiD85b] is modeled as a K^{th} order Markov chain. The path is defined as in (5.14), and the transitions from a given node s are restricted to only three possibilities: Left, Direct, and Right, That is, the alphabet set A = $\{$L, D, R$\}$. The state of the chain X_i is described by the previous K-connected nodes before reaching the present node, or by the ordered set of the last K-letters $\{a_{i-j}, 1 \leq j < K\}$, that is,

$$X_i = a_{i-K+1}, a_{i-K} \ldots a_i \qquad a_i \in A. \tag{5.20}$$

The probability density of the path **p** is related to the state transition probabilities of the Markov chain by the following equations.

$$P(\mathbf{p}) = P(a_1, a_2 \ldots a_n) \tag{5.21.A}$$

$$= P(x_N|x_{N-1})\ P(x_{N-1}|x_{N-2}) \ldots P(x_1) \tag{5.21.B}$$

$$= \prod_{i=2}^{N} P(x_i|x_{i-1}) \cdot P(x_1). \tag{5.21.C}$$

Let H_1 denote the true edge hypothesis, and H_0 be the random (no edge) hypothesis. Let $p_1(G_r = g_r)$ be the probability that the field observation at node r is g_r conditioned on hypothesis H_1, and $p_0(g_r = g_r)$ be the probability that the field observation at node r is g_r conditioned on the null hypothesis (no edge node), that is,

$$p_1(Gr = g_{r)} = P(Gr = y|H_1), \tag{5.22}$$

$$p_0(Gr = g_r) = P(Gr = y|H_0). \tag{5.23}$$

For N observations $(g_{r_i}, i = 1, 2, \ldots, N)$ along a path **p**, the likelihood ratio is defined as follows:

$$l(g) = p_1(g_{r_1}, g_{r_2}, \ldots, g_{r_N})/p_0(g_{r_1}, g_{r_2}, \ldots, g_{r_N}). \tag{5.24}$$

The path metric is defined by the following equation:

$$\gamma(\mathbf{p},g) = ln\left(P(\mathbf{p}) \cdot l(g)\right). \tag{5.25}$$

If the observations are independent, the metric can be expressed in a recursive form which can be easily calculated. In this case the likelihood function in (5.25) can be written as follows:

$$l(g) = \frac{p_1(g_{r_1}) \cdot p_1(g_{r_2}) \ldots p_1(g_{r_N})}{p_0(g_{r_1})p_0(g_{r_2}) \ldots p_0(g_{r_N})}. \tag{5.26}$$

The path metric becomes:

$$\gamma(p,g) = \sum_{i=1}^{N} \left(\frac{ln\ p_1(g_{r_i})}{ln\ p_0(g_{r_i})} + ln\ P(x_i | x_{i-1}) \right). \tag{5.27}$$

Extending the observations by another node, the metric in (5.27) at node (N + 1) can be easily written as follows:

$$\gamma_{N+1} = \gamma_N + \left(\frac{ln\ p_1(g_{r_{N+1}})}{ln\ p_0(g_{r_{N+1}})} + ln\ P(x_{N+1} | x_N) \right). \tag{5.28}$$

The metric in (5.27) is formed of two components. The first is the ratio of the two probability measures in (5.22) and (5.23) defined from the random field model describing the image data. The second component is defined by the path model and it is a measure of the a priori probability that the edge path proceeds in a certain direction, given the last K-branch directions. The metric in (5.27) possesses the drift characteristic, that is, its value increases over correct paths and decreases over wrong paths. It is also unbiased by path length [EiD85a].

Again, the fact that pixel data are dependent on each other cannot be overemphasized (Chapter 2). The structural (syntactic) information in the image imposes this dependency. It has been shown that any meaningful image model relates the image pixel value to the surrounding pixels in a certain neighborhood (e.g., [Bar75], [Bes74], [Kas81]). The dependency between pixels, however, decreases with distance.

To elevate the condition of independence, Eichel and Delp [EiD85a] assumed an ARMA-model structure for paths in the random field. The basic assumption was that the random field G is homogeneous, isotropic, has a zero mean, is Gaussian, and its autocorrelation function is exponential of the distance between nodes, and the distance measure is the city block. This random field was denoted by the discrete step isotropic model (D-SIM) and its autocorrelation function has the following form:

$$R_g\,(k,\, l) = R_g(|k| + |l|) = R_g(d), \tag{5.29}$$

where the distance between nodes k and l is given by $d = d(k,l) = |k| + |l|$. The p^{th} order D-SIM random field is a homogeneous field with an autocorrelation function of the form:

$$R_g(k,\, l) = \sigma^2\, [c_1 e^{-a_1(|k| + |l|)} + \ldots + c_p e^{-a_p(|k| + |l|)}] \tag{5.30}$$

where $\sigma^2 = R_g(0,\, 0)$ and $\sum\limits_{i=1}^{p} c_i = 1.$

The path in SEL was defined as an ordered sequence of random field nodes such that, for a sequence of three nodes, the angle between the first two nodes and the second two is less than or equal to $\pi/4$. Figure 5.6 shows an example of the paths defined by the SEL algorithm. The field observations along a given path in the set Y in (5.17) forms a linearly ordered set. This enables the use of classical linear prediction models for the paths in Y, e.g., ARMA models. The general one-dimensional ARMA equation is given by

$$g_i - \sum_{k=1}^{p} \phi_k\, g_{i-k} = \zeta_i - \sum_{l=1}^{q} \theta_l\, \zeta_{i-1}, \tag{5.31}$$

where the ϕ_k are the p autoregressive coefficients and the θ_l are the q moving average coefficients. The process $\{\zeta_i\}$ is called the innovation process and has the property

$$E\ \{\zeta_i\, \zeta_j\} = \sigma_\zeta^{\,2}\, \delta_{ij}. \tag{5.32}$$

A correlated Gaussian random field which is D-SIM makes it possible to use a path metric of the form (5.27 - 5.28). This can be stated by the following two theorems [EiD85a].

Theorem 5.1: A path $\mathbf{p}\ \varepsilon\ Y$ in a *m^{th}* order D-SIM random field induces a super sequence g' such that (5.31) holds for this g'. □

Theorem 5.2: With g' as in theorem 5.1, it is possible to construct a sequence $\hat{g}$ from past values of g such that:

$$\hat{g}_i = \sum_{k=1}^{p} \phi_k\, g'_{(i)-k} + \sum_{l=1}^{q} \theta_l\, \zeta_{(i)-1} \tag{5.33}$$

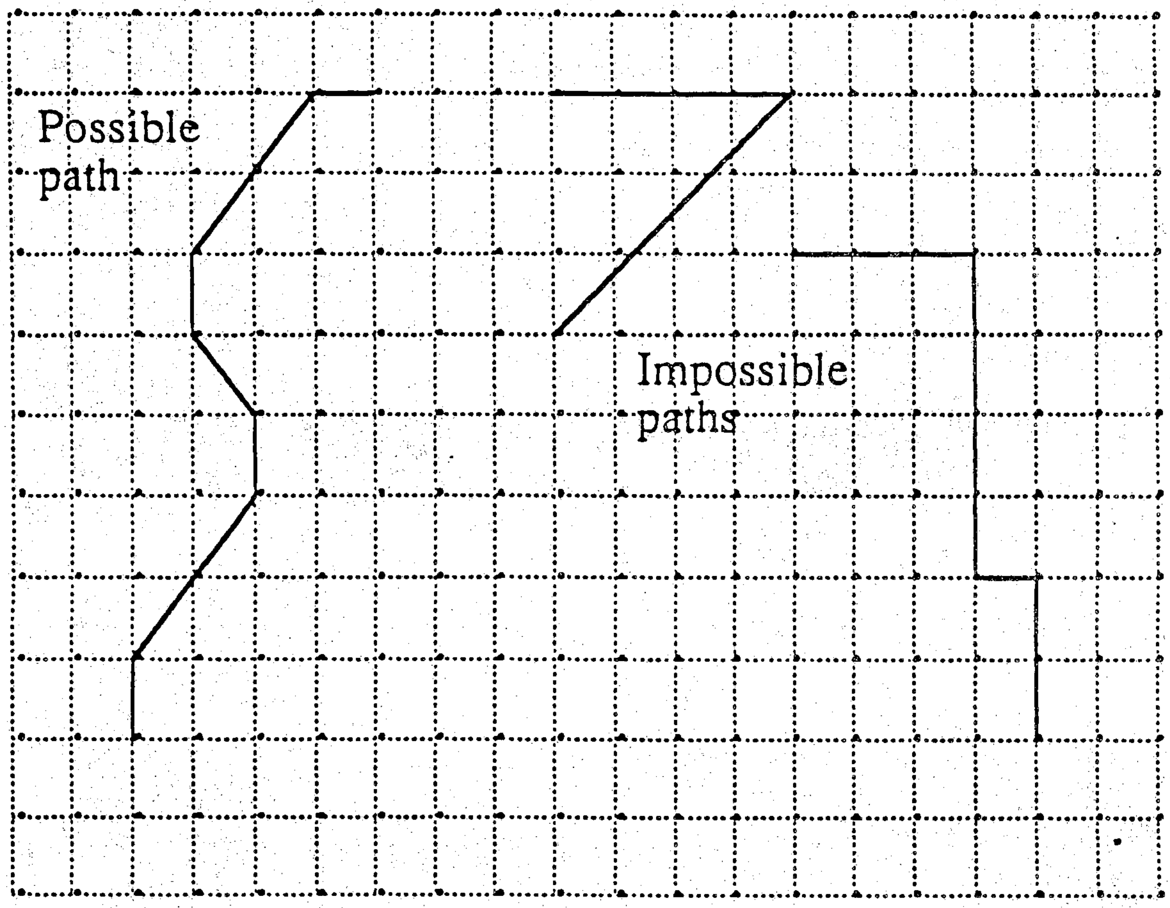


Figure 5.6. Examples for path definition in the SEL algorithm.

$$g_i - \hat{g}_i = \zeta_{(i)}$$

$$E\{\zeta_{(i)}\zeta_{(j)}\} = \sigma_\zeta^2\delta_{(i)(j)}.\square$$

Theorems 5.1 and 5.2 state that if a random field is a D-SIM random field, then the paths in the set Y have an ARMA-like structure. Thus, it is possible to construct a super sequence g′ from g such that uncorrelated innovations are generated which allow the evaluation of the path branch metric recursively.

Among the problems with the above metric are first, that the adequacy of the D-SIM model cannot be easily justified. Since it is defined by an autocorrelation function, a D-SIM random field cannot be synthesized (simulated) except in the Gaussian case. Second, no clear procedure for deciding on the order of the D-SIM model and estimating the coefficients in the autocorrelation function is presented. Finally, no clue for evaluating the measures $p_1(\cdot)$ and $p_0(\cdot)$ in (5.22) and (5.23) is provided.

In the following subsection, we examine the metric in (5.25) and introduce a method for estimating it from the image data without the independence assumption.

5.3.4. A Suboptimal ML Path Metric

According to the path definitions in [Coo79] and [EiD84], a path of length N leads to a search space of size 3^N. The path metric should greatly reduce the number of nodes actually searched in the search space. We will implement the metric in (5.25) on a small neighborhood rather than on the whole graph. A recursive metric formulation will be obtained which will be locally optimal (on the neighborhood) and globally suboptimal. As before, the path process is modeled by a K^{th} order Markov chain. Each node in the tree is, therefore, described by K^{th} dimensional Markov state defined by the boundary generation process. A likelihood value is assigned to each node in the tree and it corresponds to the maximum of the likelihoods for all paths leading to that node from a predetermined start node. We define a goal node to be a boundary state within a neighborhood of the start node.

Our goal is to evaluate the metric $\gamma(\cdot)$ in (5.25) on a small number of observations (within some neighborhood). Since this metric will be suboptimal, we will add the super script ^ "hat" to $\gamma(\cdot)$ in (2.25), that is,

$$\hat{\gamma}(\mathbf{p},\mathbf{g}) = ln\left(\hat{\mathrm{P}}(\mathbf{p})\right) + ln\left(\hat{\mathrm{l}}(\mathbf{g})\right), \qquad (5.34)$$

where the estimates $\hat{\mathrm{P}}(\mathbf{p})$ and $\hat{\mathrm{l}}(\mathbf{g})$ are to be evaluated on the neighborhood η. Two issues need to be addressed: First a method for selecting η must be provided and, second a method for estimating the quantities in (5.25) over this η must be developed. A discussion of these two problems follows.

5.3.4.1. Neighborhood Selection

To reduce the size of the search space further, and hence shorten the execution time, we limit the area (admissible area) of the search to be inside a swath defined by a hypothesized boundary. This boundary can be selected by a variety of ways. For example, an edge operator can be applied to the image and the resulting contour will be the hypothesized boundary. Or a priori information can be used to construct the hypothesized boundary. This is particularly useful if a prototype boundary is available (e.g., chest X-ray images used in [AsM78]). The approach we propose here is to apply the $\nabla^2 G$ operator to the original image and consider the zero-crossings loci to be our hypothesized boundary. The advantage of this approach is that the $\nabla^2 G$ provides a closed boundary; thus, an easier definition of the swath is possible. We will use the hypothesized boundary for two purposes: To define the swath of important boundary information, and to provide a measure for our a priori knowledge about the object's boundaries in the scene.

As we indicated in Chapter 4, studies on the $\nabla^2 G$ operator (e.g., [Ber84] and [Cla89]) lead to the following conclusions about the displacement from the ideal edge locations: For linear intensity variations (except at discontinuities), the displacement of the contours is less than $\sqrt{2}\sigma_f$, and the magnitude of the gradient is generally small if the displacement is comparable to σ_f. Nonlinear illuminations lead to spurious double contours if the second derivative is positive (i.e. opposite to the second derivative at the center of the smoothing Gaussian kernel). These spurious contours correspond to minima in the magnitude of the gradient, and the slope of the Laplacian at the zero-crossings is comparable to that for proper edge contours [Ber84]. Classic methods of setting a minimum threshold value for the slope of the

Laplacian at computed edge contours will not discard the spurious contours in the nonlinear illumination case. This is why a maximum likelihood approach is better than thresholding. We will limit the size of the neighborhood to be within ± $2\sigma_f$ of the zero-crossings contour.

Definition 5.3: The boundary swath b is a subset of the observation g which is formed of those gray scale pixels on the edge enhanced image separated by a maximum distance of $2\sigma_f$ from the zero-crossings contour obtained from the $\nabla^2 G$ operator. □

Once the hypothesized contour is selected, the sequential search is performed over portions of the image (edge enhancement image). This will be denoted by a sectioned search to distinguish it from the global search which uses all the data in the enhanced edge image at each node in the tree. Figure 5.7 illustrates the swath of important edge information surrounding a hypothesized contour.

Now, in accordance with the SEL path definition, we can identify all possible paths to be searched in a window, given the start node and direction. Figure 5.8 shows the possible paths for windows of widths 3 and 5. In each of these graphs, it is assumed that the start node and start direction are known. In Figure 5.8, the start node is denoted by × and the start direction is assumed to be horizontal. As expected, the number of possible paths increases with the window width. For a search window of dimensions 3×3, we have only three possible paths (from a preselected start node and start direction) and for a 5×5 search window, we have nine possible paths. Notice that the center of the neighborhood η is separated by one pixel from the start node.

5.3.4.2. Transition Probabilities Estimation

A Markov chain is usually specified by a state transition diagram and a state transition probability matrix. The selection of the neighborhood determines the number of possible states. Again, we will consider the path definition as in SEL algorithm [EiD84], [EiD85b]. Consider a 5×5 neighborhood. As shown in Figure 5.8b, if the start direction is known to be horizontal, we have nine possible sequences from the current location. Similarly, we have nine possible sequences if the start direction is to the left

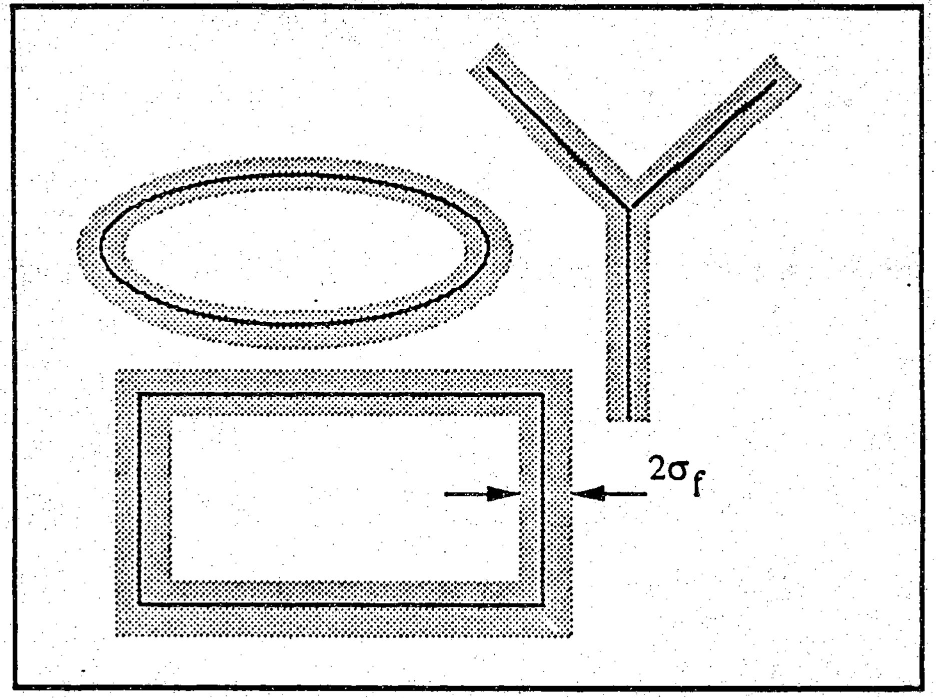
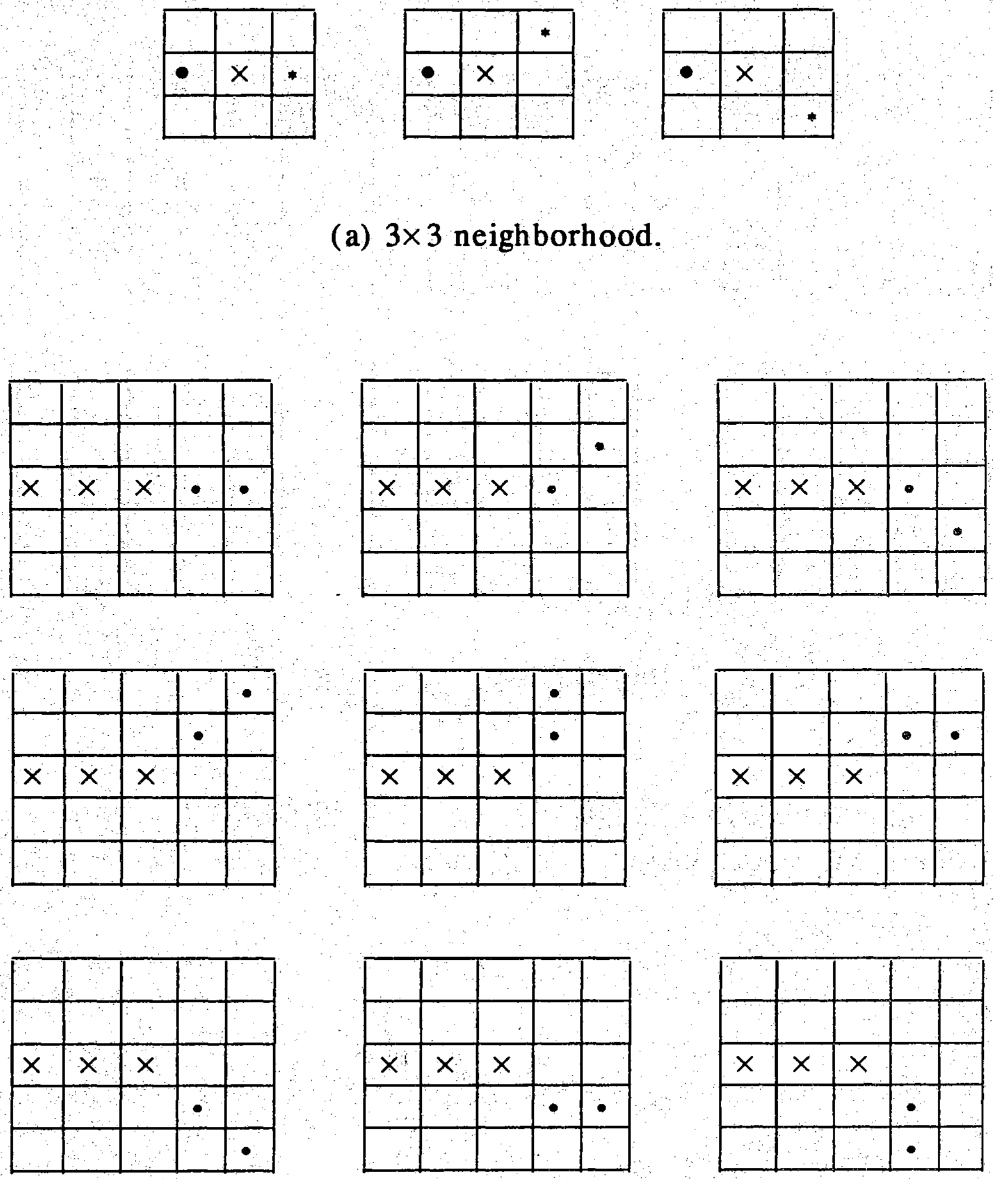


Figure 5.7. Definition of the swath of important edge information.



(a) 3×3 neighborhood.

(b) 5×5 neighborhood.

Figure 5.8. Possible paths for 3×3 and 5×5 neighborhoods.
(The start node is denoted by × and the start direction is assumed hoizontal.)

and another nine possible sequences if the start direction is to the right, etc. Recall that a node on the tree is represented by K-dimensional state. Now suppose the first node of the tree (i.e., a start state represented by a start node and a sequence of transitions) was given (e.g., all straight: D,D,D); then, the next node of the tree will be chosen from nine possible states. The nine possible states for this situation are shown in Figure 5.8b.

Empirical techniques have been used in the literature to estimate the state transition probabilities of the Markov chain (e.g., [ElC82] and [EiD85a]). Our experiments at the end of this chapter use the state transition tables in [EiD85b]. These tables, however, were constructed for first- and second-order Markov chains (i.e., $K = 1$ and 2); thus, they are suitable only for 3×3 neighborhoods. Using the chain rule of probability, the transition probability for a higher order Markov chain also can be calculated using these tables. For example, on a 5×5 neighborhood, the transition probability for a state $X = \{b_1, b_2, b_3, b_4, b_5\}$ is evaluated as follows:

$$P(b_1\, b_2\, b_3\, b_4\, b_5) = \prod_{i=2}^{4} P(b_i | b_{i-1})\; P(b_1). \tag{5.35}$$

Once the initial node and initial direction are known, the values $P(b_i | b_{i-1})$ are readily obtained from the empirical tables in [EiD85b] and the probability of the state can, therefore, be estimated.

5.3.4.3. Likelihood Ratio Estimation

We examine here a technique for the estimation of the likelihood ratio in (5.24) based on the Auto-normal Markov random field model [Bes74]. We have shown in Chapter 2 that the observations on a Markov chain form a Markov random field. Consider a path **p** of length n defined on the lattice. The gray levels on the pixel locations describing this path are assumed to be an observation from a Markov random field; that is, the set $\{g_0, g_1, \ldots, g_{n-1}\}$ is considered a sample function of a Markov random field. Now a site i, $(1 \leq i \leq n-2)$ has neighbors $(i-1)$ and $(i+1)$ while the boundary sites 0 and $(n-1)$ have the single neighbors 1 and $(n-2)$, respectively. This neighborhood will be denoted by η_c to distinguish it from the neighborhood η that defines the swath of important edge information.

From a given initial state on a neighborhood η we know the subset of possible transitions, as we discussed in the previous section. We fit an Autonormal model for the path observation on each possible transition. In Chapter 2 we studied this class of Markov random field models. The random variables G_i, $i \in \eta$ are assumed to be jointly Gaussian. Note that the lattice under consideration is a small portion of the image determined by the neighborhood η. We assume that each possible path segment has a length *n*. Over the neighborhood η_c defined by the Markov chain, the random variable $\{G_i|G_j, j \in \eta_c\}$ has a variance σ^2 and a mean μ given by:

$$\mu = E\{G_i|G_j, j \in \eta_c\} = \mu_i + \sum_{0 \le i < j \le n-1} \beta_{ij}(g_i - u_j). \tag{5.36}$$

The conditional probability is

$$p_{G_i|G_j}(g_i|g_j, j \in \eta_c) = \frac{1}{\sigma\sqrt{2\pi}} \exp(g_i - \mu_i)^2/2\sigma^2. \tag{5.37}$$

By the chain rule, the joint probability is

$$p_G(g = g_0, g_1, \ldots, g_{n-1}) = \frac{1}{(2\pi\sigma^2)^{-n/2}} |B|^{1/2} \exp(-\frac{1}{2\sigma^2}[g - \bar{g}]^t B[g - \bar{g}]), \tag{5.38}$$

where

$$\bar{g} = [\mu_0 \, \mu_1, \ldots, \mu_{n-1}]^t \tag{5.39}$$

is an $n \times 1$ vector and B is an $n \times n$ symmetric matrix with elements b_{ij} defined below.

$$b_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\beta_{ij} & \text{if } i \neq j \end{cases} \tag{5.40}$$

The likelihood ratio in (5.25) is estimated by the natural logarithm of (5.39) and we consider only the term in the exponent, that is, we take $\hat{l}(\cdot)$ to be:

$$\hat{l}(g) = (-\frac{1}{2\sigma^2}[g - \bar{g}]^t B[g - \bar{g}]). \tag{5.41}$$

The parameters μ_i, σ, and β_{ij} are estimated using the methods discussed in Chapter 2. In the experiments reported in this chapter, we assumed a common mean μ, that is estimated from the ensemble gray level values on each possible path. The estimates of the two components in (5.34) are scaled such that they will have an equal weight on the metric value.

The steps used in carrying out the sequential search using the metric estimation procedure we described above can be summarized as follows. (1) As in SEL, the input to the search algorithm is an edge enhanced image formed from two maps, an edge magnitude and an edge angle map. In the examples in this chapter, we used the ∇G operator for edge enhancement, and the Stack algorithm for edge linking. (2) A hypothesized boundary for the objects in the image is obtained by the $\nabla^2 G$ operator, and the swath of important edge information is determined as before. (3) The search is performed in the normal way starting from a node on the hypothesized contour. The starting direction is arbitrarily selected. The metric $\hat{\gamma}(\mathrm{p,g})$ is estimated as above on data values inside the swath.

The above procedure is quite simple to apply on test as well as real world images. The search algorithm that uses the above suboptimal metric, instead of the metric in [EiD85a], will be denoted by SEL2 to distinguish it from SEL. Several experimental results are provided at the end of this chapter.

5.4. A Path Metric Based on the Linear Model

In order to use existing fast algorithms for sequential search (e.g., the A^* and Stack algorithms) in edge linking, a metric to guide the search is necessary. This quantity is a measure of differentiation between various paths in the enhanced edge map.

Let $\mathrm{p}^{(i)} = \{r_0, r_1^{(i)}, r_2^{(i)}, \ldots, r_{Q-1}^{(i)}\}$ be a sequence of nodes along the i^{th} path up to level Q in the tree, and let $\beta_j^{(i)}$ be a cost measure for the transition from node r_{j-1} to node r_j on this path. The cost of the transitions along the sequence of nodes $\mathrm{p}^{(i)}$ will be the cumulative cost for the Q-transitions along the i^{th} path from a start node r_0. The path metric is a function of this cost. If significant a priori information about edge paths in the image is known, it should be included in the path metric. Examples of a priori information can be the ratio of horizontal to diagonal edges, the types of objects in the image, etc. We propose a metric of the following form:

$$\gamma_Q(\mathbf{p}^{(i)}) = \sum_{j=1}^{Q} \beta_j^{(i)} + h_j(\mathbf{p}^{(i)}), \tag{5.42}$$

where $\beta_j^{(i)}$ is a measure for the selection of one of the three possible transitions along the j^{th} branch of the path $\mathbf{p}^{(i)}$, and $h_j(\mathbf{p}^{(i)})$ is a measure of a priori information about this particular branch that can be extracted (in some fashion) from the edge enhanced image. The first component of the path metric in (5.42) will be derived from the linear model equation, which is considered next.

5.4.1. The Linear Model

Let the observed edge enhanced image be a sample function of a random field (two-dimensional random process) $G = \{G_s, s \in S\}$. Consider another random field **E** in which the random variables $\{E_s, s \in S\}$ have zero-mean Gaussian distribution with common variance σ^2. The linear model equation is ([Gra76], [Men87]):

$$\underset{\sim}{\mathbf{g}} = \mathbf{A}\Theta + \underset{\sim}{\mathbf{e}}, \tag{5.43}$$

where $\underset{\sim}{\mathbf{g}}$ and $\underset{\sim}{\mathbf{e}}$ are $L \times 1$ vectors representing an observation from the random fields G and E, respectively. The matrix **A** is of dimension $L \times k$ with $L > k$, and Θ is a $k \times 1$ vector of unknown parameters.

We will consider the case where the coefficients of the matrix **A** are deterministic. Let us define the two hypothesis:

$$H_1 \colon \mathbf{B}\Theta = \underset{\sim}{\mathbf{c}}, \tag{5.43}$$

and

$$H_2 \colon \mathbf{B}\Theta \neq \underset{\sim}{\mathbf{c}}, \tag{5.44}$$

where **B** is a given $q \times k$ matrix of rank $q \geq k$, $\underset{\sim}{\mathbf{c}}$ is a given $q \times 1$ vector, and $\mathbf{B}\Theta = \underset{\sim}{\mathbf{c}}$ is a consistent set of linear equations.

Given the values of the parameters Θ and σ^2, the random vector $\underset{\sim}{g}$ is Gaussian with mean $A\Theta$ and diagonal covariance matrix, $\sigma^2\mathbf{I}$ where, $\mathbf{I}$ is L × L identity matrix, i.e.,

$$p(\underset{\sim}{g}|\Theta,\sigma^2) = \left(\frac{1}{2\pi\sigma^2}\right)^{L/2} \exp\left(- \frac{1}{2\sigma^2}(\underset{\sim}{g} - A\Theta)^t(\underset{\sim}{g} - A\Theta)\right). \qquad (5.45)$$

Taking the natural logarithm of the above equation, we obtain

$$\psi(\sigma^2,\Theta) = ln\, p(\underset{\sim}{g}|\Theta,\sigma^2) \qquad (5.46)$$

$$\psi(\sigma^2,\Theta) = -\frac{L}{2} ln\, 2\pi - \frac{L}{2} ln\, \sigma^2 - \frac{1}{2\sigma^2}(\underset{\sim}{g} - A\Theta)^t(\underset{\sim}{g} - A\Theta). \qquad (5.47)$$

The maximum likelihood estimates of σ^2 and Θ can be obtained from the differentiation of (5.47) and equating the derivatives to zero, that is,

$$\frac{\partial}{\partial\sigma^2}\psi(\sigma^2,\Theta) = -\frac{L}{2}\,\frac{1}{\sigma^2} + \frac{1}{2\sigma^4}(\underset{\sim}{g} - A\Theta)^t(\underset{\sim}{g} - A\Theta) = 0. \qquad (5.48)$$

Hence,

$$\hat{\sigma}^2 = \frac{1}{L}(\underset{\sim}{g} - A\Theta)^t(\underset{\sim}{g} - A\Theta). \qquad (5.49)$$

Similarly,

$$\frac{\partial}{\partial\Theta}\psi(\sigma^2,\Theta) = (\underset{\sim}{g} - A\Theta)^t A - A^t(\underset{\sim}{g} - A\Theta) = 0. \qquad (5.50)$$

Hence,

$$\hat{\Theta} = \frac{1}{2}\,(A^tA^{-1})^{-1}(A^t\underset{\sim}{g} - \underset{\sim}{g}^tA). \qquad (5.51)$$

Substituting (5.49), (5.51) into (5.45), and simplifying, we obtain

$$p(\underset{\sim}{g}|\Theta,\sigma^2) = (2\pi\hat{\sigma}^2)^{-L/2} \exp\left(- \frac{L}{2}\right), \qquad (5.52)$$

which is independent of the parameter Θ.

Now, define the following likelihood ratio test (LRT):

$$LRT = \frac{p(\underset{\sim}{g}|(\Theta,\, \sigma^2)_{H_1})}{p(\underset{\sim}{g}|(\Theta,\, \sigma^2)_{H_2})}. \qquad (5.53)$$

From (5.52), LRT simplifies to:

$$\mathrm{LRT} = \left(\frac{\hat{\sigma}_2}{\hat{\sigma}_1}\right)^{L} . \tag{5.54}$$

The estimation of $\hat{\sigma}_1$ and $\hat{\sigma}_2$ will be addressed shortly. For the parameter β_j, we use a function of LRT, which is defined as follows:

$$\Lambda = \left(\mathrm{LRT}^{-2/L} - 1\right)\left(\frac{L-k}{q}\right) = \left(\frac{\hat{\sigma}_1^{\,2} - \hat{\sigma}_2^{\,2}}{\hat{\sigma}_2^{\,2}}\right)\left(\frac{L-k}{q}\right). \tag{5.55}$$

The statistic Λ has an F-distribution with $(q, (L - k))$ degrees of freedom ([KeS77], [KaL86]).

5.4.2. Definition of Edge Hypothesis

On a 3 × 3 neighborhood, edge directions are usually quantized into eight directions (multiples of 45^0). Hence, we can define the set $\{\mathbf{H}, \mathbf{V}, \mathbf{D}_1, \mathbf{D}_2\}$ of four edge models (edge hypotheses) on this neighborhood as shown in Figure 5.9. To use the statistic in (5.55), we need to specify L, k, and the parameters $\hat{\sigma}_1$ and $\hat{\sigma}_2$. The matrix A is specified by fitting a linear model to each edge model (edge hypotheses). Figure 5.10 shows the components of the linear model fit to edge model $\mathbf{D}_2$. The linear model equation for this edge model is written as follows:

$$\underset{\sim}{\mathbf{g}} = \mathbf{A}\,\underset{\sim}{\mathbf{h}} + \underset{\sim}{\mathbf{e}}, \tag{5.56}$$

where

$$\underset{\sim}{\mathbf{g}} = [g_1 \;\; g_2 \;\; g_3 \;\; \dots \;\; g_9]^t \tag{5.57}$$

$$\underset{\sim}{\mathbf{e}} = [e_1 \;\; e_2 \;\; e_3 \;\; \dots \;\; e_9]^t \tag{5.58}$$

$$\mathbf{A}^t = \begin{bmatrix} 0.5 & 1 & 1 & 0 & 0.5 & 1 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0 & 1 & 0.5 & 0 & 1 & 1 & 0.5 \end{bmatrix} \tag{5.59}$$

$$\underset{\sim}{\mathbf{h}} = [h_1\, h_2]^t, \tag{5.60}$$

and the script t in (5.57)-(5.60) denotes matrix transposition.

5.4.3. Parameter Estimation

We now address the problem of parameter estimation. As we indicated before, for proper definition of the linear model, it is necessary that the matrix **B** and the vector $\underset{\sim}{\mathbf{c}}$ in (6-7) be such that the $q \times k$ matrix **B** must have a full rank (i.e., the rank $= k$), and the set of equations $\mathbf{B}\Theta = \underset{\sim}{\mathbf{c}}$ must have a consistent set of solutions. A simple choice for the matrix **B** and the vector $\underset{\sim}{\mathbf{c}}$ which satisfies these requirements is the following:

$$\mathbf{B} = [1 \;\; -1], \tag{5.61}$$

$$\underset{\sim}{\mathbf{c}} = 0. \tag{5.62}$$

This choice for **B** and $\underset{\sim}{\mathbf{c}}$ is also similar to that in [KaL86].

With the above choice for **B** and $\underset{\sim}{\mathbf{c}}$, the hypothesis H_1 and H_2 are now written as follows:

$$H_1 : [1 \;\; -1]\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = 0, \tag{5.63}$$

$$H_2 : [1 \;\; -1]\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \neq 0. \tag{5.64}$$

From the above specifications of **A**, **B**, and $\underset{\sim}{\mathbf{c}}$, we have $L = 9$, $k = 2$, and $q = 1$. Hence the statistic Λ in (5.55) is

$$\Lambda = 7\left(\frac{\hat{\sigma}_1{}^2 - \hat{\sigma}_2{}^2}{\hat{\sigma}_2{}^2}\right). \tag{5.65}$$

The maximum likelihood estimation (MLE) for the parameters $\hat{\sigma}_2$ and $\hat{\sigma}_1$ can be easily calculated as follows: For the above choice of **B** and $\underset{\sim}{\mathbf{c}}$, H_1 is the no-edge hypothesis, that is, $h_1 = h_2 = h$. Hence, (5.42) becomes

$$\underset{\sim}{\mathbf{g}} = \mathbf{A}' + \underset{\sim}{\mathbf{e}}, \tag{5.66}$$

where

$$\mathbf{A}' = \mathrm{A}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\mathrm{h}. \tag{5.67}$$

Equation (5.66) is the classic deterministic signal and additive noise problem. The MLE is

$$\hat{\sigma}_1^2 = \frac{1}{9}\sum_{i=1}^{9}(g_i - \hat{\mu})^2, \tag{5.68}$$

where

$$\hat{\mu} = \frac{1}{9}\sum_{i=1}^{9} g_i. \tag{5.69}$$

For hypothesis H_2, the edge hypothesis, the MLE for σ_2 is obtained from the minimization of the natural logarithm in (5.46), which can be written in the following quadratic form:

$$J = \sum_{i=2,3,6}(g_i - h_1)^2 + \sum_{i=4,7,8}\left(g_i - \frac{h_1 + h_2}{2}\right)^2 + \sum_{i=1,5,9}(g_i - h_2)^2. \tag{5.70}$$

Differentiating (5.70) with respect to h_1 and h_2 and equating the derivatives to zero, it is simple to show that the MLE for h_1 and h_2 are as follows:

$$\hat{h}_1 = w_1{}^t \underset{\sim}{g}, \tag{5.71}$$

$$\hat{h}_2 = w_2{}^t \underset{\sim}{g}, \tag{5.72}$$

where

$$w_1 = \frac{1}{18}[2 \quad 5 \quad 5 \quad -1 \quad 2 \quad 5 \quad -1 \quad -1 \quad 2]^t \tag{5.73}$$

and

$$w_2 = \frac{1}{18}[2 \quad -1 \quad -1 \quad 5 \quad 2 \quad -1 \quad 5 \quad 5 \quad 2]^t. \tag{5.74}$$

Hence, the estimate $\hat{\sigma}_2$ has the following form:

$$\hat{\sigma}_2^2 = \sum_{i=2,3,6}(g_i - \hat{h}_1)^2 + \sum_{i=4,7,8}\left(g_i - \frac{\hat{h}_1 + \hat{h}_2}{2}\right)^2 + \sum_{i=1,5,9}(g_i - \hat{h}_2)^2. \tag{5.75}$$

The values for other edge models are similarly calculated. Table 5.1 provides a summary of the equations needed to calculate the statistic (5.55) for the four edge models.

Table 5.1. The test statisitic Λ for 4-edge models.

$\Lambda = 7\left(\frac{\hat{\sigma}_1^2 - \hat{\sigma}_2^2}{\hat{\sigma}_2^2}\right) \qquad \hat{\sigma}_1^2 = \frac{1}{9}\sum_{i=1}^{9}(g_i - \bar{u})^2 \qquad \hat{\mu} = \frac{1}{9}\sum_{i=1}^{9} g_i$ $\hat{h}_i = \mathbf{w}_i^t \underset{\sim}{\mathbf{g}} \qquad i \in [1,2] \qquad \underset{\sim}{\mathbf{g}} = [g_1\ g_2 \cdots g_9]^t$
Edge model H
$\hat{\sigma}_2^2 = \sum_{i=1,2,3}(g_i - h_1)^2 + \sum_{i=4,5,6}\left(g_i - \frac{h_1 + h_2}{2}\right)^2 + \sum_{i=7,8,9}(g_i - h_2)^2$ $\mathbf{w}_1 = \frac{1}{18}[5\ \ 5\ \ 5\ \ 2\ \ 2\ \ 2\ \ -1\ \ -1\ \ -1]^t$ $\mathbf{w}_2 = \frac{1}{18}[-1\ \ -1\ \ -1\ \ 2\ \ 2\ \ 2\ \ 5\ \ 5\ \ 5]^t$
Edge model V
$\hat{\sigma}_2^2 = \sum_{i=1,4,7}(g_i - h_1)^2 + \sum_{i=2,5,8}\left(g_i - \frac{h_1 + h_2}{2}\right)^2 + \sum_{i=3,6,9}(g_i - h_2)^2$ $\mathbf{w}_1 = \frac{1}{18}[5\ \ 2\ \ -1\ \ 5\ \ 2\ \ -1\ \ 5\ \ 2\ \ -1]^t$ $\mathbf{w}_2 = \frac{1}{18}[-1\ \ 2\ \ 5\ \ -1\ \ 2\ \ 5\ \ -1\ \ 2\ \ 5]^t$
Edge model D_1
$\hat{\sigma}_2^2 = \sum_{i=1,2,4}(g_i - h_1)^2 + \sum_{i=3,5,7}\left(g_i - \frac{h_1 + h_2}{2}\right)^2 + \sum_{i=6,8,9}(g_i - h_2)^2$ $\mathbf{w}_1 = \frac{1}{18}[5\ \ 5\ \ 2\ \ 5\ \ 2\ \ -1\ \ 2\ \ -1\ \ -1]^t$ $\mathbf{w}_2 = \frac{1}{18}[-1\ \ -1\ \ 2\ \ -1\ \ 2\ \ 5\ \ 2\ \ 5\ \ 5]^t$
Edge model D_2
$\hat{\sigma}_2^2 = \sum_{i=2,3,6}(g_i - h_1)^2 + \sum_{i=4,7,8}\left(g_i - \frac{h_1 + h_2}{2}\right)^2 + \sum_{i=1,5,9}(g_i - h_2)^2$ $\mathbf{w}_1 = \frac{1}{18}[2\ \ 5\ \ 5\ \ -1\ \ 2\ \ 5\ \ -1\ \ -1\ \ 2]^t$ $\mathbf{w}_2 = \frac{1}{18}[2\ \ -1\ \ -1\ \ 5\ \ 2\ \ -1\ \ 5\ \ 5\ \ 2]^t$

5.4.4. Metric Evaluation

The proposed metric in (5.42) is formed of two components: A measure for differentiation between possible transitions on a certain branch of the path (i.e., to select one of the three possible extensions of a certain node on the path), and a measure for a priori information about this specific branch. These two components were represented by the quantities $\beta_j^{(i)}$ and $h_j^{(i)}$, respectively. Evaluation of the quantity $h_j()$ will be addressed in the following section. The quantity $\beta_j()$ is evaluated from the linear model as follows:

(1) Knowing the predecessor of the current node, select the next node by choosing the edge model (among the three possible models) that has maximum value of Λ in (5.55).

(2) Set $\beta_j^{(i)}$ to be equal to the value of Λ in the previous step.

For example, in the situation shown in Figure 5.8a, we need to evaluate Λ for edge models $\{\mathbf{H}, \mathbf{D}_1, \mathbf{D}_2\}$ and choose the maximum. Ties are selected either arbitrarily or according to the value of the a priori information measure (the second part of (5.42)), but always in favor of goal nodes.

Ignore for a moment the a priori information part in (5.42); that is, consider the following path metric:

$$\gamma_Q(\mathbf{p}^{(i)}) = \sum_{j=1}^{Q} \beta_j^{(i)}. \tag{5.76}$$

The cost of the i^{th} path having Q-nodes is the additive cost of all the nodes forming it. The cost associated with node j (the j^{th} branch) is obtained from (5.65). It is obvious that only local calculations are needed to obtain this cost and, therefore, the metric in (5.76) is very easy to calculate. While we have not yet proved that the metric in (5.76) possesses the desired drift characteristic or that it is unbiased with respect to the path length, our experimental results (using only the metric in (5.76)) seem to indicate that the metric does not overestimate the cost of a certain path (as evident in the resulting accurate edge localization.) Now, since the A* algorithm (which was used in our experiments) always finds the optimal path if the cost is not overestimated (see [DeP88]), therefore, we can conclude that our path metric is very adequate for implementing the sequential search to link the edges.

Finally, we should point out that Kay and Lemay [KaL86] have also used the linear model in their study. However, our approach differs from theirs in four main points. First, they consider only two edge models, the horizontal and the vertical edge models $\{H, V\}$. Second, and most importantly, they detect edges solely based on the value of the ratio in (5.55), that is, a threshold is set, and an edge pixel is declared if the ratio exceeds this threshold. As a result, their approach suffers from the well known problems associated with edge detection by thresholding (e.g., [FaD86]). Third, their approach gives no consideration to edge orientation; thus, the issue of edge localization is totally ignored. This, in addition to the previous point, explains why the authors had poor edge detection results. Finally, our approach uses the linear model as a part of the linking algorithm on enhanced edges and not on the original image as in [KaL86].

5.4.5. A Priori Information Measure

We first obtain a gross estimate of the actual object boundaries in the image. This was obtained by the zero-crossing contours of the $\nabla^2 G$ [MaH80] operator. The selection of the $\nabla^2 G$ rather than, say, the ∇G [Can86] operator was due to the fact that the zero-crossing contours are continuous. Therefore, an estimate of the actual boundaries can be obtained at every point in the image. These boundaries are called prototype or typical boundaries. The quantity $h_j(p^{(i)})$ in (5.42) is evaluated as follows: Over the support of the edge models discussed before (3 × 3 windows), the following distance measure is calculated for each of the two nodes off the center:

$$d_m = \sum_{n=1}^{9} |x_m - x_n^0| + |y_m - y_n^0|, \qquad m = 1,2, \qquad (5.77)$$

where $x_m$and y_m are the coordinates of the m^{th} node off the center of the current node under consideration, and $x_n^0$and y_n^0 are the coordinates of the n^{th} node of the typical boundary (within the 3 × 3 neighborhood of the current node) obtained from the $\nabla^2 G$ operator. The distance measure in (5.77) is to be used in conjunction with the angle information to decide, for a given edge model, which node is to be extended from the center node, i.e., backward or forward.

For example, in the **H**-model, let the coordinates of the center node be r = (x,y). The distance measure is calculated as follows:

$$d_1 = \sum_{n=1}^{9} |x - x_n{}^0| + |(y-1) - y_n{}^0|, \tag{5.78}$$

and

$$d_2 = \sum_{n=1}^{9} |x - x_n{}^0| + |(y+1) - y_n{}^0|. \tag{5.79}$$

Now if $d_1 < d_2$, the node to the left is chosen, provided that the angle information is not conclusive (and unless the node to the right is a goal node) and vice versa. Similarly, if the likelihood ratios of any two edge models are the same, then the distance measure evaluated for the nodes of the two models (off the center) decides where the next node is to be extended.

Our use of a prior information, as outlined above, is merely a guide in breaking ties. Other heuristics are possible (see [ElC82], [Mar76], and [AsM78]). The search algorithm that uses the metric we just described will be denoted by LINK.

5.5. The Linking Algorithm (LINK)

The sequential linking algorithm that uses the metric in (5.42) and the A^* algorithm (or the Stack algorithm) is outlined in the following steps.

Algorithm:

Step 1. Perform edge enhancement using a gradient operator of suitable width.

Step 2. Choose a starting node (a pixel with maximum gradient magnitude) and find the corresponding initial direction from the gradient angle.

Step 3. Transitions on a path i are carried out by the A^* algorithm (or the Stack algorithm) depending on the value of the metric $\gamma(\cdot)$ in (5.42).

Step 4. Stop the search when all goal nodes have been examined, when a stack overflow occurs, or when the paths discovered intersect each other.

So far, we have examined in detail the issue of a quantitative path metric. To use graph search algorithms in edge linking, we need also to define the following quantities: (1) A representation of the edge information at each node; (2) A starting (root) node; (3) A class of goal nodes. We used the ∇G operator for edge enhancement. Therefore, the edge information (gradient vector) is represented in terms of a magnitude map and angle map. This is typical with the majority of the edge linking techniques in the literature (e.g., [AsM78] and [EiD85b]). In the rest of this section we will discuss, briefly, the issues of root node selection and goal nodes definition.

5.5.1. Root Node Selection

A quantitative measure for the root node selection is not always possible. A general rule, however, is that this node should lie on an actual edge. This will, in turn, provide a good start for the search algorithm and reduce the overall search time. An attempt to quantify the selection of the root node was given by Eichel and Delp [EiD85b]. Briefly, they used the classical solution of the signal detection problem involving two signal levels plus an additive noise. The minimum probability of error solution selects the root node such that the probability of false alarms is minimized; thus, the node of the highest gradient magnitude is selected as the starting node.

If significant a priori information is known about the edges in the scene, empirical methods can be used to determine the root node. For example, in the experiments of [AsM78] on angiocardiograms, a prototype for the boundaries of the heart was used as a means of limiting the candidates for the root node to those who best fit the prototype model. The accuracy of this approach, however, is a function of the adequacy of the a priori information.

In this chapter, we will also select the root node to be the point of highest gradient magnitude.

5.5.2. Goal Nodes Definition

No specific rules can be adopted for all applications. Again, a priori information and the general definition of edge elements are often used to determine goal nodes (e.g., [AsM78], [CoE80], and [EiD85b]). If the objects in the scene have circular (or closed) boundaries, the class of goal nodes can be assigned such that the best path closes on itself. If the scene contains long objects, the goal nodes can be selected such that the search is terminated when the best path runs off the image boundaries.

If the path metric is such that beyond an abrupt edge its value decreases, a criterion for goal node selection or search termination based on the behavior of the metric can be used [EiD85b]. This criterion is summarized as follows: When the metric of the best path falls below some specified fraction of the highest metric along that path, the search is terminated. While this approach seems reasonable for abrupt intensity edges, other edges that have smaller values or that are slowly fading might not be well characterized by the behavior of the path metric. In this case, empirical methods must be used to assign goal nodes.

In this chapter, the goal nodes were selected such that: (i) The nodes considered by the search algorithm have a corresponding gray level pixel values (gradient magnitude) within 30% of the maximum (that defines the root node); (ii) Paths do not intersect the image boundary; and (iii) Short paths (less than 18 pixels long) are neglected.

5.6. Results

The two algorithms, SEL and LINK, have been applied to various test and natural scenes. We will compare the relative performance of the two algorithms, as well as their performance, to the nonmaximal suppression approach of Canny [Can86]. The comparison, to be of any value, must start from a certain step common to all the algorithms. The output of nonmaximum suppression and thresholding is, usually, broken contours (streaks). Therefore, its comparison with the SEL and LINK algorithms is, essentially, to test the adequacy of the path models used in the algorithms to represent connected object boundaries. Two types of test images are used in these experiments: A step test image [Pra78], and a discs test image [KiR81].

We adopt the following procedure for comparison of the algorithms on test images:

(1) Edge enhancement is accomplished by the ∇G operator. The edge enhanced map is in terms of an edge magnitude MAG and an angle ANG file.

(2) Nonmaximum suppression is performed on the edge enhanced maps. The threshold used is selected by an analytical method, e.g., the iteration method [RiC78] or the entropic thresholding method [Pan81]. (See [FaD86] for more on these methods as well as various threshold selection techniques.)

(3) The SEL and LINK algorithms are run on the original edge enhanced map. The threshold from step (2) is used as an upper bound for the two linking algorithms.

(4) In the case of the step test image, the Pratt figure of merit [Pra78] is used as a heuristic to test edge localization and the number of edge pixels detected by each algorithm. This figure of merit will be used to test the output of the linking algorithms as well as the nonmaximal suppression approach.

(5) Since the SEL algorithm in [EiD85a] was implemented using a first order Markov chain model for the path, and since LINK uses an LRT on a 3 × 3 window, the comparison should provide an evaluation of the metrics used in SEL and LINK. No a priori information has been used in this comparison. That is, the value of the parameter $\mathbf{h}_j(\mathbf{p}^{(i)})$ in (5.42) was set to zero.

In the following discussion, first the edge enhancement procedure is described. Then, the performance of the algorithms on test images is examined. Finally, we discuss how the algorithms compare (visually) on natural scenes.

5.6.1. Generation of the Edge Enhancement Map

In Chapter 4 we studied optimal edge enhancement filters and we also compared the relative performances for two such filters, namely, the ∇G and the $\nabla^2 G$ operators. We concluded that the ∇G operator outperforms the $\nabla^2 G$ operator in terms of edge localization. The SEL and LINK algorithms were tested on edge enhancement maps generated by the ∇G operator.

A practical implementation for edge enhancement by the ∇G operator is now described.

(1) Estimate the smoothing filter parameter σ_f (Chapter 4).

(2) Since the Gaussian kernel is separable, only one-dimensional convolution is necessary; thus, a gradient magnitude MAG and direction ANG files are constructed as follows:

(a) Sample the one-dimensional Gaussian kernel such that the total span is about $7\sigma_f$.

(b) Construct two files g_h, g_v from the horizontal and vertical convolutions of the digital image with the 1-d Gaussian kernel.

(c) Calculate an estimate of the gradient magnitude and direction using the following equations:

$$\mathrm{MAG} = [g_h{}^2 + g_v{}^2]^{0.5} \tag{5.80}$$

$$\mathrm{ANG} = \tan^{-1}(\frac{g_v}{g_h}). \tag{5.81}$$

(d) Quantize the direction map into eight possible values, as in Figure 5.2b.

5.6.2. Performance on Test Images

We used two test images to evaluate the performance of the linking algorithms: The step image of Pratt [Pra78] and the discs image of Kitchen and Rosenfeld [KiR81]. These images have been used in the literature to test the performance of edge operators. Zero-mean Gaussian noise with standard deviation σ is added to each test image. The signal-to-noise ratio (SNR) is defined as follows:

$$\mathrm{SNR} = (\frac{\Delta}{\sigma})^2 \tag{5.82}$$

where Δ is the image contrast (step height).

The Pratt figure of merit [Pra78] is used as a qualitative measure for the performance of the three algorithms on the step image. This figure of merit (R) penalizes an edge detection algorithm for wrong edge points and for missing an edge point. It is defined as follows [Pra78]:

$$R = \frac{1}{I_N} \sum_{i=1}^{I_A} \frac{1}{1 + \alpha d^2}, \tag{5.83}$$

where $I_N = \max(I_I, I_A)$ and I_I and I_A represent the number of ideal and actual edge map points, α is a scaling constant, and d is the separation distance of an actual edge point normal to a line of ideal edge points. The above measure is normalized such that $R = 1$ for perfectly detected edge points. The scaling factor α may be adjusted to penalize edges that are localized but offset from the actual edge position. Normalization by the maximum of the ideal and actual edge elements insures a penality for smeared or fragmented edges [Pra78].

1. Step Image

The step test image is a 128 × 128 image formed such that the first half has a gray level value of 115, the second half has a gray level value of 140, and the one-pixel wide column located in the center has a gray level value of 128. A zero-mean Gaussian noise is added to this image. The variance of the Gaussian noise was selected such that the SNR defined in (5.82) takes on the values 100, 10, 1, and 0.5. Figure 5.11 shows the original image for various SNR.

In Figure 5.12, the top row shows the results of NONMAX for various SNR, the second row shows the results of SEL for various SNR, the third row shows the results of the SEL2 algorithm, and the fourth row shows the results of LINK. Threshold values used in these graphs were 3, 9, 22, 28 for SNR = 100, 10, 1, 0.5, respectively. Edge enhancement was performed by the ∇G operator of size 7 × 7 (i.e., the standard deviation of the Gaussian kernel $\sigma_f = 1.0$).

Figure 5.13 is similar to Figure 5.12, except the size of the ∇G operator was 9 × 9 instead (i.e., the standard deviation of the Gaussian kernel $\sigma_f = 1.3$). Threshold values used in these graphs were 2, 6, 17, 21 for

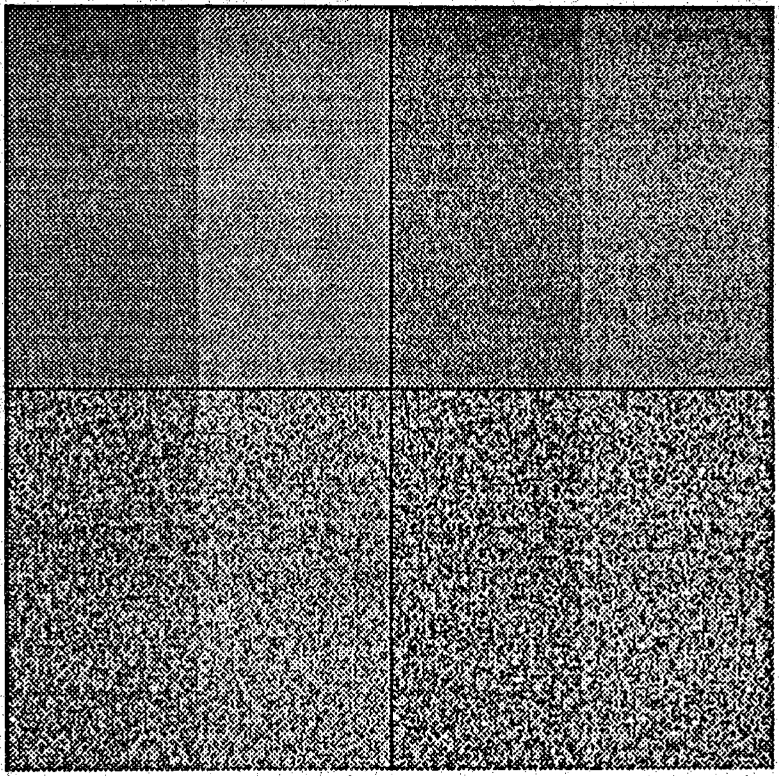


Figure 5.11. A noisy step test image.
(Upper left: SNR = 100; Upper right: SNR = 10
Lower left: SNR = 1; Lower right: SNR = 0.5)

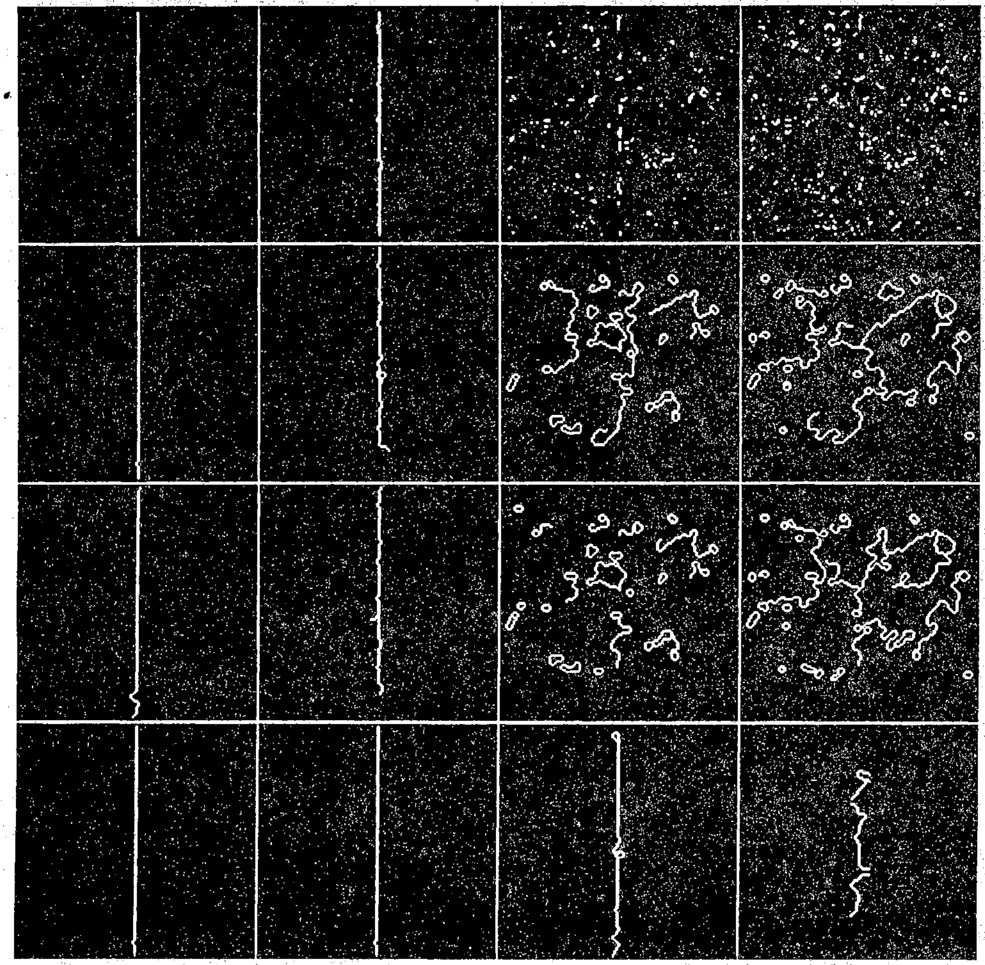


Figure 5.12. Performance of the linking algorithms on the step image.
Edges enhanced by a 7 × 7 ∇G operator
(Top row: Edge elements extracted by NONMAX.
Second row: Edge linking by the SEL algorithm.
Third row: Edge linking by the SEL2 algorithm.
Bottom row: Edge linking by the LINK algorithm.)

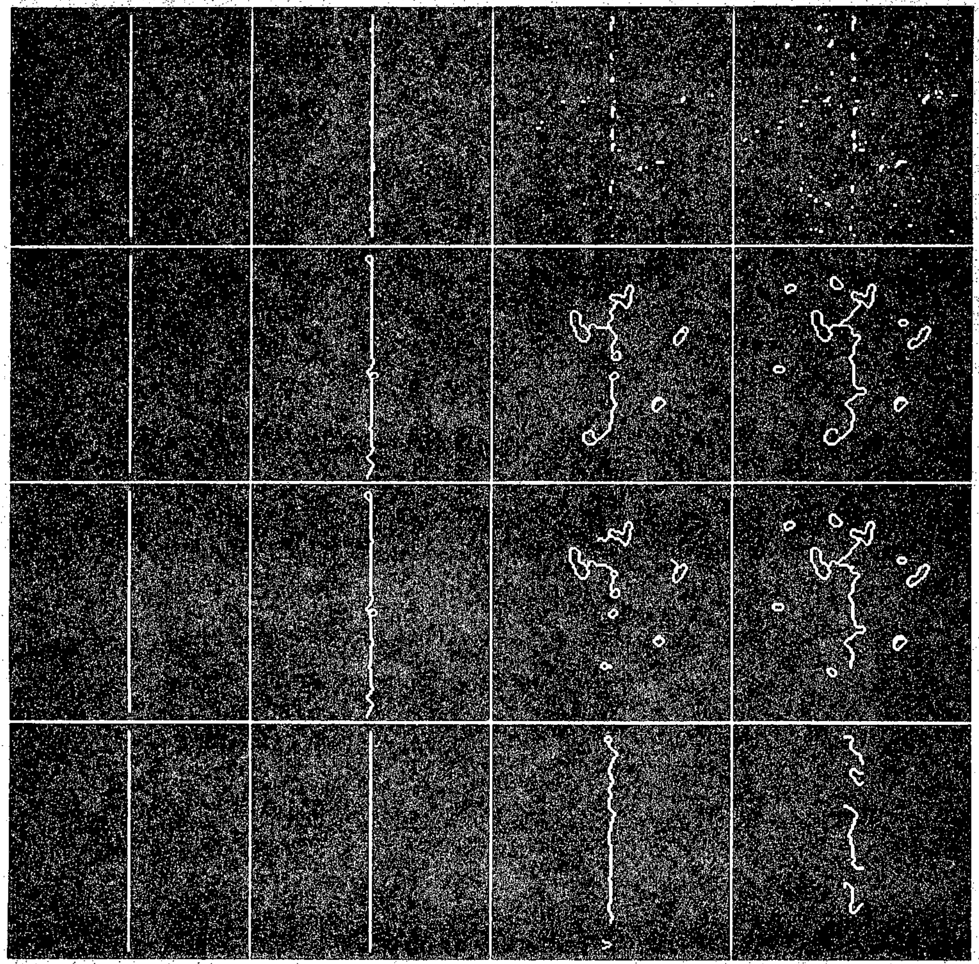


Figure 5.13. Performance of the linking algorithms on the step image.
Edges enhanced by a 9 × 9 ∇G operator
(Top row: Edge elements extracted by NONMAX.
Second row: Edge linking by the SEL algorithm.
Third row: Edge linking by the SEL2 algorithm.
Bottom row: Edge linking by the LINK algorithm.)

SNR = 100, 10, 1, 0.5, respectively.

All algorithms performed well at SNR = 100, 10. The contour obtained from the SEL oscillated slightly around the actual edge location, however. At SNR = 1, NONMAX provided a broken edge contour and the number of false edge points increased as we lowered the threshold. SEL also provided an oscillatory and broken contour, while LINK provided the fewest oscillations of the three algorithms. The same conclusion can be drawn at SNR = 0.5. The Pratt figure of merit for the three algorithms is shown in Table 5.2. for the 7 × 7 window and in Table 5.3 for the 9 × 9 window. The tables indicate that LINK performed best, especially at lower SNR. The results of SEL and SEL2 are very close on the step image. The main advantages of using the suboptimal metric in (5.34)-(5.41) is that the metric is estimated from the actual image, thus we can isolate the errors due the metric from the errors due to the search algorithm.

2. Discs Image

The discs image is a 128 × 128 image formed as in [KiR81]. Briefly, the image is formed originally from a 512 × 512 image consisting of two gray levels, 115 (dark) and 140 (light). The image contains a dark circle of radius 64 at its center surrounded by six concentric circles of width 32 with alternating dark and light gray levels. The final 128 × 128 image is obtained by compressing the original image by a factor of four. The contrast Δ is 25, the same as in the step image. Zero-mean Gaussian noise was also added to the original image. Figure 5.14 shows the original image for various SNR.

Figure 5.15 shows the results of the algorithms on the discs image. The threshold values were 8, 8, 21, 30 for SNR = 100, 10, 1, 0.5, respectively. The spatial support of the ∇G was 7 × 7. Figure 5.16 shows the results of the algorithms on the discs image. The threshold values were 6, 6, 15, 21 for SNR = 100, 10, 1, 0.5, respectively. The spatial support of the ∇G was 9 × 9.

At SNR = 10, SEL produced an incomplete contour in parts of the second circle from the outside. On the other hand, few false boundaries were detected by LINK because the threshold set on Λ was slightly low. As the SNR is reduced, the performance of the algorithms deteriorates. This was

Table 5.2. The Pratt figure of merit for 7 × 7 window.

SNR	Pratt Figure of Merit			
	NONMAX	SEL	SEL2	LINK
100	100%	99%	100%	100%
10	85%	85%	85%	94%
1	51%	50%	49%	53%
0.5	45%	25%	24%	39%

Table 5.3. The Pratt figure of merit for 9 × 9 window.

SNR	Pratt Figure of Merit			
	NONMAX	SEL	SEL2	LINK
100	100%	100%	100%	100%
10	96%	94%	94%	94%
1	51%	52%	50%	62%
0.5	45%	30%	29%	49%

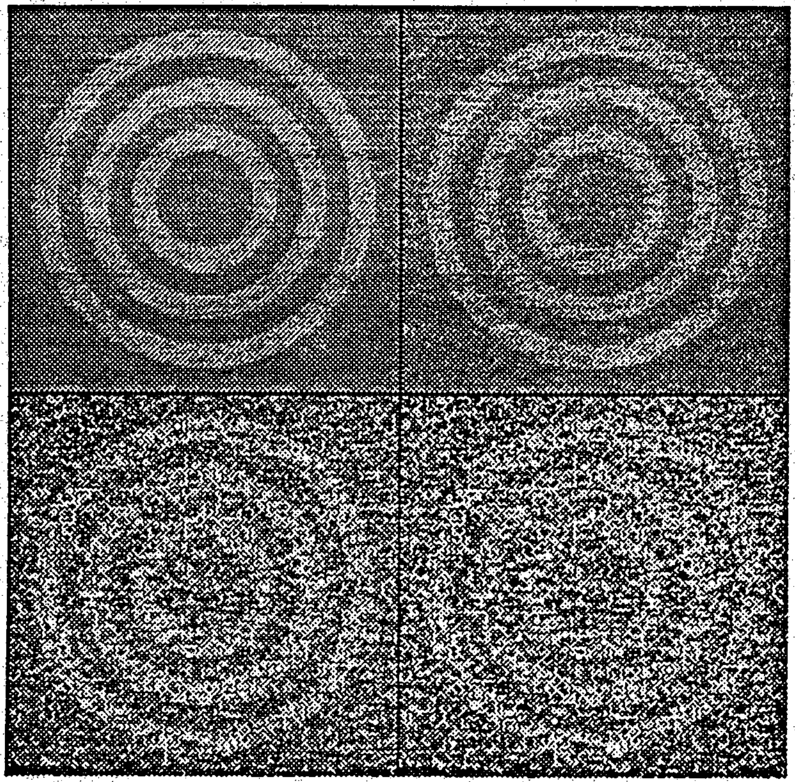


Figure 5.14. A noisy discs test image.
(Upper left: SNR = 100; Upper right: SNR = 10
Lower left: SNR = 1; Lower right: SNR = 0.5)

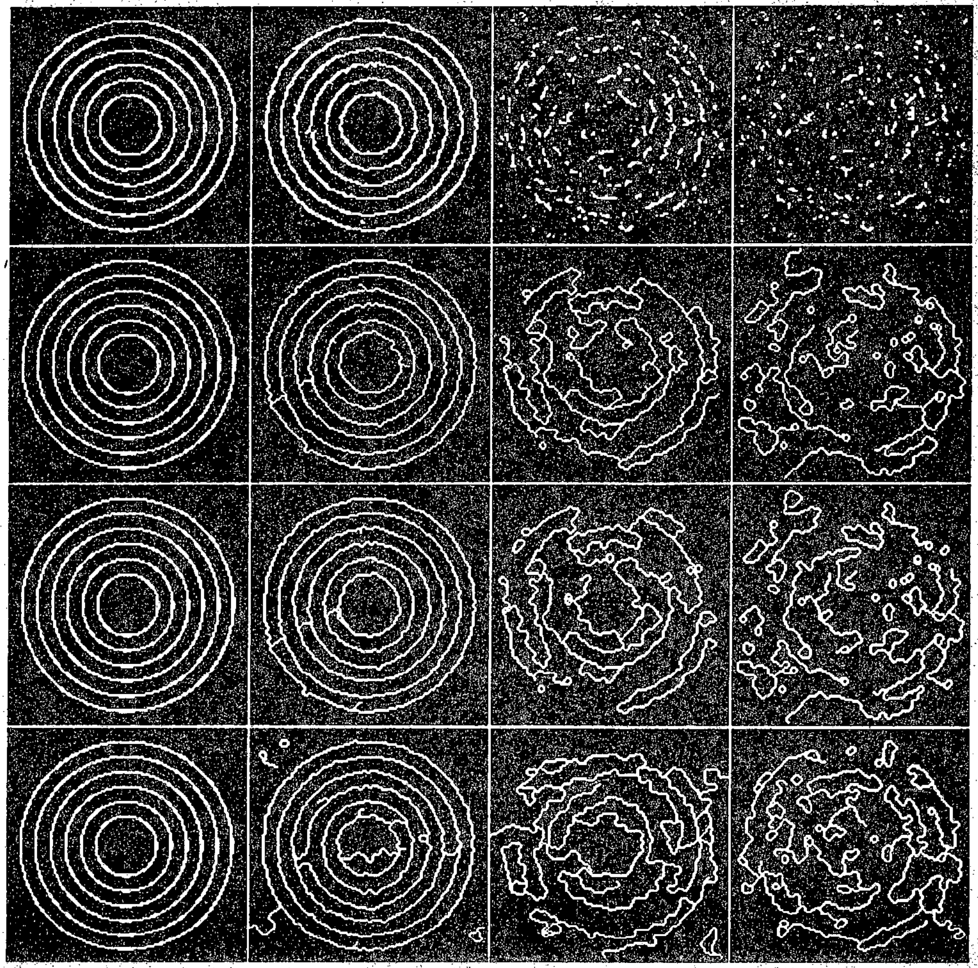


Figure 5.15. Performance of the linking algorithms on the discs image.
Edges enhanced by a 7 × 7 ∇G operator
(Top row: Edge elements extracted by NONMAX.
Second row: Edge linking by the SEL algorithm.
Third row: Edge linking by the SEL2 algorithm.
Bottom row: Edge linking by the LINK algorithm.)

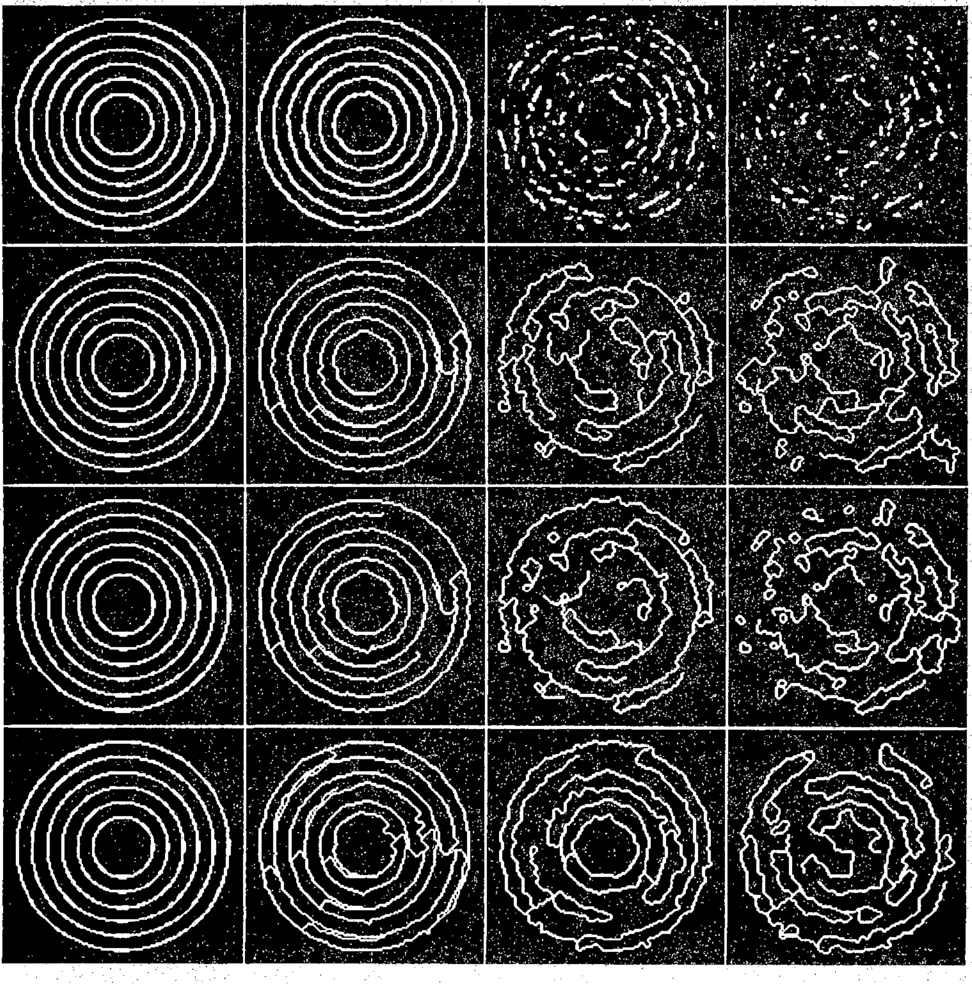


Figure 5.16. Performance of the linking algorithms on the discs image.
Edges enhanced by a 9 × 9 ∇G operator
(Top row: Edge elements extracted by NONMAX.
Second row: Edge linking by the SEL algorithm.
Third row: Edge linking by the SEL2 algorithm.
Bottom row: Edge linking by the LINK algorithm.)

the case for either enhancement filter spatial supports.

The following conclusions can be drawn from the above experiments on the step image and the discs image: First, the performance of the algorithms depends to a large extent on the accuracy of the enhancement step. At this point we should emphasize that a larger spatial support of the ∇G operator might not always be a good choice and the classical compromise between spatial and temporal resolutions should always be addressed. Secondly, the path definitions used in SEL (also adopted in our experiments) might not be suitable for highly oscillating edge boundaries.

Since the results of SEL and SEL2 were very comparable on test images, we will compare the results of LINK vs NONMAX and SEL only on real scenes.

5.6.3. Performance on Real Scenes

In this section we discuss the relative performance of LINK, NONMAX, and SEL on some real world scenes. The following description is common to all figures: The original image is shown in the upper left-hand corner, NONMAX results in the upper right-hand corner, SEL results in the lower left-hand corner, and the results of LINK are shown in the lower right-hand corner.

Figure 5.17 shows the results on the airport image. Edge enhancement was accomplished by a ∇G operator of spatial support 7×7, (i.e., $\sigma_f = 1.0$). The SEL missed totally the airplane located on the topmost, right-hand side of the original image. This specific airplane was detected by LINK; a few extra details along with the airplane's boundaries were also detected, however. The upper bound for the threshold used in the three algorithms was 30.0.

Figure 5.18 shows the results on the mechanical parts (rods) image. Enhancement was performed by a ∇G operator of spatial support 14×14 (i.e., $\sigma_f = 2$). The upper bound threshold was 100. The three algorithms performed approximately the same. SEL did not provide a complete outline for the contours while LINK contours were better localized. A smaller spatial support for the enhancement algorithm should be more appropriate for this

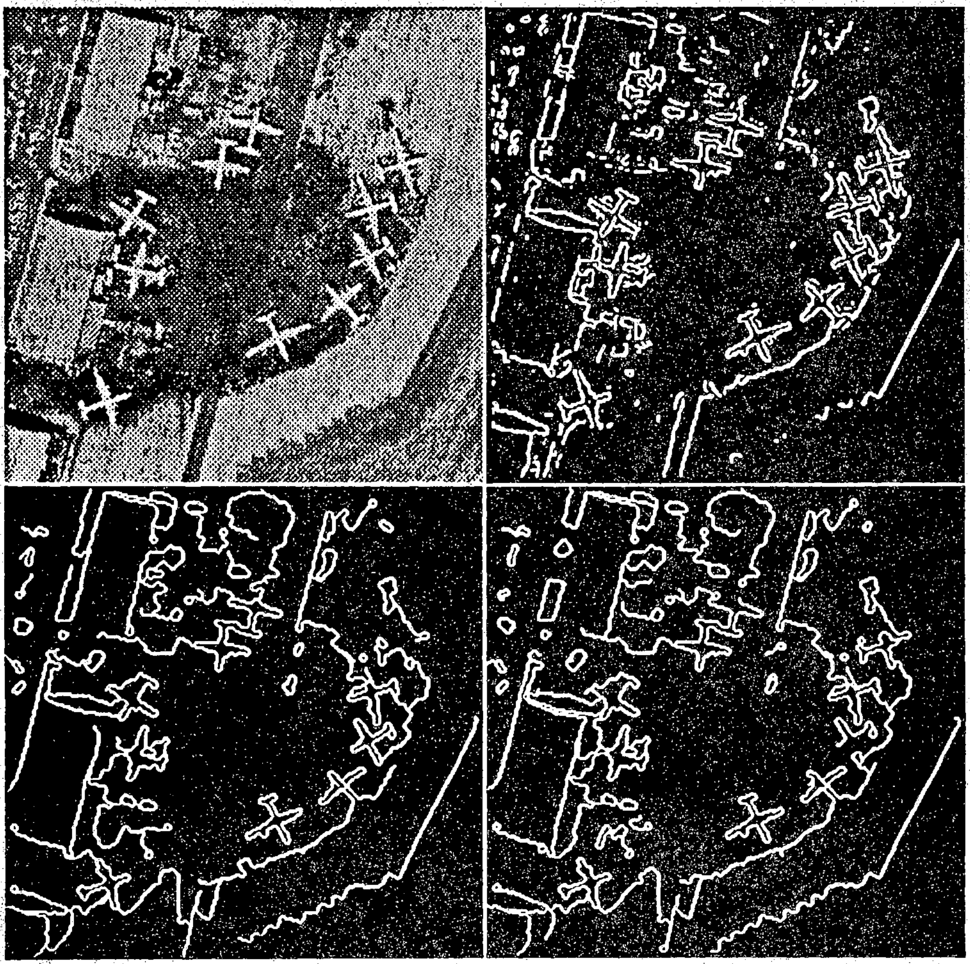


Figure 5.17. Performance of the linking algorithms on the airport image. (Upper left: original, upper right: NONMAX, lower left: SEL, lower right: LINK)

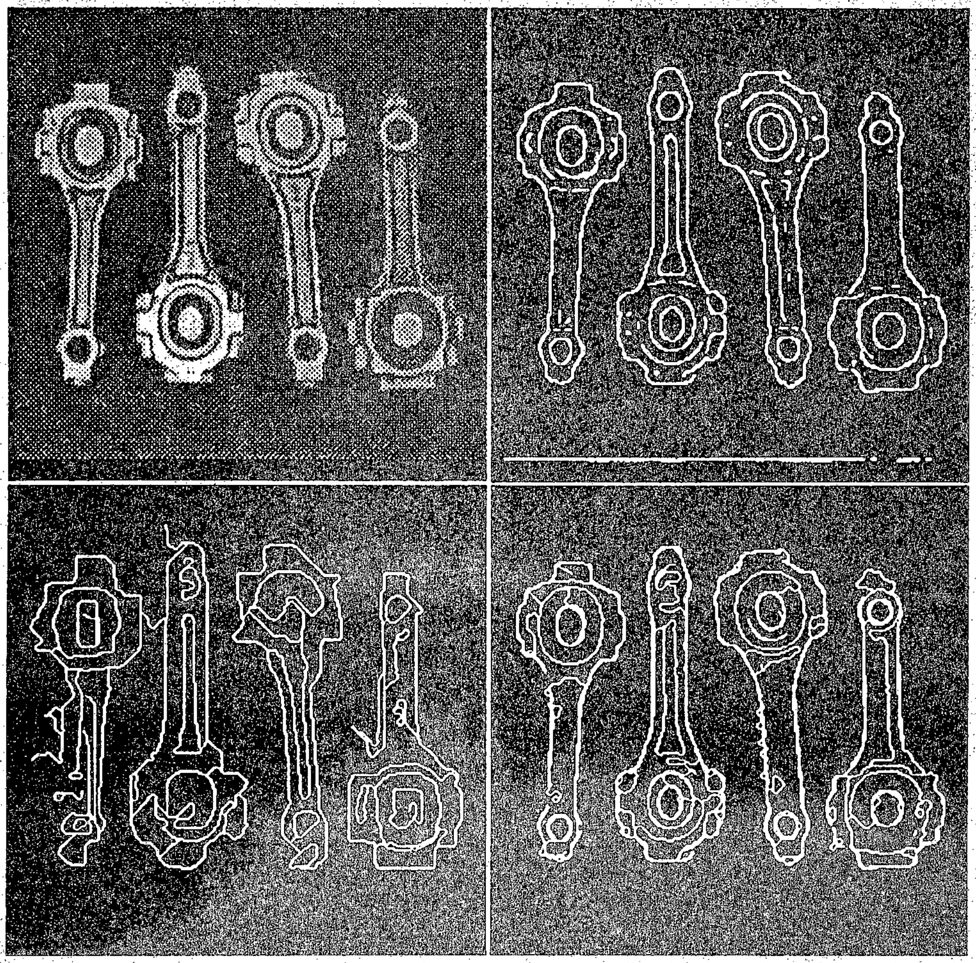


Figure 5.18. Performance of the linking algorithms on the rods image. (Upper left: original, upper right: NONMAX, lower left: SEL, lower right: LINK)

image.

Figure 5.19 shows the results on the outdoor scene enhanced by a ∇G operator of spatial support 7 × 7 and a threshold upper bound of 25. Many of the details in the image were preserved by the algorithms, in spite of its overcrowding. Note that the trees, particularly those on the upper left-hand portion of the original image, were not picked up by the SEL because of the inherent tendency to emphasize long edges. Also note that LINK boundaries were not completely closed, especially in the car image. This is an example of where the metrics in the linking algorithms cannot adequately allocate the object boundaries. A priori information of the scene can be useful in such situations.

Finally, Figure 5.20 shows the performance on the girl image enhanced by a ∇G operator of spatial support 7 × 7 and an upper bound threshold of 23 for the three linking algorithms. Notice the distinct improvement on this picture compared to Figure 5.19. The picture here was less crowded, and the contrast between the object and the background was great which resulted in a very good enhancement map. This was reflected on the response of the three linking algorithms. LINK shows, however, too many details of the image. A higher threshold for the Λ used in the algorithm would reduce these details.

The following conclusions can be drawn form the results of the algorithms on real scenes. First, LINK performed at least as well as the SEL on all of the images. This indicates that the assumptions and the heuristics included with these algorithms are adequate. Secondly, when the contrast is low or when the image is crowded, only LINK provided adequate details about the object's boundaries in the scene. Finally, LINK performed as fast as the SEL because of the restrictions which were included on the size of the search domain.

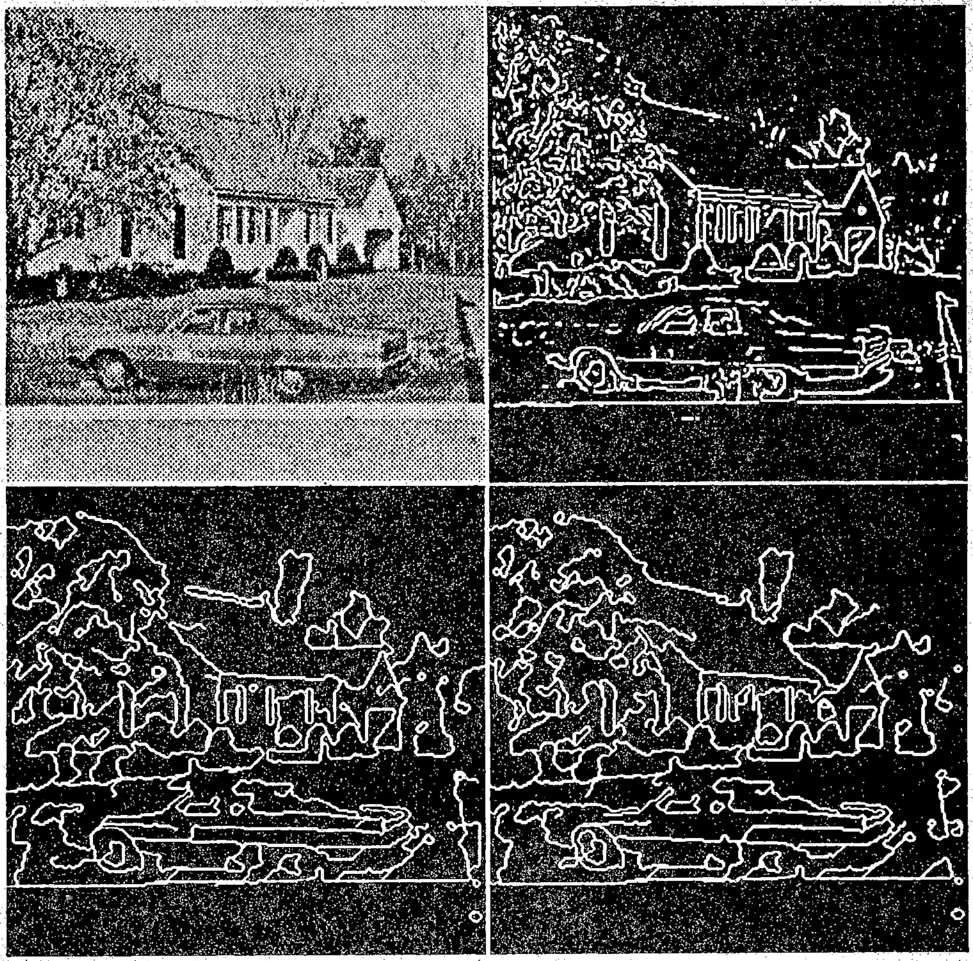


Figure 5.19. Performance of the linking algorithms on the outdoor image.
(Upper left: original, upper right: NONMAX, lower left: SEL, lower right: LINK)

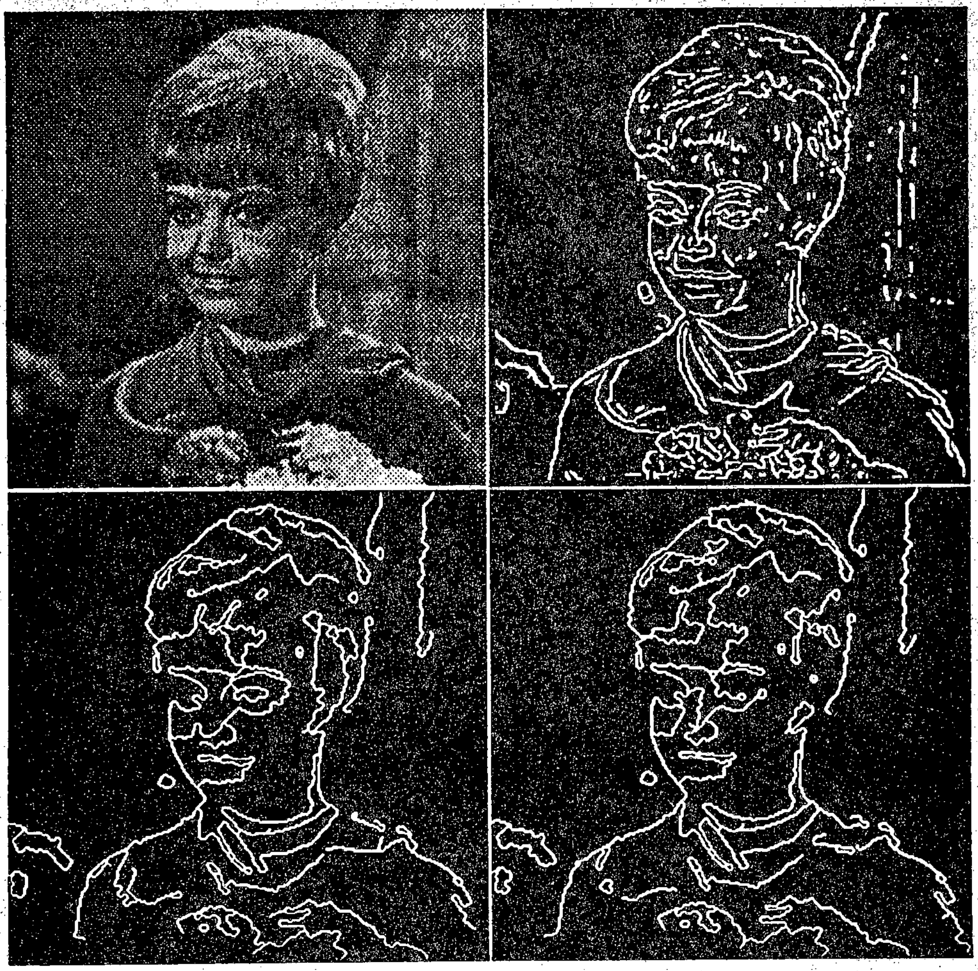


Figure 5.20. Performance of the linking algorithms on the girl image.
(Upper left: original, upper right: NONMAX, lower left: SEL, lower right: LINK)

5.7. Summary

In this Chapter we examined the application of sequential search techniques for edge linking. Various important issues of sequential search were studied. A new metric based on the linear model was developed and analyzed. A search algorithm (LINK) that used this metric and the Stack algorithm for edge linking was constructed. Results of LINK compare well with respect to nonmaximal suppression and thresholding [Can86] and also with respect to the SEL algorithm [EiD85b], and [EiD88]. The metric is easy to compute and provides more accurate results than the nonmaximal suppression technique, and at least as accurate results as the SEL algorithm with comparable execution time.

CHAPTER 6
CONCLUSIONS

6.1. Problem Review

In this thesis we have developed two image segmentation algorithms based on recent results in random field theory. As we pointed out, the purpose of image segmentation is to isolate objects in a scene from the background. This is a very important step in any computer vision system since various tasks, such as shape analysis and object recognition, require accurate image segmentation. Image segmentation can also produce tremendous data reduction. Edge-based and region-based segmentation have been examined and two new algorithms have been developed.

The edge-based segmentation algorithm uses the pixel gray level intensity information to allocate object boundaries in two stages: Edge enhancement, followed by edge linking. Edge enhancement is accomplished by maximum energy filters used in one-dimensional bandlimited signal analysis. The issue of filter spatial support was examined on ideal edge models. Edge linking is performed by quantitative sequential search using the Stack algorithm. A probabilistic search metrics is introduced and its performance is evaluated on test as well as real scenes. Compared to other methods, this algorithm is shown to produce more accurate allocation of object boundaries.

Region-based segmentation was modeled as a MAP estimation problem in which the actual (unknown) objects were estimated from the observed (known) image by recursive algorithm. The observed image was modeled by an autoregressive (AR) model whose parameters were estimated locally, and a Gibbs-Markov random field (GMRF) model was used to model the unknown scene. A computational study was conducted on images having various types of texture images. The issues of parameter estimation, neighborhood selection, and model orders were examined.

6.2. Conclusions

The following are the main conclusions of this research. First, it was shown that a number of edge enhancement filters belong to the class of maximum energy filters. The link between these filters is used to study the problem of filter spatial support selection. We have shown that for ideal edges (e.g., edges in the step or discs test images), an accurate formula relating the filter width to the edge structure was possible. This is not, however, possible in general because the actual edge structure cannot be known a priori in actual scenes. Gaussian smoothing before taking the derivative (Laplacian or gradient) provides the necessary regularization to change the ill-posed numerical differentiation problem into a well-posed problem.

Secondly, the problem of sequential search as applied to edge linking was further quantified. We have developed a suboptimal metric to guide the search in the SEL algorithm. The proposed metric is globally optimal (i.e., it corresponds to the metric in [EiD85a]) and locally suboptimal since it uses data on a neighborhood of a pre-selected boundary. The optimality in question relates mainly to the desired drift property; that is, the metric will tend to have a smaller magnitude on wrong (random) edge paths than its value on correct edge paths [EiD85a], [EiD88]. We have developed estimation techniques for this metric that uses the gradient magnitude in the edge enhanced image by a gradient operator (e.g., the ∇G operator). Through this suboptimal metric, the SEL algorithm becomes fully data-driven. A novel metric based on the linear model was also introduced. By fitting a linear model to four possible edge models defined on a 3×3 windows of the edge enhanced image, a likelihood ratio test (LRT) was derived to guide the process of node extension in sequential search. The resulting metric is easy to implement and provides a very comparable results to the SEL metric on test as well as real world images. The new metric is data-driven, and the execution time of the search algorithm with this metric is lower than its value with the SEL metric.

Finally, we conclude that the MAP approach for region segmentation generally works well on images having a large content of microtextures, which can be properly modeled by both AR and GMRF models. On these texture images, second-order AR and GMRF models were shown to be adequate.

6.3. Suggestions for Further Work

Sequential search relies on the ability of the evaluation function (path metric) to discover only those paths having a higher probability of being object boundaries. In adjusting the edge linking problem to sequential tree search, heuristics were introduced. Since this, in general, is problem related, the need exists for an optimum set of rules which are not problem dependent. This can be done by further quantifying the edge enhancement step such that edge definitions are specifically included in the linking algorithm. Although this was done in our current work, the question of edge definition to a great extent remains open.

The search metrics developed in this thesis depend upon the random field model fitted to the data window. A class of random field models can be used in this regard. An accurate criterion for the selection between the various members of this class is needed. This is still an open research area of both theoretical and practical importance. Also, modern stochastic optimization techniques such as simulated annealing might be useful in getting the MAP solution instead of the recursive approach we implemented in this thesis.

Finally, a quantitative comparison between the more recent model-based techniques for edge detection techniques, whether parallel (e.g., Geman et al. [GeG88]) or sequential (e.g., the algorithms developed in this thesis), should be researched. Further study will help us understand the advantages and limitations of using random field concepts in edge detection. Further research is also needed in the area of image segmentation using random field models. The majority of the algorithms in the literature, thus far, have centered on texture images.

LIST OF REFERENCES

LIST OF REFERENCES

[AbP79] I. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proceedings of the IEEE,* Vol. 67, No. 5, pp. 753-763, May 1979.

[AnD88] N. Ansari and E. J. Delp, *Shape Recognition: A Landmark-Based Approach,* TR-EE 88-31, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, July 1988.

[AsM78] G. Ashkar and J. Modestino, "The contour extraction problem with biomedical applications," *Computer Graphics and Image Processing,* Vol. 7, pp. 331-355, 1978.

[BaB82] D. H. Ballard and C. M. Brown, *Computer Vision,* Prentice Hall, New Jersey, 1982.

[BaC88] L. Basano, B. Caprile, E. De Micheli, A. Geminiani, and P. Ottonello, "Edge-detection schemes highly suitable for hardware implementation," *The Journal of Optical Society of America,* Vol. 5, No. 7, pp. 1170-1175, July 1988.

[BaE81] M. Basseville, B. Espian, and J. Gasiner, "Edge detection using sequential methods for change in level-- Part 1: A sequential edge detection algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. ASSP-29, No. 1, pp. 24-31, February 1981.

[Bar75] M. S. Bartlett, *The Statistical Analysis of Spatial Pattern,* Chapman and Hall, London, 1975.

[BeP88] M. Bertero, T. Poggio, and V. Torre, "Ill-posed problems in early vision," *Proceedings of the IEEE,* Vol. 76, No. 8, pp. 869-889, August 1988.

[Ber84] V. Berzins, "Accuracy of the Laplacian edge detector," *Computer Vision, Graphics, and Image Processing,* Vol. 27, pp. 195-210, 1984.

[Bes72] J. E. Besag, "Nearest neighbor systems and the auto-logistic model for binary data," *Journal of the Royal Statistical Society,* London, Series B, Vol. B-35, pp. 75-83, 1972.

[Bes74] J. E. Besag, "Spatial interaction and statistical analysis of lattice systems," *Journal of the Royal Statistical Society,* London, Series B, Vol. B-35, pp. 192-236, 1974.

[Bes75] J. E. Besag, "Statistical analysis of non-lattice data," *The Statistician,* Vol. 24, pp. 179-95, 1975.

[Bes78] J. E. Besag, "Some methods of statistical analysis for spatial data," *Bulletin of the International Statistical Institute,* Vol. 47, pp. 77-92, 1978.

[Bes86] J. E. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society,* London, Series B, Vol. B-48, pp. 259-302, 1986.

[Bie86] J. Biemond, "Stochastic linear image restoration," *Advances in Computer Vision and Image Processing,* ed. S. T. Huang, Vol. 2, Chapter 5, JAI Press, Greenwich, Connecticut, 1986.

[BoL88] C. Bouman and B. Liu, "Segmentation of textured images using a multiple resolution approach," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* New York City, New York, pp. 1124-1127, April 1988.

[BoL89] C. Bouman and B. Liu, "Multiple resolution segmentation of textured images," *Preprint,* 1989.

[Can83] J. Canny, "Finding edges and lines in edges," Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, June 1983.

[Can86] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 6, pp. 679-698, November 1986.

[Car72] L. Carlucci, "A formal system for texture languages," *Pattern Recognition,* Vol. 4, pp. 53-72, 1972.

[Cer85] V. Cerny, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications,* Vol. 45, pp. 41-51, 1985.

[ChD88] H. Chu and E. J. Delp, *The Analysis of Image Sequence Data with Applications to Two-Dimensional Echocardiography,* TR-EE 88-32, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, July 1988.

[Che81] R. Chellappa, *Stochastic Models in Image Analysis and Processing,* Ph.D. dissertation, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, August 1981.

[Che85] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," *Progress in Pattern Recognition 2,* ed. L.N. Kanal and A. Rosenfeld, Elsevier Science Publishers B. V. (North-Holland), 1985.

[ChK82] R. Chellappa and R. L. Kashyap, "Digital image restoration using spatial interaction models," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. ASSP-30, No. 3, pp. 461-472, June 1982.

[Cla89] J. J. Clark, "Authenticating edges produced by zero-crossing algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-11, No. 1, pp. 43-57, January 1989.

[CoA79] G. Coleman and H. C. Andrews, "Image segmentation and clustering," *Proceedings of the IEEE,* Vol. 67, No. 5, pp. 773-785, May 1979.

[CoE80] D. B. Cooper, H. Elliott, F. Cohen, L. Reiss, and P. Symosek, "Stochastic boundary estimation and object recognition," *Computer Graphics and Image Processing,* Vol. 12, pp. 326-356, 1980.

[Coo79] D. Cooper, "Maximum likelihood estimation of Markov-process blob boundaries in noisy images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-1, No. 4, pp. 372-384, October 1979.

[CrJ83] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-5, No. 1, pp. 149-163, January 1983.

[Dav75] L. S. Davis, "A survey of edge detection techniques," *Computer Graphics and Image Processing,* Vol. 4, pp. 248-270, 1975.

[DeE87] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-9, No. 1, pp. 39-55, January 1987.

[DeK79] E. J. Delp, R. L. Kashyap, and O. R. Mitchell, "Image data compression using autoregressive time series models," *Pattern Recognition,* Vol. 11, pp. 313-323, 1979.

[DeM78] K. Deguchi and I. Morishita, "Texture characterization and texture based image partitioning using two-dimensional linear estimation techniques," *IEEE Transactions on Computers,* C-27, pp. 738-745, August 1978.

[DeP88] R. Dechter and J. Pearl, "The optimality of A*," *Search in Artificial Intelligence,* ed. L. Kanal and V. Kumar, Springer Verlag, New York, pp. 166-199, 1988.

[DiS77] F. M. Dickey and K. S. Shanmugam, "Optimum edge detection filter," *Applied Optics,* Vol. 16, No. 1, pp. 145-148, January 1977.

[DoF86] H. S. Don and K. S. Fu, "A parallel algorithm for stochastic image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 5, pp. 594-603, September 1986.

[DuH73] R. Duda and P. Hart, *Pattern Classification and Scene Analysis,* J. Wiley, New York, 1973.

[EiD84] P. H. Eichel and E. J. Delp, "Sequential edge Linking," *Proceedings of the 22nd Allerton Conference on Communications, Control, and Computers,* Monticello, Illinois, pp. 782-791, October 1984.

[EiD85a] P. H. Eichel and E. J. Delp, "Sequential detection in correlated random fields," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* San Francisco, pp. 14-21, June 1985.

[EiD85b] P. H. Eichel and E. J. Delp, *Sequential Detection of Linear Features in Two-Dimensional Random Fields,* TR-EE 85-13, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, July 1985.

[EiD88] P. H. Eichel, E. J. Delp, K. Koral, and A. J. Buda, "A method for fully automatic detection of coronary arterial edges for cineangiograms," *IEEE Transactions on Medical Imaging,* Vol. MI-7, No. 4, pp. 313-320, December 1988.

[ElS81] H. Elliott and L. Srinivasn, "An application of dynamic programming to sequential boundary estimation," *Computer Graphics and Image Processing,* Vol. 17, pp. 291-314, 1981.

[ElC82] H. Elliott, D. Cooper, F. Cohen, and P. Symosek, "Implementation, interpretation, and analysis of a suboptimal boundary finding algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-4, No. 2, pp. 167-182, March 1982.

[Eom86] Kie-Bum Eom, *Robust Image Models with Application,* Ph.D. dissertation, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, August 1986.

[FaD86] A. A. Farag and E. J. Delp, " Some experiments with histogram-based segmentation," *Proceedings of the 1986 Conference on Intelligent Systems and Machines,* Oakland University, Rochester, Michigan, pp. 251-256, 1986.

[Fra81] L. Franks, *Signal Theory,* Dowden & Culver, Inc., Stroudsburg, Pennsylvania, 1981.

[FrC77] W. Frie and C. Chen, "Fast boundary detection: A generalization and a new algorithm," *IEEE Transactions on Computers,* Vol. C-26, No. 10, pp. 988-998, 1977.

[Fre61] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computer,* Vol. EC-10, pp. 260-268, June 1961.

[FuM81] K. S. Fu and J. K. Mui, "A survey on image segmentation," *Pattern Recognition,* Vol. 13, pp. 3-16, 1981.

[Fu82] K. S. Fu, *Syntactic Pattern Recognition and Applications,* Prentice-Hall, New Jersey, 1982.

[GeG84] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Baysian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-6, No. 6, pp. 721-741, 1984.

[GeG86] S. Geman and C. Graffigne, "Markov random field image models and their applications to computer vision," *Proceedings, International Congress of Mathematicians,* Berkeley, California, pp. 1496-1517, 1986.

[GeG87] D. Geman, S. Geman, and C. Graffigne, "Locating texture and object boundaries," *Pattern Recognition Theory and Applications,* ed. P. V. Devijver and J. Kittler, Springer Verlag, Berlin, pp. 165-177, 1987.

[GeG88] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *Preprint,* 1988.

[Gem87] D. Geman, "A stochastic model for boundary detection," *Image and Vision Computing,* pp. 61-65, May 1987.

[GoW87]R. Gonzalez and P. Wintz, *Digital Image Processing,* 2nd ed. Addison Wesley, Reading, Massachusetts, 1987.

[Gra76] F. A. Graybill, *Theory and Applications of the Linear Model,* Duxbury Press, Massachusetts, 1976.

[Gra87] C. Graffigne, *Experiments in Texture Analysis and Segmentation,* Ph.D. dissertation, Division of Applied Mathematics, Brown University, Providence, Rhode Island, May 1987.

[GrH85] W. E. Grimson and E. C. Hildreth, "Comments on: Digital step edges from zero crossings of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-7, No. 1, pp. 121-127, January 1985.

[Gri76] D. Griffeath, "Introduction to random fields," *Denumerable Markov Chains,* ed. Kemeny, Knapp, and Snell, Springer Verlag, New York, pp. 425-458, 1976.

[HaH64] J. M. Hammersely and D. C. Handscomb, *Monte Carlo Methods,* J. Wiley, New York, 1964.

[Hal79] E. Hall, *Computer Image Processing and Recognition,* Academic Press, New York, 1979.

[HaN68] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics,* SSC-4, No. 2, pp. 100-107, 1968.

[Har79] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE,* Vol. 67, No. 5, pp. 786-804, May 1979.

[Har81] R. M. Haralick, "A facet model for image data," *Computer Graphics and Image Processing,* Vol. 15, pp. 113-129, 1981.

[Har84] R. M. Haralick, "Digital step edges from zero crossings of second directional derivatives," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-6, No. 1, pp. 58-68, January 1984.

[Har85] R. M. Haralick, "Digital step edges from zero crossings of second directional derivatives: Author's reply," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-7, No. 1, pp. 127-129, January 1985.

[Har86] R. M. Haralick, "Statistical image texture analysis," *Handbook of Pattern Recognition and Image Processing,* ed. Young and Fu, Chapter 11, Academic Press, New York, 1986.

[HaS80] M. Hassner and J. Sklansky, "The use of Markov random fields as models of texture," *Computer Graphics and Image Processing,* Vol. 12, pp. 357-370, 1980.

[HaS85] R. Haralick and L. Shapiro, "Survey: Image segmentation techniques," *Computer Vision, Graphics, and Image Processing,* Vol. 29, pp. 100-132, 1985.

[Hil80] E. C. Hildreth, "Implementation of a theory of edge detection," Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, April 1980.

[Hue71] R. Hueckel, "An operator which locates edges in digital pictures," *Journal of Association of Computation Machinery,* Vol. 18, pp. 113-125, 1971.

[Hun73] B. R. Hunt, "The application of constrained least squares estimation to image restoration by digital computer," *IEEE Transactions on Computers,* Vol. C-22, No. 9, pp. 805-812, August 1973.

[Jai81] A. K. Jain, "Advances in mathematical models for image processing," *Proceedings of the IEEE,* Vol. 69, No. 5, pp. 502-528, May 1981.

[Jel69] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development,* Vol. 13, pp. 675-685, November 1969.

[KaC83] R. L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial interaction models of images," *IEEE Transactions on Information Theory,* Vol. IT-29, pp. 60-72, January 1983.

[KaE88] R. L. Kashyap and K. Eom, "Robust image models and their applications," *Advances in Electronics and Electron Physics,* ed. P. W. Hawkes, Vol. 72, Academic Press, 1988.

[KaE89] R. L. Kashyap and K. Eom, "Texture boundary detection based on the long correlation model," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-11, No. 1, pp. 58-67, January 1989.

[KaK86] R. L. Kashyap and A. Khotanzad, "A model-based method for rotation invariant texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 4, pp. 472-481, July 1986.

[KaK88] L. Kanal and V. Kumar, eds., *Search in Artificial Intelligence*, Springer Verlag, New York, 1988.

[KaL86] S. M. Kay and G. J. Lemay, "Edge detection using the linear model," *IEEE Transactions on Acoustics, Speech, and Signal Processing* Vol. ASSP-34, No. 5, pp. 1221-1227, 1986.

[KaS88] A. K. Katsaggelos and B. J. Sullivan, "Regularized edge detection," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* New York City, New York, pp. 1048-1051, April 1988.

[Kas80] R. L. Kashyap, "Random field models on torus lattices for finite images," *Proceedings of the 5th International Conference on Pattern Recognition,* Miami, Florida, March 1980.

[Kas81] R. L. Kashyap, "Analysis and synthesis of image patterns by spatial interaction models," *Progress in Pattern Recognition,* ed. L. N. Kanal and A. Rosenfeld, North-Holland Publishing Co., 1981

[Kas86] R. L. Kashyap, "Image models," *Handbook of Pattern Recognition and Image Processing,* ed. Young and Fu, Chapter 12, Academic Press, New York, 1986.

[KeS77] M. Kendall and A. Stuart, *The Advanced Theory of Statistics,* Vol. 2, MacMillan, New York, 1977.

[Kho83] A. Khotanzad, *Stochastic Model-Based Techniques for Classification and Segmentation of Textures,* Ph.D. dissertation, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, December 1983.

[KiG83] S. Kirkpatrick, C. D. Gelett, and M. P. Vecchi, "Optimization by simulated annealing," *Science,* Vol. 220, pp. 621-680, May 1983.

[Kir71] R. Kirsch, "Computer determination of the constituent structure of biological images," *Computer and Biomedical Research,* Vol. 4, No. 3, pp. 315-328, June 1971.

[KiR81] L. Kitchen and A. Rosenfeld, "Edge evaluation using local edge coherence," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-11, No. 9, pp. 597-605, September 1981.

[KaS85] J. N. Kappur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing,* Vol. 29, pp. 273-285, 1985.

[LaD89] S. Lakshamanan and H. Derin, "Simultaneous parallel estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-11, No. 8, pp. 799-813, August 1989.

[Lei60] R. Leipnik, "The extended entropy uncertainty principle," *Information and Control,* Vol. 3, pp. 18-25, 1960.

[Log77] B. F. Logan Jr., "Information in the zero crossings of bandpass signals," *Bell Systems Technical Journal,* Vol. 56, No. 4, pp. 487-510, 1977.

[LuB86a] W. H. Lunscher and M. Beddoes, "Optimal edge detector design I: Parameter selection and noise effects," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 2, pp. 164-177, March 1986.

[LuB86b] W. H. Lunscher and M. Beddoes, "Optimal edge detector design II: Coefficient quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 2, pp. 178-187, March 1986.

[LuF78] S. Y. Lu and K. S. Fu, "A syntactic approach to texture analysis," *Computer Graphics and Image Processing,* Vol. 7, pp. 303-330, 1978.

[Lun83] W. H. Lunscher, "The asymptotic optimal frequency domain filter for edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-5, No. 6, pp. 678-680, Nov. 1983.

[LyM88] E. Lyvers and O. R. Mitchell, "Precision edge contrast and orientation estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-10, No. 6, pp. 927-937, November 1988.

[MaH80] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society,* London, Vol. B 207, pp. 187-217, 1980.

[Mal82] N. A. Malik, *One and Two Dimensional Maximum Entropy Spectral Estimation,* Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, February 1982.

[MaM82] T. Matsuyama, S. Miura, and M. Nagao, "Structural analysis of natural texture by Fourier transformation," *Computer Graphics and Image Processing,* Vol. 19, pp. 347-362, 1982.

[Mar72] A. Martelli, "Edge detection using heuristic search methods," *Computer Graphics and Image Processing,* Vol. 1, pp. 169-182, 1972.

[Mar76] A. Martelli, "An application of heuristic search methods to edge and contour detection," *Communications of Association of Computing Machinery,* Vol. 19, No. 2, pp. 73-83, 1976.

[Mas72] J. Massey, "Variable-length codes and the Fano metric," *IEEE Transactions on Information Theory,* Vol. IT-18, pp. 196-198, January 1982.

[Men87] J. M. Mendel, *Lessons in Digital Estimation Theory,* Prentice Hall, New Jersey, 1987.

[MoB75] P. A. Moran and J. E. Besag, "On the estimation and testing of spatial interaction in Gaussian lattice," *Biometrika,* Vol. 62, pp. 555-562, 1975.

[MoF77] J. Modestino and R. Fries, "Edge detection in noisy images using recursive digital filtering," *Computer Graphics and Image Processing,* Vol. 6, pp. 409-433, 1977.

[Mor73] P. A. Moran, "A Gaussian Markovian process on a square lattice," *Journal of Applied Probability,* Vol. 10, pp. 54-62, 1973.

[NaB86] V. Nalwa and T. O. Binford, "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 6, pp. 699-714, November 1986.

[Nal87] V. Nalwa, "Edge-detector resolution improvement by image interpolation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-9, No. 3, pp. 446-451, May 1987.

[NaJ77] N. E. Nahi and M. H. Jahanshahi, "Image boundary estimation," *IEEE Transactions on Computers,* Vol. C-26, No. 8, pp. 772-781, August 1977.

[NaM78] N. E. Nahi and S. Lopez-Mora, "Estimation-detection of object boundaries in noisy images," *IEEE Transactions on Automatic Control,* Vol. AC-23, No. 5, pp. 834-845, October 1978.

[NeB80] R. Nevatia and K. Babu, "Linear feature extraction and description," *Computer Graphics and Image Processing,* Vol. 13, pp. 257-269, July 1980.

[Nev86] R. Nevatia, "Image Segmentation," *Handbook of Pattern Recognition and Image Processing,* ed. Young and Fu, Chapter 9, Academic Press, New York, 1986.

[Nil71] N. J. Nilsson, *Problems-Solving Methods in Artificial Intelligence,* McGraw-Hill, New York, 1971.

[Nil80] N. J. Nilsson, *Principals of Artificial Intelligence,* Tioga, Palo Alto, California, 1980.

[Ots79] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-9, No. 1, pp. 62-66, January 1979.

[PeM82] T. Peli and D. Malah, "A study of edge detection algorithms," *Computer Graphics and Image Processing,* Vol. 20, pp. 1-21, 1982.

[Pra78] W. Pratt, *Digital Image Processing,* J. Wiley, New York, pp. 530-542, 1978.

[Pre70] J. M. Prewitt, "Object enhancement and extraction," *Picture Processing and Psychopictorics,* ed. Lipkin and Rosenfeld, Academic Press, New York, 1970.

[Pun81] T. Pun, "Entropic thresholding, a new approach," *Computer Graphics and Image Processing,* Vol. 16, pp. 210-239, 1981.

[ReA83] A. Reeves, M. Akey, and O. R. Mitchell, "A moment-based two-dimensional edge operator," *Proceedings 1983 Computer Society Symposium on Computer Vision and Pattern Recognition,* pp. 150-155, June 1983.

[RiA77] E. M. Riseman and M. A. Arbib, "Computational techniques in the visual segmentation of static scenes," *Computer Graphics and Image Processing,* Vol. 6, pp. 221-276, 1977.

[RiC78] T. W. Riddler and S. Calvard, "Picture thresholding using an iterative selection method," *IEEE Transactions on Systems, Man, Cybernetics,* Vol. SMC-8, pp. 630-632, August 1978.

[Rob65] L. G. Roberts, "Machine perception of three dimensional objects," *Optical and Electro-Optical Information Processing,* ed. Tippet et al., MIT press, Cambridge, Massachusetts, pp. 159-197, 1965.

[RoK82] A. Rosenfeld and A. Kak, *Digital Picture Processing,* Vol. 1 & 2, Academic Press, New York, 1982.

[Ros77] A. Rosenfeld, "Iterative methods in image analysis," *Proceedings, Conference on Pattern Recognition and Image Processing,* Troy, New York, pp. 14-20, 1977.

[Ros89] A. Rosenfeld, "Image analysis and computer vision 1988," *Computer Vision, Graphics, and Image Processing,* Vol. 46, pp. 196-264, May 1989.

[RoT71] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers,* Vol. C-20, No. 5, pp. 562-569, 1971.

[Rud76] W. Rudin, *Principles of Mathematical Analysis,* 3rd ed., McGraw-Hill, New York, 1976.

[ShD79] K. S. Shanmugam, F. M. Dickey, and J. A. Green, "An optimal frequency domain filter for edge detection in digital picture," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-1, No. 1, pp. 37-49, January 1979.

[SiC88] T. Simchony and R. Chellappa, "Stochastic and deterministic algorithms for MAP texture segmentation," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* New York City, New York, pp. 1120-1123, April 1988.

[Sle65] D. Slepian, "Some asymptotic expansions for prolate spheroidal wavefunctions," *Journal of Mathematical Physics,* (MIT), Vol. 44, pp. 99-140, 1965.

[SlP61] D. Slepian and H. O. Pollak, "Prolate spheroidal waveforms, Fourier analysis and uncertainty - I," *Bell Systems Technical Journal,* Vol. 40, pp. 43-63, 1961.

[Spi71] F. Spitzer, "Markov random fields and Gibbs ensembles," *American Mathematical Monthly,* Vol. 78, pp. 142-154, 1971.

[Str68] W. Streifer, "Uniform asymptotic expansions for prolate spheroidal wavefunctions," *Journal of Mathematical Physics,* (MIT), Vol. 47, pp. 407-415, 1968.

[TaM78] H. Tamura, S. Mori, and T. Yamawaki, "Texture feature corresponding to visual perception," *IEEE Transactions on Systems, Man, Cybernetics,* Vol. SMC-8, pp. 460-473, June 1978.

[TiA77] A. N. Tikhonov and J. P. Arsenin, *Solutions of Ill-Posed Problems,* V. H. Winston & Sons, Washington, D. C., 1977.

[ToP86] V. Torre and T. A. Poggio, "On edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 2, pp. 147-163, March 1986.

[ToS82] T. Tomita, Y. Shirai, and S. Tsuji, "Description of textures by a structral analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-4, No. 2, pp. 183-191, March 1982.

[VaD85] L. Van Gool, P. Dewaele, and A. Oosterlink, "Texture analysis anno 1983, survey," *Computer Vision, Graphics, and Image Processing,* Vol. 29, pp. 336-357, 1985.

[Wes78] J. S. Weszka, "A survey of threshold selection techniques," *Computer Graphics and Image Processing,* Vol. 21, pp. 259-265, 1978.

[Whi54] R. Whittle, "On stationary processes in the plane," *Biometrika,* Vol. 41, pp. 434-449, 1954.

[Woo72] J. W. Woods, "Two-dimensional discrete Markovian random fields," *IEEE Transactions on Information Theory,* Vol. IT-18, pp. 232-240, March 1972.

[Woo81] J. W. Woods, "Two-dimensional Kalman filtering," *Two-Dimensional Digital Signal Processing,* Vol. 1, ed. T. S. Huang, Springer-Verlag, New York, 1981.

[ZhV89] Y. T. Zhou, V. Venkateswar, and R. Chellappa, "Edge detection and linear feature extraction using a 2-D random field model," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-11, No.1, pp. 84-95, January 1989.

[Zig66] K. Zigangirov, "Some sequential decoding procedures," *Problemy Peredachi Informatsii,* Vol. 2, pp.13-25, 1966.

[Zuc76a] S. W. Zucker, "Region growing: childhood and adolescence," *Computer Graphics and Image Processing,* Vol. 7, pp. 382-399, 1976.

[Zuc76b] S. W. Zucker, "Toward a model of texture," *Computer Graphics and Image Processing,* Vol. 5, No. 2, pp. 190-202, 1976.

APPENDIX

APPENDIX
TEXTURE SYNTHESIS

Monte Carlo methods [HaH64] can be used for generating sample functions of a random field once the probability distribution of the field is specified. It has been used by various researchers for texture synthesis (e.g. [HaS80], [CrJ83], and [DeE87]). In general, this method is computationally expensive. For finite random fields, we saw in Chapter 2 that a simpler description results from using the torus assumption ([Mor73], [KaC83]). With this assumption, a matrix formulations for the Markov random fields (MRF) and simultaneous autoregressive (SAR) models result; these formulations have a convenient structure for fast calculations by the FFT algorithm. A simulation technique based on the torus assumption was developed by Kashyap [Kas80] that has been used in various texture studies (e.g. [Che81], [KaC83], [Kho83], and [KaK86]). We will use the Kashyap algorithm for generating the synthetic textures used in Chapter 3. A description for this algorithm for the MRF and the SAR random fields now follows.

1. Kashyap Algorithm for MRF Texture Synthesis

Consider a square lattice $S = \{(i,j): 0 \leq i,j \leq M-1\}$, and a symmetric neighbor system $\eta \in S$. The random field **G** is described by $(S, F, P, \{G_s\})$. Consider also another stationary Gaussian random field $\mathbf{E} = \{E_s;\ s \in S\}$ with correlation structure as below (Equ. 2.34)

$$E\{E_sE_r\} = \begin{cases} \nu & \text{if } s = r \\ -\theta_{(s-r)}\nu & \text{if } (s-r) \in \eta \\ 0 & \text{otherwise.} \end{cases} \tag{A.1}$$

Using the torus assumption, we can write the finite lattice form MRF equation in the following form (Equ. 2.38):

$$G_s = \sum_{r \in \eta} \theta_r G_{(s \oplus r)} + E_s. \tag{A.2}$$

In matrix form, (A.2) can be written, using the lexicographic representations (Equ. 2.2), as follows:

$$H(\underset{\sim}{\theta})\underset{\sim}{g} = \underset{\sim}{e}, \qquad (A.3)$$

where $\underset{\sim}{g}$ and $\underset{\sim}{e}$ are $M^2 \times 1$ column vectors representing the gray level information at pixel sites and the the innovation (non white), respectively. The matrix $H(\underset{\sim}{\theta})$ is $M^2 \times M^2$ block circulant and symmetric. Equation (A.3) can be written, equivalently, as follows [Kas81]:

$$\sqrt{H}(\underset{\sim}{\theta})\underset{\sim}{g} = \sqrt{v}\,\underset{\sim}{\eta}, \qquad (A.4)$$

where $\underset{\sim}{\eta}$ is zero mean, unit variance, *iid* Gaussian random numbers. The matrix $\sqrt{H}(\underset{\sim}{\theta})$ is also block-circulant and positive definite; thus its inverse can be obtained by Fourier calculations. The vector $\underset{\sim}{g}$, can be written as below.

$$\underset{\sim}{g} = \frac{1}{M^2}\sum_{s}\frac{\underset{\sim}{y_s}\,x_s}{\sqrt{\mu_s(\underset{\sim}{\theta})}}, \qquad (A.5)$$

where

$$x_s = \sqrt{v}\,\sum \underset{\sim}{y_s}^{*t}\underset{\sim}{\eta}$$

$$\mu_s(\underset{\sim}{\theta}) = 1 - 2\underset{\sim}{\theta}^t\underset{\sim}{\phi}_s,$$

where the script t stands for transposition, and the vectors $\underset{\sim}{y_s}$ and $\underset{\sim}{\phi_s}$ are defined as follows for s = (i,j):

$$\underset{\sim}{y_s} = [1, \lambda_i\underset{\sim}{t_j}, \lambda_i^2\underset{\sim}{t_j} \cdots \lambda_i^{M-1}\underset{\sim}{t_j}]^t \quad M^2 \times 1 \text{ vector}$$

$$\underset{\sim}{t_j} = [1, \lambda_j, \lambda_j^2 \cdots \lambda_j^{M-1}]^t$$

$$\lambda_i = \exp(\sqrt{-1}\,\frac{2\pi}{M}(i-1))$$

$$\underset{\sim}{\phi_s} = [\cos\frac{2\pi}{M}(s-1)^t\mathbf{r}, \mathbf{r} \in \eta_s]^t,$$

where η_s is the asymmetric neighbor set (see Chapter 2.)

The Kashyap algorithm for MRF, as described above, can be implemented as follows:

(1) Generate M^2 random numbers for the vector $\underset{\sim}{\boldsymbol{\eta}}$. Arrange these numbers in an $M \times M$ matrix $\mathbf{x}$ and get its two-dimensional discrete Fourier transform (DFT). Call this $\mathbf{X}$.

(2) Scale each entry $X_s \in \mathbf{X}$, $s = (i,j)$ by the factor $\sqrt{\dfrac{\nu}{\mu_s(\underset{\sim}{\boldsymbol{\theta}})}}$.

(3) Obtain the inverse DFT for the results in step (2).

It should be pointed out that the above algorithm generates strictly stationary MRF configurations when the noise is Gaussian. For arbitrary distribution of the noise, the result is wide sense stationary Markov [Che81].

The convention used for the vector $\underset{\sim}{\boldsymbol{\theta}}$ for up to the third-order MRF (η is symmetric) is as follows:

$$\underset{\sim}{\boldsymbol{\theta}} = [\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_2 \mid \boldsymbol{\theta}_3], \tag{A.6}$$

where

$$\boldsymbol{\theta}_1 = [\theta_{(i,j-1)}, \theta_{(i+1,j)}, \theta_{(i,j-1)}, \theta_{(i,j+1)}\ ]^t \tag{A.7.a}$$

$$\boldsymbol{\theta}_2 = [\theta_{(i-1,j-1)}, \theta_{(i+1,j+1)}, \theta_{(i-1,j+1)}, \theta_{(i+1,j-1)}]^t \tag{A.7.b}$$

$$\boldsymbol{\theta}_3 = [\theta_{(i-2,j)}, \theta_{(i+2,j)}, \theta_{(i,j-2)}, \theta_{(i,j+2)}]^t. \tag{A.7.c}$$

In the above equations, $[\boldsymbol{\theta}_1]$ is the set of coefficients for the first-order MRF, $[\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_2]$ is the set of coefficients for the second-order MRF, and $[\boldsymbol{\theta}_1 \mid \boldsymbol{\theta}_2 \mid \boldsymbol{\theta}_3]$ are the third-order MRF coefficients.

2. Kashyap Algorithm for SAR Texture Synthesis

Consider, as before, a square lattice $S = \{(i,j): 0 \le i,j \le M-1\}$, a neighbor system $\eta \in S$. The random field $\mathbf{G}$ is described by $(S, F, P, \{G_s\})$. Consider another random field $\mathbf{W} = \{\mathbf{W}_s, s \in S\}$ in which the random variables $\mathbf{W}_s$ have a distribution with zero mean and unit variance. Recall that there is no restriction on the neighbor system η, that is, it need not be symmetric (Chapter 2.) The SAR model for finite lattice is described by a difference equation as follows (see 2.55):

$$G_s = \sum_{r \in \eta} \theta_r\, G_{(s+r)} \oplus \sqrt{\rho}\; W_s, \tag{A.8}$$

where η is an arbitrary neighbor set, $\{W_s\}$ is a sequence of zero mean uncorrelated random variables with unit variance, and $s = (i,j) \in S$ and $r = (k,l) \in \eta$ specify pixel locations on the image.

Now, using the toroidal assumption, we can write the finite lattice form corresponding to (A.8) as follows:

$$B(\underset{\sim}{\theta})\; \underset{\sim}{g} = \sqrt{\rho}\; \underset{\sim}{w}, \tag{A.9}$$

where the matrix $B(\theta)$ is block circulant involving at most M distinct blocks [Kas81]. Since the matrix $B(\underset{\sim}{\theta})$ is block circulant, its inverse can be obtained by decomposition of the eigen values evaluated by the FFT algorithm. Equation A.9 can be written as follows:

$$\underset{\sim}{g} = \frac{\sqrt{\rho}}{M^2} \sum_{S} \underset{\sim}{y}_s \underset{\sim}{y}_s^{*t}\; w\; / \mu_s(\underset{\sim}{\theta}), \tag{A.10}$$

where w is an $M \times M$ matrix of random numbers, $\underset{\sim}{y}$ is the Fourier vector defined in (A.5), and $\mu_s(\underset{\sim}{\theta})$ is defined as follows:

$$\mu_s(\underset{\sim}{\theta}) = \left(1 - \underset{\sim}{\theta}^t \underset{\sim}{\psi}_s\right)$$

$$\underset{\sim}{\psi}_s = \left[\exp\left(\sqrt{-1}\,\frac{2\pi}{M}(s-1)^t r\right),\; r \in \eta\,\right]^t.$$

The term $\left((s-1)^t r\right)$ in the above equation and in the definition of the vector $\underset{\sim}{\psi}_s$ is calculated as follows: let $s = (i,j)$ and $r = (k,l)$. Then this term will be equal to $\left((i-1) \times k + (j-1) \times l\right)$

A configuration from an SAR random field can be generated by the Kashyap algorithm as follows:

(1) Generate M^2 random numbers for the vector $\underset{\sim}{w}$. Arrange these numbers in an $M \times M$ matrix w, and obtain its two-dimensional discrete Fourier transform (DFT). Call this W.

(2) Scale each entry $W_s \in X$, $s = (i,j)$ by the factor $\sqrt{\dfrac{\nu}{\mu_s(\underset{\sim}{\theta})}}$.

(3) Obtain the inverse DFT for the results in step (2).

We use the same convention for η as described above in the MRF. If η is not symmetric, we put the zeros for the missing elements in the vector $\underset{\sim}{\theta}$.

VITA