PROBABILISTIC SOLUTION OF INVERSE PROBLEMS

by

Jose Luis Marroquin

Submitted to the Department of Electrical Engineering
and Computer Science on September 6, 1985 in partial fulfillment
of the requirements for the degree of Doctor of Philosophy

ABSTRACT

In this thesis we study the general problem of reconstructing a function, defined on a finite lattice, from a set of incomplete, noisy and/or ambiguous observations. The goal of this work is to demonstrate the generality and practical value of a probabilistic (in particular, Bayesian) approach to this problem, particularly in the context of Computer Vision. In this approach, the prior knowledge about the solution is expressed in the form of a Gibbsian probability distribution on the space of all possible functions, so that the reconstruction task is formulated as an estimation problem. Our main contributions are the following:

1. We introduce the use of specific error criteria for the design of the optimal Bayesian estimators for several classes of problems, and propose a general (Monte Carlo) procedure for approximating them. This new approach leads to a substantial improvement over the existing schemes, both regarding the quality of the results (particularly for low signal to noise ratios) and the computational efficiency.

2. We apply the Bayesian approach to the solution of several problems, some of which are formulated and solved in these terms for the first time. Specifically, these applications are: the reconstruction of piecewise constant functions from noisy data; the reconstruction of piecewise continuous surfaces from sparse and noisy observations; the reconstruction of depth from stereoscopic pairs of images and the formation of perceptual clusters.

3. For each one of these applications, we develop fast, deterministic algorithms that approximate the optimal estimators, and illustrate their performance on both synthetic and real data.

4. We propose a new method, based on the analysis of the residual process, for estimating the parameters of the probabilistic models directly from the noisy observations. This scheme leads to an algorithm, which has no free parameters, for the restoration of piecewise uniform images.

5. We analyze the implementation of the algorithms that we develop in non-conventional hardware, such as massively parallel digital machines, and analog and hybrid networks.

Thesis Supervisors:

Dr. Sanjoy K. Mitter
Professor of Electrical Engineering

Dr. Tomaso Poggio
Professor of Psychology

PROBABILISTIC SOLUTION OF INVERSE PROBLEMS

by

Jose Luis Marroquin

B.S., Universidad Nacional Autonoma de Mexico
(1968)

M.S., Massachusetts Institute of Technology
(1976)

Submitted to the
Department of Electrical Engineering
and Computer Science
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September, 1985

Signature of Author_____
Department of Electrical Engineering and Computer Science
September 6, 1985

Certified by____
Professor Sanjoy K. Mitter
Thesis Supervisor

Certified by____
Professor Tomaso Poggio
Thesis Supervisor

Accepted by_____
Professor Arthur C. Smith
Chairman, Departamental Committee on Graduate Students

ACKNOWLEDGEMENTS

I wish to express my deep gratitude to Professors Sanjoy Mitter, Tomaso Poggio and Whitman Richards. From all of them, I have received encouragement and support, excellent guidance and true friendship.

Thanks also go to Dr. Manuel Cerrillo, who provided the impulse that brought me here; to Dr. Stuart Geman, whose work has been for me a source of inspiration; to Drs. Christof Koch and Alan Yuille, for many useful and stimulating conversations; to Dr. Eric Grimson, who kindly provided the data for some of the experiments that are presented here; to my academic advisors, Professors William Schreiber and Alan Willsky, and to my thesis reader, Professor Dimitri Bertsekas.

I gratefully acknowledge the partial support that the Mexican Petroleum Company (PEMEX), the National University of Mexico (UNAM) and the Massachusetts Institute of Technology provided during the course of my graduate studies. The research reported in this thesis was supported in part by the U.S. Army Research Office under grant DAAG29-84-K-0005 and by the Air Force Office of Sponsored Research under grant AFOSR 82-0135B.

Finally, and most importantly, my special gratitude to my dear wife, Rosa Maria. Without her love, friendship and unfailing support, this thesis would never have been completed.

To Rosa Maria and Daniel

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

A fundamental problem in the design and analysis of systems endowed with perceptual abilities is the construction of internal representations of the physical structures in the external world. The precise form of these representations is not well understood, and is the subject of much current research in Artificial Intelligence and Psychology. It is clear, however, that these representations should integrate prior generic knowledge about the physical properties of the external world with measurements from a number of different sensory modalities. Furthermore, in order to be effectively action–oriented, the representations should provide compact descriptions of the physical structures of interest at different levels of detail.

This problem is not exclusive of biological perceptual systems; it arises whenever information from a set of sensors has to be processed, stored and retrieved in an efficient way. Thus, it is of fundamental importance, for example, in the design of computer vision systems; in the reconstruction of subterranean geological structures from geophysical data and in the design of biomedical imaging systems. The motivation for this thesis is to increase our understanding of the principles underlying the process of integrating prior generic constraints with the available observations, for the construction of these representations. In particular, we will address the problem of reconstructing, in a way that is consistent with the available sensory data, the value of certain properties of the physical structure of interest over a discretized region of space.

To define these early perceptual processes in a more precise way, let us model the specific properties of the physical structure as functions f that map a (compact) region $\Omega \subseteq \mathcal{R}^n$ into $\mathcal{R}^m$. In the most interesting cases, f will be either a scalar ($m = 1$) or a vector field ($m = 2$) defined on a two–dimensional region. This is

the case, for example, of the problems of image restoration and segmentation, and of the recovery of: depth from stereo; lightness; shape from shading; and the computation of optical flow in computer vision, as well as many problems in the recovery of geological structure from geophysical measurements.

We will assume that the available data consists of several sets of qualitatively different measurements $\{g_1, \ldots, g_M\}$ that in general are modeled as:

$$g_i = H_i(f, Df, D^2f, \ldots, n_i)$$

where Df denotes the derivative of the property f; n_i is a noise process, and H_i is some operator (for example, in vision problems, the different measurements may correspond to: stereo disparity; brightness; color, etc.). We will also assume that this information is collected with different sampling patterns $\{S_1, \ldots, S_M\}$, that is, the observations g_i are defined only on the finite set $S_i \subseteq \Omega$. Since most physical phenomena consist of events that occur at a variety of scales, and in general, events at widely different scales have little influence on one another, the numerical descriptions of the behavior of a property over a range of scales can be used effectively to produce a physically meaningful hierarchical decomposition of the original structure into individual substructures ("objects") which can be subsequently described in symbolic forms that are more compact and easy to manipulate (see Marr, 1976 and 1982; it is not surprising that there is psychophysical evidence suggesting the presence of a multiscale processing hierarchy in the human visual system; see Campbell and Robson, 1977, and Marroquin, 1976).

Thus, the solutions we are looking for consist on a family $\{\hat{f}_\alpha\}$ of numerical descriptions of the function f at different scales (indexed by α) at the sites of some lattice $L \subseteq \Omega$ (the finest scale representation should correspond to the best estimate of the actual value of f at the sites of L). To illustrate this idea, in figure 1-a we present a binary pattern, and in figures 1-b through 1-e, its numerical representation at increasingly coarser scales. This family of descriptions was generated by the algorithm described in section 5 of chapter 4.

In general, the observation processes g_i do not determine the value of f in a unique and stable way (that is to say, these problems are ill–posed in the sense of

Figure 1. Representation of the binary pattern (a) at increasingly coarser scales.

Hadamard; see Poggio and Torre, 1984). Therefore, the algorithms we are looking for should be able to regularize the problem by incorporating constraints on the solution generated by some prior knowledge about its general characteristics.

Finally, because of the large number of variables involved, reasonable speed of performance will usually require that these algorithms be distributed, and thus, efficiently implementable in parallel hardware.

1. Regularization Analysis and Cooperative Algorithms.

Among the most successful solutions to these type of problems are those that formulate them as variational problems, where the measurement and generic constraints are separately represented in the following way:

Let us consider the case of only one set of "perfect" measurements (i.e., with no noise) g defined on the set S, and suppose that the constraints that they impose on the solution can be expressed in the form:

$$\int_S A(f, g) = 0$$

where A is a positive definite, real valued function that measures the incompatibility of the value of the property f with the observations g. In general, the observations will not be perfect, and so, we will only require that the error $\int_S A(f,g)$ be small. However, there may be a large number of configurations that minimize the error. To find a unique solution, an assumption about the *global* smoothness of f is introduced by means of some positive definite, real valued function $P(f, Df, \ldots)$ which measures the "jaggedness" of f. If both A and P are convex, the desired solution will be the unique minimizer of the "energy" functional:

$$U(f,g) = \int_S A(f,g) + \lambda \int_\Omega P(f, Df, \ldots) \qquad (1)$$

where λ is a parameter.

This approach has been applied with varying degrees of success to the problems of surface interpolation (Grimson, 1981b, 1982; Terzoupulos, 1983, 1984a); computation of visual motion (Horn and Schunk, 1981; Hildreth, 1984a,b); recovery of shape from shading information (Ikeuchi and Horn, 1981); computation of subjective contours (Ullman, 1976; Brady et al., 1980; Horn, 1981); lightness (Horn, 1974), and edge detection (Torre and Poggio, 1983).

In a recent paper, Poggio and Torre (1984) have shown how functionals of the form of equation (1) can be derived in a rigorous and systematic way using regularization methods (Tikhonov, 1963; Tikhonov and Arsenin (1977); Wahba (1980); in this context $\int_\Omega P$ is called a stabilizing functional, and λ, the regularization parameter).

Once the functional (1) is specified, its minimization can be carried out by standard variational methods (Courant and Hilbert, 1953). Since usually one is interested in the value of f only at the discrete set of points L, the solution of the resulting Euler–Lagrange partial differential equations can be obtained as the fixed point of a relaxation (cooperative) algorithm of the form:

$$f_i^{(k+1)} = F_i(f^{(k)}) \qquad i \in L \qquad (2)$$

This algorithm can be efficiently implemented in parallel hardware using a network of locally connected processors (one for each site i), or even by some analog network (see Poggio and Koch, 1984).

It is interesting to note that it is also possible, and sometimes easier, to embed the prior knowledge about the solution, and the constraints imposed by the observations, directly in a cooperative network of a given form, without explicitly defining a global variational principle. This approach has been used by Marr and Poggio (1976) for the stereo matching problem. We will have more to say about it in chapter 6.

It is also possible, in principle, to incorporate qualitatively different measurements into a single cooperative process, by a simple modification of the energy functional:

Suppose that we have M sets of measurements, and that each set g_i places some constraints on f (and/or its derivatives) which can be expressed by the functionals:

$$\int_{S_i} A_i(g_i, f, Df, \ldots) = 0 \qquad i = 1, \ldots, M$$

The solution will now be constructed as the global minimizer of the functional:

$$U(f) = \sum_{i=1}^{M} a_i(f, g) \int_{S_i} A_i + \lambda \int_{\Omega} P(f, Df, \ldots) \tag{3}$$

where the parameters a_i measure the relative weight we wish to assign to each set of measurements.

If all the functions A_i are convex, the solution will again be unique, and the minimization of (3) may be carried out by means of a cooperative network (this approach has been used by Terzopoulos (1985) for the surface interpolation problem, when the depth value f is known at some set S_1 of sites, and the slope (Df) at a different set S_2).

The approach we have been discussing — which we will call the "standard regularization method" is very attractive: it provides a unified framework for the formulation of a variety of problems, and it leads to computationally efficient algorithms. However, it has some important limitations (some of them pointed out by Poggio and Torre):

(i) Very often the assumption that the solution f is smooth over the whole domain Ω is not justified. What is more commonly true is that Ω can be

partitioned into a small set of disjoint connected regions, and that while f is smooth in the interior of each of them, it has discontinuities along the boundaries between regions (which in turn are piecewise smooth curves). This limitation is a serious one, because very often the discontinuities of f, which the regularization methods tend to hide, are the most important parts of the surface, in particular if one is trying to compute a symbolic representation for it.

(ii) The meaning of the parameters of the energy functional is not always clear, and they often have to be selected on a purely empirical basis.

(iii) In many cases, the choice of the particular (often quadratic) form for the functions A and P is arbitrary, and is determined mainly by the tractability of the uniqueness problem for the solution, and by the simplicity of the (linear) minimization algorithm (in some cases, of course, there may be other theoretical or experimental considerations that justify this choice).

(iv) The interaction between qualitatively different observations is purely additive. One would like to be able to include more realistic non–linear modes of interaction.

2. Probabilistic Formulation.

A different approach is to model the function f, whose reconstruction solves a perceptual problem, as a random field that has to be estimated from a set of noisy, and possibly ambiguous measurements. Within this formulation, one can adopt a Bayesian viewpoint (see Good, 1983), and assume that the best way of expressing the prior knowledge about the nature of the solutionis in the form of a (prior) probability distribution P_f. This distribution, together with a probabilistic description of the noise that corrupts the observations, allows one to use Bayes theory to compute the posterior distribution $P_{f|g}$, which represents the likelihod of a solution f given the observations g. In this way, we can solve the reconstruction problem by finding the estimate $\hat{f}$ which either maximizes this likelihood (the so called Maximum a Posteriori or MAP estimate), or minimizes the expected value (with respect to $P_{f|g}$) of an appropriate error function. This formulation has several advantages over the "Standard Regularization" approach:

1. Flexibility.

With simple modifications in the prior probabilistic model for f, one can generate algorithms that reconstruct not only smooth, but piecewise constant or piecewise continuous functions. It is also possible to include explicitly into the model prior knowledge about the geometry of the curves that bound the smooth patches (i.e., about the discontinuities) of f.

2. Generality.

This approach provides a general framework for the formulation of a wide variety of perceptual problems. We will show, for instance, how it can be used for: image segmentation; surface reconstruction from sparse data; modeling of perceptual grouping processes; stereo matching, etc. Furthermore, the incorporation of qualitatively different measurements into a single cooperative estimation process can be made in a natural way: if the noise processes $n_1, n_2, \ldots, n_M$ associated with the sets of measurements $g_1, \ldots g_M$ are independent, the joint posterior distribution $P(f \mid g_1, \ldots g_M)$ will be simply:

$$P(f \mid g_1, \ldots g_M) = \frac{P_f(f) \prod_{i=1}^{M} P(g_i \mid f)}{\prod_{i=1}^{M} P(g_i)}$$

3. Precise Interpretation.

The parameters that appear in the reconstruction algorithms that are derived using this approach have a precise statistical interpretation (for example, the relative weight of the evidence provided by each set of observations, will be determined by the variance of the associated noise process); also, the plausibility of the prior assumptions about the behavior of the solution can be explicitly verified by generating sample functions of the random field defined by P_f, by means of an appropriate Monte Carlo procedure. Finally, one can choose the precise loss function whose expected value will be minimized by the Bayesian estimator.

3. Computational Efficiency.

As we will see, if the random field defined by P_f is Markovian (i.e., if the probabilistic dependencies are local), the estimation algorithms will be distributed, so that it will be possible to implement them efficiently in parallel hardware.

3. Goals of this Thesis.

The objective of this work is to apply the probabilistic approach we have just described to the solution of a general class of perceptual problems. In particular, we will:

1. Present a class of random fields with local probabilistic dependencies, that can be used very effectively to model the behavior of a wide variety of functions.

2. Develop appropriate loss functions, and the corresponding optimal estimators for different classes of problems.

3. Develop general distributed algorithms for computing these estimates.

4. Apply the above results to several specific problems, to illustrate the generality and practical value of this approach.

5. Develop more efficient algorithms for each of these particular cases.

We now present a list of our main contributions:

3.1. Summary of our Main Contributions.

1. Optimal Bayesian Estimators.

Several researchers have used Bayes theory and Markov random field (MRF) models for the restoration of piecewise uniform images. It has been implicitly assumed by most of them that the maximization of the posterior probability (which leads to the Maximum a Posteriori or MAP estimator) is the best possible performance criterion. We introduce the use of different specific error criteria for the design of the optimal Bayesian estimators for several classes of problems, and propose a general procedure (which is based on some existing Monte Carlo techniques, such as the Metropolis algorithm) for approximating them. We show, both theoretically and experimentally (in particular for the case of the restoration of piecewise uniform images) that this new approach leads to a substantial improvement

over the existing methods, both regarding the quality of the results (particularly for low signal to noise ratios) and the computational efficiency.

2. Novel Applications.

Throughout this thesis we present several examples of the application of the probabilistic approach, and of the optimal estimation procedures that we have derived, to several problems, some of which are formulated and solved in these terms for the first time. The results that we get show that this approach can provide a unified framework for the integration of a variety of related perceptual tasks into a single cooperative process. Also, these results represent, in several cases, a significant improvement over those obtained using existing schemes. Specifically, these new applications are the following:

a) Surface Interpolation.

The problem of reconstructing a piecewise continuous surface from sparse and noisy data is formulated using a Bayesian approach, using two coupled MRF's to model the behavior of the smooth patches, and of the curves (discontinuities) that bound them. Although this type of coupled model has been used before (in the context of the restoration of piecewise uniform, noisy images), its adaptation to this problem requires some non-trivial modifications: the local interactions between the elements of the fields have to be redefined in an appropriate way, and the general estimation algorithm has to be modified to make it computationally feasible. The practical value of the resulting algorithm is illustrated using both synthetic and real data.

b) Signal Matching.

This problem consists in finding the corresponding points in two signals when one is obtained from the other by shifting it by a variable amount. We study in detail a specific instance: the reconstruction of depth from a stereoscopic pair of images, and show how to formulate it using our general framework. The performance of the algorithms that we construct is also illustrated by means of synthetic and real examples.

c) Formation of Perceptual Clusters.

We suggest that the process of formation of perceptual clusters of certain dot patterns can be modeled in terms of the estimation of binary images corrupted by multiplicative noise, and illustrate the application of our estimation algorithms to this task.

3. Efficient Algorithms.

Although the Monte Carlo procedure that we have developed for approximating the optimal estimates is perfectly general, for each particular application it is often possible to design alternative (some times deterministic) algorithms that improve significantly the computational efficiency. It has been our concern in this work to develop such alternative fast algorithms for each one of the applications that we present. Specifically, we have developed the following algorithms:

a) Estimation of One–Dimensional Signals.

We present a new deterministic algorithm of minimal complexity which computes (exactly) the MAP estimate of binary, one–dimensional MRF's, and a rigorous proof of its optimal performance. We also develop an alternative scheme for the same purpose, based on dynamic programming principles, which can be extended to handle more general situations (such as the MAP estimation of piecewise constant one–dimensional signals).

b) Estimation of Two–Dimensional, Binary MRF's.

We heuristically motivate and develop a new deterministic algorithm for approximating the optimal Bayesian estimator of two–dimensional MRF's. We find, experimentally, that the quality of the results produced by this scheme is equivalent to those obtained by the general Monte Carlo procedure, and the computational efficiency (execution time) is improved at least by an order of magnitude.

For the case of the MAP estimation of binary patterns, we develop a modification to the "Simulated Annealing" procedure, which improves its computational efficiency. It is based on the computation of "coarse solutions" (formed by aggregating the elements of the field into blocks) which are then progressively refined.

c) Reconstruction of Piecewise Continuous Surfaces.

In this case, we also develop a heuristic, deterministic scheme whose experimental performance is practically equivalent to that of the Monte Carlo procedure, and improves significantly on its computational efficiency.

d) Stereo Matching.

We propose a new algorithm for solving the stereo matching problem in some simple cases. This scheme is based on the direct implementation of the local constraints (generated by the probabilistic model) in a highly distributed cooperative network of a particular form: a "Winner–Take–All" network. We show rigorously that, for noise–free observations, the state of this network will converge to the correct solution, and estimate the maximum number of required iterations (which is usually very small). The application of this technique to the reconstruction of the depth of real objects from stereoscopic photographs is discussed, and some modifications to the algorithm are introduced, which permit us to produce results whose quality is comparable to those of other "state of the art" algorithms.

4. Parameter Estimation.

In the context of the estimation of two–dimensional, binary fields, we study the case where the parameters that characterize the field model and the noise are not known, and have to be estimated from the noisy observations, a situation that, so far, has never been treated. We present a maximum likelihood procedure, which based on an analysis of the residual ("innovations") process, permits the simultaneous estimation of the field and the parameters of the system. We apply this technique to the construction of an algorithm, which does not have any free parameters, for the reconstruction of piecewise uniform images, and perform experiments to demonstrate its performance.

5. Parallel Implementations.

An important issue regarding the practical value of the algorithms that we develop is their possible implementation in certain non–conventional hardware,

such as massively parallel digital machines; hybrid and analog computers, etc. In this connection, we make the following contributions:

a) Monte Carlo Procedures.

We analyze the parallel implementation of the general Monte Carlo procedure for approximating the optimal Bayesian estimators. We show that the convergence of certain widely used algorithms (such as the Metropolis and Heat Bath schemes) cannot be guaranteed in this case. We justify the selection of an appropriate algorithm (the "Gibbs Sampler"), and present an estimate of its computational complexity.

b) Reconstruction of Piecewise Continuous Surfaces.

The parallel implementation of both the modified Monte Carlo procedure and the deterministic algorithm that solve this problem are analyzed, and their computational complexity is estimated. We also propose schemes for the construction of hybrid (digital/analog) and analog networks that implement these procedures, and perform digital simulations to evaluate experimentally their performance.

c) Estimation of Two–Dimensional Binary Fields.

The computational complexity of the parallel implementation of the fast deterministic algorithm that performs this task, is estimated and compared with that of the general Monte Carlo scheme.

We also propose the adaptation of a class of analog networks proposed by Hopfield and Tank (1985), so that we can obtain an approximation to the optimal estimate of the field from the equilibrium state of this system. The performance of this scheme is assessed experimentally by means of numerical simulations.

3.2. Thesis Overview.

This thesis is organized in the following way:

In chapter two we will introduce the basic concept of a Markov random field; show how to compute the corresponding probability distribution, and present Monte Carlo procedures for generating sample functions. In chapter three, we develop loss functionals for the image segmentation and surface reconstruction problems, and derive the corresponding optimal Bayesian estimators. We also present general algorithms for computing these estimates, and discuss their implementation in parallel hardware.

These results are applied, in chapter four, to the problem of segmenting piecewise constant images given noisy observations. For the particular case of binary images, a very efficient distributed algorithm is developed, and we present a procedure for the case when the model and the noise parameters are not known, and have to be estimated from the noisy data. Also in this chapter, we show how these principles can be applied to the problem of computing the perceptual clusters that are formed in some dot patterns.

In chapter five, we treat the problem of reconstructing piecewise smooth surfaces from sparse and noisy data, without blurring the boundaries between continuous regions; we discuss the use of Markov random field models to embody the prior knowledge about the shape and location of the discontinuities, and show how to adapt the general reconstruction algorithms developed in chapter three to this problem. We also develop a special purpose efficient algorithm for this case, and discuss its parallel implementation.

Chapter six is devoted to the problem of the reconstruction of depth from stereoscopic images. As in the previous cases, we first present a probabilistic formulation of the problem, and extend the general methods of chapter three for implementing a solution. Then, we develop special purpose algorithms that improve the computational efficiency. The performance of these algorithms is illustrated using both synthetic and "real" images.

Finally, in chapter seven, we summarize our results, and suggest areas where future research may be fruitful.

Chapter 2

LOCAL SPATIAL INTERACTION MODELS

1. Introduction.

The key to the success in the use of the probabilistic (and in particular, Bayesian) approach for the solution of the class of reconstruction problems in which we are interested, is our ability to find a class of stochastic models (that is, random fields) that have the following characteristics:

(i) The probabilistic dependencies between the elements of the field should be spatially localized. This condition is necessary if the field is to be used to model surfaces that are only piecewise smooth; besides, if it is satisfied, the reconstruction algorithms will be distributed, and thus, efficiently implementable in parallel hardware.

(ii) The class should be rich enough, so that a wide variety of qualitatively different behaviors of the desired solutions can be modeled.

(iii) The relation between the parameters of the models and the characteristics of the corresponding sample fields should be relatively transparent, so that the models are easy to specify.

(iv) It should be possible to represent the prior probability distribution P_f explicitly, so that Bayes theory can be applied.

(v) It should be possible to specify an efficient Monte Carlo procedure for generating sample fields from the distribution, so that the ability of the model to represent our prior knowledge can be verified.

Fortunately, there is a class of models that satisfies these characteristics: the class of Markovian Random Fields (MRF) on lattices. We will describe them in this chapter, and we will also show how they satisfy the required conditions. To do this, we will need two important results: the Hammersley–Clifford theorem, which is related to conditions (iii) and (iv), and the Metropolis and Gibbs–sampler algorithms, which will permit us to satisfy condition (v).

2. Markov Random Fields.

The concept of a MRF is a direct extension of the concept of a Markov process to higher dimensions and originated in the work of Ising (1925) on the construction of models for ferromagnetic phenomena. The definition for a two dimensional continuous MRF was introduced by Wong (1968), following Levy (1956) (see also Dobrushin, 1968), and in intuitive terms it says that a random field is Markovian if for any closed curve that separates the space into two regions, the knowledge of the value of the field along the curve, makes the field in these regions mutually independent.

More useful for our purposes (since usually we will be interested only in reconstructing the field at the sites of a regular lattice) is the definition of a discrete MRF, a generalization of the concept of a Markov chain. A discrete Markov random field on a finite lattice is defined as a collection of random variables, which correspond to the sites of the lattice, whose probability distribution is such that the conditional probability of a given variable having a particular value, given the values of the rest of the variables, is identical to the conditional probability given the values of the field in a small set of sites, which we will call the *neighborhood* of the given site. In formal terms we have the following (see Geman and Geman, 1983, and also Woods, 1972 for an alternative definition):

Let S be a finite set of N sites, and $G = \{G_s, s \in S\}$ be a neighborhood system for S, i.e., a collection of subsets of S for which:

(i) $s \notin G_s$ for all $s \in S$.

(ii) $s \in G_r$ if and only if $r \in G_s$, for all $r, s \in S$.

Let $F = \{F_s, s \in S\}$ be any family of random variables indexed by $s \in S$, and suppose, for simplicity, that these variables take values on some finite sets $\{Q_s\}$ (the definition can be extended, with some technical modifications, to the case of continuous state space). We will call any possible sample realization f :

$$(f_{s_1}, \ldots, f_{s_N}) \quad , \quad f_{s_i} \in Q_{s_i}$$

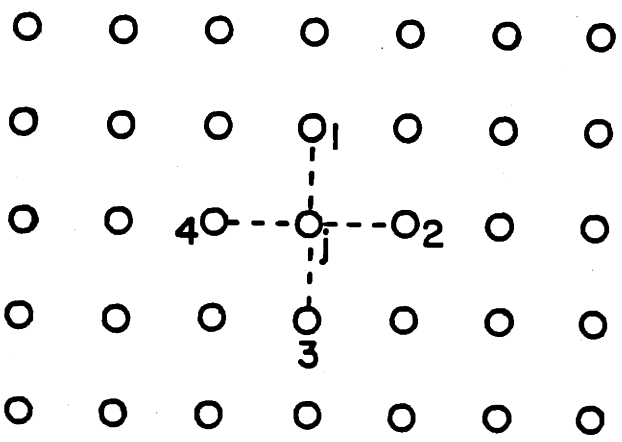


Figure 2. Sites 1, 2, 3 and 4 are the neighborhood of site j

a *configuration* of the field. Let Ω be the set of all possible configurations (i.e., the sample space), and let P be a probability measure in Ω. F is a MRF with respect to G if:

(i) $P(F = f) > 0$, for all $f \in \Omega$ ($(F = f)$ denotes the event: $(F_s = f_s$ for all $s \in S)$).

(ii) $P(F_s = f_s \mid F_r = f_r \quad r \neq s) = P(F_s = f_s \mid F_r = f_r \quad r \in G_s)$.

for every $s \in S$.

It is clear from this definition , that if the size of the neighborhoods is small, a MRF will satisfy the first condition we required from our class of models. The direct specification of a MRF from this definition (i.e., in terms of the conditional probabilities), however, is not very convenient because of the following reasons:

Firstly, the functions defining valid conditional distributions for a MRF cannot be chosen arbitrarily, since they have to satisfy a set of consistency conditions (that

result from Bayes' rule; see Besag, 1972), and are, in general, very difficult to specify directly. Secondly, although the joint probability distribution P_f can be uniquely determined from the conditional probabilities, its computation is, in general, a highly non–trivial task. Finally, there is no obvious intuitive relation between the form of the conditional probability distributions and the qualitative behavior of the sample fields.

To overcome these difficulties, we need an alternative way of defining a MRF. This is done as follows.

2.1. Markov–Gibbs Equivalence.

First, we need the following definition:

Given a system of neighborhoods on a lattice, we define a "clique" C as either a single site, or a set of sites of the lattice, such that all the sites that belong to C are neighbours of each other. For example, on a 4–connected lattice (Fig. 2), the sites 1, 2, 3 and 4 form the neighborhood of site j, and the cliques are sets consisting either of single sites, or of two (vertically or horizontally) adjacent sites (nearest neighbours; see Fig. 3).

The result we are looking for is contained in the Hammersley–Clifford theorem (Hammersley and Clifford, 1971) which states that if F is a MRF on a lattice S with respect to the neighborhood system G, the probability distribution of the configurations (sample functions) generated by it will always have a definite form, which is that of a Gibbs distribution:

$$P_f(f) = \frac{1}{Z} e^{-\frac{1}{\beta}U(f)}$$

where Z is a normalizing constant, β is a parameter, and the "Energy function" $U(f)$ is of the form:

$$U(f) = \sum_C V_C(f)$$

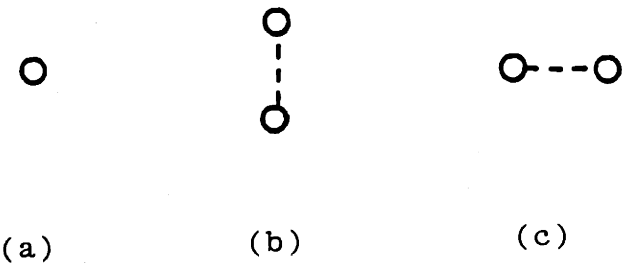


Figure 3. Cliques for the 4-connected lattice of Fig. 2.

where C ranges over the cliques associated with the given neighborhood system, and the potentials $V_C(f)$ are functions supported on them. Thus, in our example of a 4-connected lattice, U would be of the form:

$$U(f) = \sum_i V_a(f_i) + \sum_{i,j \in N_H} V_b(f_i, f_j) + \sum_{i,j \in N_V} V_c(f_i, f_j)$$

where N_H and N_V denote the sets of all horizontal and vertical nearest neighbor pairs of sites of the lattice (figure 3 (b) and (c)), respectively, and V_a, V_b and V_c are some functions.

A simple proof of this important result can be found in Besag (1972). We present here a brief sketch:

Without loss of generality, we may assume that 0 (the configuration with $f_i = 0$ for all i) belongs to Ω (otherwise, we simply perform a translation of the origin).

Since F is a MRF, we have that

$$P(0) > 0$$

so that the quantity

$$\frac{P(f)}{P(\mathbf{0})}$$

is well defined.

The key step is to note that we can always write:

$$\frac{P(f)}{P(\mathbf{0})} = e^{Q(f)}$$

with

$$Q(f) = \sum_i f_i G_i(f_i) + \sum_i \sum_j f_i f_j G_{ij}(f_i, f_j) + \ldots$$
$$+ f_1 \ldots f_n G_{ij\ldots n}(f_1, \ldots f_n)$$

for some functions $G_i, G_{ij}, \ldots$.

Now, for any configuration f and any selected site i, we define the configuration $f^{(i)}$ as being equal to f everywhere, except possibly at site i, where it is equal to $\mathbf{0}$:

$$f^{(i)} = \{f_1, \ldots f_{i-1}, \mathbf{0}, f_{i+1}, \ldots, f_n\}$$

Using Bayes rule we find that:

$$\frac{P(f)}{P(f^{(i)})} = \frac{P(f_i \mid f_j, j \neq i) \cdot P(f_j, j \neq i)}{P(\mathbf{0} \mid f_j, j \neq i) \cdot P(f_j, j \neq i)} =$$

$$= \frac{P(f_i \mid f_j, j \neq i)}{P(\mathbf{0} \mid f_j, j \neq i)} = \exp[Q(f) - Q(f^{(i)})] =$$

$$= \exp[f_i G_i(f_i) + \sum_j f_i f_j G_{ij}(f_i, f_j) + \ldots]$$

Note that because of the Markov property, the above quotient of conditional probabilities can depend only on the value of f at those sites which are neighbors of site i.

Now, suppose l is not a neighbor of i, and consider a particular configuration f which is equal to $\mathbf{0}$ everywhere, except at sites i and l. By the above considerations, we have that:

$$Q(f) - Q(f^{(i)}) = f_i G_i(f_i) + f_i f_l G_{il}(f_i, f_l)$$

depends only on f_i, which means that $G_{il}(f_i, f_l) = 0$.

By a similar reasoning, one can show that $G_{i,j,\ldots,m}(f_i, \ldots, f_m)$ can be different from 0 only if the sites $i, j, \ldots, m$ are neighbors of each other, i.e., if they belong to the same clique. The proof is completed by defining:

$$-\frac{1}{\beta}V(f_i, \ldots, f_m) = f_i, \ldots f_m G_{i,\ldots m}(f_i, \ldots f_m)$$

It is important to note that whereas the functions defining valid conditional probabilities for a MRF cannot be chosen arbitrarily, the form of the potentials V_C is not restricted in any way, and can be used freely to specify the required behaviour of the field f (which is what one does in practice). The relation between these potentials and the conditional probabilities is given by the following formula (which follows from Bayes rule):

$$P(F_i = f_i \mid F_j = f_j, j \neq i) = \frac{\exp[-\frac{1}{\beta}\sum_{C:i\in C} V_C(f)]}{\sum_{q\in Q_i}\exp[-\frac{1}{\beta}\sum_{C:i\in C} V_C(f^q)]} \qquad (1)$$

where Q_i is the set of allowable values for the state of F_i, and f^q is the configuration which is equal to q at site i, and coincides with f everywhere else.

There are other ways of representing certain classes of MRF's. For example, Woods (1972) has shown that every homogeneous Gaussian MRF defined on a finite lattice satisfies a difference equation of the form:

$$f_{nm} = \sum_{D(P)} h_{kl} f_{n-k,m-l} + u_{nm}$$

where f_{nm} is the value of the field at site nm and u is a (non–white) stationary Gaussian field whose autocorrelation function satisfies:

$$E[u_{nm}u_{00}] = \begin{cases} c, & m = n = 0 \\ -h_{mn}c, & (m,n) \in D(P) \\ 0, & \text{elsewhere} \end{cases}$$

where

$$D(P) = \{(k,l) \quad : \quad 0 < k^2 + l^2 \leq P^2\}$$

and also

$$E[f_{nm}u_{kl}] = \begin{cases} c, & \text{if } n = k \text{ and } m = l \\ 0, & \text{otherwise} \end{cases}$$

the numbers h_{kl} can be interpreted as the coefficients of the linear minimum mean square error estimator of f_{mn} given its neighbors out to distance P, and u as the estimation error.

This representation (called a "Conditional Markov" (CM) model by Kashyap (1983)) can then be used to generate sample functions (Woods, 1972 also presents an algorithm, based on the discrete Fourier transform, for the generation of sample realizations of the field u, and for the computation of the joint distribution for f). A field that satisfies a difference equation of the form:

$$f_{nm} = \sum_{D(P)} h_{kl} f_{n-k,m-l} + w_{nm}$$

where $\{w_{nm}\}$ are independent random variables, is called a "Simultaneous Autoregressive" (SAR) model by Kashyap (a similar representation can be obtained for fields with exponential autocorrelation functions; see Habibi, 1972). Although it is claimed that for any homogeneous SAR model it is possible to find a MRF with the same spectral density, albeit with a different neighborhood structure, it is in general very difficult to compute the joint distribution explicitly from the SAR representation. On the other hand, the Gibbs representation has the following advantages:

(i) It is perfectly general: it applies to discrete valued fields, and it can be easily generalized to the case of continuous valued ones.

(ii) It is easy to generate sample functions from the distribution (we will discuss algorithms for doing this in the next section).

(iii) Since the posterior distribution is also a Gibbs measure, the optimal estimates can be obtained directly from the posterior energy function.

For these reasons, this is the representation that we will adopt.

3. Generation of Sample Configurations of MRF's.

3.1. The Metropolis and Gibbs–Sampler Algorithms.

The earliest successful Montecarlo procedure for the generation of sample functions of MRF's was developed by Metropolis et al. (1953) for the numerical computation of thermodynamic properties of many–particle systems in thermal equilibrium. To describe it, let us consider a system with N particles, each of which may be in any one of a finite number of allowable states. Let f_j denote the state of the j^{th} particle (we will refer to the N –vector f as the global configuration of the system), and let $U(f)$ be the corresponding energy.

The basic idea of the algorithm is to construct a Markov chain whose states correspond to the global configurations of the system at discrete time intervals $1, \ldots, n$. It is a well known fact, from statistical physics, that when the physical system is at thermal equilibrium at a given temperature T, its configurations will be distributed according to the Gibbs measure:

$$\pi(f) = \frac{1}{Z} \exp[-\frac{U(f)}{T}] \tag{2}$$

Therefore, we want $\pi(f)$ to be the invariant measure for our chain. If the chain is regular (i.e., if it is possible to go between any two states in some fixed number of steps), $\pi(f)$ will be the unique vector satisfying:

$$\pi P_C = \pi$$

where P_C is the transition matrix of the chain (see Kindermann and Snell, 1980).

Also, since a system in equilibrium looks the same if we reverse the time direction, we require that the associated chain be reversible, that is,

$$\Pr(f(n+1) = j \mid f(n) = i) = \Pr(f(n-1) = j \mid f(n) = i)$$

For a regular chain, reversibility is equivalent to the "detailed balance" condition:

$$\pi(f)P_C(f, f') = \pi(f')P_C(f', f) \tag{3}$$

where f and f' are any two global configurations. This condition means that, if we consider a large collection of isolated, identical systems, each one in thermal

equilibrium at the same temperature (the so called "Canonical Ensemble"), the number of systems going from state f to f' must equal the number of systems going from f' to f. This condition is also sufficient for the convergence of the chain to the desired Gibbs measure.

The algorithm proposed by Metropolis generates a regular chain that satisfies (3). It is as follows:

Suppose that we visit the particles of the system (i.e., the sites of the lattice) in some random sequential order (for example, we choose the next site to be visited at random with uniform distribution). When a particle j is visited, we update its state as follows:

(i) Choose a new state $\hat{f}_j$ randomly from the set of allowable states using a uniformly distributed random number.

(ii) Compute the increment in energy ΔE_j that results from moving the state of the j^{th} particle from f_j to $\hat{f}_j$.

(iii) If $\Delta E_j \leq 0$, make the move, i.e., set $f_j = \hat{f}_j$.

If $\Delta E_j > 0$, generate a new random number r, uniformly distributed between 0 and 1.

If $r \leq e^{-\Delta E_j/T}$, set $f_j = \hat{f}_j$.

If $r > e^{-\Delta E_j/T}$, leave f_j unchanged.

If we denote by $q(f, \hat{f})$ the probability of proposing the state $\hat{f}$ when the system is at state f (i.e., the probability of visiting particle j, and selecting the state $\hat{f}_j$ for it; note that q must be a symmetric, irreducible stochastic matrix, so that $q(f, \hat{f}) = q(\hat{f}, f)$, by construction), we have that

$$P_C(f, \hat{f}) = q(f, \hat{f}) \min(1, e^{-\Delta U/T})$$

$$P_C(\hat{f}, f) = q(\hat{f}, f) \min(1, e^{\Delta U/T})$$

where

$$\Delta U = U(\hat{f}) - U(f)$$

Therefore, if $\Delta U < 0$,

$$P_C(f, \hat{f}) = q(f, \hat{f}) \text{ and } P_C(\hat{f}, f) = q(f, \hat{f}) e^{\Delta U/T}$$

and if $\Delta U > 0$,

$$P_C(f, \hat{f}) = q(f, \hat{f}) e^{-\Delta U/T} \text{ and } P_C(\hat{f}, f) = q(f, \hat{f})$$

Clearly, in both cases, (3) is satisfied.

This is not the only chain that satisfies (3). Another possibility is to set:

$$P_C(f, \hat{f}) = q(f, \hat{f}) \frac{\pi(\hat{f})}{\pi(f) + \pi(\hat{f})} =$$

$$= q(f, \hat{f}) \frac{1}{1 + e^{\Delta U/T}}$$

in which case we get the "heat bath" algorithm (see Gidas (1984) and Hastings (1982)).

A different construction, called the "Gibbs sampler" has been proposed by Geman and Geman (1983) (see also Besag (1972)). In this scheme, too, at each iteration only one site is modified; its new state, $\hat{f}_j$ is selected at random from the conditional distribution given by equation (1). These authors show that provided only that we keep visiting every site, (i.e., that we update its state "infinitely often") the resulting chain is ergodic, and its invariant measure is given by (2) (note that reversibility is not required in this case). It is not difficult to see that for binary systems this method is equivalent to the heat bath algorithm.

3.2. Statistical Mechanics Interpretation.

To get an intuitive grasp on the way these algorithms work, it is useful to recall some results from statistical mechanics (see, for example, Reif, 1965). When a macroscopic system (i.e., a system with a large number of degrees of freedom) is in thermal equilibrium at a given temperature T, its state f will be such that the Gibbs free energy F is minimized. The relation between $F(f)$ and the internal energy $U(f)$ of the system is given by:

$$F(f) = U(f) - TS$$

where the entropy S is:

$$S = \ln \Omega(U)$$

and $\Omega(U)$ is the total number of feasible configurations of the system with energy equal to U.

From this relation it is clear that at high temperatures, a system in equilibrium will adopt a disordered, high energy configuration (which will have a high value of S), while at low temperatures, the dominant tendency will be towards low energy states. The probability distribution of the equilibrium energy is given by:

$$P_U(U) = \frac{1}{Z} e^{-U/T} \Omega(U)$$

where Z is a constant. Since $\Omega(\cdot)$ is a rapidly increasing function of U, and the negative exponential is rapidly decreasing, P_U will be sharply peaked around a value $U^*(T)$. Using the fact that $\Omega(U) = O(U^n)$, where n is the number of degrees of freedom of the system, one can show that the relative width ΔU of this peak will be inversely proportional to the square root of n:

$$\frac{\Delta U}{U^*} \approx \frac{1}{\sqrt{n}}$$

(This result holds, in fact, not only for the energy, but for other related thermodynamical properties as well). This means that, for large n, the Metropolis (or Gibbs sampler) chain will generate (asymptotically) configurations whose energy is very close to $U^*(T)$, which is an increasing function of T.

To illustrate this, let us consider a binary system on a four–connected square lattice, whose energy function is given by:

$$U(f) = \frac{1}{T} \sum_C V_C(f_i, f_j)$$

with

$$V_C(f_i, f_j) = \begin{cases} -1, & \text{if } f_i = f_j \\ 1, & \text{otherwise} \end{cases}$$

Figure 4. Sample patterns of the two–dimensional Ising model at 0.8 (left), 1.0 (center) and 1.2 (right) times the critical temperature.

where C ranges over all the nearest neighbor cliques of the lattice (this is the two dimensional Ising model with "free boundaries" — since the only interactions that contribute to the energy are those between elements of the field that belong to the lattice — which we will later discuss in detail).

In figure 4 we present typical equilibrium configurations generated at three different temperatures using the Metropolis algorithm with random updating order. The temperatures used correspond to 0.8, 1.0 and 1.2 times the critical temperature for this model (the critical temperature is defined as the maximum value of the temperature for which the effect of fixed conditions at the boundary of a square lattice is felt at the center, no matter how large the lattice is. For the two–dimensional Ising model it equals 2.273).

In the limit of very large lattices, the equilibrium energy per spin (which is proportional to the total length of the boundaries between "black" and "white" regions) is given by (see Wannier, 1959):

$$\frac{U^*(T)}{n} = -\frac{2}{k}\coth\frac{1}{T}[1 \pm \frac{2}{\pi}(1-\alpha^2)^{1/2}K(\alpha)]$$

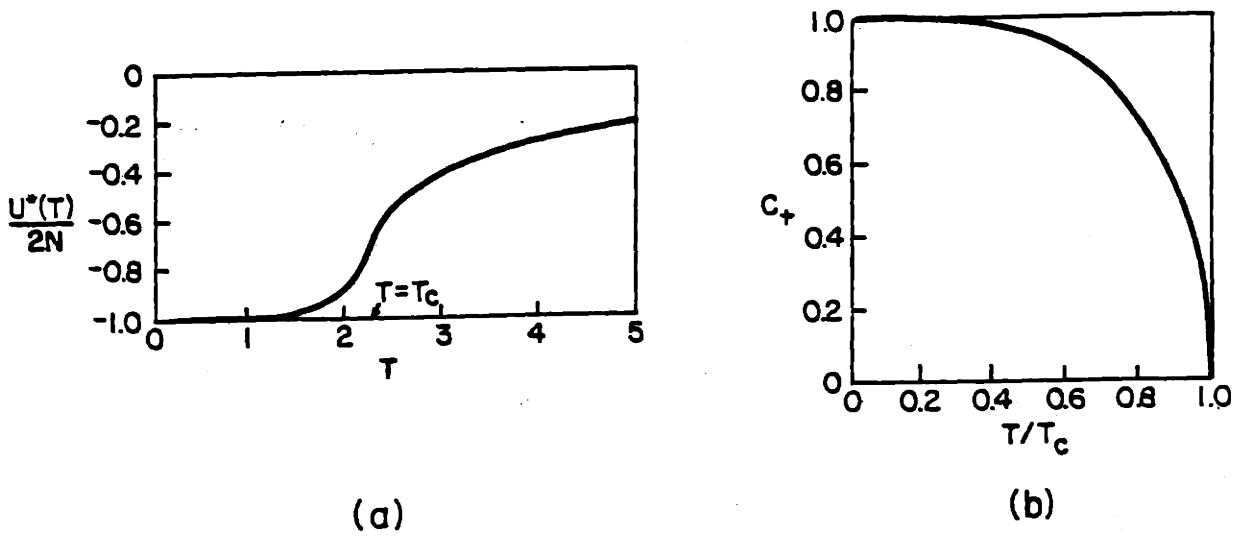


Figure 5. Equilibrium values of the energy (a) and average density (b) for an infinite Ising net (from Wannier, 1959)

where we take the + or − sign, above and below T_c, respectively. k is the Boltzmann constant; α is given by:

$$\alpha = \frac{2\sinh(1/T)}{\cosh^2(1/T)}$$

and $K(\cdot)$ is the complete elliptic integral of the first kind (see, for example, Hildebrand, 1976).

The average density of "black" elements can be computed by the expression:

$$C_+(T) = \frac{1}{2}[1 + \frac{\cosh^2(1/T)}{\sinh^4(1/T)}(\sinh^2(1/T) - 1)]^{1/8}$$

The shape of these functions is illustrated in figure 5.

From a qualitative viewpoint, one can see that the temperature, which is the only free parameter of this model, controls the granularity (average cluster size and cluster density) of the sample patterns.

Other examples of patterns generated with these algorithms (or some variations of them) may be found in Cross and Jain (1983) and Hassner and Sklansky (1980), where they are used as models for texture; in Geman and Geman (1983) as models

for piecewise constant images, and in Grenander (1983), where they are used to produce more complex patterns.

3.3. Continuous Valued State.

Any of the two algorithms presented in section 3.1 can be generalized to the case where the state of each particle can take any real value on a compact set (e.g., a closed interval) at the expense of their computational efficiency. A different approach that seems promising is based on the fact that a vector f which obeys the stochastic differential equation:

$$df = -\text{grad}U(f)dt + \sqrt{2T}dw \tag{4}$$

where w is a vector Wiener process with unit variance (a collection of independent Brownian motion processes), will be, under suitable smoothness conditions on U, distributed asymptotically (as $t \uparrow \infty$) according with the Gibbs measure (1) (see Grenander 1984; Geman and Hwang, 1984). This means that we can use a numerical simulation of (4) (see Wong and Zakai (1965)) to generate the desired patterns. This approach has two interesting advantages, that result from the fact that, in a numerical simulation, the increments dw are approximated by independent, identically distributed Gaussian random variables:

(i) We only need to generate Gaussian random numbers, for which efficient algorithms exist.

(ii) All sites can be updated at the same time, so that efficient parallel implementations can be adopted.

The probability distribution of the configurations generated by the system at any given time can, in principle, be obtained by solving an appropriate system of partial differential equations (i.e., the Kolmogorov equations; see for example, Karlin and Taylor, 1981); this will not be practical in most cases, however, so that the rate of convergence of this algorithm will have to be assessed in an experimental way.

We will now describe how an extension of the techniques presented in this section can be used to find the global minimum of arbitrary energy functionals. As we will show in the next chapter, this method will be particularly useful for minimization in the variational principles which represent the Maximum a Posteriori estimated solution to a reconstruction problem.

4. Simulated Annealing and Global Minimization.

Simulated annealing is a new technique, developed by Kirkpatrick et al (1983) for the solution of combinatorial optimization problems. It is based on the idea that any cost functional of N variables, each of which can take values on some finite set, can be considered as the energy function of a physical system whose state corresponds to a particular value of these variables. Therefore, we can use, say, the Metropolis algorithm to generate, at any given "temperature" T (which now becomes a parameter of the optimization process) samples from the corresponding Gibbs measure. Since as $T \downarrow 0$ this measure converges to an impulse (or set of impulses) corresponding to the state (or states) of minimum energy, the state of the system in thermal equilibrium at zero temperature will correspond to the value of f that minimizes $U(f)$ globally.

One serious difficulty, however, is that attaining thermal equilibrium might take a very long time at low temperatures. Kirkpatrick's idea was to start at a relatively high temperature (where thermal equilibrium is reached very fast), and then, to slowly cool the system, until "freezing" occurs and the state stops changing.

4.1. Discrete Valued State.

Geman & Geman (1983) were able to show that if the temperature is lowered at the rate:

$$T = \frac{C}{\log(n+1)} \tag{5}$$

where n is the number of iterations, and C is a constant, this algorithm (using the Gibbs sampler) will in fact converge (in probability) to the set of states of minimal

energy. They also showed that this chain is asymptotically ergodic in the sense that for any real valued function Y of the global state at time t, $f(t)$, we have:

$$\lim_{n\uparrow\infty} \frac{1}{n} \sum_{t=1}^{n} Y(f(t)) = \int_{\Omega} Y(\omega) dP_f(\omega)$$

where Ω is the set of allowable global states. This means that we can use time averages to estimate ensemble averages. Similar results have been obtained by Gidas (1984) for the Metropolis and heat bath algorithms.

The minimal value of the constant C in equation (5) for which convergence can be guaranteed has not been determined in general. The value found by Geman and Geman is:

$$C = N\Delta$$

where N is the total number of sites in the lattice, and Δ is the largest absolute difference in energies associated with pairs of global configurations that differ at only one site. This value, however, is too large to be of any practical use in most applications. Gidas (1984) has shown that if U has not more than two local minima, C can be computed as:

$$C = \frac{1}{\Delta'}$$

where Δ' is the minimal energy change between a local minimizer and a neighboring (in the sense that it differs at exactly one site) configuration. He also conjectures that this expression holds in general, but this result has not been confirmed.

In a recent paper, White (1984) characterizes the initial annealing temperature in terms of the standard deviation of the "density of states" (the number of possible states of the system, per unit energy, for each value of the energy) when this function is approximately Gaussian (which seems to be the case for a large class of systems). In some particular cases this value can be determined analytically from the structure of the problem, but in general, it has to be computed numerically from a simulation of the system at high temperature.

For the class of systems in which we are interested, we have found, by a trial and error procedure, that a value of C equal to 1.5 times the natural temperature of

the system (i.e., the temperature associated with the Gibbs distribution of the prior MRF model) produces a reasonable convergence behaviour (of the order of 500 iterations), but clearly, more research, both theoretical and experimental is needed in this area.

Another important factor which determines the computational efficiency of simulated annealing is related to the difficulty in computing the increment in energy ΔU_j associated with a change in the state of the j^{th} variable. If the energy function comes from the probability measure of a MRF, the computation of ΔU_j will require only the states of the variables in the neighborhood of j. Suppose now that we color the sites of the lattice in such a way that any two neighbors will always be of different color. In a parallel implementation we can, in principle, update the states of all the sites that are of the same color in a simultaneous way. The minimum number of colors needed to satisfy this condition is called the "Chromatic Number" of the graph that describes the neighborhood structure of the MRF, and it is bounded below by the size of the largest clique of the system. This number, then, determines the minimum number of steps that are needed in a parallel machine to update the state of the whole lattice. We will analyze these implementations in more detail for some particular examples in the next chapters.

4.2. Continuous Valued State.

All the available convergence results for the annealing algorithm hold only for the case where the set of allowable values for the state of each variable is finite. If this set is infinite, but compact, we can still use these results to find approximate solutions by discretizing it. However, the computational complexity will increase as we increase the resolution of this discretization. An attractive alternative is to generalize the approach discussed in section 2.2 by making T in equation (4) time dependent. A convergence proof for this modified scheme, for smooth energy functions that satisfy appropriate boundary conditions, can be found in Geman and Hwang, 1984.

5. Discussion.

We have presented a class of probabilistic models with local dependencies which can represent prior generic knowledge about the solution of a reconstruction problem: the class of MRF's on finite lattices. We have seen how they can be completely specified by defining arbitrary "potential functions" which are supported on the cliques of the associated neighborhood system. It is thus easy to define families of fields with a wide range of different behaviors. For example, if the only prior knowledge that we have is that the reconstructed surface should be piecewise constant, we may use a 4–connected lattice with Ising potentials:

$$V_C(f_i, f_j) = \begin{cases} -1, & \text{if } |i-j| = 1 \text{ and } f_i = f_j \\ 1, & \text{if } |i-j| = 1 \text{ and } f_i \neq f_j \\ 0, & \text{otherwise} \end{cases}$$

In this case, the natural temperature of the system will index a one parameter family of fields with varying degrees of granularity.

Smooth surfaces can be modeled using the same neighborhood system, but with quadratic potentials:

$$V_C(f_i, f_j) = \begin{cases} (f_i - f_j)^2, & \text{if } |i-j| = 1 \\ 0, & \text{otherwise} \end{cases}$$

More complicated, non–isotropic patterns can also be modeled, using slightly larger neighborhoods (as in Cross and Jain, 1983). Also, as we will see in chapter 5, an appropriate choice of the lattice and the neighborhood system, permits one to use a MRF to model sets of piecewise smooth curves on the plane. Using this construction, it is possible to model the behavior of a piecewise smooth function defined on a two–dimensional lattice (a "piecewise smooth surface") by coupling two MRF's: one for the smooth portions, and another for the curves that bound them.

We showed how the probability distribution of the configurations generated by a MRF has the same form as the one associated with a macroscopic physical system in thermal equilibrium, so that one can use Monte Carlo procedures that simulate

the behavior of such systems to generate sample functions of arbitrary MRF's. The Markovian property of the models imply that the computations performed by these procedures are local in nature (the updating rule for each site depends only on the states of its neighbors), so that, in principle, efficient parallel schemes can be designed for their implementation. We will examine this question in detail in the next chapter, where we discuss the use of MRF models and Bayes theory for the optimal solution of reconstruction problems.

Chapter 3

OPTIMAL BAYESIAN ESTIMATORS

1. Introduction.

The use of the Bayesian approach for the solution of reconstruction problems requires the development of the following items:

(i) A prior probabilistic model for the functions to be reconstructed.

(ii) Stochastic models for the observation processes.

(iii) Appropriate loss (error) criteria.

(iv) Estimators that are optimal with respect to (i), (ii), and (iii).

(v) Efficient algorithms for the computation of these estimates.

In the previous chapter, we discussed item (i), and presented a class of probabilistic models that can be used very effectively to encode prior generic constraints about the solutions of reconstruction problems. In this chapter we will develop the remaining necessary ingredients that are necessary to perform optimal reconstructions in the general case.

First of all, let us formulate the class of problems of interest in a precise way, and present a general stochastic model for the observation process.

2. Problem Formulation.

We mentioned in chapter 1 that there is an important class of perceptual problems whose solution can be found by reconstructing a function $f : R^n \mapsto R^m$ on a finite set of points that lie inside a compact domain $\Omega \subseteq R^n$. Although the methods that we will develop are, in principle, perfectly general, for the sake of clarity we will confine ourselves to the important particular case when $n = 2$ and

$m = 1$. We are, therefore, interested in reconstructing the value of a function f at each one of the N sites of a lattice L (we will denote the value of the function at site $i \in L$ by f_i).

2.1. Stochastic Model for the Observations.

Let us assume that we have a set of observations g on a subset S of the sites of L, and that the process by which these observation are obtained can be modeled by:

$$g_j = \Psi(H_j(f), n_j) \quad , \quad j \in S \tag{1}$$

Here , $H_j(\cdot)$ is an operator with local support that represents some kind of (in general non–invertible) degrading operation (such as blurring); Ψ is an operation invertible with respect to n_j (so that $n_j = \Psi^{-1}(g_j, H_j(f))$); it may represent, for example, noise addition or multiplication followed by a memoryless non–linear transformation. n_j represents a scalar noise process with known probability distribution P_{nj}. We will assume that n_j is independent of n_i, for all $i \neq j$, and also that it is independent of f.

Given f, the conditional probability distribution for the observations $P_{g|f}$ will be given by:

$$P_{g|f}(g; f) = \prod_{i \in S} P_{ni}(\Psi^{-1}(g_i, H_i(f))$$

Assuming that $P_{ni}(n_i) > 0$ for all i, and all possible values of n_i, we can define the functions Φ_i by:

$$\Phi_i(f, g_i) = -\ln P_{ni}(\Psi^{-1}(g_i, H_i(f)) \tag{2}$$

so that we can write the conditional distribution as:

$$P_{g|f}(g; f) = \exp[-\sum_{i \in S} \Phi_i(f, g_i)] \tag{3}$$

As an example, consider the case of additive , zero mean white Gaussian noise. We have:

$$H_i(f) = f_i$$

$$\Psi(a,b) = a + b$$

$$P_{ni}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-x^2/2\sigma^2]$$

$$P_{g|f}(g;f) = \prod_{i \in S} \frac{1}{\sqrt{2\pi}\sigma} \exp[-(f_i - g_i)^2/2\sigma^2] =$$

$$= \exp[-\sum_{i \in S} \{\ln(\sqrt{2\pi}\sigma) + \frac{1}{2\sigma^2}(f_i - g_i)^2\}]$$

2.2. Posterior Probability Distribution.

Since we are using a MRF model for the field f, its prior distribution will be of the form:

$$P_f(f) = \frac{1}{Z_0} \exp[-\frac{1}{T_0} U_0(f)] \tag{4}$$

with

$$U_0(f) = \sum_C V_C(f)$$

where C ranges over the cliques of the neighborhood system of f.

Using Bayes rule, we find that the posterior distribution is:

$$P_{f|g}(f;g) = \frac{P_f(f) P_{g|f}(g;f)}{P_g(g)}$$

Using the expressions (3) and (4) for P_f and $P_{g|f}$, and recognizing that $P_g(g)$ is a constant for a given set of observations, we get that the posterior probability will also follow a Gibbs distribution:

$$P_{f|g}(f;g) = \frac{1}{Z_P} \exp[-U_P(f;g)] \tag{5}$$

with

$$U_P(f;g) = \frac{1}{T_0} U_0(f) + \sum_{i \in S} \Phi(f, g_i) \tag{6}$$

Where Z_P is a constant, and the functions Φ_i are defined by (2).

We can now provide a physical interpretation of the posterior distribution, by considering that, while the prior distribution (4) describes the behavior of a free field in thermal equilibrium (see section 3.2 of chapter 2), the distribution (5) describes the behavior of the same field coupled with a fixed (but spatially varying) external field whose value is given by g. The functions Φ_i, whose magnitude depends on the noise variance, can then be interpreted as the coupling strengths between the two fields. This coupled system is also Markovian, and if

$$H_i(f) = H_i(f_i) \quad \text{for all } i \in S$$

its neighborhood structure will be identical to that of the original field.

The importance of this interpretation lies in the fact, which will be proved in the following sections, that the optimal estimate for f can be obtained *simply by observing the equilibrium behavior of this coupled field.* Before considering this question in detail, let us define the appropriate cost functionals for the applications we are interested in.

3. Cost Functionals.

The Bayesian approach to the solution of reconstruction problems has been adopted by several researchers. In most cases, the criterion for selecting the optimal estimate has been the maximization of the posterior probability (the Maximum a Posteriori or MAP estimate). It has been used, for example, by Geman and Geman (1984) for the restoration of piecewise constant images; by Grenander (1984) for pattern reconstruction, and by Elliot et. al. (1983) and Hansen and Elliot (1982) for the segmentation of textured images (a similar criterion — the maximization of a suitably defined likelihood function — has been used by Cohen and Cooper (1984) for the same purposes).

Since the use of this criterion defines the optimal estimator as the global minimizer of the posterior energy U_P (equation 6), it is closely related to the standard regularization method that we discussed in chapter 1. Indeed, if we assume quadratic potentials for the prior MRF model, the term $U_0(f)$ corresponds to a

global smoothness assumption (the "stabilizing functional"), and if the observations are corrupted by additive Gaussian noise, the term $\sum \Phi_i(f, g_i)$ will also be quadratic, so that U_P will have a unique minimum. For more general prior and observation models, the MAP estimator may be considered as an extension of the standard regularization approach. Thus, the variational principle proposed by Blake (1983), on a purely pragmatic basis, for the reconstruction of piecewise constant images is very similar to the one derived by Geman and Geman (1984). Even in this case, however, the precise probabilistic formulation in the latter case is preferable, since it provides a precise interpretation of the parameters, and a practical means for verifying the adequacy of the prior assumptions (via the experimental analysis of sample fields).

In some other cases, a performance criterion, such as the minimization of the mean squared error has been implicitly used for the estimation of particular classes of fields. For example, for continuous–valued fields with exponential autocorrelation functions, corrupted by additive white Gaussian noise, Nahi and Assefi (1972) and Habibi (1972) have used causal linear models and optimal (Kalman) linear filters for solving the reconstruction problem.

The minimization of the expected value of error functionals, however, has not been used as an explicit criterion for designing optimal estimators in the general case. We will show that this design criterion is in fact more appropriate in our case, for the following reasons:

(i) It permits one to adapt the estimator to each particular problem.

(ii) It is in closer agreement with one's intuitive assessment of the performance of an estimator.

(iii) It leads to attractive computational schemes.

We will now propose design criteria for two particular problems: image segmentation and surface reconstruction.

3.1. Error Criterion for the Segmentation Problem.

Consider a field f with N elements each of which can belong to one of a finite

set Q_i of classes. Let f_i denote the class to which the i^{th} element belongs. The segmentation problem is to estimate f from a set of observations $\{g_1, \ldots, g_p\}$. Note that f_i does not necessarily correspond to the image intensity. It may represent, for example, the texture class for a region in the image (as in Elliot et. al., 1983), etc.

A reasonable criterion for the performance of an estimate $\hat{f}$ is the number of elements that are not classified correctly. Therefore, we define the segmentation error e_s as:

$$e_s(f, \hat{f}) = \sum_{i=1}^{N} (1 - \delta(f_i - \hat{f}_i)) \qquad , f_i, \hat{f}_i \in Q_i \tag{7}$$

where

$$\delta(a) = \begin{cases} 1, & \text{if } a = 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

3.2. Error Criterion for the Reconstruction Problem.

In this case, we also consider a field f with N elements which can take values on finite sets $\{Q_i\}$, but now we assume specifically that f_i represents the intensity of an image (or the height of a surface) at site i. This suggests that an estimate $\hat{f}$ should be considered "good" if it is close to f in the ordinary sense, so that the total squared error:

$$e_r(f, \hat{f}) = \sum_{i=1}^{N} (f_i - \hat{f}_i)^2 \tag{9}$$

will be a reasonable measure for its performance.

Let us now derive the optimal estimators for these error criteria.

4. Optimal Bayesian Estimators.

To derive the optimal estimators with respect to the criteria stated above, we first present the general result (which can be found, for example in Abend, 1968) which states that if the posterior marginal distributions for every element of the field are known, the optimal Bayesian estimator with respect to any additive, positive

definite cost functional C may be found by independently minimizing the marginal expected cost for each element.

In more precise terms, we will consider cost functionals $C(f, \hat{f})$ of the form:

$$C(f, \hat{f}) = \sum_{i \in L} C_i(f_i, \hat{f}_i) \tag{10}$$

with

$$C_i(a, b) \begin{cases} = 0, & \text{if } a = b \\ > 0, & \text{if } a \neq b \end{cases}, \text{ for all } i$$

We will assume that the value of each element f_i of the field f is constrained to belong to some finite set Q_i (the generalization to the case of compact sets is straightforward). The Optimal Bayesian estimator $\hat{f}^*$ with respect to the cost functional C is defined as the global minimizer of the expected value of C over all possible f and g:

$$\bar{C}(\hat{f}^*) = \int_{f,g} C(f, \hat{f}^*) dP_{f,g}(f, g) =$$

$$= \inf_{\hat{f}} \int_{f,g} C(f, \hat{f}) dP_{f,g}(f, g) \tag{11}$$

We now have:

Theorem 1:

The optimal estimate of a field f with respect to the positive definite cost functional C can be found by minimizing independently the marginal expected cost for each element, i.e.,

$$\hat{f}_i^* = q \quad : \quad \sum_{r \in Q_i} C_i(r, q) P_i(r \mid g) \leq \sum_{r \in Q_i} C_i(r, s) P_i(r \mid g)$$

for all $s \neq q$, and for all $i \in L$.

$P_i(r \mid g)$ is the posterior marginal distribution of the element i:

$$P_i(r \mid g) = \sum_{f : f_i = r} P_{f \mid g}(f; g) \tag{12}$$

Proof:

First, we note that since C is positive definite, and since

$$P_{f,g}(f,g) = P_{f|g}(f;g)P_g(g)$$

where $P_g(g)$ is a constant for a given set of observations, we can write, from (11):

$$\sum_f C(f,\hat{f}^*)P_{f|g}(f;g) = \inf_{\hat{f}} \sum_f C(f,\hat{f})P_{f|g}(f;g)$$

Using (10), we rewrite the right hand side as:

$$\inf_{\hat{f}} \sum_f \sum_i C(f_i,\hat{f}_i)P_{f|g}(f;g) =$$

$$= \inf_{\hat{f}} \sum_i \sum_f C(f_i,\hat{f}_i)P_{f|g}(f;g) =$$

$$= \inf_{\hat{f}} \sum_i \sum_{r\in Q_i} \sum_{f:f_i=r} C(r,\hat{f}_i)P_{f|g}(f;g)$$

From (12), we find that this expression is equal to:

$$= \inf_{\hat{f}} \sum_i \sum_{r\in Q_i} C_i(r,\hat{f}_i)P_i(r \mid g)$$

which, since C is positive definite, we can rewrite as:

$$\sum_i \inf_{\hat{f}_i} \sum_{r\in Q_i} C_i(r,\hat{f}_i)P_i(r \mid g) \quad \blacksquare$$

The optimal estimators for the error criteria defined in section 3, can be easily derived from this result:

In the case of the segmentation problem, we put

$$C_i(f_i,\hat{f}_i) = 1 - \delta(f_i - \hat{f}_i)$$

and get that

$$\sum_{r\in Q_i} (1 - \delta(r,\hat{f}_i^*)P_i(r \mid g) = 1 - P_i(\hat{f}_i^* \mid g)$$

and therefore,

$$\hat{f}_i^* = q \in Q_i \quad : \quad P_i(q \mid g) \geq P_i(s \mid g)$$

$$\text{for all } s \neq q \tag{13}$$

We will call this estimate the "Maximizer of the Posterior Marginals" ($\hat{f}_{MPM}$).

For the reconstruction problem, we set:

$$C_i(f_i, \hat{f}_i) = (f_i - \hat{f}_i)^2$$

now,

$$\sum_{r \in Q_i} (r - q)^2 P_i(r \mid g) \leq \sum_{r \in Q_i} (r - s)^2 P_i(r \mid g)$$

implies that

$$-2q\bar{r} + q^2 \leq -2s\bar{r} + s^2$$

or equivalently,

$$(\bar{r} - q)^2 \leq (\bar{r} - s)^2$$

where

$$\bar{r} = \sum_{r \in Q_i} r P_i(r \mid g)$$

so that the optimal estimate is:

$$\hat{f}_i^* = q \in Q_i \quad : \quad (\bar{f}_i - q)^2 \leq (\bar{f}_i - s)^2$$

$$\text{for all } s \neq q \tag{14}$$

We will call this estimate the "Thresholded Posterior Mean" ($\hat{f}_{TPM}$).

Note that these results still hold if the sets Q_i of allowable values for each element, or the individual cost criteria C_i are not the same for all i. In particular, we may assume that the index i varies over the union of two lattices:

$$i \in L_1 \bigcup L_2$$

and let the field at the sites of L_1 represent the height of a piecewise smooth surface, and at the sites of L_2, take an integer value to indicate the presence (and possibly the direction) of a boundary between two adjacent continuous patches (see Geman and Geman, 1984; we will explain this construction in detail in chapter 5). If we now define a mixed error functional:

$$e_m(f,\hat{f}) = \sum_{i\in L_1}(f_i-\hat{f}_i)^2 + \lambda \sum_{i\in L_2}(1-\delta(f_i-\hat{f}_i))$$

for any positive value of λ, the optimal estimate will be:

$$\hat{f}_i^* = \begin{cases} \hat{f}_{TPM}(i), & i\in L_1 \\ \hat{f}_{MPM}(i), & i\in L_2 \end{cases}$$

The main obstacle for the practical application of these results, lies in the formidable computational cost associated with the exact computation of the marginals and the mean of the posterior distribution given by (5), even for lattices of moderate size. In the next section we will present a general distributed procedure that will permit us to approximate these quantities as precisely as we may want.

5. Algorithms.

The algorithms that we will propose are based on the use of the Metropolis or Gibbs Sampler schemes that we presented in chapter 2, to simulate the equilibrium behavior of the coupled MRF described by equation (5). We recall that the Markov chain generated by these algorithms is regular, and their invariant measure is the posterior distribution $P_{f|g}$. The law of large numbers for regular chains (see, for example, Kemeny and Snell, 1960) establishes that the fraction of time that the chain will spend on a given state f will tend to $P_{f|g}(f;g)$ as the number of steps gets large, independently of the initial state. This means that we can approximate $\bar{f}$ by:

$$\bar{f}_i \approx \frac{1}{n-k}\sum_{t=k}^{n} f_i^{(t)} \tag{15}$$

and the posterior marginals by:

$$P_i(q \mid g) \approx \frac{1}{k-n}\sum_{t=k}^{n} \delta(f_i^{(t)}-q) \tag{16}$$

where $f^{(t)}$ is the configuration generated by the Metropolis algorithm at time t, and k is the time required for the system to be in thermal equilibrium. From these values, $\hat{f}_{MPM}$ and $\hat{f}_{TPM}$ can be easily computed using (13) and (14).

This procedure is related to the use of simulated annealing (see section 4 of chapter 2) for finding the global minimum of U_P (i.e., the MAP estimate: see Geman and Geman, 1984). In our case, however, we are interested in gathering statistics about the equilibrium behavior of the coupled field *at a fixed temperature* $T = 1$, rather than in finding the ground state of the system. This fact gives our procedure some distinct advantages:

1. It is difficult to determine in general the descent rate of the temperature (annealing schedule) that will guarantee the convergence of the annealing process in a reasonable time (it usually involves a trial and error procedure). Since we are running the Metropolis algorithm at a fixed temperature, this issue becomes irrelevant.

2. Since in our case we are using a Monte Carlo procedure to approximate the values of some integrals, we should expect a nice convergence behavior, in the sense that coarse approximations can be computed very rapidly, and then refined to an arbitrary precision (in fact, it can be proved (see Feller, 1950) that the expected value of the squared error of the estimates (15) and (16) is inversely proportional to n).

The main disadvantage of this procedure is that in the case of the segmentation problem, a large amount of memory might be required if the number of classes per element m is large (we need to store the $N(m-1)$ numbers that define the posterior marginals).

With respect to the relative performance, we point out that in many cases, particularly for high signal to noise ratios, the MAP estimate is usually close to the optimal one. If the noise level is high, however, the difference in the performances of the two estimators may be dramatic. This is illustrated in the example portrayed in figure 6: panel (c) represents the MAP estimate of the binary MRF (a) from the noisy observations (b); it is clear that the approximations to the MPM estimates

shown in panels (d) and (e) are better than the MAP from almost any viewpoint. An intuitive explanation for this behavior comes from the fact that the MAP estimator is implicitly minimizing the expected value of a cost functional $C_{MAP}(f, \hat{f})$ which is equal to zero only if $f_i = \hat{f}_i$ *for all* i, and is equal to, say, M otherwise. If the signal to noise ratio is sufficiently high, the expected value of the optimal segmentation error will be very close to zero, so that $\hat{f}_{MPM}$ and $\hat{f}_{MAP}$ will coincide. In a high noise situation, however, the MAP estimator will tend to be too conservative, since from its viewpoint it is equally costly to make one or one thousand mistakes. The MPM estimator, in contrast, can make a better (although more risky) guess, since making a few mistakes has only a marginal effect on the expected cost. We will return to this example, and analyze in detail the relative performance of both estimates in the next chapter.

6. Computational Complexity and Parallel Implementations.

We have seen how the optimal solutions of reconstruction problems , for a large class of cost criteria, can be obtained from the observation of the evolution of the Markov chain generated by the algorithms presented in chapter 2. In this section, we will discuss the following questions:

(i) Which of these algorithms is the best one to use on a serial machine, from the viewpoint of the computational efficiency.

(ii) Which one is best suited for an implementation in parallel hardware.

We will also describe a parallel machine that is currently under construction at Thinking Machines Corporation and at the MIT Artificial Intelligence Laboratory: the "Connection Machine" (Hillis, 1985), and present estimates for the execution time of these algorithms in that particular piece of hardware.

6.1. Serial Complexity.

Suppose we are running our algorithms on a serial machine. In the three cases (Metropolis, Heat Bath and Gibbs Sampler), we first have to select the next site

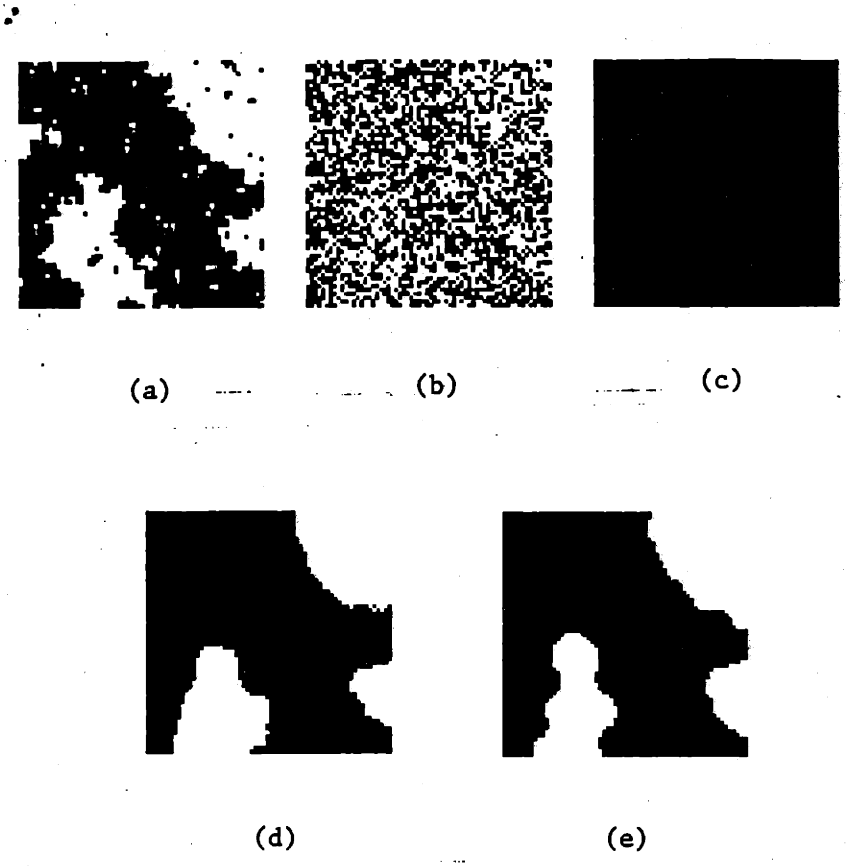


Figure 6. (a) Sample function of a binary MRF. (b) Output of a binary symmetric channel (error rate: 0.4) (c) MAP estimate. (d) Monte Carlo approximation to the MPM estimate. (e) Deterministic approximation to the MPM estimate.

whose state has to be updated. Assume it is site i. Let ΔU_q denote the increment in the posterior energy associated with replacing the value of the state of the i^{th} element by the value q. Using (6) and the expression for U_0 of (4), we get:

$$\Delta U_q = \frac{1}{T_0} \sum_{C:i \in C} (V_C(f^{(q)}) - V_C(f)) + \Phi_i(f^{(q)}, g_i) - \Phi_i(f, g_i) \tag{15}$$

where

$$f_j^{(q)} = \begin{cases} f_j, & j \neq i \\ q, & j = i \end{cases} \tag{16}$$

Let $C(\Delta U)$ denote the computational cost of evaluating (15).

The necessary steps for updating the state of site i are, in the Metropolis scheme (see section 3.1 of chapter 2):

(i) Select the candidate state q from the set Q_i (generate a uniform pseudo–random number in the range $(0, |Q_i|]$, with cost $C(prn)$, and load q from a table, with cost $C(load)$).

(ii) Compute ΔU_q.

(iii) Check if $\Delta U_q > 0$ (cost: $C(comp)$). If not, set $f_i = q$. Otherwise, go to (iv):

(iv) Compute $exp[-\Delta U_q]$ (cost: $C(exp)$).

(v) Generate a new uniform pseudo–random number in the range $(0, 1)$.

(vi) Compare it with $\exp[-\Delta U_q]$.

Therefore, we have that the total updating cost for the Metropolis scheme, C_M, satifies:

$$C_M > C(\Delta U) + C(prn) + C(comp) + C(load)$$

$$C_M \leq C(\Delta U) + 2C(prn) + C(exp) + 2C(comp) + C(load) \qquad (17)$$

For the Heat Bath scheme, steps (i), (ii) and (iv) are identical, and step (iii) is deleted. The remaining steps are in this case:

(v) Generate a new uniform pseudo–random number r in the range $(0, 1 + \exp[-\Delta U_q]]$

(vi) If $r > 1$, set $f_i = q$; otherwise, leave f_i unchanged.

The updating cost for the Heat Bath scheme, C_{HB} is then:

$$C_{HB} = C(\Delta U) + 2C(prn) + C(exp) +$$

$$C(comp) + C(add) + C(load) \qquad (18)$$

and in general, it will be higher than C_M, since

$$C(exp) >> C(comp)$$

For the Gibbs Sampler, we select the new state by generating a pseudo–random number which takes values on Q_i, with probabilities given by the conditional distribution (equation (1) of chapter 2). To do this efficiently, we rewrite this

equation as:

$$P_i(q \mid f) = \frac{\exp[-\Delta U_q]}{\sum_{r \in Q_i} \exp[-\Delta U_r]}$$

(Note that $\Delta U_{f_i} = 0$).

Let $Q_i = \{q_1, \ldots, q_M\}$. We now generate an array a, by putting:

$$a_0 = 0$$

$$a_j = a_{j-1} + \exp[-\Delta U_{q_j}] \quad , \quad j = 1, \ldots, M$$

The new state $\hat{f}_i$ is now computed by generating a uniform pseudo–random number r in the range $(0, a_M]$, and putting:

$$\hat{f}_i = q_j \quad ; \quad r \in (a_{j-1}, a_j]$$

The computational cost will be:

$$C_{GS} = (M-1)[C(\Delta U) + C(exp) + 2C(add) + 4C(load) + C(comp)] +$$

$$+C(prn) \tag{19}$$

note that we are including the overhead cost incurred by the use of the auxiliary array a.

If N is the size of the lattice, and we perform n iterations to compute our estimate, the total cost will be:

$$C_T = N \cdot n \cdot (C(update) + C(select) + C(overhead)) \tag{20}$$

where $C(select)$ is the cost associated with the selection of the next site whose state is going to be updated. This selection involves the generation of 2 uniform pseudo–random numbers in the first two cases, whereas for the Gibbs sampler it requires only a couple of additions, since in this case we can select the next site using a deterministic rule, such as lexicographic order (see section 6.3 below).

Since $C(update)$ is the dominant cost, apparently one should conclude that the Metropolis algorithm is the most efficient. It must be considered, however, that as the size of the state space (i.e., $M = |Q_i|$) increases, the number of iterations needed to get an estimate with an equivalent degree of precision will increase much faster in the Metropolis or Heat Bath cases, than in the Gibbs sampler, since in the latter case we are using an "importance sampling" procedure, versus the uniform sampling of the former (see Hammersley and Handscomb, 1965).

A rigorous analysis of the tradeoffs involved is not easy, and is highly dependent on the nature of the particular problem, so that an experimental analysis might be needed to clarify these questions in each case. In the more interesting case of a parallel implementation, however, the Gibbs sampler becomes the obvious choice. We will justify this assertion in the following sections.

6.2. Parallel Updating.

A necessary condition for the convergence of the probability measures of the Markov chains defined by the Metropolis, Heat Bath or Gibbs Sampler algorithms to the Gibbs measure is that if two sites belong to the same clique, they are never updated at the same time. As we will show in the next section, this condition is also sufficient only for the case of the Gibbs sampler. In this case it is possible to update simultaneously the states of all non–neighboring sites, by implementing the algorithm in a parallel architecture in which a processor is assigned to each site. The total execution time will then be reduced by a factor of

$$\frac{N}{K}$$

where K is the so called "chromatic number" of the graph that describes the neighborhood structure, and it is equal to the minimum number of colors needed to color the sites of the lattice in such a way that no two neighbors are the same). Note that if the state of every site is allowed to take real (continuous) values, we may use a numerical simulation of the stochastic differential equation:

$$df = -\text{grad}U(f)dt + \sqrt{2T}dw$$

to generate sample configurations from the desired distribution (see section 3.3 of chapter 2). In this case, all sites can be updated at the same time, so that a parallel implementation can reduce the complexity by a factor of N.

6.2.1. Convergence of the Gibbs Sampler.

Geman and Geman (1984) established that the measure of the Markov chain defined by the Gibbs sampler will converge to the Gibbs measure independently of the initial state, independently of the order in which the sites are updated (provided only that we keep visiting every site, i.e., that we update its state infinitely often). The convergence of the parallel implementation, therefore, follows from this general result for which we present here a simple alternative proof:

First, we note that from the definition of a MRF, it follows that for every site i, every value $q \in Q_i$, and every configuration f, the conditional probability,

$$\Pr(f_i = q \mid f_j \quad , \quad j \neq i) > 0$$

Since by hypothesis every site is visited infinitely often, this implies that any two states of the chain will be mutually accessible (with positive probability) in a finite number of steps, which means that the Gibbs sampler defines a regular chain.

On the other hand, the Gibbs measure $\pi(f)$ is an invariant probability vector of the chain. To see this, suppose that at time t, just before updating site i, the possible configurations of the field $F(t)$ are distributed according with the Gibbs measure:

$$\Pr(F(t) = f) = \pi(f)$$

After the update we have:

$$\Pr(F(t+1) = f) = \Pr(F_i(t+1) = f_i \mid F_j(t) = f_j \quad , \quad j \neq i) \cdot$$

$$\cdot \Pr(F_j(t) = f_j \quad , \quad j \neq i) =$$

$$= \pi(f_i \mid f_j \quad , \quad j \neq i) \cdot \pi(f_j \quad , \quad j \neq i) = \pi(f)$$

because, by the definition of the algorithm, the new state of the i^{th} element is selected randomly according with the conditional Gibbs distribution. The proof is now completed by remembering a well known theorem for finite Markov chains (see Kemeny and Snell, 1960) that establishes that every regular Markov chain:

(i) Has a *unique* invariant probability measure.

(ii) The measure of the chain will converge (with probability 1) to this invariant measure independently of the initial probability distribution of the states.

Note that, unlike the Metropolis and Heat Bath algorithms, the convergence of the Gibbs sampler does not depend on the reversibility of the chain (or equivalently, on the satisfaction of the "detailed balance" condition given by equation (3) of chapter 2), although this condition will hold if we use it with a random updating order. We will now see that the reversibility will not hold in general if we use a parallel updating scheme, which will make the first two algorithms unsuitable for parallel implementations.

6.2.2. Breakdown of Reversibility for Parallel Updating.

To show why this condition is violated (by the three algorithms) when a parallel updating scheme is used, we will consider a first order, binary MRF on a lattice L with Ising potentials, that is,

$$f_i \in \{0, 1\} \quad \text{for all } i \in L$$

$$V_C(f_i, f_j) = \begin{cases} -1, & \text{if } |i-j| = 1 \text{ and } f_i = f_j \\ 1, & \text{if } |i-j| = 1 \text{ and } f_i \neq f_j \\ 0, & \text{otherwise} \end{cases}$$

To implement a parallel updating scheme, we divide the sites of the lattice into two non–overlapping sets, which we will call B and W (the sets of "black" and "white" sites, respectively) as illutrated in figure 7.

Let f_W, f_B denote the state of the elements belonging to W and B, respectively,so that $f = \{f_W, f_B\}$. The parallel updating scheme consists in updating first, say, all

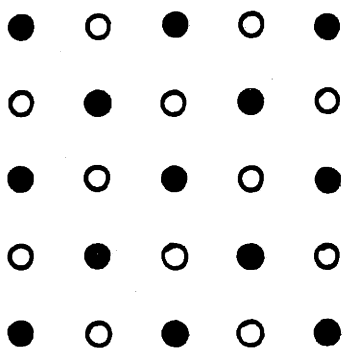


Figure 7. Non–overlapping sets for parallel updating (see text)

the white sites, and then all the black ones. Note that the random variables associated with any two sites of the same color are conditionally independent (given the state of the elements of the other color), which means that the order in which their state is updated is immaterial, so that, in fact, they can be updated simultaneously.

Let P_W, P_B, denote the transition probabilities corresponding to an update of all the white and black sites, respectively. Note that both Markov chains with transition probabilities P_W and P_B satisfy the detailed balance condition (although they are clearly not regular), so that for a fixed f_B, we have:

$$P_W(\{f_W, f_B\}, \{\hat{f}_W, f_B\}) = \frac{\pi(\hat{f}_W, f_B)}{\pi(f_W, f_B)} P_W(\{\hat{f}_W, f_B\}, \{f_W, f_B\})$$

and similarly, for a fixed f_W,

$$P_B(\{f_W, f_B\}, \{f_W, \hat{f}_B\}) = \frac{\pi(f_W, \hat{f}_B)}{\pi(f_W, f_B)} P_B(\{f_W, \hat{f}_B\}, \{f_W f_B\})$$

where π is the Gibbs measure of the complete configuration $f = \{f_W, f_B\}$.

Now, let $P_{WB}(f, \hat{f})$ be the transition probability associated with a complete "white–black" update (where the white elements are updated first). We have:

$$P_{WB}(f, \hat{f}) = P_W(\{f_W, f_B\}, \{\hat{f}_W, f_B\}) P_B(\{\hat{f}_W, f_B\}, \{\hat{f}_W, \hat{f}_B\}) =$$

$$= P_W(\{\hat{f}_W, f_B\}, \{f_W, f_B\}) \frac{\pi(\hat{f}_W, f_B)}{\pi(f_W, f_B)} \cdot$$

$$\cdot P_B(\{\hat{f}_W, \hat{f}_B\}, \{\hat{f}_W, f_B\}) \frac{\pi(\hat{f}_W, \hat{f}_B)}{\pi(\hat{f}_W, f_B)} =$$

$$= \frac{\pi(\hat{f})}{\pi(f)} P_{BW}(\hat{f}, f)$$

where P_{BW} is the transition probability of the converse "black–white" update (black sites visited first).

Now, consider the particular configuration:

$$f_i = \begin{cases} 0, & i \in W \\ 1, & i \in B \end{cases}$$

and let

$$\hat{f}_i = 1 \quad \text{for all } i \in L$$

Clearly,

$$P_{BW}(\hat{f}, f) > P_{WB}(\hat{f}, f)$$

and so,

$$\pi(f) P_{WB}(f, \hat{f}) > \pi(\hat{f}) P_{WB}(\hat{f}, f)$$

so that the detailed balance condition does not hold.

The above argument can be easily generalized to show that if we use any prescribed updating order (such as lexicographic order), the Markov chain generated by any of the three algorithms will also become irreversible. These chains, however, will remain regular, which means that in each case, the probability distribution of the configurations generated by the chain will converge towards a unique invariant distribution. In general, however, it will not be possible to guarantee the coincidence of this invariant measure with the desired Gibbs distribution, except in the case of the Gibbs sampler.

An example of a situation in which the invariant distribution is not the Gibbsian measure, can be obtained by running the Metropolis algorithm, either

with lexicographic or "black–white" updating order for the Ising model discussed in section 2 of chapter 2. If the natural temperature is below the critical temperature of the infinite lattice, the algorithm will produce equilibrium configurations that are almost completely uniform, and therefore, inconsistent with the theoretical predictions (and with the behavior of the same algorithm when random updating order is used). The Gibbs Sampler (which in this case is equivalent to the Heat Bath scheme), on the other hand, produces consistent results, as expected.

6.3. Discussion.

The previous results mean that the expected computational cost (execution time) for the solution of a reconstruction problem on a large parallel machine, using our general Monte Carlo procedure, will be given by:

$$C_P = n \cdot K \cdot C_{GS} \tag{21}$$

where n is the number of (global) iterations; K is the chromatic number of the graph of the underlying Markov model, and C_{GS} is the updating cost of the Gibbs Sampler, given by equation (19).

An example of such a massively parallel architecture is the "Connection Machine" (Hillis, 1985). This machine was originally designed for the parallel processing of structured symbolic expressions, such as frames and semantic networks. It is a "Single Instruction Multiple Data" (SIMD) array processor consisting of 256,000 processing units (each with a single bit Arithmetic/Logical unit, and about 4K bits of storage) organized in a four–connected lattice that is 512 elements square. Besides this nearest–neighbor connectivity, it will also be possible (although computationally more expensive), to connect any two processors in the array using a "Cross Omega" router network (Knight, in Winston, 1984).

At each cycle of the machine, for which we will assume a duration of one microsecond, an instruction is executed by each processor, and a single bit is transmitted to its neighbors. This means that the updating scheme can be implemented most efficiently if the field is first order Markov, but higher

order processes can also be implemented without using the router by successively propagating the transmited state (the execution time, therefore, will grow linearly with the order of the field).

To make these results more concrete, consider, as an example, the problem of finding the optimal estimate for an M–ary, first order MRF with Ising potentials (i.e., the segmentation of a piecewise constant image) from noisy observations (we will analyze this problem in detail in the next chapter). Let us assume that the estimator is to be implemented in the "Connection Machine", and suppose that by the use of appropriate scaling factors, all the numbers can be represented as 16–bit integers. We will use the following conservative assumptions: We assume that 16 cycles of a single 1–bit processor are needed to perform 16–bit addition, substraction or comparison; 16^2 cycles to perform multiplication or division; 2×16^2 cycles for generating a pseudo–random number with uniform distribution on a given interval; 16 cycles for memory transfer operations, and 6×16^2 cycles for computing an exponential.

Assuming that we run 250 iterations of the system, and ignoring the overhead time we get, from (19) and (21),

$$C_P \approx 1.4(M-1) \text{ seconds} \tag{22}$$

Although this execution time may be reasonable in many cases, it is clear that this approach becomes impractical as M becomes large. In this case, it might be more convenient to approximate the field by one in which the state at each site takes continuous values in a compact set and, provided that U_P satisfies the appropriate smoothness conditions, use the stochastic differential equation:

$$df = -\text{grad}U_P\, dt + \sqrt{2T}dw \tag{23}$$

where w is a Wiener process, to simulate the behavior of the system (see chapter 2, section 2.2).

This scheme will not work, however, if some of the variables are intrinsically discrete (e.g., binary variables indicating the presence or absence of a boundary). In

this case, it might still be possible to use a mixed scheme in which the state of the discrete variables is updated using the Gibbs Sampler, and that of the continuous ones using equation (23), but the precise form of such mixed schemes has not been determined, nor their convergence properties established.

These considerations provide us with a strong motivation for finding alternative ways of solving these problems. In particular, much more research is needed in the following directions:

(i) Design of more efficient (possibly deterministic) algorithms for approximating the optimal estimators for particular classes of problems.

(ii) Design of analog and hybrid networks for implementing these kinds of algorithms.

We will study these possibilities in detail, in the context of specific problems in the following chapters.

Chapter 4

RECONSTRUCTION OF PIECEWISE CONSTANT FUNCTIONS

1. Introduction.

In this chapter we will apply the optimal Bayesian estimators that we have developed, to the problem of reconstructing piecewise constant functions from noisy observations. The efficient solution of this problem is relevant for several reasons:

(i) Binary images (or images consisting of only a few grey levels) are directly useful in many interesting applications (for example, object recognition and manipulation in restricted (industrial) environments).

(ii) Several perceptual problems, such as the segmentation of textured images (Elliot, et. al. (1983); Hansen and Elliot (1982); Cohen and Cooper (1984)), or the formation of perceptual clusters (O'Callahan (1974); Marroquin (1976)), can be reduced to the problem of reconstructing a piecewise constant surface.

(iii) As we will see in the next chapter, where we treat the reconstruction of piecewise smooth surfaces, the boundaries between continuous patches can be adequately modeled by binary fields coupled with continuous valued processes. These coupled systems are very difficult to analyze in a rigorous way. We hope to increase our understanding of them by studying first the estimation of binary fields.

2. Problem Formulation.

Following Geman and Geman (1984), we will model the behavior of piecewise constant functions using first order MRF models on a finite lattice with generalized Ising potentials:

$$V_C(f_i, f_j) = \begin{cases} -1, & \text{if } |i-j| = 1 \text{ and } f_i = f_j \\ 1, & \text{if } |i-j| = 1 \text{ and } f_i \neq f_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$f_i \in Q_i = \{q_1, \ldots, q_M\} \quad \text{for all } i$$

We will use a free boundary model, so that the neighborhood size for a given site will be: 4, if it is in the interior of the lattice; 3, if it lies at a boundary, but not at a corner, and 2 for the corners.

The Gibbs distribution:

$$P_f(f) = \frac{1}{Z} \exp[-\frac{1}{T_0} U_0(f)]$$

$$U_0(f) = \sum_{i,j} V(f_i, f_j) \tag{2}$$

defines a one parameter family of models (indexed by T_0) describing piecewise constant patterns with varying degrees of granularity.

Using the general stochastic model for the observation process presented in section 2.1 of chapter 3, we get the posterior distribution given by equation (6) of that chapter:

$$P_{f|g}(f; g) = \frac{1}{Z_P} \exp[-U_P(f; g)]$$

with

$$U_P(f; g) = \frac{1}{T_0} U_0(f) + \sum_{i \in S} \Phi(f, g_i) \tag{3}$$

Of particular interest will be the case of binary fields ($M = 2$) with the observations taken as the output of a binary symmetric channel (BSC) with error rate ϵ (Gallager, 1975), so that:

$$P(g_i \mid f_i) = \begin{cases} (1-\epsilon), & \text{for } g_i = f_i \\ \epsilon, & \text{for } g_i \neq f_i \end{cases}$$

In this case, the posterior energy reduces to:

$$U_P(f; g) = \frac{1}{T_0} \sum_{i,j} V(f_i, f_j) + \alpha \sum_i (1 - \delta(f_i - g_i)) \tag{4}$$

where $f_i \in \{q_1, q_2\}$;

$$\delta(a) = \begin{cases} 1, & \text{if } a = 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

and

$$\alpha = \ln\left(\frac{1-\epsilon}{\epsilon}\right) \tag{6}$$

Note that in this case (and also in the case of additive white Gaussian noise), by modifying the constant Z_P, and applying a suitable linear transformation to the variables $\{f_i\}$, so that $Q_i = \{-1, 1\}$, we can write the posterior energy in the form:

$$U_P(f; g) = \frac{1}{T_0} \sum_{i,j:|i-j|=1} f_i f_j + \alpha \sum_i f_i g_i \tag{7}$$

which corresponds to the Hamiltonian of an Ising ferromagnet coupled with a spatially varying external magnetic field (whose magnitude is proportional to g). The importance of this connection is twofold: on the one hand, it means that the tools developed for the equilibrium behavior of these systems — which is what the estimation process is about — may be relevant for the physicists. On the other hand, it is conceivable that one could use physical ferromagnets to construct special purpose "quantum" computers that could solve estimation problems at atomic speeds.

In the following sections, we will study the relative performance of different Bayesian estimators, and design efficient algorithms for approximating them in some important particular cases.

3. Relative Performance of Bayesian Estimators for Binary Fields.

Once the posterior energy has been determined, one can solve the reconstruction problem by finding the optimal Bayesian estimate of the field f. As we discussed in chapter 3, however, we have several possible choices for the optimality criterion. To understand the differences in their performance, we will now analyze in detail the estimation of binary fields, when the observations are the output of a BSC with error rate ϵ.

Since the field is binary, the MPM and TPM estimators (defined by equations (13) and (14) of chapter 3, respectively) coincide. The question is: how do the performances of the MAP, and say, TPM estimates compare with respect to the error criterion:

$$\bar{e} = E[e_s(f, \hat{f})]$$

with

$$e_s = \sum_{i=1}^{N} (1 - \delta(f_i - \hat{f}_i))$$

where N is the size of the lattice, and the expectation is taken over all possible configurations f and g.

In particular we are interested in the ratio:

$$r = \frac{\bar{e}_{MAP}}{\bar{e}_{TPM}} =$$

$$= \frac{\sum_{f,g} \exp[-U_P(f;g)] e_s(f, \hat{f}_{MAP}(g))}{\sum_{f,g} \exp[-U_P(f;g)] e_s(f, \hat{f}_{TPM}(g))}$$

The numerical evaluation of this expression is feasible only for small values of N.

In figure 8 we show a plot of the ratio r for a 2×2 lattice, for different values of the error rate ϵ and the natural temperature T_0. As expected, r is never less than 1. In the worst case (for $\epsilon = 0.1$ and $T_0 = 0.2$) the error of the MAP estimate is 1.17 times that of the MPM estimate; if T_0 is not too small and ϵ is not too large, both estimates coincide, and as ϵ approaches 0.5 (low signal to noise ratio), the MPM estimate is consistently better than the MAP. An experimental analysis of larger lattices reveals a similar qualitative behavior, but the values of r are much larger in this case (see table 1).

3.1. Example.

We now return to the example presented in figure 6 of chapter 3, and examine it in more detail. Panel (a) represents a typical realization of a 64×64 Ising net with free boundaries, using a value of $T_0 = 1.74$ (0.75 times the critical temperature

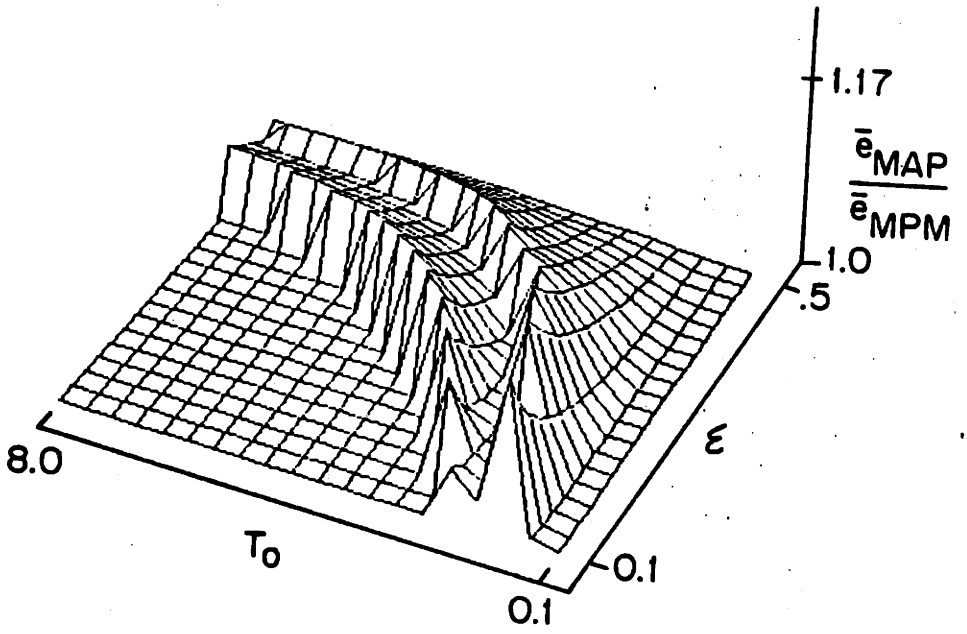


Figure 8. Ratio of the average errors of the MAP and MPM estimators for a 2×2 Ising net.

of the lattice); panel (b), the output of a binary symmetric channel with error rate $\epsilon = 0.4$; panel (c) the MAP estimate, and panel (d) an approximation to the MPM estimate (which we will label "MPM (M.C.)") obtained using the Metropolis algorithm and equation (10) to estimate the posterior density. The corresponding values of the posterior energy U_P (equation (13)) and the relative segmentation error ($e_s/64^2$) are shown on table 1.

Table 1

	f	g	$\hat{f}_{MAP}$	$\hat{f}_{MPM}(M.C.)$	$\hat{f}_{MPM}(Det.)$
Energy	−5594.8	−226.0	−6660.9	−6460.0	−6427.0
Seg. Error	–	0.4	0.33	0.128	0.124

4. Exact Algorithms for the MAP Estimator.

From the discussion of the previous section, it is clear that if the signal to noise ratio is not too low, the MAP criterion may be an appropriate choice, if one can design efficient algorithms for computing it. As we will now show, in the case of one–dimensional binary fields, one can in fact construct an algorithm which computes (exactly) the MAP estimate with computational complexity which is $O(N)$ (the length of the lattice) in a serial machine: at most $22N$ operations are needed, and the storage requirements are also $O(N)$. The algorithm can also be distributed in a parallel architecture, making its execution time independent of the lattice length.

To simplify the notation, we will assume that $f_i \in \{-1, 1\}$ for all i (there is no loss of generality in this asumption, since any binary process can be brought into this form by a reversible linear transformation). Also, assuming the noise process is stationary, we introduce the notation:

$$\Psi_{f_i}(g_i) = \frac{T_0}{2}\Phi_i(f_i, g_i)$$

where T_0 is the natural temperature of the field.

From equations (1) and (3), it is clear that the MAP estimation problem is equivalent to the minimization of:

$$U_P(f) = n + \sum_i \Psi_{f_i}(g_i)$$

where n is the number of places where $f_i \neq f_{i+1}$ (the number of odd bonds of the configuration). From this expression, it follows that the MAP estimation process can be reduced to the problem of finding the optimal value for n, and the best locations for the odd bonds (which we will also call "boundaries" between constant–valued blocks). We will now present a procedure for performing this task.

Description of the Algorithm.

The idea in which this method is based is the following:

We start scanning the sequence $\{g_i\}$, say, from the left, with some initial estimate $k \in \{-1, 1\}$ for the value of f in the block that starts at l_0 (a pointer that is initially set to 1).

Whenever we process a new observation g_j, we ask if we can get a lower energy by putting a boundary in j and in the best possible location l within the interval $[l_0, j]$, that is, we ask if:

$$U_b + 1 < U_p$$

where

$$U_p = \sum_{i=l_0}^{j} \Psi_{+k}(g_i)$$

$$U_b = 1 + \sum_{i=l_0}^{l} \Psi_{+k}(g_i) + \sum_{i=l+1}^{j} \Psi_{-k}(g_i)$$

As we will see below, the optimal boundary location l (which is initially set equal to l_0) needs to be updated only if the conditions:

$$f_j^{ML} \neq k$$

$$f_j^{ML} \neq f_{j-1}^{ML}$$

$$\sum_{i=l_0}^{j} \Psi_{+k}(g_i) - \Psi_{-k}(g_i) < \sum_{i=l_0}^{l} \Psi_{+k}(g_i) - \Psi_{-k}(g_i)$$

hold simultaneously, in which case l is set equal to $j - 1$. Here, f^{ML} denotes the maximum likelihood estimate; since we are using a white noise model, it is given

by:

$$f_j^{ML} = \begin{cases} 1, & \text{if } \Psi_{+1}(g_j) < \Psi_{-1}(g_j) \\ -1, & \text{otherwise} \end{cases}$$

If we get a lower energy by putting a boundary at l, we set $f_i = k$ for $i \in [l_0, l]$; update the value of the pointer l_0 by setting it equal to $l + 1$, and set the new estimate for the value of f, in the block that starts at l_0, equal to $-k$.

Otherwise, we just set $f_j = k$, and continue to process the next observation.

When we reach g_N, we take f_N as the initial estimate and run the same process backwards to get the final solution (in fact, one can show that it is possible to make this backward run as soon as we get the second boundary). This means that we can implement the algorithm in a distributed fashion, by processing in parallel overlaping subsequences of $\{g_i\}$, provided that the length of each of these subsequences is greater than twice the length of the largest constant-valued block in f. The final solution is then obtained by pasting together these partial estimates.

Formally, the algorithm is as follows:

Definition of Variables.

i: Current position.

l_0: Pointer to the beginning of the current region.

l: Current optimal location of the boundary in the interval $[l_0, i]$.

k: Current estimate for $f([l_0, l])$.

U_p: Energy increment associated with the assignment $f([l_0, i]) = k$.

U_m: Energy increment associated with the assignment $f([l_0, i]) = -k$.

U_b: Energy increment associated with the assignment $f([l_0, l]) = k; f((l, i]) = -k$.

si: Best local (maximum likelihood) estimate for f_i.

$sim1$: Best local (maximum likelihood) estimate for f_{i-1}.

U_{pl}: Energy increment associated with the assignment $f([l_0, l]) = k$.

U_{ml}: Energy increment associated with the assignment $f([l_0, l]) = -k$.

U_{temp}: Temporary storage register.

M: A very large positive number.

K_0: Switch indicating the method for estimating f_1.

Algorithm $A1(K_0)$:

1: Initialization.

Set $l_0 = l = 1; U_p = U_m = U_{ml} = 0; U_b = 1; U_{pl} = M$.

Set $k = 1$, if $K_0 = 0$ and $\Psi_{+1}(g_1) < \Psi_{-1}(g_1)$;
-1, if $K_0 = 0$ and $\Psi_{+1}(g_1) \geq \Psi_{-1}(g_1)$;
K_0, if $K_0 \neq 0$.

Set $sim1 = k$

2: Main Loop: For i from 1 to N do:

Begin

Set $si = 1$, if $\Psi_{+1}(g_i) < \Psi_{-1}(g_i)$;
-1, otherwise.

2.1: See if the optimal boundary location needs to be updated:

If ($si \neq k$ and $si \neq sim1$ and $U_p - U_{pl} - U_m + U_{ml} < 0$) do :

Update boundary location:

Set :

$l = i - 1$
$U_{pl} = U_p$
$U_{ml} = U_m$
$U_b = U_p + 1$

2.2: Update energy increments:

Set :

$$U_p = U_p + \Psi_{+k}(g_i)$$
$$U_m = U_m + \Psi_{-k}(g_i)$$
$$U_b = U_b + \Psi_{-k}(g_i)$$

2.3: See if a new boundary has to be introduced:

If $(U_b + 1 < U_p)$ do :

Introduce a new boundary:

For j from l_0 to l do : Set $f_j = k$

Set :

$$k = -k$$
$$l_0 = l + 1$$
$$U_{temp} = U_p - U_{pl}$$
$$U_p = U_m - U_{ml}$$
$$U_m = U_{temp}$$
$$U_{pl} = M$$
$$U_b = U_m + 1$$

2.4: Set $sim1 = si$

End

3: See if the last boundary has to be introduced:

If $(U_b < U_p)$ do :

3.1: For j from l_0 to l set $f_j = k$.

3.2: Set $l_0 = l + 1$.

3.3: Set $k = -k$.

4: Fill the last region:

For j from l_0 to N set $f_j = k$.

End.

The proof of the fact that this algorithm will in fact find the global minimizer of (7) is presented in appendix 4.A.

In appendix 4.B we present an alternative approach to this minimization, which is based on dynamic programming ideas. The resulting algorithm is less efficient than the one we have just presented for the case of binary fields, but it has the advantage of being extensible to handle more general situations. Also in this appendix, we compute the probability distribution for the number of odd bonds, and discuss the relationship between the dynamic programming procedure, and the use of linear filters to produce multi-scale descriptions of piecewise constant signals.

5. Estimation of Two-Dimensional Binary Fields.

The techiques developed in the last section for the exact computation of the MAP estimate cannot be extended to the two-dimensional case; the main difficulty here is that the geometry of the boundaries between uniform regions (which in the one dimensional case are simply points), causes a combinatorial explosion of the number of possible configurations compatible with a given total boundary length. The question, then, is whether it is possible to find algorithms that approximate the optimal estimates (with respect to the selected error criterion), that are more efficient than the general Monte Carlo procedures presented in chapter 3.

5.1. MAP Estimator.

In the case of the MAP estimator, the efficiency of the Simulated Annealing algorithm for the minimization of U_P can be improved by defining large "blocks" of sites (in a manner that is reminiscent of the "block-spin" strategy used by Wilson (1975) in connection with the renormalization group approach to the study of critical phenomena); the optimal estimate for the average value of the field in each of these blocks is found, and then progressively refined by subdividing the blocks in successive annealing stages. We will now show that, if we use a maximum entropy assumption, the structure of the MAP estimation process for Ising models

is invariant under the "blocking" transformation; this means that the ground state (i.e., the MAP estimator) of the aggregated process (with blocks of size L) also corresponds to that of an Ising model with a coupled external field, in which the natural temperature is scaled by a factor of $1/L$, and the noise (coupling) parameter by a factor of L^2. As a consequence of this scaling, the final temperature for the simulated annealing of this smaller network will be approximately L times larger than for the original problem.

Let us consider a binary Ising net f with the observations taken as the output of a binary symmetric channel with error rate ϵ. From section 2, we know that the posterior energy will be:

$$U_P = \frac{1}{T_0} \sum_{i,j} V(f_i, f_j) + \alpha \sum_i q(f_i, g_i) \tag{8}$$

with

$$q(f_i, g_i) = \begin{cases} 0, & \text{if } g_i = f_i \\ 1, & \text{if } g_i \neq f_i \end{cases}$$

and

$$\alpha = \ln\left(\frac{1-\epsilon}{\epsilon}\right)$$

Notice that equation (8) can also be written in the form:

$$U_P = \frac{1}{T_0} \sum_{i,j} V_C(f_i, f_j) + \alpha \sum_i q_C(f_i, g_i) \tag{8'}$$

where V_C, q_C are continuous functions satisfying:

$$V_C(x, y) = V(x, y) \quad \text{and}$$

$$q_C(x, y) = q(x, y) \quad \text{for} \quad x, y \in \{0, 1\}$$

We will now derive an expression for the energy in the "block spin" case. Let us partition the original lattice $\mathcal{L}$ into square blocks of side L. The "block observations" g_L will now be the density of 1's on each block, i.e.,

$$g_L(i) = \frac{1}{L^2} \sum_{j \in B_i} g_j$$

where B_i is the i^{th} block. The "block field" f_L is defined in a similar way.

For a given f_L, we compute the energy by assuming a maximum entropy configuration, which occurs when the 1's that correspond to the given density $f_L(i)$ are randomly distributed within the block. The energy will have three terms:

1. Interactions between adjacent blocks:

The interaction between two adjacent blocks i and j will be:

$$I_{ij} = [-1 \cdot (P_{11} + P_{00}) + 1 \cdot (P_{10} + P_{01})] \cdot L$$

where P_{kl} is the probability of having an element with state k on block i adjacent to an element with state l on block j:

$$P_{11} = f_l(i) f_L(j)$$

$$P_{01} = f_L(j)(1 - f_L(i))$$

$$P_{10} = f_L(i)(1 - f_L(j))$$

$$P_{00} = (1 - f_L(i))(1 - f_L(j))$$

Substituting these values we get:

$$I_{ij} = L[2(f_L(i) + f_l(j)) - 4 f_L(i) f_L(j) - 1]$$

2. Interactions within each block:

This term depends on the relative frequencies of the clique configurations 11,10,01 and 00 (p_{11}, p_{10}, p_{01} and p_{00}, respectively) on each block (note that there are $2L(L-1)$ different cliques). Since the 1's are randomly distributed we get:

$$p_{11} = f_L(i)^2$$

$$p_{10} = p_{01} = f_L(i)(1 - f_L(i))$$

$$p_{00} = (1 - f_L(i))^2$$

so that the internal interaction I_i is:

$$I_i = 2L(L-1)(-4f_L(i)^2 + 4f_L(i) - 1)$$

3. Interaction with the observations:

Assuming that the 1's in the observations and in the field are independently distributed we get:

$$I_{obs}(i) = \alpha L^2[f_L(i)(1-g_L(i)) + (1-f_L(i))g_L(i)] =$$

$$= \alpha L^2[f_L(i) + g_L(i) - 2f_L(i)g_L(i)]$$

Finally, the energy takes the form:

$$U_L(f_L) = \frac{1}{T_0}\sum_{i,j} I_{ij} + \sum_i (\frac{1}{T_0} I_i + I_{obs}(i)) =$$

$$= L\{\frac{1}{T_0}\sum_{i,j}[2(f_L(i) + f_L(j)) - 4f_L(i)f_L(j) - 1] +$$

$$+\frac{2}{T_0}(L-1)\sum_i(-4f_L(i)^2 + 4f_L(i) - 1) +$$

$$+\alpha L\sum_i(f_L(i) + g_L(i) - 2f_L(i)g_L(i)\}$$

note that the sums are taken over pairs of adjacent blocks, and over all the blocks, respectively. For $L = 1$, this expression reduces to (8') with

$$V_C(f_i, f_j) = 2(f_i + f_j) - 4f_if_j - 1$$

$$q_C(a, b) = a + b - 2ab$$

For $L > 1$, the quadratic terms of U_L are:

$$\frac{L}{T_0}[-4\sum_{i,j} f_L(i)f_L(j) - 8(L-1)\sum_i f_L(i)^2]$$

and since

$$-2\sum_{i,j} f_L(i)f_L(j) + 2\sum_i f_L(i)^2 = \sum_{i,j}(f_L(i) - f_L(j))^2 \geq 0$$

it follows that

$$\sum_i f_L(i)^2 \geq \sum_{i,j} f_L(i) f_L(j)$$

and

$$-4 \sum_{i,j} f_L(i) f_L(j) - 8(L-1) \sum_i f_L(i)^2 <$$
$$< -(4 + 8(L-1)) \sum_i f_L(i)^2 \leq 0$$

which implies that U_L is negative definite for $L > 1$, and therefore, its minima, constrained to the hypercube $[0,1]^{N_L}$ (N_L is the total number of blocks) will always lie in a corner of such hypercube, which means that we can use simulated annealing to find the global minimum of U_L, constraining the search to $\{0,1\}^{N_L}$. In this case, the energy to be minimized takes the simpler equivalent form (up to an additive constant):

$$U_L = \frac{1}{T_0/L} \sum_{i,j} V(f_L(i), f_L(j)) + \alpha L^2 \sum_i q(f_L(i), g_L(i))$$

The minimum energy solutions for each L can be interpreted as "coarse scale" representations of the original pattern f. Once a solution is obtained, the next refinement (for blocks of size $L/2$) can be efficiently obtained using the previous solution as a starting point, and initiating the annealing process at a lower temperature (the MAP estimates presented in this chapter were obtained using this technique). At present, however, we do not have a good method (other than trial and error) for determining the optimal values for these initial temperatures.

Also in this connection, the work of Blake (1983, 1985) should be mentioned. This author proposed the minimization of an energy function similar to U_P as a pragmatic criterion for restoring piecewise constant images. He also proposed an algorithm, based on the successive approximation of U_P by a family of convex envelopes to find an approximation to the global minimizer.

The relative performance and computational efficiency of these various schemes should be assessed experimentally.

5.2. MPM Estimator.

In the case of the MPM estimate, it is possible to construct a fast deterministic algorithm whose experimental performance (in terms of the average segmentation error) is equivalent to the Monte Carlo method discussed above. It is based on the following ideas:

First, we recall that for a binary pattern, the MPM and TPM estimates coincide. We will approximate the posterior mean of (3) by that of a Gaussian distribution P_G with the property:

$$P_G(h) = \frac{1}{Z_P} e^{-U_P(h)} \qquad \text{for all } h \in \{0,1\}.$$

In particular, we use:

$$P_G(h) = \frac{1}{Z_G} \exp[-\frac{1}{T_0} \sum_i \sum_{j \in N_i} (h_i - h_j)^2 - \alpha \sum_i (h_i - g_i)^2].$$

where

$$N_i = \{j \in L \quad : \quad \|i - j\| = 1\}.$$

For this distribution, $\bar{h}$ is the (unique) minimizer of the convex function:

$$U_G(h) = \frac{1}{T_0} \sum_i \sum_{j \in N_i} (h_i - h_j)^2 + \alpha \sum_i (h_i - g_i)^2$$

which corresponds to the unique fixed point of the system:

$$h_i^{(k+1)} = \frac{\sum_{j \in N_i} h_j^{(k)} + \alpha T_0 g_i}{|N_i| + \alpha T_0} \tag{9}$$

We could now approximate our estimate by putting:

$$\hat{f}_i^* = \Theta(\bar{h}_i)$$

where

$$\Theta(x) = \begin{cases} 1, & \text{if } x \geq \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

There is an additional consistency condition that $\hat{f}^*$ must satisfy, however. It can be shown that when the posterior distribution has the form given by (3) and (4), the MPM estimate $\hat{f}$, which by definition satisfies:

$$P_{i|g}(\hat{f}_i; g) > P_{i|g}((1-\hat{f}_i); g)$$

also satisfies:

$$P_{i|g}(\hat{f}_i; \hat{f}) > P_{i|g}((1-\hat{f}_i); \hat{f}) \tag{11}$$

which means that if we replace the observations by the MPM estimate, and compute a new MPM estimate for this modified problem, we should get the same result (the proof is included in appendix 4.C). Translating this condition to the case of $\hat{f}^*$, we get that it must satisfy:

$$\hat{f}_i^* = \Theta(h_i^*) \tag{12}$$

where h^* satisfies:

$$h_i^* = \frac{\sum_{j \in N_i} h_j^* + \alpha T_0 \Theta(h_i^*)}{|N_i| + \alpha T_0}$$

In practice, we get h^* as the fixed point of the system:

$$h_i^{*(k+1)} = \frac{\sum_{j \in N_i} h_j^{*(k)} + \alpha T_0 \Theta(h_i^{*(k)})}{|N_i| + \alpha T_0} \tag{13}$$

with

$$h^{*(0)} = \bar{h}$$

Note that the function:

$$U_h(h) = \sum_{i,j \in N_i} (h_i - h_j)^2 + \alpha T_0 \sum_i (h_i - \Theta(h_i))^2$$

acts as a Lyapunov function for the system (13), which is therefore (locally) stable (Vidyasagar, 1978).

This algorithm can be visualized as operating in two steps: In the first one, we extract all the information that we need from the observations and encode it in

$\bar{h}$ (which is continuous-valued), and in the second one, we find the closest binary pattern that satisfies the consistency condition (11).

To illustrate the performance of this approximation, we show $\hat{f}^*$, for the example discussed above, in panel (e) of figure 1, and its corresponding energy and segmentation error in the last column of table 1 (labeled "MPM det.").

5.2.1. Parallel Implementation.

The dynamical systems defined by equations (9) and (13) can be implemented directly in a parallel architecture, such as the "Connection Machine", by assigning a processor to each site, and updating the state of all sites at the same time. Each update will require, for both systems, at most 10 (16-bit) additions and two multiplications, that is, a total of 672 cycles of a 1-bit processor. We have found experimentally that in most cases, less than 50 iterations of (9), and 100 of (13) are needed for convergence, so that, using the figures of chapter 3, we estimate the total execution time as approximately 0.1 seconds, an improvement of one order of magnitude over the general Monte Carlo procedure described in that chapter.

5.3. Analog Networks.

Hopfield and Tank (1985) (see also Hopfield, 1982 and 1984) have studied the behavior of "neural" analog networks of non-linear amplifiers interconnected by resistors, whose dynamics can be described by the differential equations:

$$\frac{du_i}{dt} = \sum_{j \in N_i} T_{ij} f_j - \frac{u_i}{\tau} + I_i \qquad (14)$$

$$f_i = \Theta(u_i)$$

Here, N_i is the neighborhood of node i; u_i and f_i denote the input and output voltage of the i^{th} amplifier; T_{ij} is the conductance of the link between the nodes i and j; I_i is a fixed current injected at node i, and τ, a constant depending on the internal resistance and capacitance of each amplifier. The gain function of the

amplifiers, $\Theta(\cdot)$ is chosen as a sigmoid function that restricts the output to the interval $(0, 1)$, and has a form similar to the observed response of biological neurons (hence the term "neural"). In particular, one can put:

$$\Theta(u) = \frac{1}{1 + \exp[-\beta u]} \tag{15}$$

where β is called the "gain parameter".

These researchers have proved that the system (14) is always stable, provided we have $T_{ij} = T_{ji}$ for all i, j, and in the high gain limit (for $\beta >> 1$), the stable fixed points will be local minima of the "energy" function:

$$E(f) = -\frac{1}{2}\sum_{i,j} T_{ij} f_i f_j - \sum_i f_i I_i \tag{16}$$

Note that we can write (14) as:

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial f_i} - \frac{u_i}{\tau} \tag{18}$$

$$f_i = \Theta(u_i)$$

They have also pointed out that if one uses the gain function (15), the fixed points of (18) will satisfy:

$$f_i = \frac{1}{1 + \exp[-\beta \tau H_i(f)]} \tag{19}$$

with

$$H_i(f) = -\frac{\partial E}{\partial f_i} = \sum_{j \in N_i} T_{ij} f_j + I_i \tag{20}$$

These equations will also be satisfied by the mean field approximation (see Reif, 1965) to the ensemble averages of a binary process f ($f_i \in \{0, 1\}$) with respect to the Gibbs measure generated by the energy (16) at a temperature $T = 1/\beta\tau$. This can be shown as follows:

The mean field approximation is obtained by assuming that the local energy at node i, which is:

$$E_i(f) = -f_i[\sum_{j \in N_i} T_{ij} f_j + I_i] = -f_i H_i(f)$$

can be approximated by:

$$E_i \approx -f_i[\sum_{j \in N_i} T_{ij}\bar{f}_j + I_i] = -f_i H_i(\bar{f})$$

where $\bar{f}_j$ denotes the ensemble average of f_j. Since $\{\bar{f}_j\}$ are constants for a given temperature, we can compute $\bar{f}_i$ as:

$$\bar{f}_i = \frac{\sum_{f_i=0,1} f_i \exp[-H_i(\bar{f})/T]}{\sum_{f_i=0,1} \exp[-H_i(\bar{f})/T]} =$$

$$= \frac{1}{1 + \exp[-H_i(\bar{f})/T]}$$

This means that there is a fixed point of equation (18) that can be interpreted as an approximation of the ensemble average of a corresponding binary MRF (note that in general this fixed point will not be unique, and will depend on the selection of the initial conditions; the lack of an adequate criterion for making this selection in the general case represents, at this point, a serious limitation of this approach).

In the case of the posterior energy (4), if we require that $f_i \in \{0, 1\}$, we can write it in the equivalent form (up to an additive constant):

$$U_P(f) = -\frac{4}{T_0}\sum_i \sum_{j \in N_i} f_i f_j - \sum_i [-\frac{2|N_i|}{T_0} + \alpha(2g_i - 1)]f_i$$

so that

$$H_i = -\frac{\partial U_P}{\partial f_i} = \frac{4}{T_0}\sum_{j \in N_i} f_j + \alpha(2g_i - 1) - \frac{2|N_i|}{T_0}$$

In this form, one can construct directly the system (18), and defining the initial state as $f_i^{(0)} = u_i^{(0)} = 0.5$ for all i, find the stable fixed point that will approximate $\bar{f}$. Since for a binary system the MPM and TPM estimators are equivalent, we can approximate the optimal estimate by:

$$\hat{f}_i = \begin{cases} 0, & \text{if } \bar{f}_i < 0.5 \\ 1, & \text{otherwise} \end{cases}$$

We have performed digital simulations of the system (18), and have found very good performances for relatively high signal to noise ratios. For high error rates,

the behavior of this approximation is similar to that of the MAP estimator. We will have to say more about this approach at the end of the next chapter.

6. Simultaneous Estimation of the Field and the Parameters.

To apply the estimation procedures described in the previous sections, the parameters that characterize, both the prior model of the field (the natural temperature T_0), and the noise process, (the error rate ϵ, or the variance σ^2) have to be known. In most practical cases, however, we are only given the noisy observations g and general qualitative information about the structure of the field and the noise, so that f, α (which stands for either $\log[(1-\epsilon)/\epsilon]$ or σ) and T_0 have to be simultaneously estimated.

In principle, one could use again a Bayesian approach, and assuming prior independent uniform distributions for α and T_0 (in the ranges $[\alpha^0, \alpha^1]$ and $[T_0^0, T_0^1]$, respectively), find those $\hat{\alpha}, \hat{T}_0$ and $\hat{f}$ which jointly maximize the posterior distribution:

$$P(f, \alpha, T_0 \mid g) = \frac{\exp[-U_P(\alpha, T_0, f)]}{(\alpha^1 - \alpha^0)(T_0^1 - T_0^0) Z(T_0) P_g(g)}$$

The main difficulty here is the extraordinary computational complexity of the partition function:

$$Z(T_0) = \sum_f \exp[-\frac{1}{T_0} U_0(f)]$$

which makes this approach impractical, except for very small lattices.

An alternative approach is based on the following considerations (we will study in detail the case of a BSC; other noise models can be analyzed in a similar way):

Equations (9) and (13), which describe the deterministic approximations to $\hat{f}_{MPM}$ depend on the parameters of the system, ϵ and T_0, only through the product:

$$\gamma = \alpha T_0 = T_0 \log\left(\frac{1-\epsilon}{\epsilon}\right) \tag{21}$$

which means that the behaviour of the algorithm is completely characterized by the single parameter γ. In the case of the Monte Carlo approximation, if we fix the

value of γ, the value of T_0 cannot be chosen arbitrarily, since it has to satisfy the consistency condition:

$$T_0 = \frac{\gamma}{\hat{\alpha}}$$

with

$$\hat{\alpha} = \log\left(\frac{1-\hat{\epsilon}}{\hat{\epsilon}}\right) \quad ;$$

$$\hat{\epsilon} = \frac{1}{N}\sum_{i=1}^{N} z_i \tag{22}$$

where z is the residual process defined as:

$$z_i = \begin{cases} 1, & \text{if } \hat{f}_i \neq g_i \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

This means that, given γ, the correct value of T_0 can, in principle, be determined in an adaptive way, so that in this case too, the behaviour of the approximation depends effectively only on γ.

For a given value of γ, we can approximate the corresponding MPM estimate $\hat{f}$ using the methods developed in the previous section, and compute the residual process z and the conditional (on γ) Maximum Likelihood Estimate of the error rate ϵ using equations (22) and (23). The corresponding conditional estimate for T_0 will be:

$$\hat{T}_0 = \frac{\gamma}{\hat{\alpha}} \tag{24}$$

To measure the "likelihood" of the estimate $\hat{f}$, we use the degree of uniformity (or "whiteness") of the residual process z. This property can be quantified by the variance of the local noise density, which we estimate as follows:

We cover the lattice with a set $\{S_j\}$ of m non–overlapping squares (say, 8 pixels wide). For each square S_j, the relative variance of the noise density is:

$$\sigma_j = \left(\frac{\bar{z}_j - \hat{\epsilon}}{\hat{\epsilon}}\right)^2 \tag{25}$$

with

$$\bar{z}_j = \frac{1}{|S_j|} \sum_{i \in S_j} z_i$$

where $|S_j|$ is the area of the j^{th} square.

The desired likelihood function is:

$$L(\hat{f}) = -\sum_{j=1}^{m} \sigma_j \tag{26}$$

which is equivalent to a χ^2 criterion (Cramer, 1946) normalized to take into account the sample size.

Alternatively, one can use directly the likelihood that the residuals come from a uniform distribution. To compute it, we note that the quantities:

$$\nu_j = \sum_{i \in S_j} z_i$$

are distributed according to the multinomial law:

$$P(\nu_1, \ldots \nu_m) = \frac{n!}{\nu_1! \ldots \nu_m!} \left(\frac{1}{m}\right)^n$$

with

$$n = N\hat{\epsilon} = \nu_1 + \ldots \nu_m$$

Using the Stirling approximation we get the log–likelihood:

$$L(\nu_1, \ldots, \nu_m) = \log P(\nu_1, \ldots, \nu_m) \approx -\sum_{i=1}^{m} \nu_i \log \nu_i +$$

$$+n \log\left(\frac{n}{m}\right) + \frac{1}{2} \log\left(\frac{n}{\nu_1 \ldots \nu_m}\right) + K \tag{27}$$

where K is a constant. We have found experimentally that both likelihood measures (26) and (27) have a similar behavior when n is large. When n is relatively small, or when for some i, $\nu_i = 0$, however, (26) is preferable, and so, it is the one we adopt.

Note that a more conventional likelihood function, such as the conditional likelihood proposed by Besag (1972), will not work in this case; this function is defined as:

$$L(\hat{f}) = \frac{L_1(\hat{f}) + L_2(\hat{f})}{2} \qquad \text{with}$$

$$L_k(\hat{f}) = \prod_{i \in C_k} P(\hat{f}_i \mid \hat{f}_j, j \in N_i, \hat{T}_0) =$$

$$= \prod_{i \in C_k} \left[\frac{\exp[-\frac{1}{\hat{T}_0} \sum_{j \in N_i} V(\hat{f}_i, \hat{f}_j)]}{\exp[-\frac{1}{\hat{T}_0} \sum_{j \in N_i} V(\hat{f}_i, \hat{f}_j)] + \exp[-\frac{1}{\hat{T}_0} \sum_{j \in N_i} V(1 - \hat{f}_i, \hat{f}_j)]} \right]$$

$$= \prod_{i \in C_k} (1 + \exp[\frac{2}{\hat{T}_0} \sum_{j \in N_i} V(\hat{f}_i, \hat{f}_j)])^{-1}, \qquad k = 1, 2$$

where the "codings" C_1 and C_2 are the sets:

$$C_1 = \{i \ : \ (x_i \text{ is odd and } y_i \text{ is even}) \text{ or } (x_i \text{ is even and } y_i \text{ is odd})\}$$

$$C_2 = \{i \ : \ (x_i \text{ is odd and } y_i \text{is odd}) \text{ or } (x_i \text{ is even and } y_i \text{ is even})\}$$

with (x_i, y_i) denoting the row and column indices of site i (notice that, given the value of the field at the sites of C_1, the random variables associated with any pair of sites of C_2 become independent, and viceversa). In our case, we find that as γ decreases, $\hat{f}$ becomes more and more uniform, while $\hat{T}_0$ remains almost constant. It is not difficult to see that as a result, the conditional likelihood L will decrease monotonically with γ, which renders it useless for our purpose.

The range of values $[\gamma_0, \gamma_M]$ of the parameter γ that corresponds to the class of systems of interest can be determined as follows:

One can show that for $\gamma > 8$ we will always have $\hat{f}_{MPMi} = g_i$ for all i, so that we can use $\gamma_M = 8$. The value of γ_0 can be obtained from an upper bound for ϵ and a lower bound for T_0. For example, assuming that $\epsilon \leq .45$ and $T_0 \geq .5T_c$, we get $\gamma_0 = .23$. (Note that when the natural temperature T_0 of a first order, isotropic MRF is below 0.5 times T_c (the critical temperature of the lattice; see Kindermann and Snell, 1980), the patterns become practically uniform (i.e., f_i =constant for all i), while for values of T_0 greater than $1.5T_c$, we get patterns that are practically

indistinguishable from white noise. Therefore, the assumption $T_0 \geq .5T_c$ covers practically all the interesting cases).

The complete estimation procedure is as follows:

Maximum Likelihood Estimation Algorithm:

1: Sample the interval $[\gamma_0, \gamma_M]$ at the points

$$\gamma_0 \leq \gamma_1, \ldots \gamma_n \leq \gamma_M$$

2: For each $\gamma \in Q = \{\gamma_1, \ldots \gamma_n\}$:

2.1: Find $\hat{f}(\gamma)$ using (9) and (13).

2.2: Compute z using (23).

2.3: Compute $\hat{\epsilon}$ using (22). If $\hat{\epsilon} = 0$, set $L(\hat{f}(\gamma)) = -\infty$ and proceed with the next value of γ. Otherwise, compute $\hat{\alpha}$ and go to 2.4.

2.4: Compute $\hat{T}_0$ using (24).

2.5: Compute $L(\hat{f}(\gamma))$ using (25) and (26).

3: Compute the optimal estimate $\hat{f}^*$ using:

$$\hat{f}^* = \hat{f}(\gamma^*) \quad : \quad L(\hat{f}(\gamma^*)) = \sup_{\gamma \in Q} L(\hat{f}(\gamma)) \tag{28}$$

The corresponding $\hat{\epsilon}^*, \hat{T}_0^*$ will be the optimal estimates for ϵ and T_0, respectively.

Remarks:

1. This estimation algorithm allows us to reconstruct a binary pattern f from the noisy observations g *without having to adjust any free parameters.* The only prior assumptions correspond to the qualitative structure of the field f (first order, isotropic MRF) and to the nature of the noise process, but neither the natural

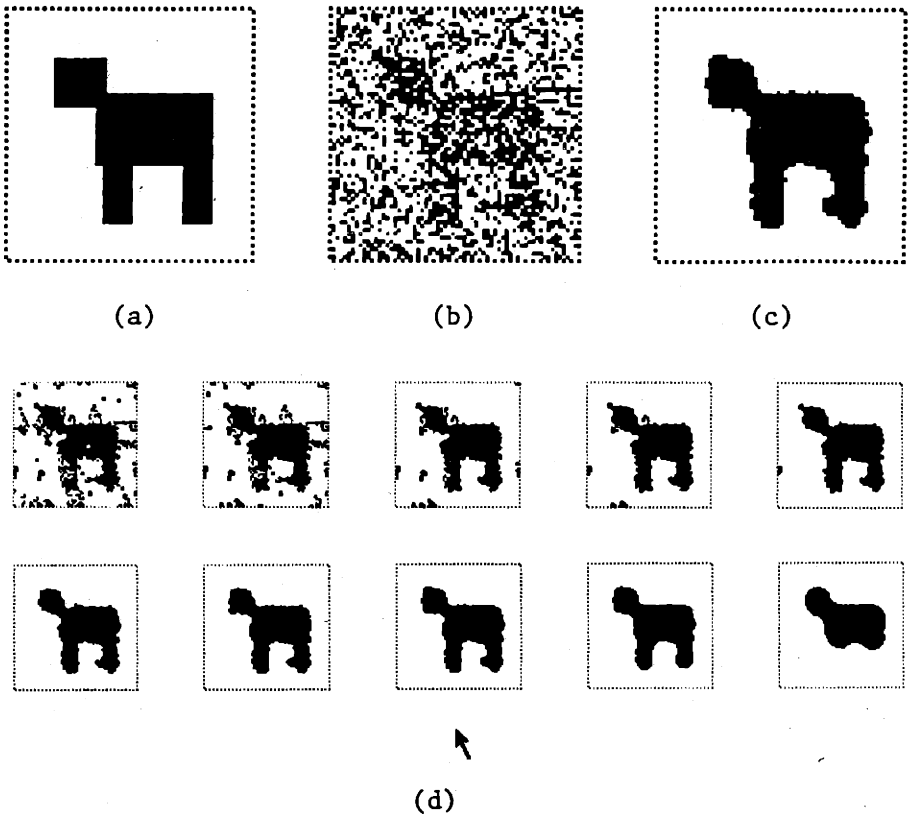
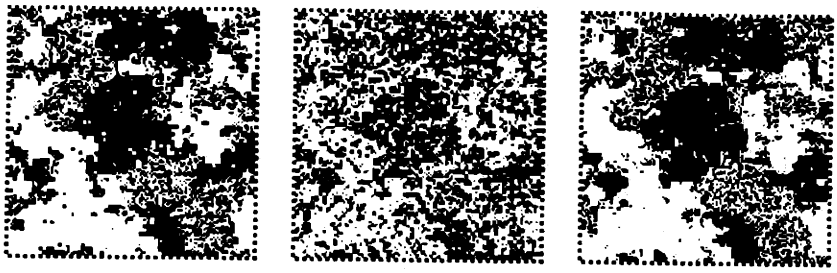


Figure 9. (a) Synthetic image. (b) Noisy observations. (c) Maximum Likelihood Estimate. (d) A complete series of estimates. The optimal estimate (for $\gamma = 2.9$) is indicated by an arrow.

temperature T_0 nor the error rate ϵ have to be known in advance. In practice, this means that we can apply it to restore any binary image with uniform granularity, even if it has not been generated by a Markov random process. We have used this algorithm to reconstruct a variety of binary images with excellent results. In figure 9 we show such a restoration. The observations (b) were generated from the synthetic image (a) with an actual error rate of .35 (assumed unknown). The MLE for f is shown in (c). A complete series of estimates $\hat{f}(\gamma)$, with γ varying from .5 to 3.5 is shown in panel (d).

2. This procedure can be easily extended to handle any one-parameter noise corruption process (such as zero mean, additive white Gaussian noise). The extension to the case of N-ary fields, i.e., to the restoration of piecewise constant images, is also straightforward (using the general algorithm described in chapter 3 instead of (9) and (13) in step 2.1). As an example, we present in figure 10 the optimal



(a) (b) (c)

Figure 10. (a) Original ternary MRF. (b) Noisy observations (additive Gaussian noise). (c) Optimal (Maximum likelihood) estimate.

restoration of a ternary pattern corrupted by additive white Gaussian noise.

3. We have found that the likelihood function (26) is reasonably well behaved as a function of γ. This permits us to perform the one-dimensional search for its supremum in an economical way, by first determining its approximate location using a coarse sampling pattern, and then refining its position by a fine sampling of a reduced interval. In practice, it is possible to get very good results using on the order of 15 samples.

4. The whole procedure is highly distributed, so that it is possible to implement it in parallel hardware in a very efficient way.

7. Formation of Perceptual Clusters.

At the heart of a general purpose perceptual system, one must have a mechanism for deciding which parts of an image should be considered to "belong" together (Marroquin, 1976). A simple instance of this problem is the grouping of dots in an image into perceptual clusters. Some heuristic schemes have been proposed to model this phenomenon (see for example, O'Callahan, 1974). We will show, however, how

this problem can be formulated in an elegant way that is also biologically motivated, as a particular case of the reconstruction of binary patterns from noisy observations.

The conceptual model for this formulation is as follows:

Let us consider the dots that form the original pattern as patches belonging to some objects of uniform color that are partially hidden, say, by some foliage. In this way, the formation of clusters is equivalent to the problem of reconstructing these objects (whose cohesive nature is modeled by a first order MRF with Ising potentials) from observations that are formed according with the following model:

Suppose that $f_i = 1$ only if an object overlaps the i^{th} site of the lattice. We assume that the "foliage" will hide this point (i.e., make $g_i = 0$) with probability ϵ, and that spurious values of $g_i = 1$ can appear in sites where $f_i = 0$ with a very small probability ρ:

$$g_i = \begin{cases} 1, & \text{with prob. } (1-\epsilon), \text{ if } f_i = 1 \\ 0, & \text{with prob. } \epsilon, \text{ if } f_i = 1 \\ 0, & \text{with prob. } (1-\rho), \text{ if } f_i = 0 \\ 1, & \text{with prob. } \rho, \text{ if } f_i = 0 \end{cases}$$

with $\rho << 1$. The posterior energy is:

$$U_P(f;g) = \frac{1}{T_0} U_0(f) + \alpha \sum_{i:f_i=1} (1 - \delta(1-g_i)) + \\ + M \sum_{i:f_i=0} (1 - \delta(g_i)) \qquad (29)$$

where $U_0(f)$ is given by (1) and (2):

$$V_C(f_i, f_j) = \begin{cases} -1, & \text{if } |i-j| = 1 \text{ and } f_i = f_j \\ 1, & \text{if } |i-j| = 1 \text{ and } f_i \neq f_j \\ 0, & \text{otherwise} \end{cases}$$

$$U_0(f) = \sum_{i,j} V(f_i, f_j) \quad ;$$

δ and α are defined in (5) and (6):

$$\delta(a) = \begin{cases} 1, & \text{if } a = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha = \ln\left(\frac{1-\epsilon}{\epsilon}\right)$$

and M is a very large number.

The clustering task is now equivalent to the problem of estimating f and the parameters α and T_0 from the noisy observations g. To accomplish this, we can use the method described in the previous section, except that in this case, only those sites for which $\hat{f}_i = 1$ will be useful for the estimation of the residual density and its local variance. This means that equation (22) has to be modified to:

$$\hat{\epsilon} = \frac{1}{|A|} \sum_{i \in A} z_i$$

where

$$A = \{i \ : \ \hat{f}_i = 1\}$$

and z_i is defined in (23). Also, the sets S_j used to compute the relative variance of the residual density in (25) should now be taken as the intersection of the squares that cover the lattice with the set A.

With these modifications, the Maximum Likelihood algorithm can be used for clustering. Its performance is illustrated in figure (11) where we show: the original dot pattern (upper left) and the recontructed objects for decreasing values of $\gamma = \alpha T_0$. The maximizer of the likelihood is marked with an arrow. We believe that these preliminary results are encouraging, although, clearly, more numerical and psychophysical experiments are needed to assess the plausibility of this scheme to model human perceptual processes.

8. Discussion

In this chapter we have addressed the problem of reconstructing piecewise constant functions from noisy observations. We showed that the optimal solution to this problem can be obtained from the observation of the equilibrium behavior of a generalized Ising net coupled with a spatially varying (but fixed in time)

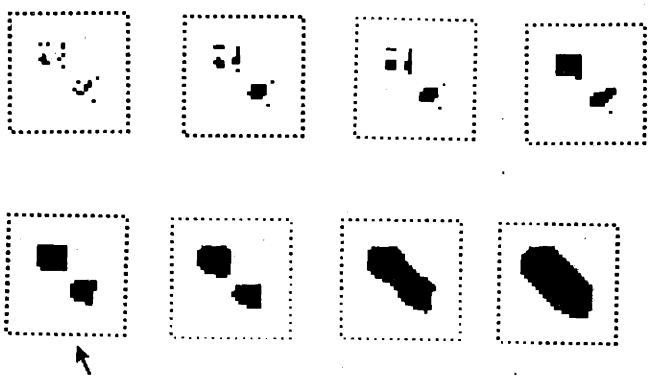


Figure 11. Formation of perceptual clusters. We show: the original dot pattern (upper left) and the reconstructed objects for decreasing values of $\gamma = \alpha T_0$. The maximum likelihood estimate (i.e., the optimal clustering) is marked with an arrow.

external field. If we use the minimization of the expected segmentation error as a criterion, the optimal estimate is the maximizer of the posterior marginals (the MPM estimator which was described in chapter 3).

We compared the relative performance of the MAP and MPM estimators, and found that for moderate signal to noise ratios, they are practically equivalent, but as the noise level increases, the MPM estimate is (sometimes dramatically) superior. A consequence of this analysis is that, if the noise level is not too high, the MAP estimator may be a reasonable choice in those cases where it is computationally advantageous. This is the case, for example, of the reconstruction of one–dimensional binary signals, where we derived a very efficient algorithm for its exact computation.

In the two–dimensional case, however, the situation is different: the general Monte Carlo procedure for the approximation of the MPM estimator is in fact more efficient, from a computational viewpoint, than the corresponding one for the MAP (Simulated Annealing), and in the case of binary fields, we derived a much faster

deterministic scheme with excellent experimental performance.

We also showed how these estimation procedures can be extended to the more interesting case where the parameters of the system are not known in advance. In this case, a maximum likelihood estimation algorithm can be derived, which, using a likelihood function that is computed from the residuals, allows for the simultaneous estimation of the field and the parameters.

We point out that although, for the sake of simplicity, we have concentrated on the case of binary fields sent through binary symmetric channels, the results that we have presented can be generalized to N-ary fields and other noise models (see figure 10).

The constructions that we have presented can be applied not only to image segmentation and restoration, but to other problems as well. As an illustration, we presented a novel application to the modeling of the process of formation of perceptual clusters. Another important problem that can be formulated in this way is the recontruction of surfaces from stereoscopic pairs of images; we will discuss it in detail in chapter 6.

Appendix 4.A

OPTIMALITY OF ALGORITHM A1

In this appendix we present a proof of the fact that the algorithm presented in section 4 of chapter 4, effectively computes the MAP estimate for a one-dimensional, binary MRF.

The optimality of the algorithm follows from the following propositions:

Proposition 1: Let $S^* = \{l_1, \ldots l_n\}$ be the optimal boundary configuration, and suppose that l_k, for $k < n$ was detected by $A1$. Then, l_{k+1} will be the next boundary detected by $A1$.

Proof:

Suppose l_k was detected by $A1$, and let L be the next boundary detected. We will assume that $L \neq l_{k+1}$ and arrive at a contradiction. We will consider three cases:

Case 1: Suppose $A1$ detects L at $j < l_{k+1}$.

Then, we must have that

$$U_p(j) > U_p(L) + U_m(j) - U_m(L) + 2$$

and therefore,

$$U(\{l_1, \ldots, l_k, L, j, l_{k+1}, \ldots\}) < U(S^*)$$

which is a contradiction.

Case 2: Suppose $A1$ detects L at $j \in (l_{k+1}, l_{k+2}]$.

This means that at j we had that L was the optimal location for the boundary. In particular,

$$U_p(l_{k+1}) + U_m(j) - U_m(l_{k+1}) > U_p(L) + U_m(j) - U_m(L)$$

which implies that

$$U_p(L) + U_m(l_{k+2}) - U_m(L) < U_p(l_{k+1}) + U_m(l_{k+2}) - U_m(l_{k+1})$$

and therefore,

$$U(\{l_1, \ldots, l_k, L, l_{k+2}, \ldots\} < U(S^*)$$

which is a contradiction.

Case 3: Suppose that $A1$ has not detected any new boundary at $j = l_{k+2} + 1$.

Then, we must have:

$$U_p(l_{k+2} + 1) < U_b(l_{k+2} + 1) + 1$$

which means that

$$U(\{l_1, \ldots l_k, l_{k+3}, \ldots\} < U(S^*)$$

which is again a contradiction. ∎

Proposition 2: If $A1$ runs from left to right starting at a point l_0, and generates the boundaries $\{l_1, l_2, \ldots\}$, then, $l_j \in S^*$ (the set of boundaries of the optimal configuration) for $j \geq 2$.

Proof:

Let f^*, f_{A1} be the optimal configuration, and the one generated by $A1$, respectively.

Let

$$L_0 = \sup\{j \in S^* \ : \ j < l_1\}$$

$$L = \inf\{j \in S^* \ : \ j > l_1\}$$

If $L_0 = l_0$, we apply proposition 1 and finish the proof; so, let us assume that $L_0 \neq l_0$, and that l_1 was detected at i. We have two cases:

Case 1: $L_0 > l_0$. We claim that in this case, $l_1 \in S^*$, and therefore, by proposition 1, $l_j \in S^*$ for $j \geq 1$. To prove this claim, we consider two subcases:

Case 1-a: $f^*((l_0, L_0)) \neq f_{A1}((l_0, L_0))$.

In this case, we have:

$$2 + U_m(i) - U_m(l_1) + U_p(l_1) < U_p(i)$$

and therefore,

$$2 + U_m(i) - U_m(l_1) + U_p(l_1) - U_p(L_0) < U_p(i) - U_p(L_0)$$

which implies that $l_1 \in S^*$.

Case 1-b: $f^*((l_0, L_0)) = f_{A1}((l_0, L_0))$.

Suppose $l_1 \notin S^*$. We have that, at location i,

$$U_p(l_1) + U_m(i) - U_m(l_1) + 2 < U_p(L_0) + U_m(i) - U_m(L_0) + 2$$

since otherwise, L_0 would have been a better location for the boundary. However, this implies that

$$U_p(l_1) + U_m(L) - U_m(l_1) < U_p(L_0) + U_m(L) - U_m(l_1)$$

which means that we can improve S^* by moving L_0 to l_1, which is a contradiction.

Case 2: $L_0 < l_0$.

Again, we consider two subcases:

Case 2-a: $f^*((L_0, l_0)) = f_{A1}((L_0, l_0))$.

Let U_+, U_- be the energy increments with respect to L_0:

$$U_+(i) = \sum_{j=L_0}^{i} \Psi_{+k}(g_j)$$

$$U_-(i) = \sum_{j=L_0}^{i} \Psi_{-k}(g_j)$$

Note that

$$U_p(i) = U_+(i) - U_+(l_0) \qquad \text{and}$$

$$U_m(i) = U_-(i) - U_-(l_0)$$

Since l_1 was detected at i, we have:

$$2 + U_m(i) - U_m(l_1) + U_p(l_1) < U_p(i)$$

and therefore,

$$2 + U_-(i) - U_-(l_1) + U_+(l_1) < U_+(i)$$

which means that $l_1 \in S^*$.

Case 2-b: $f^*((L_0, l_0)) \neq f_{A1}((L_0, l_0))$.

Using the same definitions for U_+, U_-, we have that, by the optimality of S^*, for some $j > L$,

$$U_-(j) - U_-(L) + U_+(L) + 2 < U_+(j)$$

and therefore,

$$U_-(j) - U_-(L) + U_+(L) - U_+(l_1) + 2 < U_+(L) - U_+(l_1)$$

which means that if $A1$ detects l_1, it must detect L too, unless it detected l_2 first, but in this case we have that, for some $p < j$,

$$U_-(p) - U_-(l_2) + U_+(l_2) - U_+(l_1) + 2 < U_+(p) - U_+(l_1)$$

which implies that $l_2 \in S^*$. This completes the proof. ∎

It should be clear that these results can be easily extended to the case where $A1$ runs backwards (from right to left). With this extension, we get the following complete optimal procedure:

Algorithm $A2$:

1: Run $A1$ from left to right. Detect $\{l_1, \ldots, l_n\}$.

2: Run $A1$ backwards (starting from l_2). Get either

$$\{l_2, \ldots, l_n\} \quad \text{or} \quad \{l_1', l_2, \ldots, l_n\}$$

In either case, this is the optimal solution.

The only thing that remains to be proved is that the determination of the optimal location for a boundary is in fact performed by step 2.1 of $A1$. We have the following:

Proposition 3: Suppose that $A1$ detected a boundary at (or started from) l_0. Then, the optimal location l of the next boundary has to be updated only at places where $si = -k$ and $sim1 = k$ (note that in si we have stored the value of the maximum likelihood estimate f_i^{ML}, while $sim1 = f_{i-1}^{ML}$). Suppose i is one such place. The optimal location will be:

$$l = \begin{cases} i-1, & \text{if } U_p(i-1) - U_m(i-1) < U_{pl} - U_{ml} \\ l, & \text{(the current value) otherwise} \end{cases}$$

Proof:

First, we note that a necessary and sufficient condition for l to be the optimal location of the boundary at the point i is that, for $j \in [l_0, i-1]$:

$$U_p(l) + U_m(i) - U_m(l) \leq U_p(j) + U_m(i) - U_m(j)$$

or equivalently,

$$U_p(l) - U_m(l) \leq U_p(j) - U_m(j)$$

Suppose l was the optimal location at $i-1$, and we process observation i. We consider several cases:

Case 1: $sim1 = -k$

In this case, we show that l remains the optimal location:

By construction, we have that:

$$U_p(i-1) = U_p(i-2) + \Psi_{+k}(g_{i-1})$$

$$U_m(i-1) = U_m(i-2) + \Psi_{-k}(g_{i-1})$$

Since $sim1 = -k$ we have that,

$$\Psi_{+k}(g_{i-1}) - \Psi_{-k}(g_{i-1}) > 0$$

and therefore,

$$U_p(i-1) - U_m(i-1) = U_p(i-2) - U_m(i-2) + \Psi_{+k}(g_{i-1}) - \Psi_{-k}(g_{i-1}) >$$

$$> U_p(i-2) - U_m(i-2) \geq U_p(l) - U_m(l)$$

so that l remains the optimal location.

Case 2: $sim1 = k$

In this case we have that

$$U_p(i-1) - U_m(i-1) < U_p(i-2) - U_m(i-2)$$

This means that the minimal value for $U_p(i) - U_m(i)$ on a block for which $si = k$ will be obtained at the extremal point where $si = -k$ and $sim1 = k$, and since, by theorem 1 of appendix 4.B, this is the only point where a boundary might be placed, it is sufficient to update the optimal location only at these points. So, suppose $sim1 = k$ and $si = -k$.

If

$$U_{pl} - U_{ml} < U_p(i-1) - U_m(i-1),$$

then,

$$U_{pl} - U_{ml} < U_p(j) - U_m(j) \text{ for } j \in [l_0, i-1]$$

because l was the optimal location outside the last block where $si = k$. By the same token, it is clear that if

$$U_{pl} - U_{ml} \geq U_p(i-1) - U_m(i-1),$$

the new optimal location will be $i-1$.∎

Appendix 4.B

DYNAMIC PROGRAMMING FORMULATION OF THE ONE-DIMENSIONAL MAP ESTIMATION PROBLEM

In this appendix we present an algorithm for finding the global minimum of:

$$U_P = \sum_{i=1}^{N-1} V(f_i, f_{i+1}) + \alpha \sum_{i=1}^{N-1} \Phi_{f_i}(g_i) \tag{1}$$

which, based on dynamic programming principles, reduces the problem to a sequence of one dimensional optimizations.

As we will see, this algorithm generates, as a byproduct, a family of solutions which can be considered as descriptions of the field f at different scales, so that the coarse descriptions, which are computed very fast, are progressively refined until the optimal (finest scale) configuration is found.

This approach is based on the following idea:

A configuration f is completely characterized by the value of f_1, and the set $\mathcal{L}_n$ defined by:

$$\mathcal{L}_n = \{L \ : \ f_L \neq f_{L+1}\} \ ; \ |\mathcal{L}_n| = n. \tag{2}$$

We will call the n elements of $\mathcal{L}_n$ the "boundaries" of the configuration f. Since these boundaries correspond to odd bonds between neighboring cells, we can define an equivalent energy function as:

$$U(f) = n + \frac{\alpha}{2}\hat{U}(f) \tag{3}$$

$$\text{with } \hat{U}(f) = \sum_i \Phi_{f_i}(g_i), \quad f_i \in \{k_0, k_1\} \tag{4}$$

For a fixed n, U depends only on the value of f_1, and on the position of the n boundaries, that is, on $n+1$ variables. To make this dependence more explicit, let

us define the functions

$$G(L) = \sum_{j=1}^{L} (\Phi_{k_0}(g_j) - \Phi_{k_1}(g_j)) \tag{5}$$

Let U_0 and U_1 denote the energy functions corresponding to the configurations with $f_1 = k_1$ and k_0, respectively, for a given set of boundaries

$$\mathcal{L}_n = \{L_1, \ldots L_n\}, \quad L_1 < \ldots < L_n \tag{6}$$

We have that, for n even,

$$U_0(n, \mathcal{L}_n) = n + \frac{\alpha}{2}[\sum_{j=1}^{L_1} \Phi_{k_0}(g_j) + \sum_{L_1+1}^{L_2} \Phi_{k_1}(g_j) + \ldots + \sum_{L_n+1}^{N} \Phi_{k_0}(g_j)] =$$

$$= n + \frac{\alpha}{2}[G(L_1) - G(L_2) + \ldots - G(L_n) + \sum_{j=1}^{N} \Phi_{k_0}(g_j)]$$

$$U_1(n, \mathcal{L}_n) = n + \frac{\alpha}{2}[\sum_{j=1}^{L_1} \Phi_{k_1}(g_j) + \sum_{L_1+1}^{L_2} \Phi_{k_0}(g_j) + \ldots + \sum_{L_n+1}^{N} \Phi_{k_1}(g_j)] =$$

$$n + \frac{\alpha}{2}[-G(L_1) + \ldots + G(L_n) - G(N) + \sum_{j=1}^{N} \Phi_{k_0}(g_j)] \tag{7}$$

and for n odd,

$$U_0(n, \mathcal{L}_n) = n + \frac{\alpha}{2}[G(L_1) - G(L_2) + \ldots + G(L_n) - G(N) + \sum_{j=1}^{N} \Phi_{k_0}(g_j)]$$

$$U_1(n, \mathcal{L}_n) = n + \frac{\alpha}{2}[-G(L_1) + \ldots - G(L_n) + \sum_{j=1}^{N} \Phi_{k_0}(g_j)] \tag{8}$$

(Note that $\sum_j \Phi_{k_0}(g_j)$ does not depend on f).

Let $S_n^{(0)}, S_n^{(1)}$ be the sets of boundaries that minimize U_0 and U_1, respectively. Then, the optimal energy for a given n is:

$$U_n^* = min[U_0(n, S_n^{(0)}), U_1(n, S_n^{(1)})] \tag{9}$$

We will define S_n to be the corresponding optimal set of boundaries.

The determination of $S_n^{(k)}$ is an n–dimensional optimization problem. However, as we will show below, it is possible to decompose it into a sequence of one dimensional optimizations using a dynamic programming formulation. With this approach we also get, as a bonus, the solutions $S_1^{(k)}, \ldots, S_{n-1}^{(k)}$, $k \in \{0, 1\}$, and the corresponding optimal energies. If we set $n = N$, the solution to the original problem (3), $U^*(n^*, S_n\cdot)$ can then be found by a one dimensional search. This strategy, however, can be dramatically improved by the use of the following facts:

(i) We can reduce substantially the search space for the location of the optimal boundaries $L_j \in S_n\cdot$.

(ii) The sequences $\{U_1^*, U_3^*, \ldots\}$ and $\{U_2^*, U_4^*, \ldots\}$ are unimodal. This, together with the fact that the dynamic programming algorithm uses S_{j-1} to compute S_j provides us with an efficient stopping criterion for the computation of the sequence $\{S_1, \ldots, S_n\cdot\}$.

(iii) The expected value of n^* is usually small.

We will now describe the algorithm, and analyze each one of these facts.

1. Search Space for the Optimal Boundaries.

Let

$$\mathcal{P}_M = \{M_1, M_2, \ldots\} =$$

$$= \{j: \quad G(j-1) \leq G(j) \geq G(j+1), \text{ with } G(j-1) \neq G(j+1)\} \tag{10}$$

$$\mathcal{P}_m = \{m_1, m_2, \ldots\} =$$

$$= \{j: \quad G(j-1) \geq G(j) \leq G(j+1), \text{ with } G(j-1) \neq G(j+1)\} \tag{11}$$

(Conventionally we include $j = 1$ in $\mathcal{P}_M$, if $0 < G(1) \geq G(2)$, and include it in $\mathcal{P}_m$ if $0 > G(1) \leq G(2)$). We define the set $\mathcal{P}$ as

$$\mathcal{P} = \mathcal{P}_M \bigcup \mathcal{P}_m = \{P_1, \ldots, P_r\}$$

(Note that $\mathcal{P}$ corresponds to the set of places where the sequence $\{\Phi_{k_0}(g_j) - \Phi_{k_1}(g_j)\}$ changes sign).

In what follows, we will call the elements of $\mathcal{P}_M$, $\mathcal{P}_m$ and $\mathcal{P}$, the "maxima", "minima", and "critical points" of G, respectively.

Let $S_{n^\bullet +}$ ($S_{n^\bullet -}$) denote the subsets of $S_{n^\bullet}$ formed by those boundaries L_j whose corresponding term $G(L_j)$ has positive (negative) coefficient in $U^*_{n^\bullet}$, i.e., if

$$S_{n^\bullet} = S^{(k)}_{n^\bullet} = \{L_1, \ldots L_{n^\bullet}\},$$

then,

$$S_{n^\bullet +} = \{L_{1+k}, L_{3+k}, \ldots\}$$
$$S_{n^\bullet -} = S_{n^\bullet} - S_{n^\bullet +} \tag{12}$$

With these definitions, we have:

Theorem 1: Suppose that $\Phi_{k_0}(g_j) - \Phi_{k_1}(g_j) \neq 0$, for all j (a situation that will occur with probability 1 for most observation models). Then, $S_{n^\bullet +} \subseteq \mathcal{P}_m$ and $S_{n^\bullet -} \subseteq \mathcal{P}_M$.

To see why this is true, let f^{ML} denote the maximum likelihood estimate for f obtained by:

$$f^{ML}_j = \begin{cases} k_1, & \text{if } \Phi_{k_1}(g_j) > \Phi_{k_0}(g_j) \\ 0, & \text{otherwise} \end{cases}$$

and let f^* be the optimal estimate. Suppose that for some j we have, say, $L_j \in S_{n^\bullet +} - \mathcal{P}_m$. Suppose $L_j \in (P_k, P_{k+1})$, for some $P_k, P_{k+1} \in \mathcal{P}$. Clearly, either $P_k \in \mathcal{P}_m$ or $P_{k+1} \in \mathcal{P}_m$. Suppose, for definiteness that $P_k \in \mathcal{P}_m$.

If $P_k \notin S_{n^\bullet}$, the configuration $\{L_1, \ldots L_{j-1}, P_k, L_{j+1}, \ldots L_{n^\bullet}\}$ has lower energy than $S_{n^\bullet}$ (we decrease $\hat{U}$ without altering n), which is a contradiction. If $P_k \in S_{n^\bullet}$, then either

$$f^*((P_k, L_j)) \neq f^{ML}((P_k, L_j))$$
$$\text{or } f^*((L_j, P_{k+1})) \neq f^{ML}((L_j, P_{k+1}))$$

and so, we get a lower energy configuration by deleting L_j and either P_k or P_{k+1} (we decrease simultaneously n and $\hat{U}$). A similar argument can be used if $L_j \in [1, P_1)$ or $L_j \in (P_r, N]$.∎

This result means that we can use $\mathcal{P}$ to constrain the search space for the boundaries of each subproblem (i.e., for each fixed n), which now becomes:

For $n < |\mathcal{P}|$ *fixed, find* $S_n = \{L_1, \ldots L_n\}$ *with*

$$S_{n+} \subseteq \mathcal{P}_m \text{ and } S_{n-} \subseteq \mathcal{P}_M \tag{13}$$

such that $U(n, S_n) < U(n, L_n)$ *for all* $L_n \subseteq \mathcal{P}$.

Note that theorem 1 guarantees that the constrained and unconstrained solutions will coincide only for $n = n^*$, so that for $n \neq n^*$, S_n may, in general, be suboptimal.

2. Dynamic Programming (DP) Algorithm.

From equations (7) and (8), it is clear that, for any fixed n, the determination of the optimal (constrained) configurations $S_n^{(0)}, S_n^{(1)}$ is equivalent to the solution of the optimization problems:

For $S_n^{(0)}$:

$$\text{Minimize} \quad [G(L_1) - G(L_2) + \ldots]$$

with $L_1, L_3, \ldots \in \mathcal{P}_m$, and $L_2, L_4, \ldots \in \mathcal{P}_M$.

For $S_n^{(1)}$:

$$\text{Maximize} \quad [G(L_1) - G(L_2) + \ldots]$$

with $L_1, L_3, \ldots \in \mathcal{P}_M$, and $L_2, L_4, \ldots \in \mathcal{P}_m$.

Let us consider the maximization problems. Assume, for definiteness that the first critical point of G is a maximum, i.e., $M_1 < m_1$, and define the sequences:

$$D_1(k) = \sup_{i \geq k} G(M_i)$$

$$L_1(k) = \{\min L \ : \ G(M_L) = D_1(k)\}, \quad k = 1 \ldots |\mathcal{P}_M| \tag{14}$$

Clearly, $M_{L_1(1)}$ is the optimal location of the boundary for $n = 1$ (i.e., $S_1^{(1)} = \{M_{L_1(1)}\}$), and from $D_1(1)$ we can easily compute the corresponding energy.

We now define, for $j \geq 1$:

$$D_{2j}(k) = \sup_{i \geq k}\{D_{2j-1}(i+1) - G(m_i)\}$$

$$D_{2j+1}(k) = \sup_{i \geq k}\{D_{2j}(i) + G(M_i)\}$$

and

$$L_{2j}(k) = \{\min L \quad : \quad D_{2j}(k) = D_{2j-1}(L+1) - G(m_L)\}$$

$$L_{2j+1}(k) = \{\min L \quad : \quad D_{2j+1}(k) = D_{2j}(L) + G(M_L)\} \tag{15}$$

for $k = 1, \ldots, |\mathcal{P}_M| - j$. One can check that, for n odd,

$$S_n^{(1)} = \{M_{L_n(1)}, m_{L_{n-1}(L_n(1))}, \ldots, M_{L_1(L_2(\ldots(L_n(1))\ldots)}\} \tag{16}$$

and the optimal energy is:

$$U_1(n) = n + \frac{\alpha}{2}[-D_n(1) + \sum_j \Phi_{k_0}(g_j)] \tag{17}$$

For n even, we define:

$$D_1'(k) = \sup_{i \geq k}\{-G(m_i)\} \quad , \quad k = 1, \ldots, |\mathcal{P}_m|$$

$$L_1'(k) = \{\min L \quad : \quad D_1'(k) = -G(m_L)\}$$

$$D_{2j}'(k) = \sup_{i \geq k}\{D_{2j-1}'(i) + G(M_i)\} \quad , \quad k = 1, \ldots |\mathcal{P}_m| - j + 1$$

$$L_{2j}'(k) = \{\min L \quad : \quad D_{2j}'(k) = D_{2j-1}'(L) + G(M_L)\}$$

$$D_{2j+1}' \sup_{i \geq k}\{D_{2j}'(i+1) - G(m_i)\} \quad , \quad k = 1, \ldots |\mathcal{P}_m| - j$$

$$L_{2j+1}'(k) = \{\min L \quad : \quad D_{2j+1}'(k) = D_{2j}' - G(m_L)\} \tag{18}$$

and get:

$$S_n^{(1)} = \{M_{L_n'(1)}, \ldots m_{L_1'(L_n'(\ldots L_n'(1))\ldots)}\} \tag{19}$$

$$U_1(n) = n + \frac{\alpha}{2}[-D_n'(1) - G(N) + \sum_j \Phi_{k_0}(g_j)] \tag{20}$$

For the minimization problems, that is, for the computation of $S_n^{(0)}$, assuming again that $M_1 < m_1$, we have, for n even:

$$d_1(k) = \inf_{i \geq k} \{-G(m_i)\}$$

$$l_1(k) = \{\min l \quad : \quad d_1(k) = -G(m_l)\}$$

and for $j \geq 1$,

$$d_{2j}(k) = \inf_{i \geq k} \{d_{2j-1}(i) + G(M_i)\}$$

$$l_{2j}(k) = \{\min l \quad : \quad d_{2j}(k) = d_{2j-1}(l) + G(M_l)\}$$

$$d_{2j+1}(k) = \inf_{i \geq k} \{d_{2j}(i+1) - G(m_i)\}$$

$$l_{2j+1}(k) = \{\min l \quad : \quad d_{2j+1}(k) = d_{2j}(l+1) - G(m_l)\} \tag{21}$$

with k varying in the appropriate range. The solutions are:

$$S_n^{(0)} = \{M_{l_n(1)}, \ldots, m_{l_1(l_2(\ldots(l_n(1))\ldots)}\}$$

$$U_0(n) = n + \frac{\alpha}{2}[d_n(1) + \sum_j \Phi_{k_0}(g_j)] \tag{22}$$

For n odd:

$$d_1'(k) = \inf_{i \geq k} \{G(M_i)\}$$

$$d_{2j}'(k) = \inf_{i \geq k} \{d_{2j-1}'(i+1) - G(m_i)\}$$

$$d_{2j+1}'(k) = \inf_{i \geq k} \{d_{2j}'(i) + G(M_i)\} \tag{23}$$

with the corresponding definitions for $l_j'(k)$. The solutions are:

$$S_n^{(0)} = \{M_{l_n'(i)}, \ldots, M_{l_1',(\ldots(l_n',(1))\ldots)}\}$$

$$U_0(n) = n + \frac{\alpha}{2}[d_n'(1) - G(N) + \sum_j \Phi_{k_0}(g_j)] \tag{24}$$

The case for which $m_1 < M_1$ is treated in a similar way.

The recursions (15), (18), (21) and (23), together with equations (9) and (10), allow us to compute the sequences $\{S_1, S_2, \ldots\}$ and $\{U_1^*, U_2^*, \ldots\}$ using only one

dimensional optimizations. We now turn to the problem of determining the optimal value n^* for the number of boundaries.

3. Stopping Criterion.

In this section we prove the following:

Theorem 2. Suppose that every (constrained) optimal configuration in the sequence $\{S_1, S_2, \ldots\}$ is unique (i.e., for every n, if $S_n' \neq S_n$, and $S_n' \subseteq P$, then $U(n, S_n') > U_n^*$) and that for some n, $U_{n+2}^* > U_n^*$. Then, $U_{n+2k}^* > U_n^*$, for all $k \geq 1$.

This result will provide us with an efficient stopping criterion for the dynamic programming recursions described in the previous section; since the first local minima for the subsequences $\{U_1^*, U_3^*, \ldots\}$ and $\{U_2^*, U_4^*, \ldots\}$ are the global ones, we can terminate the computations once we have found them.

To prove the theorem, we will need the following lemmas:

Lemma 1. Let $S_k = \{L_1, \ldots, L_k\}$ and $S_{k+2} = \{L_1', \ldots L_{k+2}'\}$ be the optimal boundaries (with corresponding configurations f_k and f_{k+2}) for $n = k$ and $n = k+2$, respectively. Suppose that $k+2 < |P|$. Then, $S_k \subseteq S_{k+2}$ (i.e., S_{k+2} is a refinement of S_k), provided S_k is unique.

Proof:

We will assume that for some j, $L_j \in S_k - S_{k+2}$, and arrive at a contradiction. We consider three cases:

Case 1: Suppose that for some i,

$$[L_i', L_{i+1}'] \bigcap S_k = \emptyset$$

In this case, we claim that we can find some index p such that

$$[L_p', L_{p+1}'] \bigcap S_k = \emptyset$$

and

$$f_{k+2}((L_p', L_{p+1}']) \neq f_k((L_p', L_{p+1}'])$$

Suppose that this is not the case. Then, L_i', L_{i+1}' are the only elements of S_{k+2} in some interval (L_j, L_{j+1}) (or in one of the extreme intervals $[1, L_1), (L_k, N]$) and

$$f_{k+2}((L_i', L_{i+1}']) \neq f_k((L_i', L_{i+1}])$$

Suppose

$$[L_i', L_{i+1}] \subseteq (L_j, L_{j+1})$$

By condition (13), we have that $L_j \neq L_{i-1}'$ (otherwise, L_j would be a local maximum and minimum of G at the same time). But then, since S_k is optimal, we can find a configuration with $k+2$ boundaries whose energy is lower than that of S_{k+2}, by moving L_i' to L_j (or L_{i+1}' to L_{j+1}), which contradicts the optimality of S_{k+2}. A similar argument holds if

$$[L_i', L_{i+1}'] \subseteq [1, L_1) \text{ or } (L_k, N]$$

This proves our claim.

So, suppose that

$$[L_p', L_{p+1}'] \bigcap S_k = \emptyset$$

and

$$f_{k+2}((L_p', L_{p+1}']) \neq f_k((L_p', L_{p+1}']).$$

Form

$$S_k' = \{L_1', \ldots, L_{p-1}', L_{p+2}', \ldots, L_{k+2}'\}$$

and let f_k' be the corresponding configuration, chosen in such a way that $f_k'(1) = f_k(1)$ (and therefore, $f_k'([L_p', L_{p+1}']) = f_k([L_p', L_{p+1}'])$).

Let $\Delta\hat{U}$ be the change in $\hat{U}$ (see eq. (4)) associated with setting:

$$f([L_p', L_{p+1}']) = f_{k+2}([L_p', L_{p+1}']).$$

We have that

$$\hat{U}(S_{k+2}) = \hat{U}(S_k^{'}) + \Delta\hat{U}.$$

Now, we put:

$$S_{k+2}^{'} = \{L_1, \ldots, L_j, L_p^{'}, L_{p+1}^{'}, \ldots, L_k\}.$$

Since S_k is optimal, we have that:

$$\hat{U}(S_{k+2}) = \hat{U}(S_k^{'}) + \Delta\hat{U} > \hat{U}(S_k) + \Delta\hat{U} = \hat{U}(S_{k+2}^{'}),$$

which contradicts the optimality of S_{k+2}.

Case 2:

$$([1, L_1^{'}] \bigcup [L_{k+2}^{'}, N]) \bigcap S_k = \emptyset$$

Suppose that $L_1^{'} \in [1, L_1^{'})$. We must have

$$f_{k+2}([1, L_1^{'})) \neq f_k([1, L_1^{'}))$$

Otherwise, if $L_1 = L_2^{'}$, condition (13) generates a contradiction; if $L_1 > L_2^{'}$, we are in case 1, and if $L_1 < L_2^{'}$, S_{k+2} is not optimal, since we get a lower energy configuration by moving $L_1^{'}$ to L_1.

So,

$$f_{k+2}([1, L_1^{'}]) \neq f_k([1, L_1^{'}]).$$

By a similar argument, we get that

$$f_{k+2}([L_{k+2}^{'}, N]) \neq f_k([L_{k+2}^{'}, N]).$$

Now, proceeding as in case 1, we form:

$$S_k^{'} = \{L_2^{'}, \ldots, L_{k+1}^{'}\}$$

and let $f_k^{'}$ be the corresponding configuration, chosen in such a way that $f_k^{'}(1) = f_k(1)$

Let $\Delta\hat{U}$ be the change in $\hat{U}$ associated with setting:

$$f([L_1', L_2']) = f_{k+2}([L_1', L_2']) \quad \text{and}$$

$$f([L_{k+1}', L_{k+2}']) = f_{k+2}([L_{k+1}', L_{k+2}'])$$

so that

$$\hat{U}(S_{k+2}) = \hat{U}(S_k') + \Delta\hat{U}.$$

Now, we form:

$$S_{k+2}' = \{L_1', L_1, \ldots, L_k, L_{k+2}'\},$$

Since S_k is optimal, we have that:

$$\hat{U}(S_{k+2}) = \hat{U}(S_k') + \Delta\hat{U} > \hat{U}(S_k) + \Delta\hat{U} = \hat{U}(S_{k+2}'),$$

which again contradicts the optimality of S_{k+2}.

Case 3:

$$\text{For all } i, \quad [L_i', L_{i+1}'] \bigcap S_k \neq \emptyset,$$

$$\text{and} \quad ([1, L_1'] \bigcup [L_{k+2}', N]) \bigcap S_k \neq \emptyset \qquad (*)$$

To make (*) hold, we must be able to place k boundaries in $k+3$ (ovelapping) closed intervals, without omitting any interval. Moreover, since condition (13) must hold, we cannot put $L_j = L_i'$ and $L_{j+1} = L_{i+2}'$ for any i, j. But this is impossible; so, our proof is finished. ∎

Lemma 2. Let $\Delta\hat{U}_k = \hat{U}(S_k) - \hat{U}(S_{k+2})$. Then, $\Delta\hat{U}_k \leq \Delta\hat{U}_{k-2}$, for all $k \in [3, |P|-2]$.

Proof:

Consider the optimal configurations S_k, S_{k+2}, S_{k+4}, and suppose that $\Delta\hat{U}_{k+2} > \Delta\hat{U}_k$. Using lemma 1, let

$$S_k = \{L_1, \ldots, L_k\};$$

$$S_{k+2} = \{L_1, \ldots, L_1', L_2', \ldots, L_k'\}.$$

By condition (13) and lemma 1, there are only two valid forms for S_{k+4}. We consider each case separately:

Case 1: S_{k+4} is of the form:

$$S_{k+4} = \{L_1, \ldots, L_p, L_1', L_2', L_{p+1} \ldots, L_1'', L_2'', \ldots\}$$

(i.e., the refinements corresponding to S_{k+2} and S_{k+4} are disjoint).

Then, for

$$S_{k+2}' = \{L1, \ldots, L_p, L_{p+1}, \ldots, L_1'', L_2'', \ldots\},$$

we have

$$\hat{U}(S_{k+2}') = \hat{U}(S_k) - \Delta\hat{U}_{k+2} < \hat{U}(S_k) - \Delta\hat{U}_k = \hat{U}(S_{k+2}),$$

which is a contradiction.

Case 2: S_{k+4} is of the form:

$$S_{k+4} = \{L_1, \ldots, L_j, L_1', L_1'', L_2'', L_2', \ldots\}$$

(i.e., S_{k+4} is a subrefinement of the refinement introduced by S_{k+2}).

Let

$$a = -\hat{U}(\{L_1, \ldots, L_j, L_1', L_1'', L_{j+1}, \ldots\} + \hat{U}(S_k)$$

$$b = \hat{U}(\{L_1, \ldots, L_j, L_1'', L_2'', L_{j+1}, \ldots\} - \hat{U}(S_k)$$

$$c = -\hat{U}(\{L_1, \ldots, L_j, L_2'', L_2', L_{j+1}, \ldots\} + \hat{U}(S_k)$$

We have that,

$$\Delta\hat{U}_k = a + c - b$$

$$\Delta\hat{U}_{k+2} = b.$$

By assumption,

$$b > a + c - b$$

and therefore,

$$\Delta\hat{U}_k = a + c - b < \frac{a+c}{2} \leq \max(a, c).$$

Now, let S'_{k+2} be formed from S_k by the refinement:

$$\begin{cases} L'_1, L''_1, & \text{if } a = \max(a, c) \\ L''_2, L'_2, & \text{if } c = \max(a, c) \end{cases}$$

Then,

$$\hat{U}(S'_{k+2}) = \hat{U}(S_k) - \max(a, c) < \hat{U}(S_k) - \Delta\hat{U}_k = \hat{U}(S_{k+2}),$$

which is a contradiction.∎

Now we prove theorem 2:

Suppose $U^*_{k+2} > U^*_k$. Then,

$$k + 2 + \frac{\alpha}{2}\hat{U}(S_{k+2}) > k + \frac{\alpha}{2}\hat{U}(S_k)$$

now, by lemma 2 we have:

$$U^*_{k+4} = k + 4 + \frac{\alpha}{2}\hat{U}(S_{k+4}) = k + 4 + \frac{\alpha}{2}(\hat{U}(S_k) - \Delta\hat{U}_{k+2}) >$$

$$> k + 2 + \frac{\alpha}{2}(\hat{U}(S_k) - \Delta\hat{U}_{k+2}) > k + 2 + \frac{\alpha}{2}(\hat{U}(S_k) - \Delta\hat{U}_k) =$$

$$= k + 2 + \frac{\alpha}{2}\hat{U}(S_{k+2}) = U^*_{k+2} \quad ∎$$

4. Expected Value of n^*.

First, we compute the (prior) probability density function $p(n)$ for the number n of odd bonds in the original field f.

Let $N_b = N - 1$ be the total number of bonds. We can rewrite equation (1) as:

$$P(\omega = f) = \frac{1}{Z} e^{\frac{1}{\alpha}(N_b - 2n)} \tag{25}$$

The total number of configurations compatible with a given n is $2C_n^{N_b}$, and so,

$$p(n) = \frac{2C_n^{N_b} \exp[\frac{1}{\alpha}(N_b - 2n)]}{\sum_{k=0}^{N_b} C_k^{N_b} \exp[\frac{1}{\alpha}(N_b - 2k)]} =$$

$$= C_n^{N_b} \left(\frac{e^{1/\alpha}}{e^{1/\alpha} + e^{-1/\alpha}} \right)^{N_b - n} \left(\frac{e^{-1/\alpha}}{e^{1/\alpha} + e^{-1/\alpha}} \right)^{n} \tag{26}$$

which is a binomial distribution. Therefore,

$$E[n] = N_b \left(\frac{e^{-1/\alpha}}{e^{1/\alpha} + e^{-1/\alpha}} \right)$$

$$\text{Var}[n] = N_b \left(\frac{1}{e^{1/\alpha} + e^{-1/\alpha}} \right) \tag{27}$$

We note that as $\alpha \uparrow \infty$, $E[n] \uparrow N_b/2$, and as $\alpha \downarrow 0$, $E[n] \downarrow 0$ (and $\text{var}[n] \downarrow 0$) exponentially fast. This means that if the natural temperature of the system is not too high, we can expect that n^*, the MAP estimate for n, to be relatively small.

5. Relation to Multiscale Filtering.

An interesting characteristic of the DP formulation is that the solutions to each of the subproblems (which in fact correspond to a minimization of $\hat{U}$ (eq. (4)) are independent of the value of the parameter α. The role of this parameter is to determine the number of regions (n^*) that will be present in the optimal configuration. In this sense, it can be regarded as a "scale" parameter that controls the aggregation of the subregions into larger units, and the algorithm can be used to produce multiscale descriptions (in the style of the "fingerprints" treated in Witkin (1983) and Yuille and Poggio (1983)) of the input signals. (Several other heuristic solutions to this problem have been proposed. See, for example, Blumenthal et al., 1977; Prazdny, 1982 and Pavlidis, 1973)

If we interpret the algorithm in this way, it becomes natural to ask whether a family of linear operators can do the same job in a much efficient way. Let us formulate this question in more precise form (in what follows, we will consider a "continuous time" problem obtained from the original one as a limit when $N \uparrow \infty$

(provided that the observations are different from 0 only in a finite interval), since it simplifies the notation. It should be clear that the same arguments apply to the discrete case).

Consider a family of filters $\{F_L\}$ with the following properties:

(i) Each $F_L(x)$ is a symmetric and non-negative function of x.

(ii) For each L, $F_L(x)$ is a decreasing function of $|x|$, and $F_L(x) \downarrow 0$ as $|x| \uparrow \infty$ fast enough, so that F_L can be approximated by a function with finite support.

(iii) All the filters are normalized:

$$\int_{-\infty}^{\infty} F_l(x)dx = 1, \quad \text{for all } L.$$

(iv) The filters become sharper as $L \downarrow 0$:

$$\int_0^b F_{L_2}(x)dx < \int_0^b F_{L_1}(x)dx$$

implies that $L_2 > L_1$

Particular examples of acceptable families are:

(i) The family of rectangular boxes B_L:

$$B_L(x) = \begin{cases} \frac{1}{2L}, & \text{if } |x| \leq L \\ 0, & \text{otherwise} \end{cases}$$

(ii) The family of Gaussian Kernels:

$$G_L(x) = \frac{1}{\sqrt{2\pi}L} \exp[-\frac{x}{2L^2}]$$

Suppose we convolve the function $g(x) - \frac{1}{2}$ ($g(x)$ is a continuous time approximation to the observations) with a set of filters from the family $\{F_L\}$. If we start with L large enough, the function

$$h_L = (g - \frac{1}{2}) \quad * \quad F_L$$

will be practically constant, and therefore, it will have no zeroes. As we decrease L, zero crossings of h_L will begin to appear. To each of these zero crossings, we will associate a boundary, and form the configurations $\hat{S}_1, \hat{S}_2, \ldots$ with $1, 2, \ldots$ boundaries respectively, that correspond to the first, first two, etc. zero crossings of h_L (we are ignoring, at this point, the question of the precise localization of these boundaries. With additional contraints on the family $\{F_L\}$, it is possible, in principle, to localize them by decreasing L in a continuous fashion, and then tracing the position of each zero crossing to the finest ($L = 0$) level; see Yuille and Poggio (1983). For the moment, let us assume that we can identify the zero crossings of $g - \frac{1}{2}$ that correspond to those of h_L, for all L).

The question that we ask is the following:

If $S_1, S_2, \ldots$ are the optimal boundary configurations produced by the DP algorithm,is it true that

$$S_k = \hat{S}_k$$

for all k?

As we now show, this is not the case.

Consider the signal $g(x)$ defined by:

$$g(x) = 1 \quad ,$$

$$\text{for } x \in [l_1, l_1 + 2a] \bigcup [l_2, l_2 + 2b] \bigcup [l_2 + 4b, l_2 + 6b] \bigcup$$

$$\bigcup [l_2 + 8b, l_2 + 10b] \bigcup [l_2 + 12b, l_2 + 14b] \bigcup [l_2 + 16b, l_2 + 18b] \quad ,$$

and $g(x) = 0$, otherwise. Here, l_1, l_2, a and b are some positive numbers chosen in such a way that, if L_0 is the starting L, we take $l_2 - l_1 - a >> L_0$, so that, by property (ii), there is no interaction between $[l_1, l_1 + a]$ and $[l_2, l_2 + 18b]$ (see figure 4.B.1).

Suppose that the zero crossings corresponding to $[l_1, l_1 + a]$ appear first (as a single double zero) at $L = L_1$, and those corresponding to $[l_2, l_2 + 18b]$ at $L = L_2$.

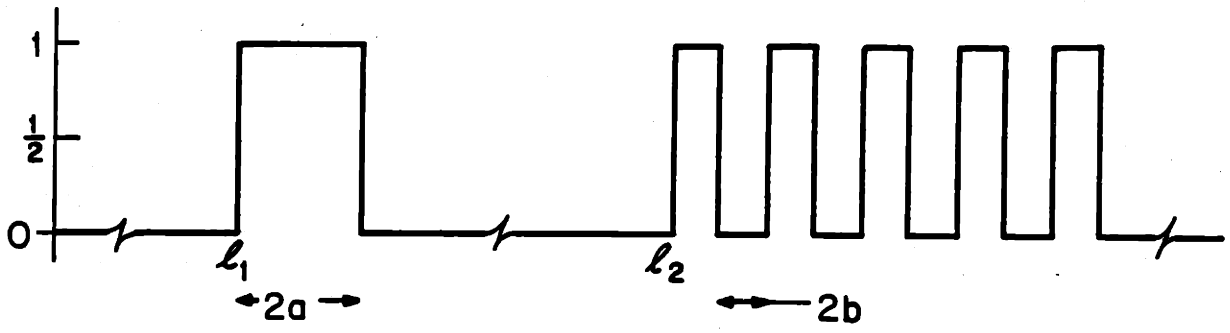


Figure 4.B.1. (See text).

Then,

$$\int_0^a F_{L_1}(x)dx = \int_a^\infty F_{L_1}(x)dx \tag{28}$$

$$\int_0^b F_{L_2}(x)dx + \int_{3b}^{5b} F_{L_2}(x)dx + \int_{7b}^{9b} F_{L_2}(x)dx =$$

$$= \int_b^{3b} F_{L_2}(x)dx + \int_{5b}^{7b} F_{L_2}(x)dx + \int_{9b}^{\infty} F_{L_2}(x)dx \tag{29}$$

Now, for $a > b$, we have:

$$\hat{U}(\{l_1, l_2\}) = 10b$$

$$\hat{U}(\{l_3, l_4\}) = 8b + 2a > \hat{U}(\{l_1, l_2\})$$

and therefore, $S_2 = \{l_1, l_2\}$.

We claim that we can find some a, b with $a > b$ such that

$$\int_0^a F_{L_2}(x)dx < \int_a^\infty F_{L_2}(x)dx$$

If this is true, we find, using (28) and conditions (iii) and (iv), that it implies that $L_2 > L_1$, and therefore, $\hat{S}_2 = \{l_3, l_4\}$.

We now prove our claim:

Let $a = b + \frac{\epsilon}{2}$, where we choose ϵ so that

$$\int_b^{b+\epsilon/2} F_{L_2}(x)dx = \int_{3b}^{5b} F_{L_2}(x)dx \tag{30}$$

(property (ii) guarantees that we can find such ϵ). From (29),

$$\int_b^{\infty} F_{L_2}(x)dx = \int_0^b F_{L_2}(x)dx + 2\int_{3b}^{5b} F_{L_2}(x)dx + 2\int_{7b}^{9b} F_{L_2}(x)dx$$

and from (30),

$$\int_a^{\infty} F_{L_2}(x)dx = \int_{b+\epsilon/2}^{\infty} F_{L_2}(x)dx = \int_b^{\infty} F_{L_2}(x)dx - \int_b^{b+\epsilon/2} F_{L_2}(x)dx =$$

$$= \int_0^b F_{L_2}(x)dx + \int_b^{b+\epsilon/2} F_{L_2}(x)dx + 2\int_{7b}^{9b} F_{L_2}(x)dx = \int_0^{b+\epsilon/2} F_{L_2}(x)dx + 2\int_{7b}^{9b} F_{L_2}(x)dx >$$

$$> \int_{7b}^{9b} F_{L_2}(x)dx = \int_0^a F_{L_2}(x)dx \quad \blacksquare$$

This result does not mean, of course, that families of linear filters cannot be used for producing useful multiscale descriptions of signals; it only means that these descriptions cannot, in general, be considered as MAP estimates of MRF models.

6. Continuous Valued Fields.

In this section we present a related problem which can, in principle, be solved using the DP approach, although, as we will see, in a less efficient way.

Let us consider the problem of estimating a piecewise constant signal corrupted by additive white Gaussian noise. We model the signal $\{f_i\}$ as a MRF with potential

$$V(f_i, f_{i+1}) = \begin{cases} -1, & \text{if } f_i = f_{i+1} \\ 1, & \text{otherwise} \end{cases} \tag{31}$$

and global states distributed according to:

$$P(F = f) = \frac{1}{Z} \exp[-\frac{1}{\alpha} \sum_{i=1}^{N-1} V(f_i, f_{i+1})]$$

The observations are given by:

$$g_i = f_i + n_i$$

where n is a white Gaussian process. The Bayesian (MAP) estimate for f is again found by minimizing eq.(4):

$$U(f) = n + \frac{\gamma}{2}\hat{U}$$

$$\hat{U} = \sum_{i+1}^{N}(f_i - g_i)^2$$

where n is the number of places where $f_i \neq f_{i+1}$, and $\gamma = \frac{\alpha}{2\sigma^2}$. Note that in this case, f_i is not restricted to $\{0, 1\}$, but can take any real value.

Proceeding as we did in section 2, we consider the sequence of subproblems obtained by putting $n = 0, 1, 2, \ldots$.

For any fixed n, $\hat{U}$ will depend only on the n integer variables that correspond to the location of the boundaries between regions of constant f, since given these boundaries $L = \{L_1, \ldots L_n\}$, the optimal estimate for f on any interval $(L_i, L_{i+1}]$ (we put $L_0 = 1$ and $L_{n+1} = N$) is:

$$f((L_i, L_{i+1}]) = \frac{1}{L_{i+1} - L_i} \sum_{j=L_i+1}^{L_{i+1}} g_j.$$

If we define $G_{k,l}$ (for $k < l$) as:

$$G_{k,l} = (1 - 2(l-k))\left(\frac{1}{l-k} \sum_{i=k+1}^{l} g_i\right)^2 \tag{32}$$

We get that:

$$\hat{U}(L_n) = \sum_{i=1}^{N} g_i^2 + \sum_{j=1}^{n+1} G_{L_{j-1}, L_j} \tag{33}$$

(note that $\sum g_i^2$ is a constant for a given set of observations). Using dynamic programming principles, we can now write the recursions:

$$F_0(k) = G_{k,N} \quad , \quad k = 0, \ldots N-1$$

$$F_{j+1}(k) = \inf_{i>k} \{G_{k,i} + F_j(i)\} \quad , \quad k = 0, \ldots N-j-1$$

$$L_{j+1}(k) = \{L \quad : \quad G_{k,L} + F_j(L) = F_{j+1}(k)\} \tag{34}$$

The optimal solution, for each given n is:

$$S_n = \{L_n(0), L_{n-1}(L_n(0)), \ldots, L_1(L_2(\ldots(L_n(0))\ldots)\}$$

and the corresponding energy,

$$U(n, S_n) = n + \frac{\alpha}{2}[\sum_{i=1}^{N} g_i^2 + F_n(0)] \tag{35}$$

The solution to our problem will be S_{n^*}, where:

$$U(n^*, S_{n^*}) = \inf_n \{U(n, S_n)\} \tag{36}$$

Unfortunately, in this case we cannot guarantee the unimodality of any subsequence of $\{\hat{U}(S_n)\}$ (although we believe that the sequence will be unimodal in many cases) and so, (36) has to be computed, in principle, by an exhaustive one dimensional search. Another unpleasantness is that, unlike the binary case, the search space for the variables L_i cannot be reduced in any obvious way.

Appendix 4.C

CONSISTENCY CONDITION FOR THE MPM ESTIMATOR

In this appendix we present a proof for the consistency condition (given by equation (11) of chapter 4) satisfied by the MPM estimator of a binary, two–dimensional Ising net:

Theorem: Let $P(f,g)$ be the posterior distribution corresponding to the estimation of the first order, binary MRF f from the observations g which are obtained as the ouput of a binary symmetric channel:

$$P(f,g) = \frac{1}{Z}\exp[-\sum_{i,j} V(f_i,f_j) - \gamma\sum_{i}(1-\delta(f_i-g_i))]$$

Let $\hat{f}$ be the MPM estimator for f. Then, for every site i,

$$\sum_{f:f_i=\hat{f}_i} P(f,g) > \sum_{f:f_i\neq\hat{f}_i} P(f,g)$$

implies that:

$$\sum_{f:f_i=\hat{f}_i} P(f,\hat{f}) > \sum_{f:f_i\neq\hat{f}_i} P(f,\hat{f})$$

Proof:

Let:

$$f_j^{(i)} = \begin{cases} f_j, & j\neq i \\ \hat{f}_i, & j=1 \end{cases}$$

$$g_j^{(i)} = \begin{cases} g_j, & j\neq i \\ \hat{f}_i, & j=1 \end{cases}$$

$$h_j^{(i)} = \begin{cases} f_j, & j\neq i \\ 1-\hat{f}_i, & j=1 \end{cases}$$

1) We first prove that for all i:

$$\sum_f P(f^{(i)}, g) > \sum_f P(h^{(i)}, g)$$

implies that:

$$\sum_f P(f^{(i)}, g^{(i)}) > \sum_f P(h^{(i)}, g^{(i)}) \quad :$$

Suppose that $g \neq g^{(i)}$ (otherwise, the above is obviously true). For any fixed f, we have that:

$$P(f^{(i)}, g) - P(h^{(i)}, g) =$$

$$= K\{\exp[-\sum_{j \in N_i} V(\hat{f}_i, f_j) - \gamma] - \exp[\sum_{j \in N_i} V(1 - \hat{f}_i, f_j)]\}$$

and

$$P(f^{(i)}, g^{(i)}) - P(h^{(i)}, g^{(i)}) =$$

$$= K\{\exp[-\sum_{j \in N_i} V(\hat{f}_i, f_j)] - \exp[\sum_{j \in N_i} V(1 - \hat{f}_i, f_j) - \gamma]\}$$

Where K is a constant. Since $\gamma > 0$, this implies that:

$$P(f^{(i)}, g) - P(h^{(i)}, g) \leq P(f^{(i)}, g^{(i)}) - P(h^{(i)}, g(i))$$

so that

$$\sum_f P(f^{(i)}, g^{(i)}) - P(h^{(i)}, g(i)) \geq \sum_f P(f^{(i)}, g) - P(h^{(i)}, g) > 0$$

2) Let $r_i = 1 - \hat{f}_i$. We now prove that if:

$$\sum_{f: f_i = \hat{f}_i} P(f, g) > \sum_{f: f_i = r_i} P(f, g)$$

then,

$$\sum_{f: f_i = \hat{f}_i} P(f, g^{(j)}) > \sum_{f: f_i = r_i} P(f, g^{(i)})$$

for all j.

For $i = j$, part (1) applies, and for $g^{(j)} = g$, the assertion is obviously true, so suppose $i \neq j$ and $g^{(j)} \neq g$. We have:

$$P(f^{(i)}, g) - P(h^{(i)}, g) =$$

$$= K_1\{\exp[-\sum_{j \in N_i} V(\hat{f}_i, f_j)] - \exp[\sum_{j \in N_i} V(1 - \hat{f}_i, f_j) - \gamma]\}$$

$$P(f^{(i)}, g^{(j)}) - P(h^{(i)}, g^{(j)}) = K_1\{\exp[-\sum_{j \in N_i} V(\hat{f}_i, f_j)] -$$

$$\exp[\sum_{j \in N_i} V(1 - \hat{f}_i, f_j) - \gamma]\} \exp[-\gamma(1 - 2(f_j - g_y))^2] \geq$$

$$\geq e^{-\gamma}(P(f^{(i)}, g) - P(h^{(i)}, g)$$

for some constant K_1, so that

$$\sum_f P(f^{(i)}, g^{(j)}) - P(h^{(i)}, g^{(j)}) \geq e^{-\gamma} \sum_f P(f^{(i)}, g) - P(h^{(i)}, g) > 0$$

The theorem is now proved by assuming that

$$\sum_{f: f_i = \hat{f}_i} P(f, g) > \sum_{f: f_i \neq \hat{f}_i} P(f, g)$$

and succesively replacing g_i by $\hat{f}_i$, for $i = 1, 2, \ldots$and using (1) and (2) to show that the corresponding inequalities hold at each step.

Chapter 5

RECONSTRUCTION OF PIECEWISE CONTINUOUS FUNCTIONS

1. Introduction.

In this chapter we will illustrate the application of local spatial interaction models and estimation techniques that we have described to the solution of the general reconstruction problem that we introduced in chapter 1. To make this discussion more specific, we will consider a particular instance of this problem: the reconstruction of piecewise continuous functions from noisy observations taken at sparse locations.

In this reconstruction, it will be important not only to interpolate smooth patches over uniform regions, but to locate and preserve the discontinuities that bound these regions, since very often they are the most important parts of the function. They may represent object boundaries in vision problems (such as image segmentation; depth from stereo; shape from shading; structure from motion, etc.); geological faults in geophysical information processing, etc.

The most successful approaches to this problem (see Terzopoulos (1984)) consist of, first, interpolating an everywhere smooth function over the whole domain; then, applying some kind of discontinuity detector (followed by a thresholding operation) to try to find the significant boundaries, and finally, to re–interpolate smooth patches over the continuous subregions.

The results that have been obtained with this technique, however, are not completely satisfactory. The main problem is that the task of the discontinuity detector is hindered by the previous smooth interpolation operation. This becomes critical when the observations are sparsely located, since in this case, the discontinuities

may be smeared in the interpolation phase to such a degree that it may become impossible to recover them in the detection phase.

One way around this difficulty is to perform the boundary detection and interpolation tasks *at the same time.* In the method we will present, this is done by using a Bayesian approach, and including in the posterior distribution our prior knowledge about the smoothness of the function and about the geometry of the discontinuities, as well as the information provided by the observations. Before describing how this is done, let us formulate the problem in a more precise way.

Consider a region Ω of the plane which is formed by a number of subregions separated by boundaries which are known to be piecewise smooth curves. Suppose that within each of these subregions, some property f (in what follows, we will refer to f as "depth") varies in a smooth fashion, presenting, at the same time, abrupt jumps across most of the boundaries. Suppose also that we have measurements for the values of f at some discrete set of sites S; these measurements will, in general, be corrupted by some form of noise.

Our problem is then to estimate the values of f on some finite lattice of points $L \subseteq \Omega$, and to find the position of the boundaries, using all the available information in an optimal way.

2. Posterior Distribution.

To apply the general recontruction algorithms developed in chapter 3 to this problem, we need to cast it in probabilistic terms. The main issue here is the representation of the concept of "piecewise continuity" in the form of a prior Gibbs distribution in a meaningful way.

This could be done, for example, by modeling the function as a first order, continuous valued MRF with nearest neighbor potentials given by:

$$V(f_i, f_j) = \begin{cases} (f_i - f_j)^2, & \text{if } |f_i - f_j| < a \text{ and } |i-j| = 1 \\ b, & \text{if } |f_i - f_j| \geq a \text{ and } |i-j| = 1 \\ 0, & \text{otherwise} \end{cases}$$

where a and b are positive constants such that $b \geq a^2$, and for every pair of neighboring sites i, j, $|f_i - f_j| < a$ if both i and j lie in the same smooth patch, and $|f_i - f_j| > a$, otherwise.

This scheme, however, has the disadvantage of not allowing for the explicit modeling of prior knowledge about the geometry of the curves that bound the smooth patches (the fact that they should be piecewise smooth curves, for example). A more flexible construction involves the use of two coupled MRF models: one to represent the function (the "surface") itself, and another to model the curves where the field is discontinuous. A coupled model of this kind was first used by Geman and Geman (1984) in the context of the restoration of piecewise constant images. We will now describe their work in detail, and define a related model that can be used for our problem.

2.1. Coupled Line and Depth Models.

In Geman and Geman's work, the intensity of the images is modeled using a first order MRF with generalized Ising potentials (see chapter 4). The boundaries between constant regions are modeled using a "line process" l, which is a MRF whose associated random variables are located at the sites of the dual lattice of lines that connect the sites of the original intensity lattice (see figure 12). These variables may be binary (indicating the presence or absence of a boundary between two pixels), or may take more values to indicate the orientation of the boundary as well. In both cases, their function is to decouple adjacent pixels, reducing the total energy if the intensities of these pixels are different.

This is done by modifying the prior energy function; the new expression is:

$$U_0(f, l) = \sum_i \sum_{j \in N_i} V_f(f_i, f_j, l_{ij}) + \sum_{C_l} V_{C_l}(l) \qquad (1)$$

where

$$V_f(f_i, f_j, l_{ij}) = \begin{cases} 0, & \text{if } l_{ij} \text{ is "on"} \\ V(f_i, f_j), & \text{otherwise} \end{cases}$$

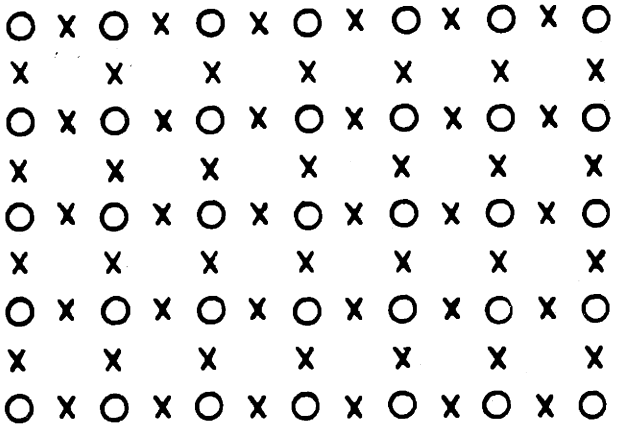


Figure 12. Dual lattice of line elements (sites denoted by x)

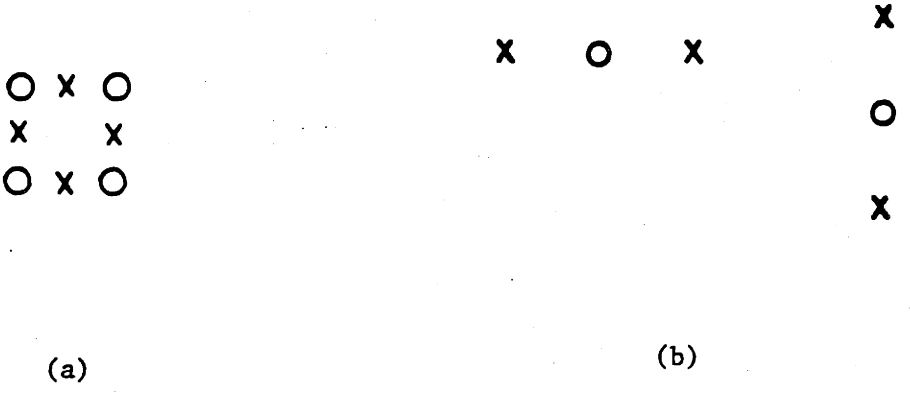


Figure 13. (a) Cliques for the line process used by Geman and Geman. (b) Additional cliques used to prevent sharp turns.

V is defined in equation (1) of chapter 4:

$$V(f_i, f_j) = \begin{cases} -1, & \text{if } |i-j| = 1 \text{ and } f_i = f_j \\ 1, & \text{if } |i-j| = 1 \text{ and } f_i \neq f_j \\ 0, & \text{otherwise} \end{cases}$$

l_{ij} is the line element between sites i and j, and the line potentials V_{C_l} have as supports cliques of size 4, such as the one shown in Fig. 13–a. Every line element (except at the boundaries of the lattice) belongs to 2 such cliques. The values of the potentials associated with each possible configuration of lines within a clique must be specified. Thus, for example, if straight horizontal and vertical boundaries are likely to be present, a binary process, with potential values as those of Fig. 14 is used (rotational invariance is assumed). In more general situations (such as piecewise smooth boundaries), we may use different values for the potentials, or we may allow more states for the line elements, corresponding to different orientations, augmenting consequently the table of values for the potentials.

2.2. Models for Piecewise Continuous Functions.

The model we have described can be adapted to our problem by modifying the choice of the potentials and the neighborhood structure of the coupled MRF's. Specifically, the following modifications are needed:

1. Since in our case the observations are sparse, it becomes necessary to expand the size of the neighborhoods of the line field, to prevent the formation of "thick" boundaries between the smooth patches (i.e., adjacent, parallel segments of active lines in these regions). In particular, we propose that the dual lattice be 8–connected, with non–zero potentials for the cliques of the form illustrated in figure 13 (a) and (b). The inclusion of the cliques of figure 13–b has the additional advantage of penalizing the occurance of sharp turns, permitting us to model the formation of piecewise smooth boundaries (a more general case) using a binary line process instead of the 4–valued process proposed by Geman and Geman. The potentials for these cliques are computed in the following way:

Let V_a, V_b denote the potentials associated with the cliques C_a, C_b of figure 13 (a) and (b), respectively, and let S_k ($k \in \{a, b\}$) denote the number of line elements

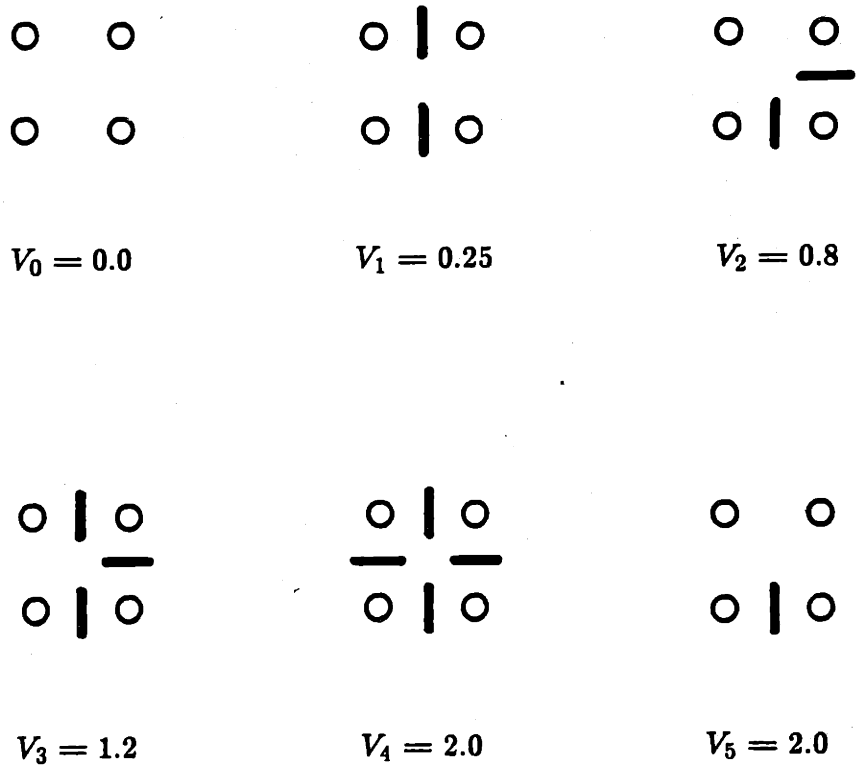


Figure 14. Potentials for the different configurations of a line process

belonging to C_k that are "on" at a given time, i.e.,

$$S_k = \sum_{i \in C_k} l_i \quad , \quad k = a, b$$

The potentials V_k are given by:

$$V_k = \beta \phi_k(S_k) \quad , \quad k = a, b \tag{2}$$

where β is a constant, and the functions ϕ_k are defined by the following tables:

S_a	0	1	2	3	4
ϕ_a	0	0.4	0.25	1.2	2.0

S_b	0	1	2
ϕ_b	0	0	10

It is not difficult to see that this choice of potentials (notice that V_a will be slightly different from the definition of figure 14) will effectively discourage both the formation of thick boundaries ($S_b = 2$) and the presence of sharp turns ($S_a = 3$ and/or $S_b = 2$).

2. The potentials of the depth process, which is now continuous–valued, have to be modified to express the more relaxed condition of piecewise continuity (instead of piecewise constancy). Specifically, we propose:

$$V_(f_i, f_j, l_{ij}) = \begin{cases} (f_i - f_j)^2(1 - l_{ij}), & \text{for } |i - j| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

(note that $l_{ij} \in \{0, 1\}$)

3. Unlike the case of piecewise constant surfaces, we now have to worry about the maximum absolute difference in the values of two adjacent depth sites that we are willing to consider as a "smooth" gradient (and not a discontinuity). This value, which in general is problem–dependent, determines the magnitude of the constant β in equation (2), which can be interpreted as the coupling strength between the two processes.

2.3. Model for the Observations.

We will adopt the general model described in section 2.1 of chapter 3 to represent the observation process. In particular, to make the discussion more specific, we will assume that the observations g correspond to samples of the surface f taken at a set $S \subseteq L$ of sparse locations, corrupted by a zero mean, white, additive

Gaussian noise process:

$$g_i = f_i + n_i$$

so that the conditional distribution is:

$$p_{f|g}(g; f) = \prod_{i \in S} \frac{1}{\sqrt{2\pi}\sigma} \exp[-(f_i - g_i)^2 / 2\sigma^2]$$

our results, however, can be extended to handle other noise models as well.

Using Bayes' rule, we can finally write the posterior distribution as:

$$P_{f,l|g}(f, l; g) = \frac{1}{Z_P} \exp[-U_P(f, l; g)]$$

with

$$U_P(f, l; g) = \frac{1}{T_0} \sum_{i,j} V(f_i, f_j, l_{ij}) +$$

$$+ \frac{1}{2\sigma^2} \sum_{i \in S} (f_i - g_i)^2 + \sum_{C_a} V_a(l) + \sum_{C_b} V_b(l) \tag{4}$$

V_a and V_b are the potentials corresponding to the "a" and "b" type cliques of the line process, and are defined by equation (2). It is convenient to introduce a function q which is equal to 1 only at those sites where there is an observation, and is equal to zero elsewhere (i.e., q is an indicator function of the set S):

$$q_i = \begin{cases} 1, & \text{if } i \in S \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Using this function, and the definition of V from equation (3) we get:

$$U_P(f, l; g) = \frac{1}{T_0} \sum_{i,j} (f_i - f_j)^2 (1 - l_{ij}) +$$

$$+ \frac{1}{2\sigma^2} \sum_{i \in L} (f_i - g_i)^2 q_i + \sum_{C_a} V_a(l) + \sum_{C_b} V_b(l) \tag{6}$$

3. Optimality Criterion.

We can now apply the general principles developed in chapter 3 to derive the optimal Bayesian estimators for the depth and line fields. As a performance criterion we will use a mixed cost functional of the form:

$$e_m(f, l, \hat{f}, \hat{l}) = \sum_{i \in L_f} (f_i - \hat{f}_i)^2 + \sum_{j \in L_l} (1 - \delta(l_j - \hat{l}_j)) \tag{7}$$

where L_f, L_l denote the depth and line lattices, respectively. This error criterion means that the reconstructed surface should be as close as possible to the true (unknown) surface, and that we should commit as few errors as possible in the assertions about the presence or absence of discontinuities.

Appllying the results of section 5 of chapter 3, we find that the optimal estimators will be the posterior mean for f and the maximizer of the posterior marginals for l. Note that these estimates must be computed by averaging over all possible values of both f and l:

$$\bar{f}_i = \sum_f \sum_l f_i P_{f,l|g}(f, l; g)$$

$$P_{l_i}(q) = \sum_f \sum_{l: l_i = q} P_{f,l|g}(f, l; g)$$

4. Monte Carlo Algorithm.

There is one serious difficulty that prevents us from applying directly the general Monte Carlo procedure that was derived in chapter 3 to the computation of these optimal estimates: since the depth variables are continuous-valued, if we discretize them finely enough to guarantee sufficient precision of the results, the computational complexity of either the Metropolis ar Gibbs Sampler algorithms will be very large. One way around this difficulty is to note that for any fixed configuration of the line field, the posterior energy becomes a non-negative definite quadratic form:

$$U(f \mid l, g) = \sum_{i,j: l_{ij} = 0} (f_i - f_j)^2 + \alpha \sum_{j \in S} (f_j - g_j)^2 + K \tag{8}$$

where α and K are constants (note that the first sum is taken only over those pairs of sites whose connecting line element is "off", and the second one over the set S). This means that the posterior distribution of the depth field is conditionally Gaussian, so that we can find the optimal conditional estimator $f^*(l)$ as the minimizer of (8) (for a Gaussian distribution, the posterior mean and the MAP estimate coincide). If l is identically zero (no lines), this function is strictly convex, and therefore it has a unique minimum. Let f_0^* be the corresponding global minimizer. For any fixed configuration l, the gradient of (8) is given by:

$$\frac{\partial U(f \mid l)}{\partial f_i} = 2 \sum_{j \in N_i} (f_i - f_j)\ell_{ij} + 2\alpha q_i (f_i - g_i) \tag{9}$$

where

$$N_i = \{j \ : \ |i-j| = 1\} \ ;$$

$$\ell_{ij} = 1 - l_{ij}$$

Setting this gradient equal to 0, we find that any minimizer of U will be a fixed point of the system:

$$f_j^{(k+1)} = \frac{\sum_{i \in N_j} \ell_{ij} f_i^{(k)} + \alpha q_j g_j}{\sum_{i \in N_j} \ell_{ij} + \alpha q_j} \qquad \text{if } \sum_{i \in N_j} \ell_{ij} + \alpha q_j \neq 0,$$

$$\text{and } f_j^{(k+1)} = f_j^{(k)} \qquad \text{otherwise} \tag{10}$$

We note that the updating scheme (10) will produce a decrease in the value of $U(f \mid l)$, regardless of the sweeping strategy. In a synchronous scheme (where all the sites are updated at the same time), the energy increment will be:

$$\Delta U(f \mid l) = U(f^{(k+1)} \mid l) - U(f^{(k)} \mid l) =$$

$$= -2 \sum_{i \in L} \Big(\sum_{j \in N_i} \ell_{ij} + \alpha q_i \Big)(f_i^{(k+1)} - f_i^{(k)})^2 -$$

$$- \sum_{i,j} \frac{\partial^2 U(f \mid l)}{\partial f_i f_j} (f_i^{(k)} - f_i^{(k+1)})(f_j^{(k)} - f_j^{(k+1)}) \leq 0$$

because U is non–negative definite. For an asynchronous strategy, where $f^{(k+1)}$ is obtained from $f^{(k)}$ by updating only the site i, we get:

$$\Delta U(f \mid l) = -4(\sum_{j \in N_i} \ell_{ij} + \alpha q_i)(f_i^{(k+1)} - f_i^{(k)})^2 \leq 0$$

Therefore, if we set

$$f^{(0)} = f_0^* \tag{11}$$

the dynamical system defined by (10) and (11) (with a given sweeping strategy) will be stable and have a unique fixed point f_l^*.

Note that, since $U(f \mid l)$ is always convex, f_l^* will be a global minimizer (see Luenberger (1973)), but in general it will not be the only one; there may be cases in which some region Q within which there are no observations is isolated from the rest of the lattice by the line process. In this case, any solution for which

$$f_j = \text{constant}\,, \qquad j \in Q$$

will also minimize $U(f \mid l)$. However, for a fixed initial state $f^{(0)}$ the deterministic dynamical system (10) will always converge to the same solution, so that the configuration $f^*(l)$ is well defined.

Let us define the set F^* as:

$$F^* = \{(f,l) \;\; : \;\; f = f_l^*\}$$

It is clear that, if $\hat{f}, \hat{l}$ are the optimal estimates for our problem, we have that:

$$(\hat{f}, \hat{l}) \in F^*$$

which suggests that we can constrain the search for the optimal estimators to this set. This can be done, in principle, by replacing the posterior energy with the function:

$$U^*(l) = U(f_l^*, l)$$

(which depends only on l), and use the standard Monte Carlo procedures to find the optimal estimator $\hat{l}$. To illustrate this idea, let us consider the following physical model:

It is a well known fact the the steady state of an electrical network that contains only (current or voltage) sources and linear resistors will be the global minimizer of a quadratic functional that corresponds to the total power dissipated as heat (Oster et al, 1971). It is therefore possible to contruct an analog network that will find the equilibrium state of the depth field for a given, fixed configuration of the line process, i.e., that will minimize the conditional energy (8) (see Poggio and Koch, 1984). This suggests a hybrid computational scheme in which the line field (whose state is updated digitally, using, say, the Metropolis or Gibbs Sampler algorithms) acts as a set of switches on the connections between the nodes of the analog network whose voltages represent the depth process. In particular, if f_i represents the voltage at node i, the hybrid network can be represented as a 4–connected lattice of nodes (see figure 15) in which:

(i) A resistance (of unit magnitude) and a switch (controlled by the line element l_{ij}) is present in every link between pairs i, j of adjacent nodes.

(ii) If an observation g_i is present at site i, a current of magnitude equal to αg_i is injected to the corresponding node, which must also be connected to a common ground via a resistance of magnitude $1/\alpha$ (see equation 8).

A direct application of Kirchoff current law shows that at each node of this network we will have:

$$\sum_{j \in N_i} (f_i - f_j)(1 - l_{ij}) + \alpha q_i f_i = \alpha q_i g_i$$

which corresponds to a fixed point of the system (10). In practice, there will always be parasitic capacitances which will prevent the instantaneous establishment of the equilibrium conditions. However, the time constant of the analog portion of the network may be made very fast, so that in fact, the probability distribution of the equilibrium states of this network will be Gibbsian with energy U^*.

This scheme can be used, in principle, to construct a special purpose hybrid computer for the fast solution of problems of this type. In a digital machine, however,

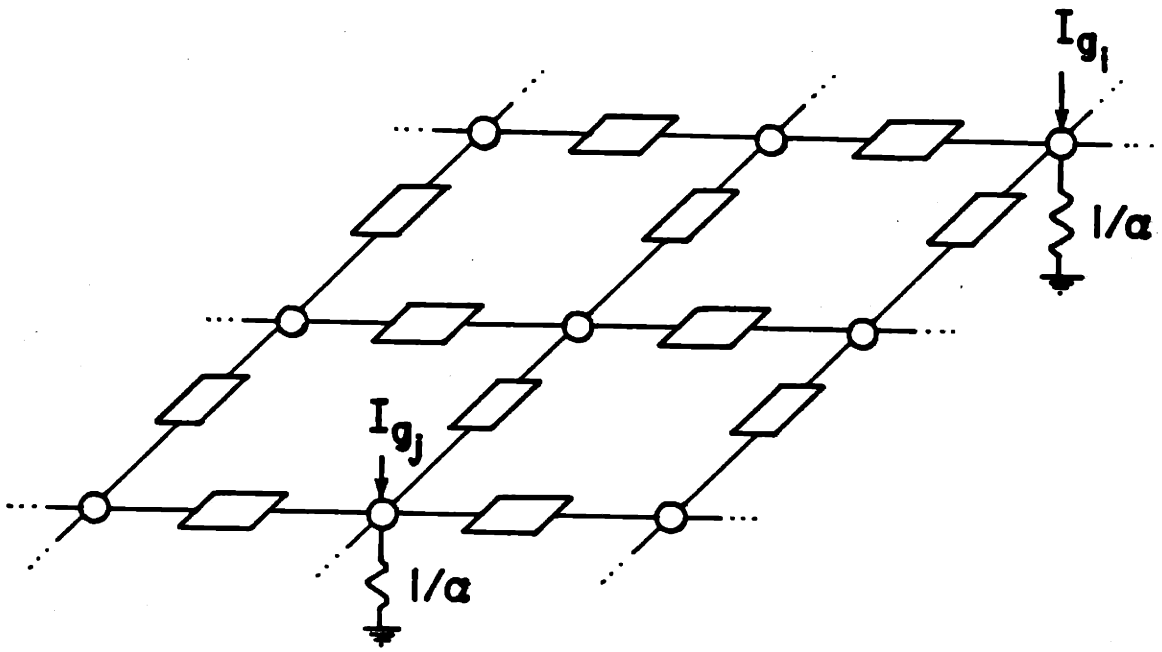


Figure 15. Hybrid network implementing the surface reconstruction algorithm of section 4. The voltage at every node represents the height of the surface. Inside every rectangular box there is a resistance of unit magnitude and a switch whose state is controlled by the corresponding line element. (see text).

the exact implementation of this strategy will in general be computationally very expensive, since f_i^* must be computed every time a line site is updated. We will now present an approximation which has an excellent experimental performance, and leads to an efficient implementation.

First, let us examine one iteration of the, say, Metropolis algorithm at a given

temperature $T > 0$ for the function U^*. When a line site is visited and its state is updated, the corresponding increment in energy ΔU_l is computed as follows:

Suppose the line site ij is visited (the line between depth sites i and j). Let l_{ij} be its current state, and $\hat{l}_{ij}$ the candidate state:

$$\hat{l}_{ij} = 1 - l_{ij}$$

Suppose that the current state of the depth process is

$$f = f_l^*$$

and let $f_{\hat{l}}^*$ be the fixed point of (10) obtained when we replace l_{ij} by $\hat{l}_{ij}$. Let us define:

$$\hat{f} = f_{\hat{l}}^*$$

and

$$\Delta V_{ij} = \sum_{C_a : l_{ij} \in C_a} V_a(\hat{l}) - V_a(l) + \sum_{C_b : l_{ij} \in C_b} V_b(\hat{l}) - V_b(l)$$

where C_a, C_b are the "a" and "b" type cliques defined in figure 13, and V_a, V_b, the corresponding potentials.

Since the depth process is at equilibrium, and we are changing only the element l_{ij}, we may assume that

$$\hat{f}_p \approx f_p \qquad \text{for } p \neq i, j \tag{12}$$

so that

$$\Delta U_l^* \approx \Delta V_{ij} +$$

$$+ \sum_{m=i,j} \left[(\hat{f}_m^2 - f_m^2)[\sum_{k \in N_m} (1 - l_{km}) + \alpha q_m] - 2(\hat{f}_m - f_m)[\sum_{k \in N_m} f_k(1 - l_{km}) + \alpha q_m g_m] \right] \tag{13}$$

Now, if the absolute difference $|f_i - f_j|$ is small, f and $\hat{f}$ will be practically identical; on the other hand, if $|f_i - f_j|$ is large, the changes in f at locations i and j will be relatively small with respect to this absolute difference. Therefore, we may

approximate ΔU_l^* by the simple expression:

$$\Delta U_l^* \approx \Delta V_{ij} + (f_j - f_i)^2(l_{ij} - \hat{l}_{ij}) \tag{14}$$

which depends only on the potentials of the cliques to which the updated line element belongs, and on the current state of the depth sites adjacent to it. If this approximation is to remain valid, the equilibrium condition on f must be mantained. This is done by performing M global deterministic iterations using (10) after every global stochastic update of the line process. We have found experimentally that the use of the approximate expression (14), and only three restoring iterations ($M = 3$) are sufficient to get a good convergence behavior.

It is also possible to use assumption (12) and the fixed point condition of the system (10) to compute a more precise approximation to ΔU^* (the corresponding formulae are derived in appendix 5.A). Our experiments indicate, however, that the simpler approximation (14) gives sufficiently good results, so that the increased complexity incurred by the use of this, more precise scheme does not seem to be justified.

An important issue in the practical implementation of this procedure is the determination of the optimal temperature for observing the equilibrium behavior of the system. We have found that this can be done effectively in an adaptive way by starting the simulation at a relatively large temperature (say, $T = 5$) and slowly decreasing it until the network shows an adequate level of activity (measured, by the fraction of sites whose state is modified in one global iteration). We have found that a level on the order of 0.1 is adequate in most cases. This technique is similar to the Simulated Annealing method for finding the global minimizer of the energy, but in that case, the cooling of the system must proceed at a slower rate, and it should be continued until the level of activity is reduced practically to 0; if we proceed in this way, the final state of the system will correspond (approximately) to the MAP estimate. Note that $(\hat{f}_{MAP}, \hat{l}_{MAP}) \in F^*$ too, so that the mixed strategy described above will also work in this case (see Marroquin, 1984). As we pointed out in the last chapter, if the signal to noise ratio is not too low, the configuration corresponding to the MAP estimate will be very similar to the optimal one $(\hat{f}_{PM}, \hat{l}_{MPM})$. From

a computational viewpoint, however, the optimal estimator is preferable, since it exhibits a faster and more consistent convergence behavior.

5. Experimental Results.

We will now present some experimental results that illustrate the performance of the optimal Bayesian estimators for surface reconstruction tasks. In these examples, we assume that we have the following prior knowledge about the nature of the surfaces we are trying to reconstruct:

(i) The region under consideration can be segmented into a small number of subregions.

(ii) Within each subregion the surface is smooth (the gradient is less than 0.5).

(iii) The boundaries between regions are piecewise smooth. There are relatively few corners.

(iv) The average height of the discontinuities across boundaries is greater than 0.8.

(v) The observations are corrupted by an additive white Gaussian noise process, and we have some estimate of its intensity.

This knowledge is embodied in the model for the line process, and in the numerical value of the parameters. For our experiments, we have used a binary process with potentials given by equation (2).

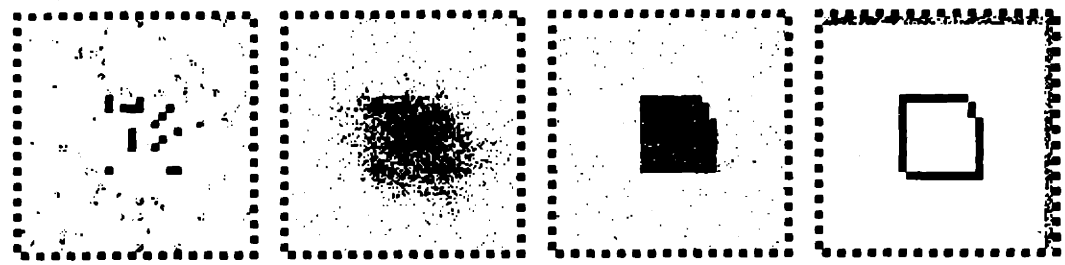
In the first set of experiments, we generated sparse observation points at 200 random locations of a 30 X 30 rectangular grid. Figures 16, 17, 18 and 19 show (with height coded by grey level) the observations (a); the configuration obtained by interpolation with no boundaries (b); the final reconstructed surface (c), and the boundaries found by the algorithm (d), for:

(i) A square at height 2.0 over a background at constant height = 1.0 (Fig. 16).

(ii) A triangle, with the same characteristics (Fig. 17).

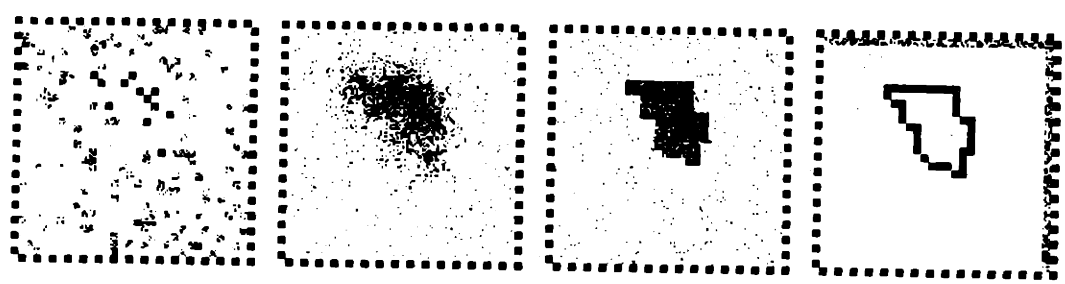
(iii) A tilted square plane (slope = 0.1) over a constant height background with white Gaussian added noise ($\sigma = 0.1$) (Fig. 18).

(iv) Three rectangles at different (constant) heights over a uniform background (Fig. 19).



(a) (b) (c) (d)

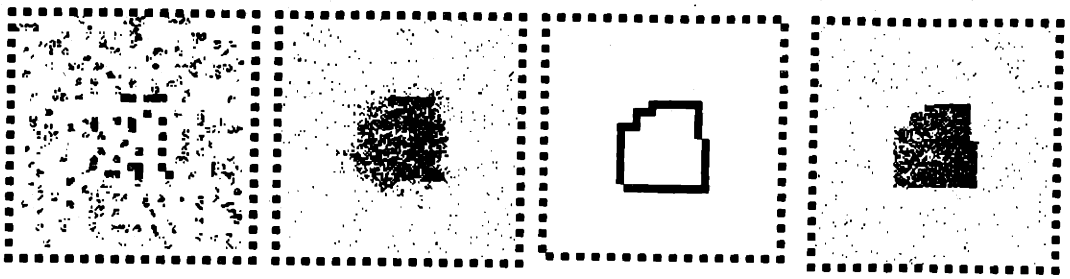
Figure 16. (a) Observations of a square at height 2.0 over a background at height 1.0 (a white pixel means that the observation is absent at that point). (b) Interpolation with no boundaries. (c)Reconstructed surface.(d) Boundaries found by the Algorithm.



(a) (b) (c) (d)

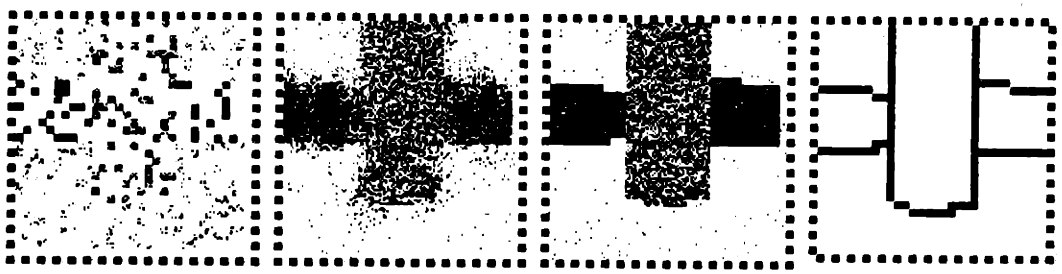
Figure 17. (a) Observations of a triangle at height 2.0 over a background at height 1.0. (a white pixel means that the observation is absent at that point). (b) Interpolation with no boundaries (c) Reconstructed surface.(d) Boundaries found by the Algorithm.

In many interesting cases, the observation sites are not randomly distributed, but rather tend to be clustered along certain curves. This is the case, for example, of



(a) (b) (c) (d)

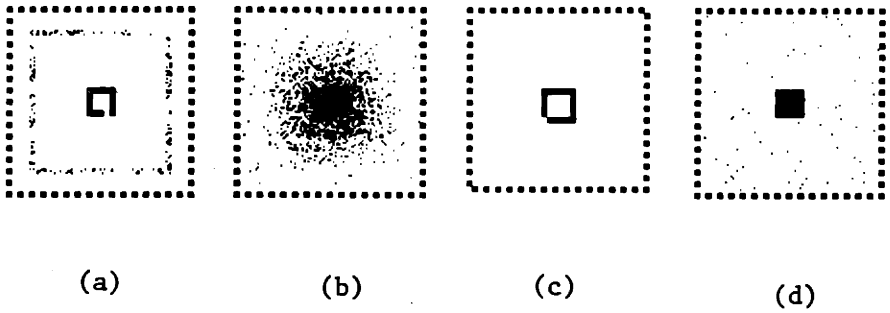
Figure 18. (a) Observations of a tilted square (slope = 0.1) over a background at height 1.0 with added white Gaussian noise ($\sigma = 0.1$) (a white pixel means that the observation is absent at that point). (b) Interpolation with no boundaries. (c) Boundaries found by the Algorithm. (d) Reconstructed surface.



(a) (b) (c) (d)

Figure 19. (a) Observations of 3 rectangles at heights 2.0, 2.0 and 3.0 over a background at height 1.0 (a white pixel means that the observation is absent at that point). (b) Interpolation with no boundaries. (c) Reconstructed surface.(d) Boundaries found by the Algorithm.

the reconstruction of geological structures from seismic data, or of certain algorithms



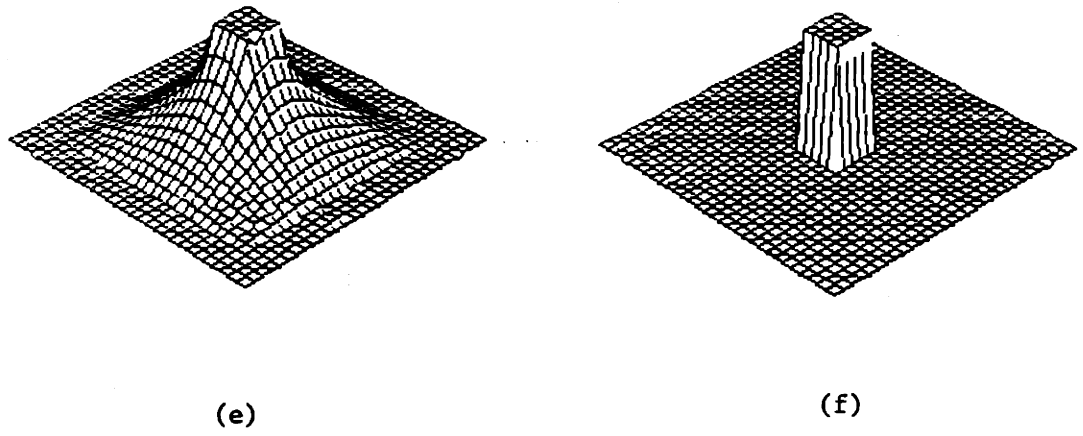


Figure 20. (a) Observations of a square at height 2.0 over a background at height 1.0 with added white Gaussian noise ($\sigma = 0.1$). White pixels denote missing observations. (b) Interpolation with no boundaries. (c) Boundaries found by the Algorithm. (d) Reconstructed surface. (e) Perspective view of (b). (f) Perspective view of (d).

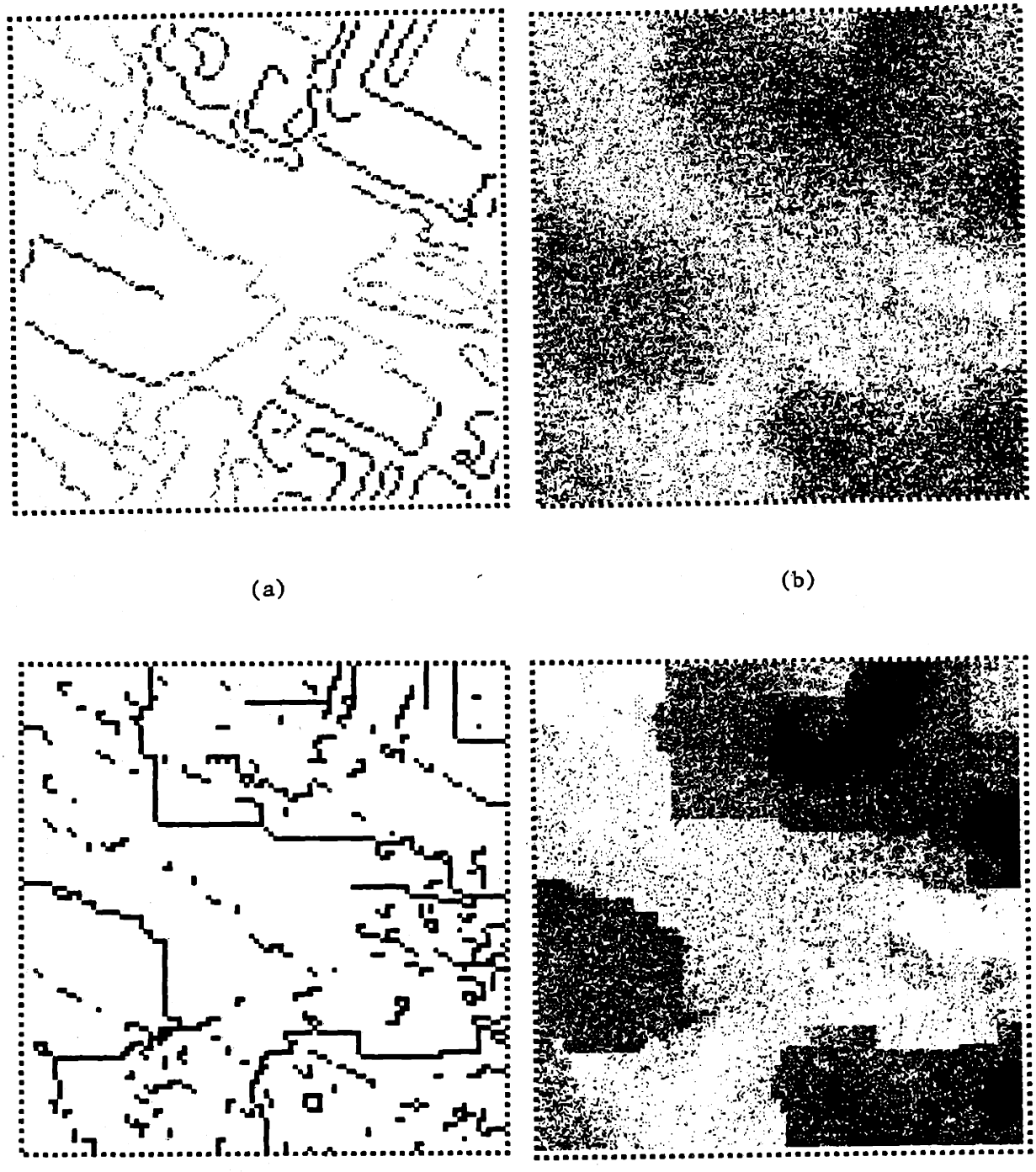


Figure 21. (a) Disparity data for a stereo pair of aerial photographs (data kindly provided by W.E.L. Grimson). (b) Interpolation with no boundaries. (c) Boundaries found by the Algorithm (d) Reconstructed surface.

for the reconstruction of surfaces from stereoscopic pairs of images, when the stereo matching is done only at the "edges" (places where the intensity gradient is large) detected in the images. The synthetic example of figure 20 illustrates this situation (here we include also a perspective representation of the recontructed surfaces, so that the difference between the smooth reconstruction and the optimal estimate can be fully appreciated). In figure 21 we illustrate a real example of this situation. It represents the interpolation of data obtained along the zero–crossing contours of the convolution of a stereo pair of aerial photographs (depicting the campus of the University of British Columbia) with a "Difference of Gaussians" operator, by Grimson's implementation of the Marr–Poggio stereo algorithm [G4,M2]. We will come back to this example when we discuss the stereo matching problem in detail in the next chapter.

We have also used a modified Simulated annealing scheme to get the MAP estimator for the same examples presented above (see Marroquin, 1984). The final configurations are very similar to the optimal ones, so we do not reproduce them here. With respect to the computational efficiency, it took, on the average, around 450 global iterations (in a global iteration the state of the complete line field is updated, and the equilibrium of the depth field is restored) for the Simulated Annealing algorithm to converge, while for the $(\hat{f}_{PM}, \hat{l}_{MPM})$ estimator, only 250 were needed. Also, in the latter, the behavior of the algorithm was more consistent in the sense that the difference in the results from successive runs with the same data were smaller than in the former case.

6. A Fast Algorithm.

The ergodicity of the "Gibbs chain" (the Markov chain generated by the Gibbs Sampler or the Metropolis algorithm at a fixed temperature) means that its time behavior mirrors the ensemble probabilistic structure. Since the probability of turning "on" a given line element depends on the difference in the values of the associated depth elements (i.e., on the current value of the gradient of the field f at that location), the configurations with active lines at points of high gradient will be generated first. These lines, in turn, will decouple the adjacent depth sites, increasing

the gradient even more, generating thus a positive feedback that stabilizes these configurations (the opposite happens in regions of low gradient, which prevents the formation of stable clusters of lines at those points).

We can see, therefore, that the behavior of the Gibbs chain can be thought of, qualitatively, as performing in time a scale separation of the discontinuities of the image. This suggests the use of a deterministic scheme that performs the same separation, but compressing the time of the Gibbs chain. A simple way of implementing this idea, is to introduce a time varying coupling between the depth and line fields, and to allow only "downhill" moves (i.e., those with negative ΔU^*) in the updating rules for the line process. Specifically, we compute the increment in energy associated with the update of the line element l_{ij} at time t using:

$$\Delta U^* = \Delta V_{ij} + K(t)(f_i - f_j)^2(l_{ij} - \hat{l}_{ij}) \tag{15}$$

instead of equation (14), and accept the candidate state only if $\Delta U^* < 0$. The coupling strength $K(t)$ is computed using:

$$K(t) = K_0 + ht$$

(where K_0 and h are positive constants) until it reaches a given value K_T, and it is held constant at this value thereafter. The state of the depth process is updated, as before, using equation (10). K_0 must be chosen in such a way that with $f = f_0^*$ and $l_i = 0$ for all i, no lines will be turned "on" in the first iteration. This means that if we use equation (2) (with $\beta = 1$, and the values of ϕ given in the corresponding tables) to compute the potentials, we must have:

$$K_0 \leq \frac{0.4}{a} \tag{16}$$

where

$$a = \sup_{i,j}(f_i - f_j)^2$$

On the other hand, the final value of $K(t)$ (i.e., K_T), must be such that no lines are introduced in the smooth regions. Let

$$b = \inf_{D}(f_i - f_j)^2$$

$$c = \sup_{Sm}(f_i - f_j)^2$$

where D is the set of neighboring pairs of sites such that each site belongs to a different smooth patch (i.e., pairs that lie across a discontinuity), and Sm is the complementary set of pairs of adjacent sites such that both sites belong to the same continuous patch. K_T must satisfy:

$$\frac{0.25}{b} < K_T < \frac{0.25}{c}$$

Note that even if we do not know the precise values of a, b and c for a given problem, usually we can estimate them accurately enough to determine "safe" values for K_0 and K_T. The value of h controls the number of iterations needed for the algorithm to reach a fixed point; if h is too large and the observations are relatively sparse, we might get suboptimal solutions where regions with no observations are completely surrounded by lines, and therefore, adopt spurious constant values. We have found experimentally that usually 50 iterations, i.e., setting

$$h = \frac{K_T - K_0}{50}$$

are enough to produce results that are indistinguishable from those produced by the Monte Carlo approximation.

This scheme has an additional advantage: the optimal value of the coupling between the depth and line fields (the constant β in equation (2)) depends on the height of the discontinuities relative to the gradient in the smooth patches. It is, therefore, a free parameter of the Monte Carlo algorithm that must be adapted to each particular problem. Since in the deterministic scheme it is varied dinamically, its adaptation to each problem is automatic, provided that we choose K_T and K_0 sufficiently large and small, respectively, so that the procedure has practically no free parameters.

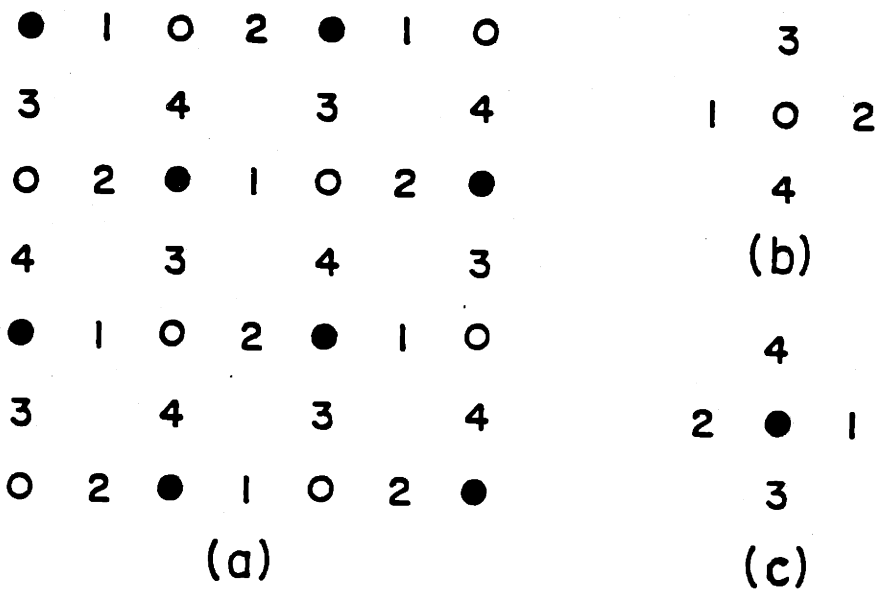


Figure 22. (a) Coloring of the coupled line–depth lattice. (b) and (c) Elements whose state is stored in each of the two types of processors of a 4–connected parallel architecture.

7. Parallel Implementations.

Both the general Monte Carlo procedure of section 5 and the deterministic algorithm of the last section can be efficiently implemented in a parallel architecture. To study this implementation, we first note that the chromatic numbers (see section 6.2 of chapter 3) of the graphs associated with the line and depth neighborhood systems are 4 and 2, respectively, which means that the coupled process has a chromatic number of 6. In figure 22 (a) we illustrate one possible "coloring".

The colors of the line process are represented by the numbers 1,2,3,4, and those of the depth process by white and black circles. The updating process can

be implemented in a 4–connected architecture such as the "Connection Machine", by assigning one processor to each depth site and its four adjacent line elements. We will thus have two different populations of processors, whose configurations are shown in figures 22 (b) and (c), respectively.

Each complete iteration consist on 6 major cycles: in the first two, the state of the white and black depth variables is respectively updated, and in the next four, the new states of the binary line variables stored in (say) the white processors are successively computed and transmitted to the corresponding memory locations of the neighboring black processors. Note that in this scheme we have some redundancy in the use of memory (each binary variable is stored twice), but the state of all the elements needed for each updating operation is always available from adjacent processors.

7.1. Connection Machine Execution Time.

The update of each depth site requires 2 (16–bit) multiplications; 5 additions and 10 1–bit comparisons, that is, about 600 cycles of a 1–bit processor. The computation of the increment in energy for the line process (equation 14) requires 1 multiplication; 5 additions and 13 1–bit operations, that is 350 cycles. For the deterministic algorithm, we require 256 additional cycles for the multiplication by the variable coupling constant, while the exponentiation and random number generation needed for the Monte Carlo updating use about 2300 additional cycles (we assume that the updating of the coupling constant is done once every complete iteration in the host computer, and the new value broadcast to the whole network).

Considering that the Monte Carlo algorithm requires about 200 iterations to converge, while only 50 are needed in the deterministic case, we get the following approximate estimates for the total execution time in the "Connection Machine" (using the same assumptions as in section 6.3 of chapter 3): 2.4 seconds for the Monte Carlo procedure, and 0.18 seconds for the deterministic algorithm.

7.2. Analog Networks.

In chapter 4 we discussed the use of the "neural" networks introduced by Hopfield (1984) (see also Hopfield and Tank, 1985) for constructing analog systems that approximate the optimal estimators of binary fields. Since for a binary system, the TPM and MPM estimates are equivalent (see chapter 3), we can, in principle, replace the digital computation of the l field in the hybrid scheme discussed above (see figure 15) by a "neural" network that approximates the optimal estimate coupled with the analog "f" network (note that the switches must be replaced by analog devices that implement a multiplication). The time constant of the "neural" network has to be adjusted so that the "f" network remains in equilibrium and the search space is effectively restricted to the set F^* (see section 4).

To implement this idea, we must define a new energy function that depends continuously on l, and whose behavior is similar to U_P for $l_i \in \{0,1\}$ (Hopfield, 1985). One such function is:

$$E(f,l) = K\sum_i \sum_{j\in N_i}(f_i - f_j)^2(1-l_{ij}) + \alpha K \sum_{i\in S}(f_i - g_i)^2 +$$

$$+c_1\sum_i \sum_{C_a: i\in C_a} l_i\Big(\sum_{k\in C_a-\{i\}} l_k - 1\Big)^2 + c_2\sum_i l_i(1-l_i) +$$

$$+c_3 \sum_{C_b: i\in C_b}\sum_{j\in C_b-\{i\}} l_i l_j \qquad (23)$$

where K, α, c_1, c_2, c_3 are constants.

Following the construction discussed in section 5 of chapter 4, we can now use an analog network that implements the dynamical system:

$$\frac{du_i}{dt} = -\frac{\partial E}{\partial l_i} - u_i$$

$$l_i = \Theta(u_i)$$

Where the function Θ, which corresponds to the gain of the non-linear amplifiers that are at the nodes of the network, is as defined in equation (15) of chapter 4 (note that in this case the network also contains non-linear elements that act as analog multipliers).

We have performed numerical simulations of this method, and the results are similar to the optimal ones if the parameters of the system are selected appropriately. The system can be made practically data-independent by making the coupling K between the two networks (see equation (23)) time-varying, in the manner that was described in section 6. We have found that a reasonable set of values for the remaining parameters is: $c_1 = .15$; $c_2 = .05$; $c_3 = 1.5$.

8. Discussion.

In this chapter we have studied the problem of reconstructing piecewise continuous surfaces from sparse and noisy data. We showed that such surfaces can be adequately modeled by two coupled MRF's: A depth field with quadratic potentials and a binary "line" field with sites in the dual lattice, and with potentials that represent our prior knowledge about the geometry of the curves that bound the smooth patches.

We pointed out that a straightforward extension of the general estimation procedures derived in chapter 3 to this problem is computationally unfeasible, due to the continuous nature of the depth field. Therefore, we proposed a modified computational strategy that is based on the fact that the search space for the optimal estimates can be restricted to those configurations in which the depth field minimizes the (quadratic) conditional posterior energy for each given line configuration. The plausibility of this scheme was demonstrated by experimental results showing the reconstruction of both synthetic and "real" surfaces.

We also derived, based on heuristic arguments, a fast deterministic algorithm with excelent experimental performance, and whose parameters can be made problem-independent, and discussed the implementation of all these procedures in parallel digital machines, and in hybrid and analog networks.

It is interesting to compare the techniques we have presented with other surface reconstruction methods that handle discontinuities. The most successful of these (see Terzopoulos, 1984) are based on the idea of interpolating a smooth surface first and then, detecting the discontinuities by a threshold mechanism. We believe

that the method that we are proposing has some advantages over this scheme which justify its use in spite of the increased computational cost:

(i) From a conceptual viewpoint, it is better to perform the interpolation and boundary detection tasks at the same time, rather than approximating an everywhere smooth surface first, since this operation hides the discontinuities that one then tries to find in the second phase.

(ii) In our method, the values of the parameters depend only on the average height of the jumps that one wants to consider as boundaries *in the reconstructed surface*, and thus, they are independent of the location of the observations. If these are sparsely located, even when the discontinuity is relatively large, the threshold method may fail.

(iii) A priori knowledge about the shape, orientation and position of the discontinuities can be easily incorporated by choice of the potentials of the line process. This fact makes our method particularly promising for integrating information from qualitatively different sources into a single unified estimation procedure.

(iv) The same algorithm can be used for surface interpolation, noise elimination (smoothing) and boundary detection.

We will now study a related problem: the reconstruction of surfaces from stereoscopic images.

Appendix 5.A

HIGHER ORDER APPROXIMATION TO ΔU^*

In this appendix we describe a higher order approximation to the energy increment ΔU^* (see section 4 of chapter 5). We will compute ΔU^* using:

$$\Delta U_l^* \approx \Delta V_{ij} +$$

$$+ \sum_{m=i,j} \left[(\hat{f}_m^2 - f_m^2)[\sum_{k \in N_m} (1 - l_{km}) + \alpha q_m] - 2(\hat{f}_m - f_m)[\sum_{k \in N_m} f_k(1 - l_{km}) + \alpha q_m g_m] \right] \tag{1}$$

using the assumption:

$$\hat{f}_p \approx f_p \qquad \text{for } p \neq i, j$$

the new equilibrium configuration $\hat{f}$ can be estimated by the following formulas, which correspond to the fixed point of:

$$f_j^{(k+1)} = \frac{\sum_{i \in N_j} \ell_{ij} f_i^{(k)} + \alpha q_j g_j}{\sum_{i \in N_j} \ell_{ij} + \alpha q_j} \tag{2}$$

when f_p, $p \neq i, j$ is held fixed:

Let:

$$\hat{\ell}_{km} = \begin{cases} 1 - \hat{l}_{km}, & \text{for } k, m = i, j \\ 1 - l_{km}, & \text{otherwise} \end{cases}$$

$$\gamma_m = \sum_{k \in N_m} \hat{\ell}_{km} + \alpha q_m$$

The new equilibrium configuration will be a fixed point of (10), and therefore, it will satisfy:

$$\hat{f}_m = \frac{1}{\gamma_m} \left[\sum_{k \in N_m} \hat{\ell}_{km} \hat{f}_m + \alpha q_m g_m \right] \qquad \text{for } m = i, j$$

If $\hat{l}_{ij} = 0$ and $\gamma_i\gamma_j \neq 1$, we get

$$\hat{f}_i = \frac{1}{\gamma_i\gamma_j - 1}\{\gamma_j\left[\sum_{k\in N_i-\{j\}} f_k\hat{\ell}_{ik} + \alpha q_i g_i\right] + \sum_{k\in N_j-\{i\}} f_k\hat{\ell}_{jk} + \alpha q_j g_j\}$$

$$\hat{f}_j = \frac{1}{\gamma_j}\left[\sum_{k\in N_j} f_k\hat{\ell}_{jk} + \alpha g_j q_j\right]$$

if $\hat{l}_{ij} = 0$ and $\gamma_i\gamma_j = 1$, it means that there are no observations, neither at i nor at j, and that these two sites are isolated from the rest of the lattice by line elements. Therefore, we use:

$$\hat{f}_i = \hat{f}_j = \frac{f_i + f_j}{2}$$

Finally, if $\hat{l}_{ij} = 1$, we put

$$\hat{f}_m = \begin{cases} \frac{1}{\gamma_m}\left[\Sigma_{k\in N_m} f_m\hat{\ell}_{km} + \alpha q_m g_m\right], & \text{if } \gamma_m \neq 0 \\ f_m, & \text{if } \gamma_m = 0 \end{cases}$$

for $m = i, j$.

Besides, if the move from l to $\hat{l}$ is accepted by Metropolis criterion, we replace

$$f_m = \hat{f}_m, \qquad \text{for } m = i, j$$

As described in chapter 5, after all l sites have been updated, M restoring iterations using equation (10) of that chapter should be applied.

Chapter 6

SIGNAL MATCHING

1. Introduction.

In all the estimation problems we have studied so far, the posterior energy function had the form:

$$U_P(f;g) = U_0(f) + \sum_i \Phi_i(f_i, g_i) \tag{1}$$

where $U_0(f)$ corresponded to the MRF model for the field f. The functions Φ_i, whose precise form depended on the particular noise model, were non–decreasing functions of the distance between f_i and g_i (see equation (2) of chapter 3):

$$\Phi_i(f, g_i) = -\ln P_{ni}(\Psi^{-1}(g_i, H_i(f))$$

There are some cases, however, when the conditional probability distribution of the observations $P_{g|f}(g;f)$ is multimodal (as a function of f) which causes the functions Φ_i to be non–monotonic, so that the solution to the problem remains ambiguous, even if the observations are dense, and the signal to noise ratio arbitrarily high. To illustrate this situation, we will study an important instance of it: the "signal matching" problem, whose one–dimensional version is as follows:

Consider two one–dimensional, real valued sequences h_L, h_R, where h_L is obtained from h_R by shifting some subintervals according to the "disparity sequence" d:

$$h_L(i) = h_R(i + d_i)$$

with

$$d_i \in Q = \{-m, -m+1, \ldots, -1, 0, 1, \ldots, m\}$$

The signal matching problem is to find d given h_L, h_R. (In a more realistic situation, we do not observe h_L, h_R directly, but rather some noise–corrupted versions g_L, g_R). Some interesting instances of this problem are the matching of stereoscopic images along epipolar lines (Marr and Poggio, 1976); the computation of the dip angle of geological structures from electrical resistivity measurements taken along a bore hole, and the matching of DNA sequences.

To make the discussion more specific, we will consider a simple example, in which the sequences h_L, h_R are binary Bernoulli sequences; we will assume that the noise corruption process can be modeled as a binary symmetric channel with known error rate, and that d is known to be a piecewise constant function. A well known instance of this problem is the matching of a row of a random dot stereogram with density ρ (Julesz (1960)), when the components of the stereo pair are corrupted by noise.

The stochastic model for the observations is then constructed by assuming that the right image is a sample function of a Bernoulli process A with parameter ρ :

$$g_R(i) = A(i)$$

The left image is assumed to be formed from the right one by shifting it by a variable amount given by the disparity function d, except at some points where an error is commited with probability ϵ. Note that some regions that appear in the right image will be occluded in the left one (see figure 23). The "occlusion indicator" ϕ_d can be computed deterministically from d in the following way:

$$\phi_d(i) = \begin{cases} 1, & \text{if } d_{i-k} \geq d_i + k, \text{ for some integer } k \in (0, m] \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The occluded areas are assumed to be "filled in" by an independent Bernoulli process B. The final model is then:

$$g_L(i) = \begin{cases} g_R(i + d_i), & \text{with prob. } 1 - \epsilon, \text{ if } \phi_d(i) = 0 \\ 1 - g_R(i + d_i), & \text{with prob. } \epsilon, \text{ if } \phi_d(i) = 0 \\ B_\rho(i), & \text{with prob. } 1, \text{ if } \phi_d(i) = 1 \end{cases} \tag{3}$$

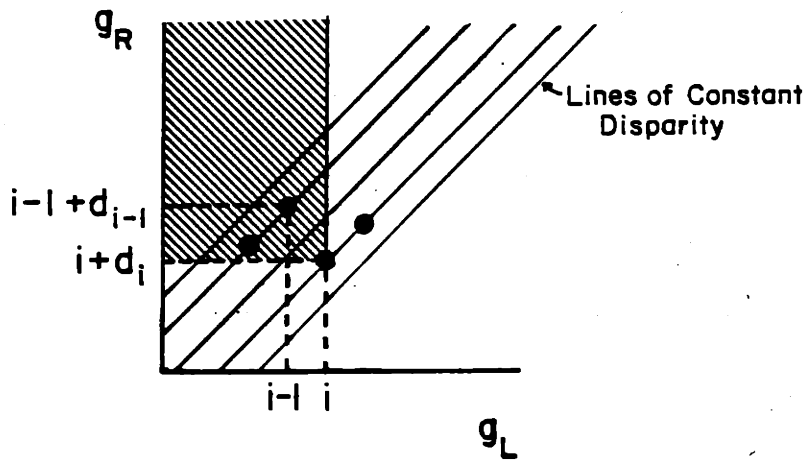


Figure 23. Occluded Regions: The horizontal and vertical axis represent points in one row of the left and right images, respectively. Matching points are represented by black circles. Any match in the shaded region will occlude the point i.

Note that in the two–dimensional case, the index i denotes a site of a lattice, and therefore it can be represented as a two–vector (i_1, i_2) whose components denote the column and row of the site, respectively. To simplify the notation, we will adopt the following convention throughout this chapter: when a scalar is added to this vector index (as in $g_R(i + d_i)$ and d_{i+k}), it will be implicitly assumed that it is multiplied by the vector $(1, 0)$ (so that the above expressions should be understood as $g_R(i + (d_i, 0))$ and $d_{i+(k,0)}$, respectively). Using this convention, the observation model of equation (3) can be applied either to the one or to the two–dimensional cases.

Notice that even if the observations are noise–free ($\epsilon = 0$) the solution of the problem remains ambiguous, and it cannot be uniquely determined unless some

prior knowledge about d (for example, in the form of a MRF model) is introduced. The use of a MRF model in the stereo matching case, corresponds to a quantification of the assumption of the existence of "dense solutions" (this term was introduced by Julesz (1960), and essentially corresponds to the assumption that the disparity d varies smoothly in most parts of the image; see also Marr and Poggio (1979)), and the use of the occlusion indicator corresponds to the "ordering constraint" (i.e., the requirement that if $i > j$, then $i + d_i > j + d_j$, see Baker (1981); we put $\phi_d = 1$ whenever this constraint is violated).

2. Bayesian Formulation.

To formulate the estimation problem, we will consider the sequence g_L as "observations", while g_R will play the role of a set of parameters. Thus, from (3), we have (assuming, for simplicity that $\rho = \frac{1}{2}$):

$$P(g_L(i) = k \mid d, g_R) = P_{g|d}(k) =$$

$$= \begin{cases} 1-\epsilon, & \text{if } \phi_d(i) = 0 \text{ and } g_R(i+d_i) = k \\ \epsilon, & \text{if } \phi_d(i) = 0 \text{ and } g_R(i+d_i) \neq k \\ \frac{1}{2}, & \text{if } \phi_d(i) = 1 \end{cases}$$

The posterior distribution $P_{d|g}$ will then be:

$$P_{d|g}(d) = \frac{P_d \cdot P_{g|d}}{P_g} =$$

$$= \frac{1}{P_g} \exp\left[-\frac{1}{T_0} \sum_{i,j} V(d_i, d_j)\right] \cdot \prod_i \{[(1-\epsilon)\delta(g_L(i) - g_R(i+d_i)) +$$

$$+\epsilon(1 - \delta(g_L(i) - g_R(i+d_i))](1 - \phi_d(i)) + \frac{\phi_d(i)}{2}\}$$

where

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases}$$

As a prior model for the disparity field, we may use a first order MRF with generalized Ising potentials, such as the one presented in chapter 4. Other models

may also be used, including the coupled depth and line fields that we discussed in the previous chapters. For the present, let us assume that the simpler Ising model is adequate. Note that even when the matching problem is one–dimensional (we are asuming that there is no vertical disparity between the images, so that the matching can be done on a row–by–row basis), the two–dimensional nature of the prior MRF model for the disparity introduces a coupling between matches at adjacent rows. The posterior energy is:

$$U_P(d;g) = \frac{1}{T_0}\sum_{i,j} V(d_i,d_j) - \sum_i \ln\{[(1-\epsilon)\delta(g_L(i)-g_R(i+d_i)) +$$

$$+\epsilon(1-\delta(g_L(i)-g_R(i+d_i))](1-\phi_d(i)) + \frac{\phi_d(i)}{2}\}$$

Using the fact that for any $a, b \neq 0$

$$\ln[a\delta(x) + b(1-\delta(x)] = \delta(x)\ln a + (1-\delta(x))\ln b$$

we can write an equivalent expression for U_P (modulo an additive constant):

$$U_P(d;g) = \frac{1}{T_0}\sum_{i,j} V(d_i,d_j) + \frac{1}{2}\sum_i \phi_d(i)\ln 2 +$$

$$+\frac{\alpha}{2}\sum_i (1-\phi_d(i))\delta(g_L(i)-g_R(i+d_i)) \qquad (4)$$

where

$$\alpha = \ln\left(\frac{\epsilon}{1-\epsilon}\right)$$

3. Optimal Estimator.

It is possible to apply the general Monte Carlo algorithms developed in chapter 3 to approximate the optimal estimate $\hat{d}$ with respect to a given performance measure (such as the mean squared error). Their use in this case, however, is complicated by the introduction of the occlusion function ϕ_d in the posterior energy: the size of the support for this function equals the total number of allowed values for the disparity (see equation (2)). If this number is large, the computation of the increment in

energy, or of the conditional distributions (if the Gibbs Sampler is used) may be quite expensive. In many cases, however, the size of the regions of constant disparity is relatively large compared with the size of the occluded areas. In these cases, one can approximate the posterior energy by:

$$U_P(d) = \frac{1}{T_0} \sum_{i,j} V(d_i, d_j) + \frac{\alpha}{2} \sum_i \delta(g_L(i) - g_R(i + d_i)) \tag{5}$$

and increase significantly the computational efficiency (we have successfully used this approach to reconstruct the disparity of random dot stereograms).

In the one–dimensional case, it is also possible to extend the dynamic programming methods described in appendix 4.B to compute the MAP estimate; this extension is described in appendix 6.A.

An alternative approach to the solution of this problem is to implement the local constraints, generated by the prior MRF model, directly in a deterministic "cooperative network" of a given form (a "Winner–Takes–All" network) whose fixed point will correspond to the optimal solution. This will be done in section 6. First we present, in section 4 the definition of a "Cooperative Algorithm", and describe and analyze, in section 5, the previous work that has been done in this connection.

4. Cooperative Algorithms.

Consider the two–dimensional signal matching problem defined in section 2, and suppose that to each site i of the lattice Ω we associate a set of binary variables: $\{f_{i,d}, d \in Q\}$ (we will call this set the "i^{th} column" of the network f, and the set: $\{f_{i,d}, i \in \Omega\}$, the "disparity layer d" of the same network).

If a particular variable $f_{i,d} = 1$, it means that we assign to site i the disparity d (note that more than one disparity may be assigned to a node at a given time).

A "Cooperative Algorithm" (Marr and Poggio, 1976; it is also known as a "Cellular automata"; see Wolfram, 1983) is a rule for updating the state of the

network f. It can be represented formally as:

$$f_{i,d}(t+1) = F_{i,d}(f(t), t)$$

with the additional requirement that the interactions should be local, that is:

$$F_{i,d}(f(t), t) = F_{i,d}(\{f_{j,s}(t): j \in N_i, s \in Q\}, t)$$

where N_i is the (two–dimensional) neighborhood of site $i \in \Omega$. The idea is to define the functions F (i.e., the connections of the cooperative network) in such a way that the following local constraints are implemented:

(i) Compatibility with the observations: Each element $f_{i,r}$ should receive an "excitatory" external input proportional to the conditional probability $\Pr(g_L(i) = g_R(i+r) \mid d_i = r)$.

(ii) Smoothness: This corresponds to an implementation of the MRF prior model for the disparity: the likelihood that an element $f_{i,d}$ is turned "on" (i.e., is set equal to 1) should increase if the elements $\{f_{j,d}, j \in N_i\}$ are "on" (N_i is the neighborhood of i in the Markov model), so that excitatory connections should exist between these elements.

(iii) Uniqueness: Since in the final configuration f^* one and only one element of each column $\{f^*_{i,d}, d \in Q\}$ should be equal to 1, each element should have "inhibitory" connections with the other elements of the same column.

The operation of the network will be Synchronous if all its elements are updated in parallel at the same time, and Asynchronous if they are updated sequentially, one at a time. Note that one synchronous iteration is equivalent to $|f|$ (the number of elements of the network f) asynchronous ones (we will refer to $|f|$ succesive iterations as a Global Iteration), and that the evolution of the asynchronous network will depend, in general, on the order in which its elements are updated.

5. "Linear Threshold" Networks.

The first successful application of this approach (although not formulated in probabilistic terms) is the algorithm developed by Marr and Poggio (1976) for the

stereo disparity computation. They proposed a binary network of the form:

$$f_i(t+1) = \sigma(p_i)$$

$$\text{with} \quad p_i = \left[\sum_j f_j(t) w_{ij} + \eta_i - \theta\right], \qquad i,j \in \Omega \times Q; \tag{6}$$

$$\sigma(p) = \begin{cases} 1, & \text{if} \quad p \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad ;$$

$$w_{ij} \text{ satisfying} \quad w_{ij} = w_{ji}, \quad \text{for all} \quad i,j \in \Omega \times Q$$

$$\text{and} \quad f_i \in \{0,1\}, \quad \text{for all} \quad i$$

The parameters w_{ij}, η_i and θ must be chosen in such a way that the constraints to the solution of our problem are implemented locally. In particular, the smoothness constraint is implemented by defining:

$$w_{x,d,y,d} = 1, \quad \text{for} \quad y \in N_x \quad ; \quad x,y \in \Omega$$

where N_x is an excitatory neighbourhood of x. The uniqueness constraint, by:

$$w_{x,d,y,d'} = -\epsilon, \quad \text{for} \quad (y,d') \in M_{x,d}$$

with $M_{x,d}$ an inhibitory neighbourhood corresponding to multiple matches at x (see Marr and Poggio (1976) for a precise definition of these neighbourhoods), and

$$w_{x,d,y,d'} = 0 \quad \text{elsewhere.}$$

The compatibility with the observations is enforced by putting

$$\eta_{x,d} = f^0_{x,d} = \begin{cases} 1, & \text{if} \quad g_R(x+d) = g_L(x) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Although it has not been possible to this date to find a rigorous proof for the convergence of this algorithm, numerical experiments and a probabilistic analysis (Marr et. al., 1978) show that the synchronous network defined above will converge to reasonably good solutions for random dot stereograms portraying piecewise constant surfaces. However, this scheme has several problems (although some modifications

to get around them are suggested in Marr and Poggio, 1976 and in Marr et. al., 1978):

In the first place, the quality of the results degrades very fast as the density of the tokens in the stereogram decreases. Besides, it is not clear how to extend this formulation to the more interesting cases of slowly varying disparities, and different types of tokens placed in points that do not correspond to a regular lattice.

5.1. Asynchronous Algorithms.

We now consider algorithms of the form (6) that operate asynchronously. In this case, it has been shown (Hopfield, 1982) that if we choose the parameters in such a way that p_i is never 0 (this can be done, for example, if w_{ij} and η_i are integers, by giving θ a non–integer value), the "Energy" function:

$$E(f) = -\frac{1}{2}\sum_{i,j} w_{ij} f_i f_j - \sum_i f_i(\eta_i - \theta) \tag{8}$$

will decrease monotonically at every global iteration of the asynchronous algorithm in which the state of every element is updated, unless the network is at a fixed point.

It is interesting to note that with the parameter definitions given above for the stereo problem, the term

$$-\frac{1}{2} f_{x,d} \sum_{y \in N_x} f_{y,d}$$

in (8) will be negative only if all the spatial neighbors of the cell x on the same disparity layer are "on", and therefore corresponds to a smoothness constraint.The term

$$-f_{x,d} f^0_{x,d}$$

corresponds to the compatibility with the observations, and the remaining terms:

$$f_{x,d}\left[\theta + \frac{\epsilon}{2} \sum_{y,d' \in M_x,d} f_{y,d'}\right]$$

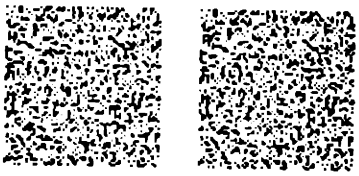
may be considered as an implementation of the uniqueness constraint, since their minimization requires that we have as few "on" cells as possible, and it penalizes explicitly the local non-uniqueness of the solution.

5.2. Experimental Performance.

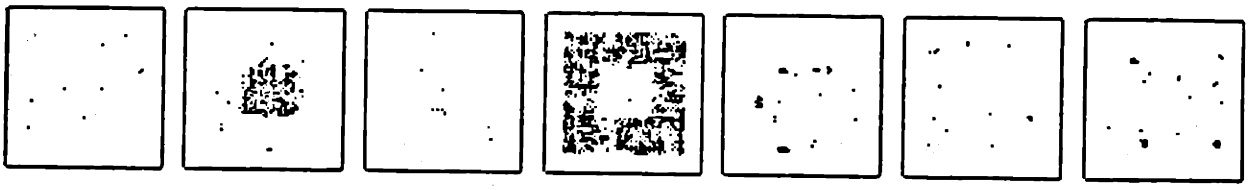
To study the performance of these algorithms, we implemented a simulator of both the synchronous and asynchronous networks. The "stimulus" used for the set of experiments performed, was a random dot stereogram portraying a square of 21×21 elements floating at disparity -2 in front of a flat background at disparity 0. Figure 24 shows this stereogram and the fixed points obtained by the synchronous and asynchronous algorithms.

In both cases, the behaviour of the algorithm shows two distinct phases: In the first iteration, most of the elements that are "on" on the wrong layers (and some on the correct ones) are turned "off" (see figure 24-b). As a result of this, at succeding iterations, the probability of having a cluster capable of growing is relatively high for the correct regions, which begin to fill in, and very small for the wrong ones, for which the remaining "on" cells are turned "off".

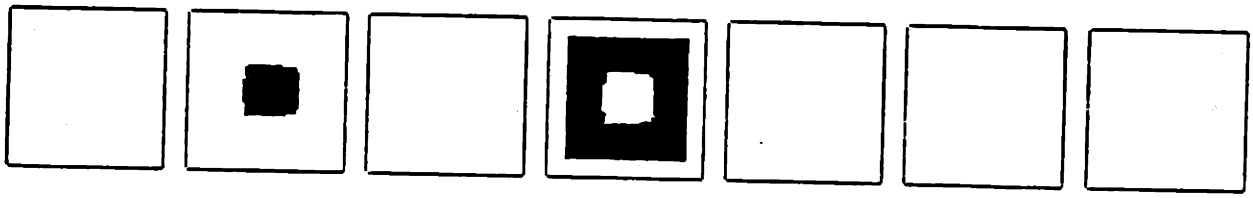
This form of operation causes that the precise shape of the boundaries between regions will depend on the exact shape and location of the random clusters that are formed after the first iteration on the correct layers. Also, it is easy to see that the form of the inhibitory neighbourhood (see Marr and Poggio (1976)) causes the cells lying on wrong layers along a narrow band near the edges of the background to be on the average less inhibited by the "on" elements in the correct layers (which in turn are less stimulated) than the interior points, making thus more likely the formation of wrong stable clusters in these regions. This effect is more pronounced in the asynchronous case, since a wrong cell that is left "on", can increase the excitation of a neighbouring one on the same global iteration, increasing the likelihood of a



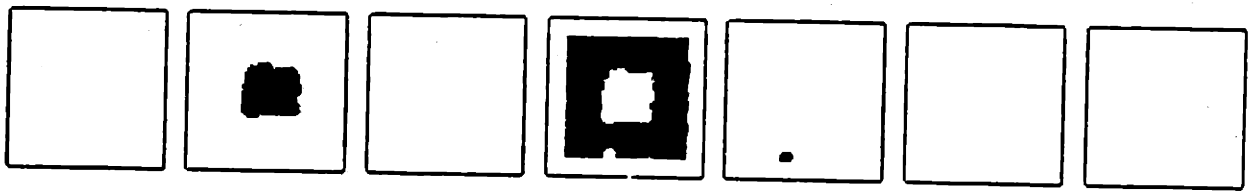
(a)



(b)



(c)



(d)

Figure 24. (a) Random dot stereogram portraying a 21×21 square at disparity -2. (b) State of the network after one iteration of the synchronous algorithm. (c) Fixed point for the Synchronous Algorithm. (d) Fixed point for the Asynchronous Algorithm.

stable cluster, whereas on the synchronous case, all the cells of the cluster must be left "on" at the same time.

For the values used for the parameters ($\epsilon = 2, \theta = 3.5$; see Marr and Poggio, 1976) the energy defined in (8) decreases monotonically at each global iteration of the asynchronous network, and thus, it converges to a configuration that is a local minimizer of this function. The correct solution will also correspond to a (different) local minimum; it is interesting to note, however, that in general it will not be the global one. It is easy to show, for example, that if the random dot stereogram portrays a region that has a ratio of area/perimeter less than a critical value (for the current value of the parameters this critical ratio is ≈ 13), this region will not be distinguished from the background in the configuration that globally minimizes the energy. This means that the use of simulated annealing to minimize (8) will not necessarily improve the solution; however, we have found that after the deterministic algorithm has converged, a few iterations of Metropolis algorithm at a moderate temperature (≈ 1) may be very effective for removing the clusters at wrong layers. This is illustrated in figure 25.

6. Winner–Takes–All (WTA) Networks.

Linear threshold networks are not the only form of local implementation of the constraints generated by the probabilistic formulation of our problem. A different possibility is to associate with each column $\{f_{x,d}, d \in Q\}$ a binary "Winner–take–all" synchronous network:

The input $u(x,d)$ to each cell corresponds to the excitatory input in the linear threshold case, that is, to the local implementation of the smoothness constraints and the compatibility with the observations.

The inhibitory terms (the uniqueness constraint) are implemented in the form of a WTA mechanism. The output (the new value of $f_{x,d}$) is given by:

$$f_{x,d} = \begin{cases} 1, & \text{if } \; u(x,d) = \max_{d' \in Q} u(x,d') \\ 0, & \text{otherwise} \end{cases} \qquad (9)$$

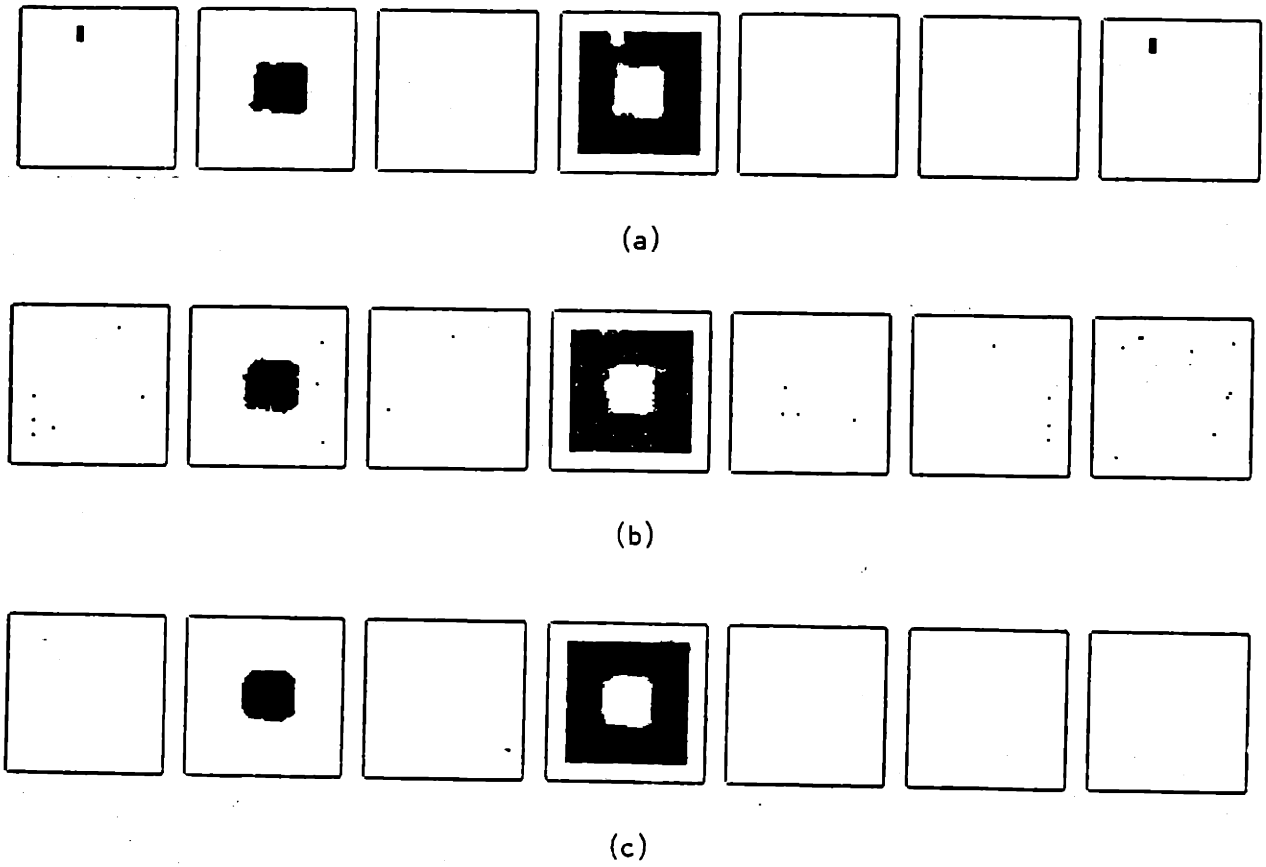


Figure 25. (a) Fixed point at $T = 0$. (b) State after 4 iterations at $T = 1$. (c) Fixed point at $T = 0$ with (b) as initial state.

This means that $f_{x,d}$ will be "on" at time $t+1$ only if it is maximally stimulated with respect to all the other elements in the same column at time t, and if it is "compatible enough" with the observations (see figure 26).

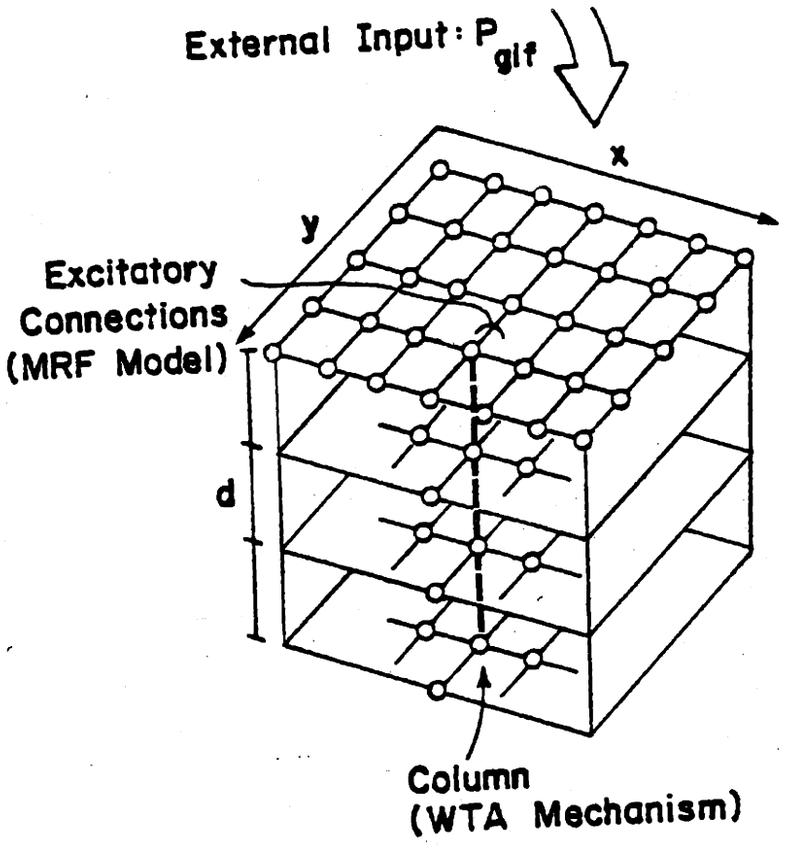


Figure 26. Winner-Takes-All network (see text).

This design has several advantages :

1. For dense stereograms, we will show that it converges to the correct solution in a small number of iterations.

2. For sparse stereograms, the algorithm will give, with high probability, the correct disparity at every location in which a matching token is present.

3. It exhibits a good performance with natural images portraying piecewise constant surfaces.

4. It is not necessary to process the whole domain Ω at the same time; a complete representation may be built up by defining local networks corresponding

to overlapping subregions that cover Ω. This feature enables the algorithm to process arbitrarily large images.

5. It can be extended in such a way that it can handle more complex situations, such as transparent and piecewise smooth surfaces.

Qualitatively, this improved performance can be explained as follows:

Unlike the linear threshold design, in the first iteration the WTA algorithm will only turn "off" cells that do not lie in the correct disparity layers. This will cause the cells that lie at the boundaries of clusters at the wrong layers to lose, in the subsequent iterations, against the corresponding strongly stimulated cells that lie in the interior of the "correct" regions. This will result in a progressive shrinking of the wrong clusters, and will end up with their disappearance.

This results in a faster convergence, since the size of the clusters that have to be killed is in general smaller than the size of the regions that the linear threshold algorithm has to fill in. Also, the boundaries between constant disparity regions will be more accurately localized.

The only situation in which this behavior will not take place, is when there is a significant overlap (due to accidental correlations in the images) between regions lying at different depths. In this case, the algorithm will not be able to solve the ambiguity correctly based only on smoothness considerations, and it will locate the boundary at a position, within the region of overlap, which will depend on the detailed shape of this region. Also, the solution will not be so clean in this case; a few cells, corresponding to different disparities at the same spatial position, may be left "on" in the final state (limit cycles involving some of these few cells are also possible).

This type of ambiguity (accidental overlap) is relatively frequent in sparse stereograms. However, the regions of overlap are typically "blank" regions (i.e., without tokens), and the algorithm will give the correct disparity at all token locations.

We will now make these considerations more precise. First, we will need some definitions.

1. Ω will be defined as the set of points lying on a finite square lattice.

2. We will use a second order MRF with Ising potentials as the prior model for the disparity field. Therefore, for each $x \in \Omega$, we define its neighborhood as:

$$N_x = \Omega \bigcap \{y \quad : \quad 0 < |x - y| < 2\} \tag{10}$$

3. Given a region $R \subseteq \Omega$, we define the set of its interior points (with respect to N_x) $I(R)$ as the set of points in R such that all its neighbors also belong to R :

$$I(R) = \{x \in R : |N_x \bigcap R| = |N_x|\}$$

In a similar way we define:

$$I^2(R) = I(I(R))$$

and so on. We call the points in R that are not interior: $x \in R - I(R)$, Boundary points of R. We will say that a region R is *connected* if, given any two sites $i, j \in R$, we can find a sequence of sites $\{i = i_0, i_1, \ldots, i_p = j\}$, with $i_k \in R$ for $k = 1, \ldots, p$, such that $i_k \in N_{i_{k+1}}$ for $k = 0, \ldots, p-1$.

4. Given a region $R \subseteq \Omega$, we define its Diameter $D(R)$ (with respect to N_x) as the smallest integer such that:

$$I^{D(R)+1}(R) = \emptyset$$

Alternatively, if we define an algorithm that deletes all the boundary points of a region at every step, the diameter of the region is the minimum number of steps necessary to completely delete the region.

5. The initial state of the network will be given by:

$$f^0_{x,d} = \begin{cases} 1, & \text{if} \quad g_R(x+d) = g_L(x) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

6. The WTA algorithm for this problem will have the particular form:

$$f_{x,d}(t+1) = \begin{cases} 1, & \text{if} \quad u_{x,d}(t) = \max_{d' \in Q} u_{x,d'}(t) \\ 0, & \text{otherwise} \end{cases}$$

$$u_{x,d}(t) = \alpha f^0_{x,d} + \sum_{y \in N_x} f_{x,d}(t) \tag{12}$$

7. We will assume that the set Ω can be covered by $M+1$ non-overlapping regions:

$$\Omega = R_1 \bigcup \ldots \bigcup R_M \bigcup O$$

and that the *correct solution* (i.e., the way the stereogram was generated) consists in assigning to every point in R_i the depth d_i:

$$f_{x,d_i} = 1 \qquad iff \qquad x \in R_i$$

The set O corresponds to the union of all the regions that are occluded in the left image (see figure 23), and therefore, for every $x \in O$, any depth assignment will be considered "correct".

8. Since we are assuming that the observations are perfect, the loading rules guarantee that

$$f^0_{x,d_i} = 1 \qquad \text{for every } x \in R_i$$

However, in many cases we will also have:

$$f^0_{x,d_j} = 1 \qquad \text{for some } x \in R_i \text{ and } d_j \neq d_i$$

due to accidental correlations in the images. A connected set W_j defined as:

$$W_j = \{x \quad : \quad f^0_{x,d_j} = 1 \text{ and } x \in R_i \text{ for some } d_j \neq d_i\}$$

will be called a *wrong cluster on layer j of R_i*.

9. We will say that a stereogram has *well defined boundaries* if there are no large wrong clusters overlapping the boundaries between adjacent regions. This means that every non-occluded point must have at least as many "on" neighbors at time 0 on the correct layer as in any other layer, i.e., for every region R_k and every point $x \in R_k$,

$$\sum_{y \in N_x} f^0_{y,d_k} \geq \sum_{y \in N_x} f^0_{y,d} \qquad \text{for all } d \neq d_k \tag{13}$$

10. A stereogram will be said to be *unambiguous* if for every region R_i and every wrong cluster W_j there is at least one point $x \in W_j \cap R_i$ which has less "on"

neighbors at time 0 on the wrong layer d_j than in the correct one d_i, i.e.,

$$\sum_{y \in N_x} f^0_{y,d_j} < \sum_{y \in N_x} f^0_{y,d_i} \tag{14}$$

We can now establish the following result:

Convergence Theorem: Given an unambiguous random dot stereogram with perfect observations (0 error rate) portraying M non–overlapping regions of constant depth with well defined boundaries, the WTA algorithm (12) with $\alpha \geq 8$ will converge to the correct solution in K iterations, where K is the diameter of the largest wrong cluster in Ω.

Proof:

1) First, we note that condition (13) guarantees that all the cells on the correct layers (which, by (11), are "on" at time 0) will remain "on" at time 1.

2) Condition (14) and the definition (12) guarantee that for every wrong cluster W_j on every region R_i there will be at least one point x that will be turned "off" in the first iteration. Then, for all points $y \in N_x \cap W_j \cap R_i$ we will have:

$$\sum_{z \in N_y} f^{(1)}_{z,d_j} < \sum_{z \in N_y} f^{(1)}_{z,d_i}$$

which implies that $f^{(2)}_{y,d_j} = 0$.

A recursive application of this reasoning establishes the theorem. ∎

Remarks:

1. For occluded regions, there will be no large clusters of "on" cells in any layer of f^0, and since the form of (12) precludes the growth over regions with $f^0 = 0$, if there are any isolated points for which $f^0_{x,d} = 1$, they will remain "on" in f^* (the fixed point of (12)); otherwise, $f^* = 0$ uniformly over these regions.

2. If the algorithm has ambiguous boundaries, we can still use this theorem to guarantee the convergence of the WTA algorithm to the correct solution outside the overlap regions. It is clear that if we define new non–overlapping regions $R'_1, \ldots R'_M$ with non ambiguous boundaries, and include the overlap areas in the set O, the theorem will guarantee that we get the correct solution in the new regions. In the overlap areas, the stable state of the network may include some leftover ambiguity ($f_{x,d} = 1$ for more than one d), and even limit cycles involving a few cells. However, these problematic areas will be confined to layers of unit width along the portions of the (final) boundaries that lie inside the overlap regions.

3. The probability of finding wrong clusters in a binary stereogram is related to the probability of finding a repeated subsequence on a Bernoulli sequence of length equal to the total number of disparity layers, and decreases exponentially with the number of cells belonging to each of these clusters. For dense sterograms (generated by a Bernoulli process with parameter $p = \frac{1}{2}$), the probability of finding a wrong cluster that contains a square of m cells per side can be bounded by

$$\Pr(\text{cluster}) \leq \frac{N_D|\Omega|}{2^{m^2+1}}$$

where N_D is the number of disparity layers, and $|\Omega|$ is the total number of cells in the lattice. On the other hand, a cluster of diameter k must contain at least a square of side $2k+1$. Thus, if $N_D = 7$ and $|\Omega| = 64^2$, for example, we can guarantee that, for dense stereograms, the algorithm will converge to the correct solution in less than 3 iterations with probability > 0.99.

4. For sparse stereograms, wrong clusters involving only "blank" areas will be very common, but those containing active tokens will be rare. This fact, together with remark 2, mean that, with high probability, tha WTA algorithm will find the correct disparity at all the sites that have active tokens. This has been confirmed by our experiments.

5. Algorithm (12) will not grow regions into occluded (uncorrelated) areas. Psychophysical experiments show that these areas should be included with the adjacent region that is at the greatest depth. It can be verified that an algorithm

such as the following:

$$f_{x,d}(t+1) = \begin{cases} 1, & \text{if } \sum_{y \in N_x} f_{y,d}(t) > 2 f_{x,d'}(t)\left[\sum_{y \in N_x} f_{y,d'}(t)\right], \quad d' \neq d \\ 0, & \text{otherwise} \end{cases}$$

with $f_{x,d}(0) = f^*_{x,d}$ (the fixed point of (12)), will converge to a solution in which these regions are correctly filled in, provided there are no wrong clusters in the occluded regions, and that each layer of constant d is allowed to converge separately, starting with $d = dmin = \min(d \in Q)$.

6. Note that even when $(x_1, x_2) \in \Omega$, $(x_1 + d, x_2)$ may lie outside Ω and so, if we load the network using (11), some cells near the boundaries of Ω may remain undefined, and (12) may give incorrect results. Therefore, we implicitly assume the existence of a larger region $\Omega_0 \supseteq \Omega$ such that for all $x \in \Omega$, $f^0_{y,d}$ is defined for $y \in N_x \cup \{x\}$ and $d \in Q$. Also, the operation of (12) should be understood in a modified sense, so that $f_{x,d}(t) = f^0_{x,d}$ for all $x \in \Omega_0 - \Omega$, all $d \in Q$, and all t.

A useful corollary establishes that it is not necessary to process all Ω at the same time, but that a complete representation can be built up by defining local networks corresponding to windows $S \subseteq \Omega$, provided that there is enough overlap between them. In particular, we will show that if we load the local network S in such a way that its initial state coincides with the initial state of the complete network at those cells, and if the algorithm operates only on the interior points of S, keeping the state of the boundary points fixed, then the final state of the local network at these interior points will correspond to the optimal solution:

Let $f^t_\Omega(x,d)$ and $f^t_S(x,d)$ be the state of the (x,d) cell at time t in the complete and local network respectively. We have:

Corollary 1: Suppose the conditions of the convergence theorem hold in Ω, and consider a set $S \subseteq \Omega$ such that the stereogram is not completely ambiguous in $S_1 = I(S)$ (i.e., condition (14) holds for every $x \in S_1$). Suppose that we load the local network f_S in such a way that for every $x \in S$, $f^0_S(x,d) = f^0_\Omega(x,d)$, for all $d \in Q$.

Then, algorithm (12), modified in such a way that $f_S^t(x,d) = f_S^0(x,d)$ for all t, all $x \in S - S_1$, and all $d \in Q$, will converge to a fixed point f_S^* for which $f_S^*(x,d) = f_\Omega^*(x,d)$ for all x belonging to unoccluded regions inside S_1.

Proof:

Consider a region R of constant disparity d such that $R' = R \cap S_1 \neq \emptyset$, and let B_1 be the intersection of R with the boundary of S_1. For every point $x \in R' - B_1$, $f_S^1(x,d) = 1$, by the same arguments as in the convergence theorem. For $x \in B_1$, $f_S^1(x,d) = 1$ too, since $f_S^0(y,d) = f_\Omega^0(y,d)$ for $y \in N_x$, and (13) holds in Ω. Therefore, for every $x \in R'$, $f_S^1(x,d) = 1$

On the other hand, for any wrong cluster $W_{d'} \subseteq R'$ in layer $d' \neq d$, since the stereogram is not completely ambiguous inside S_1, there will be at least one point $x \in W_{d'}$ such that $f_S^1(x,d') = 0$. Reasoning as we did before, we have that for all points $y \in N_x \cap W_{d'} \cap R'$ we will have:

$$\sum_{z \in N_y} f_{z,d_j}^{(1)} < \sum_{z \in N_y} f_{z,d_i}^{(1)}$$

which implies that $f_{y,d_j}^{(2)} = 0$.

Applying this reasoning recursively, we get, for every $x \in R'$, that $f_S^*(x,d) = 1$, and $f_S^*(x,d') = 0$, $d' \neq d$, which, together with the convergence theorem, completes the proof.∎

Note that $S - S_1$ defines the overlap that should exist among local windows, so that the complete representation, defined by

$$\Omega = \bigcup_j S_1^{(j)}$$

is correctly formed.

6.1. Numerical Results.

To test the performance of algorithm (12) with random dot stereograms, a simulator was implemented in a Symbolics 3600 computer. Figure 27 shows the fixed points corresponding to dense and sparse stereograms portraying a pyramid. As predicted by the theory, the convergence to the correct solution is fast (less than 4 iterations) in both cases. In the case of the sparse stereogram, the boundaries are slightly misplaced, but, as can be verified by direct inspection of the stereogram, all the dots are correctly located. The fixed point corresponding to the synchronous operation of (6) (obtained after 11 iterations) is also presented, for comparison. As we can see, the WTA algorithm (12) converges much faster to a much more precise result.

7. Recontruction of Real Images.

To apply this algorithm to the processing of real images, there are some modifications and extensions that should be made.

7.1. Neighborhood size.

It is possible to increase the robustness of algorithm (12) with respect to the presence of noise in the images by increasing the size of the excitatory neighborhood (i.e., by postulating a more global MRF prior model) and decreasing the value of the parameter α. This increased robustness is traded off by a decrease in resolution: small correct regions may be trated as "noise", and therefore disappear from the solution. Also, the shape of the piecewise constant regions may be altered (corners may be rounded and small concavities "filled in").

7.2. Token Selection.

The simple rule (11) is adequate for measuring the compatibility with the observations in the case of a synthetic image (such as a random dot stereogram). However, it will not work in the case of continuous-toned images of real objects. The reasons for this failure are manifold: the distribution of the reflected light

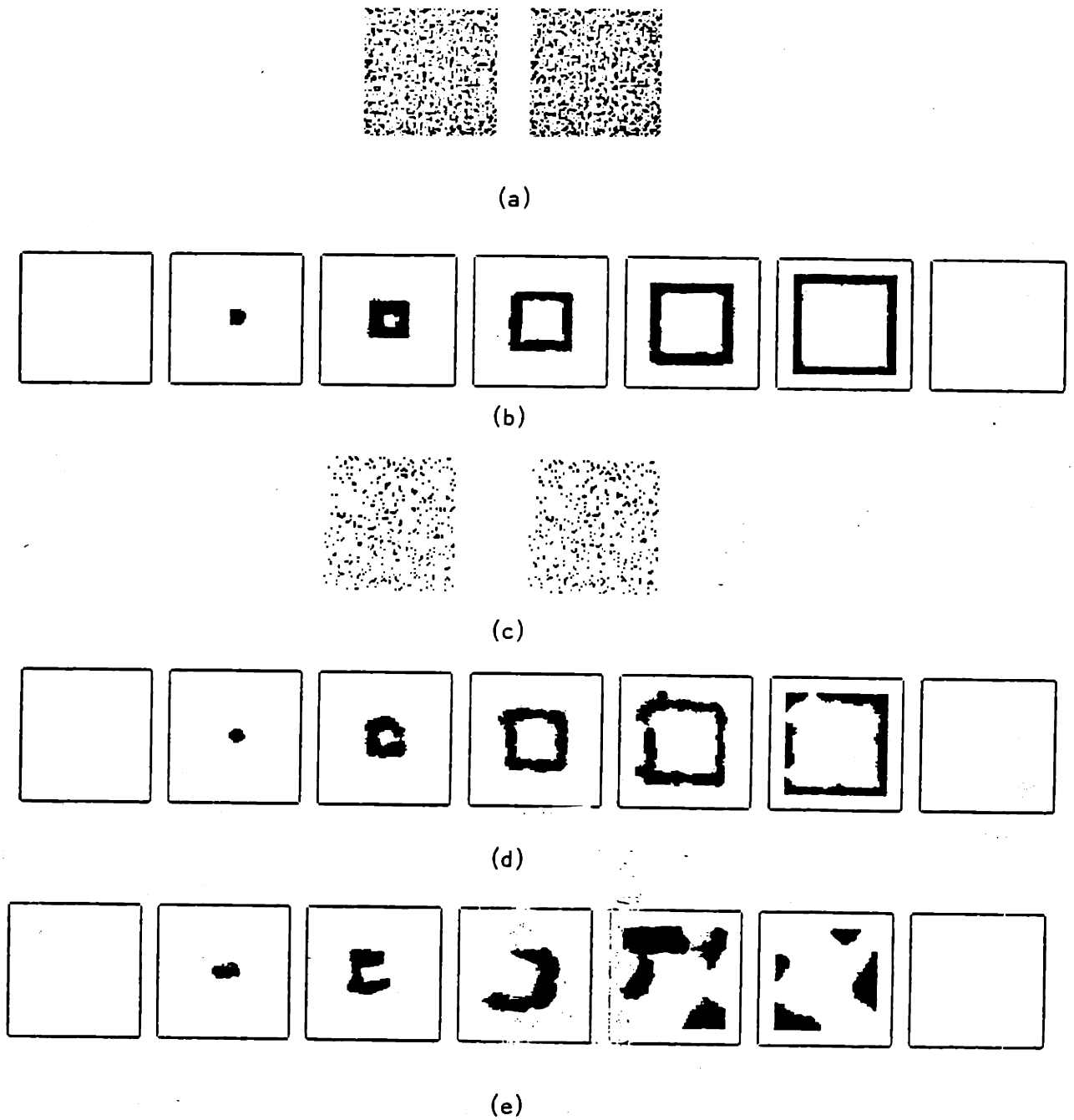


Figure 27. (a) Dense Stereogram (density = 0.4) portraying a pyramid. (b) Fixed point for algorithm (12) (c) Sparse stereogram (density = 0.1) portraying a pyramid. (d) Fixed point for algorithm (12). (e) Fixed point for the Synchronous algorithm (6).

varies as the viewpoint is changed (particularly the specular component), and the two retinas (cameras) may have different point spread functions, and be affected by independent sources of noise. This means that the model for the observation process given by equation (3) should be replaced by another that reflects the process of formation of natural images in a more realistic way. The use of a better model will cause the term $f^0_{x,d}$ in equation (12) to be replaced by a different compatibility measure $\eta_{x,d}$ which is obtained by first preprocessing the right and left images using an operator T whose output should be, ideally, invariant under the changes in viewpoint, optics, etc., and then computing a suitable defined distance D between the two processed images:

$$\eta_{x,d} = D(Tg_R(x+d), Tg_L(x)) \tag{15}$$

(note that η may be continuous-valued).

The new WTA algorithm will be:

$$f_{x,d}(t+1) = \begin{cases} 1, & \text{if } u_{x,d}(t) = \max_{d' \in Q} u_{x,d'}(t) \\ 0, & \text{otherwise} \end{cases}$$

$$u_{x,d}(t) = \alpha \eta_{x,d} + P_N(f^{(t)}, x, d) \tag{16}$$

The operator P_N is generated by the enlarged MRF model, and in general it will represent a weighted average of the values of the field in the enlarged neighborhood:

$$P_N(f, x, d) = \sum_{y \in N_x} c(|x-y|) f_{x,d} \tag{17}$$

where N_x is the extended neighborhood of x and $c(\cdot)$ denotes a set of parameters that depend only on the distance $|x-y|$, and are related to the prior MRF model for the disparity. f^0 may be chosen as:

$$f^0_{x,d} = \begin{cases} 1, & \text{if } \eta_{x,d} = \max_{r \in Q} \eta_{x,r} \\ 0, & \text{otherwise} \end{cases}$$

The convergence of this modified algorithm to the correct solution can still be guaranteed if condition (13) is replaced by the requirement that the cell corresponding

to the correct layer of every non-occluded point should be maximally stimulated at time 0, with respect to the other cells in the same column, by neighbors belonging to the same constant disparity region:

$$\alpha\eta_{x,d_i} + P_N^{(i)}(f^0, x, d_i) \geq \alpha\eta_{x,d} + P_N(f^0, x, d) \tag{18}$$

for every region R_i; every $x \in R_i$ and every $d \in Q$. $P_N^{(i)}$ is the operator P_N restricted to R_i:

$$P_N^{(i)}(f, x, d) = \sum_{y \in N_x \bigcap R_i} c(|x - y|) f_{y,d}$$

(this modification is necessary to cover the case in which a point near the boundary of a constant disparity region is partially stimulated by a wrong cluster outside this region which may disappear in succeeding iterations).

Condition (14), i.e., the requirement that every wrong cluster has less "on" neighbors at time 0 on the wrong layer than in the correct one, can now be expressed by requiring that for every region R_i and every wrong cluster W_j on layer j of R_i, there is at least one point $x \in R_i \bigcap W_j$ such that:

$$P_N(f^0, x, d_j) < P_N(f^0, x, d_i) \tag{19}$$

Under these conditions, it is easy to use the same arguments of the proof of the convergence theorem to verify the convergence of algorithm (16). It should be remarked that conditions (18) and (19) are sufficient, but by no means necessary; (16) may converge to the correct solution even if they are violated by a particular stereogram.

The determination of the optimal operators D and T in equation (15) is a difficult — and as yet unsolved problem. One scheme that has often been used is to define T as a convolution operator whose kernel is the Laplacian of a Gaussian function , and T as:

$$T(a, b) = \begin{cases} 1, & \text{if } ab > 0 \\ 0, & \text{otherwise} \end{cases}$$

(see Marr and Poggio, 1979). The rationale for this choice is that the zero crossings of the convolution with the Laplacian operator should pick the places where large intensity changes occur in both images (i.e., it acts as an "edge detector"), while the Gaussian kernel has the effect of smoothing out the "irrelevant" edges and filtering out the noise. One difficulty, however, is that if the Gaussian mask is large enough to produce the desired effect, it will also introduce errors in the localization of the zero crossings of the convolved images, which will translate into errors in the depth of the reconstructed surface (see Clark and Lawrence, 1985).

We have found that the normalized absolute value of the Laplacian of the difference between left and right images:

$$\eta_{x,d} = \frac{-v(x,d) + \max_{r \in Q} v(x,r)}{\max_{r \in Q} v(x,r) - \min_{r \in Q} v(x,r)}$$

with

$$v(x,d) = |\nabla^2 (g_R(x+d) - g_L(x))| \qquad (20)$$

has relatively good experimental behavior, but clearly, much more research is needed in this area.

It is important to note that the definition of η will affect the performance of the WTA algorithm, since it will determine the extent to which conditions (18) and (19) hold in the initial state; the structure of the WTA network, however, is independent of the choice of η, so that the experimentation with different definitions can be done very efficiently.

7.3. Uniqueness Constraint.

The definitions (12) and (16) imply the enforcement of the constraint:

"Each point in the left image should be matched by only one point in the right image".

That is to say, we are enforcing the uniqueness constraint along the left eye line of sight. It is also possible to include explicitly the corresponding constraint for

the right eye (as in Marr and Poggio, 1976). This is done by replacing (16) (or (12)) with:

$$f_{x,d}(t+1) = \begin{cases} 1, & \text{if } u_{x,d}(t) = \max_{d' \in Q} u_{x,d'}(t) \\ & \text{and } u_{x,d} = \max_{k:d+k \in Q} u_{x-k,d+k} \\ 0, & \text{otherwise} \end{cases}$$

For perfect observations, this additional constraint is redundant. If noise or other distortions are present, however, this scheme will have better performance, since the disparity of "doubtful" points will be left unassigned (the corresponding values of the disparity in these locations may be determined after convergence by the robust surface reconstruction techniques described in chapter 5).

As an example of the application of this technique, the processing of a stereo pair of aerial photographs is illustrated in figure 28 (this stereo pair is the same that was used in chapter 5; see figure 19). Although it is difficult to assess objectively the performance of an algorithm on this type of images, the quality of these results seems at least equivalent to that obtained by state-of-the-art systems (see Grimson, 1984).

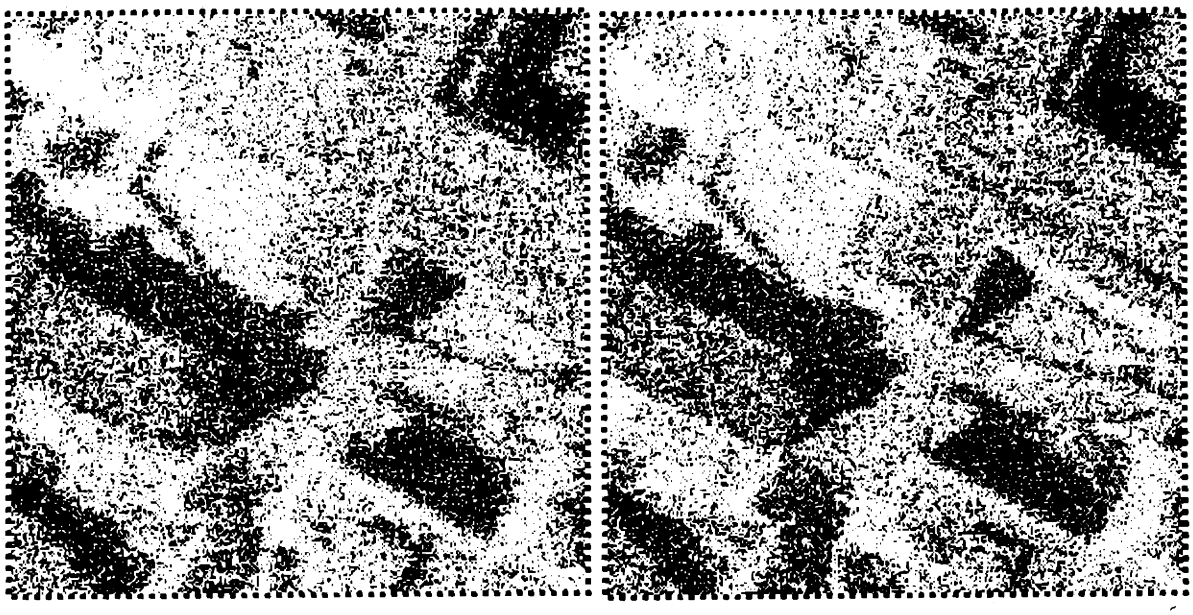
7.4. Piecewise Smooth Surfaces.

The WTA scheme can also be applied to reconstruct disparity surfaces that are piecewise smooth. To do this, it is only necessary to modify the definition of the operator P_N (equation (16)), so that cells at nearby depths are also taken into account. Notice that, in order to be consistent with the WTA mechanism, only the maximum contribution for any given column should be considered. The modified operator is:

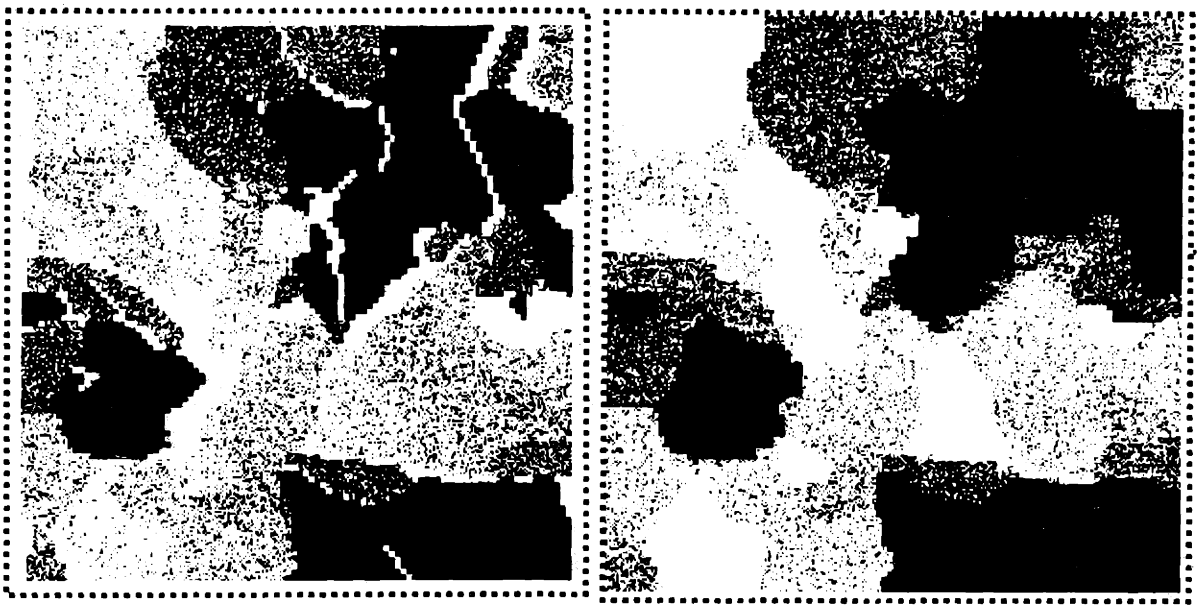
$$P_N(f,x,d) = \sum_{y \in N_x} \max_{r \in N_d} \{c(|x-y|, |d-r|) f_{y,r}\} \qquad (21)$$

where $c(\cdot,\cdot)$ is some fixed decreasing function of its arguments, and N_d is a disparity neighborhood defined as the intersection of a closed interval with the set of allowable disparities:

$$N_d = [d-p, d+p] \bigcap Q$$



(a)



(b) (c)

Figure 28. (a) Stereo pair of aerial photographs. (b) Final state of the WTA network (disparity is coded by grey level; white areas have no assigned disparity). (c) Reconstructed suface, obtained using the algorithm described in section 6 of chapter 5.

where p is a positive constant.

The sufficient conditions for the convergence of the modified algorithm, namely, that the stereogram should be unambiguous and have well defined boundaries with respect to the the modified operator P_N, can also be expressed in the form given by equations (18) and (19), but now a *wrong cluster* W_j should be defined as a connected region on the disparity layer d_j such that $f^0_{x,d_j} = 1$, and $d_j \neq d^*(x)$ for all $x \in W_j$, where $d^*(x)$ is the true disparity at point x. The proof of the convergence theorem is straightforward, but the interpretation of these conditions is not obvious, and in practice, they are very difficult to verify, so that at this point, the performance of this algorithm should be assessed experimentally.

Pradzny (1984) (see also Pollard et. al., (1984)) has obtained good results for the reconstruction of piecewise smooth and "transparent" surfaces (i.e., stereograms portraying sets of small interspersed patches that belong to two smooth surfaces, one in front of the other) using an operator of the form:

$$P_N(f, x, d) = \sum_{y \in N_x} \sum_{r \in N_d} \{c(|x - y|, |d - r|) f_{y,r}\}$$

We believe that the use of (21) should improve the performance in these cases.

8. Discussion

In this chapter we have studied a class of recontruction problems that arise when the conditional distribution of the observations is a multimodal function, which causes the solution to remain ambiguous, even for arbitrarily high signal to noise ratio. We identified the signal matching problem as one of the most important instances of this class, and gave a probabilistic formulation for it using a MRF model to model the disparity surface, so that the optimal estimation algorithms derived in chapter 3 could be applied.

We then presented a different approach to the solution of the problem in which the constraints derived both from the prior MRF model for the disparity field and from the observations are implemented directly as excitatory connections

on a three–dimensional cooperative network of processors (or "cells") with binary state space. The steady state of this network can be unambiguously interpreted as a disparity surface only if there is exactly one processor in each column whose state is equal to 1. This imposes a uniqueness constraint which can be enforced either by introducing inhibitory linear connections, or by a "Winner–take–all" mechanism that operates within each column. We showed that, for high signal to noise ratio, it is possible to define precise sufficient conditions (which are usually met in the case of synthetic images) for the convergence of the state of this "WTA" network to the correct solution in a small number of iterations.

The experimental performance of this algorithm with random dot stereograms is excellent; it produces accurate reconstructions in a very short time (in less than 5 iterations). In the case of the reconstruction of real objects from stereoscopic photographs, this algorithm — with some modifications — produces results comparable with those obtained by more complicated schemes that are considered "state of the art", and it has the advantage of being directly implementable in parallel hardware.

It should be noted that the performance of the stereoscopic vision of human beings on similar data is still dramatically superior to that of this, or any other existing artificial system. Some issues that should be addressed for the development of more effective algorithms are the following:

(i) More realistic models for the observation process that take into account the nature of the relative distortions of the left and right images should be constructed. This should lead to the definition of optimal combinations of tokens for the matching process. The precise nature of the optical system used (which may cause problems like non–horizontal epipolar lines; vertical disparities, etc.) should also be taken into account.

(ii) The use of more sophisticated prior models for the disparity field — including a coupled line field as described in chapter 5 — should be investigated.

(iii) Since the intensity edges and the regions of uniform intensity (or uniform texture) of the images are natural candidates for becoming stereo matching tokens, and the location of depth and intensity (or texture) edges is likely to be correlated in a natural scene, the integration of edge detection;

image segmentation; stereo matching and surface reconstruction into a single estimation process should produce very good results. The Bayesian approach, and the use of coupled MRF models for describing surfaces and edges that we have presented in this thesis should provide a unified framework for performing this integration. We discuss this point further in the next chapter.

Appendix 6.A

DYNAMIC PROGRAMMING APPROACH TO SIGNAL MATCHING

Consider the one-dimensional version of the signal matching problem described in section 2. To compute the MAP estimate, we need to find the global minimum of:

$$U_P(d;g) = \frac{1}{T_0}\sum_{i,j} V(d_i, d_j) + \frac{1}{2}\sum_i \phi_d(i)\ln 2 +$$

$$+\frac{\alpha}{2}\sum_i (1-\phi_d(i))\delta(g_L(i) - g_R(i+d_i)) \qquad (1)$$

(i.e., equation (4)) The use of the dynamic programming algorithm described in appendix 4.B is complicated by the fact that, given the boundaries $\mathcal{L}_n$ between regions of constant disparity, the optimal estimate for d in the interval $(L_i, L_{i+1}]$ depends on the estimate on $(L_{i-1}, L_i]$, since this last choice determines the extent of the occluded region.

However, if we assume that the size of the regions of constant disparity is relatively large compared with the size of the occluded areas (as it normally happens in most practical cases), we can estimate d given $\mathcal{L}_n$ using the formula:

$$\hat{d}((L_i, L_{i+1}]) =$$

$$= \{k: \sum_{i=L_i+1}^{L_{i+1}} \delta(g_L(i) - g_R(i+k)) < \sum_{i=L_i+1}^{L_{i+1}} \delta(g_L(i) - g_R(i+l)),\ \text{for all } l \in Q\} \qquad (2)$$

Defining:

$$G_{k,l} = \sum_{i=k+1}^{l} \delta(g_L(i) - g_R(i + \hat{d}((k,l])))) \qquad (3)$$

and

$$\Delta_i = \max(0, \hat{d}_{i-1} - \hat{d}_i)$$

(note that Δ_i corresponds to the length of the occluded region when a change in the estimated disparity occurs). we get that

$$U_P(d;g) = \frac{n}{T_0} + \hat{U}(\mathcal{L})$$

with

$$\hat{U}(\mathcal{L}) = G_{1,L_1} + \Delta_1 \ln 2 + G_{L_1+\Delta_1+1,L_2} +$$

$$+\Delta_2 \ln 2 + \ldots + G_{L_n+\Delta_n+1,N}$$

We can now perform the global minimization of U using the dynamic programming scheme of appendix 4.B. In this case, however, it is better to use "forward" recursions, (in the sense that now $F_j(k)$ will represent the cost associated with putting j boundaries, in the best possible locations, in the interval $[1,k]$), because occlusion, as we have defined it, always takes place from left to right. We have then:

$$F_0(k) = G_{1,k}$$

$$L_0(k) = 1$$

$$F_{j+1}(k) = \inf_{i<k} \{G_{i+\Delta_j+1,k} + F_j(i) + \Delta_j \ln 2\}$$

$$L_{j+1}(k) = \{L{:}G_{L+\Delta_j+1,k} + F_j(L) + \Delta_j \ln 2 = F_{j+1}(k)\}$$

The optimal location of the boundaries, for any given n is:

$$S_n = \{L_n(N), L_{n-1}(L_n(N)), \ldots, L_1(L_2(\ldots(L_n(N))\ldots)\}$$

The optimal configuration is computed using (2), and the corresponding energy, using (1).

Note that as the size of the regions of constant disparity decreases, (2) may not be well defined (the optimal estimate $\hat{d}$ may not be unique) and a more complex optimization procedure may be required.

Chapter 7

CONCLUSIONS

In this thesis we have presented a probabilistic approach to the solution of a class of perceptual problems. We showed that these problems can be reduced to the recontruction of a function on a finite lattice from a set of degraded observations, and derived the Bayesian estimators that provide an optimal solution. We have also developed efficient distributed algorithms for the computation of these estimates, and discussed their implementation in different kinds of hardware. To demonstrate the generality and practical value of this approach, we studied in detail several applications: the segmentation of noise–corrupted images; the formation of perceptual clusters; the recontruction of piecewise smooth surfaces from sparse data and the reconstruction of depth from stereoscopic measurements.

This methodology also permits, in principle, the incorporation of more than one modality of observations into a single estimation process, as well as the simultaneous estimation of several related functions from the same data set. This makes one hope that this framework could be useful in the solution of difficult problems that require such an integrated approach. We mention two examples:

1. We mentioned in chapter 6 that the stereo matching problem in real situations has not been solved yet in a satisfactory way. The same can be said of other related perceptual problems such as: edge detection; image segmentation; the recovery of the shape of an object from a single two–dimensional image (the "shape form shading" problem), and the segmentation of a scene into distinct objects, as well as the recovery of their three–dimensional structure from the analysis of images formed at successive instants of time (the "structure from motion" problem). All these problems are obviously related, and it is intuitively clear that the individual solutions that can be obtained should improve if the mutual constraints that the

solution of each individual problem imposes on the others were taken into account. Thus, the presence of a brightness edge should increase the likelihood of a depth edge, and viceversa; the depth estimated from stereo should be compatible with the shape derived from shading; points belonging to the same region in an image should move together, etc. We believe that these constraints can be incorporated in the potential functions of the corresponding MRF models (in particular, of the coupled fields that represent the "lines" or edges in each case; see chapter 5).

2. The processing and interpretation of geophysical information (as is done, for example in oil prospecting) attempts to reconstruct subterranean geological structures from information provided by a set of qualitatively different measurements, such as those obtained by: gravimetric and magnetometric surveying; reflexion seismology; measurements of physical properties taken vertically along bore holes ("well logs"), etc. Since all these measurements are obtained independently, their joint conditional probabilities can be easily determined, and since all of them refer to the same physical structures, their processing can, in principle, be integrated into a single estimation process, which should greatly increase the reliability of the results.

The above considerations may be taken one step further. Ultimately, the results one is interested in are not only the quantitative reconstruction of some surfaces, but the symbolic description of the scene in terms of functional structures or "objects". On the other hand, the prior knowledge about the occurance of a particular object or class of objects might greatly simplify the tasks of the "low level" processors (for example, a letter recognition algorithm should greatly benefit from the use of context, given the probabilities of occurance of certain letter combinations or words). The Bayesian approach provides a common "language" that may allow these low-level and high-level (or symbolic) processes to communicate and mutually enhance their performance.

As a simple example of this situation, suppose that we are interested in finding a symbolic description of a binary pattern f in terms of a set of geometric objects (such as squares, triangles, etc.) that are characterized by some parameters (such as position, orientation, size, etc.) for whose values we have some prior probability

knowledge.

Given a description $\mathcal{D}$, i.e., a list of objects with a set of particular values for their parameters, we can find a binary field q which corresponds to the boolean sum of the indicator functions of the objects included in $\mathcal{D}$:

$$q_i = \begin{cases} 1, & \text{if an object in } \mathcal{D} \text{ covers pixel } i \\ 0, & \text{otherwise} \end{cases}$$

We can now write the joint prior distribution for the field f (which represents the actual intensity of the noise-free image) and its description as:

$$P(f, \mathcal{D}) = P(f \mid \mathcal{D})P(\mathcal{D})$$

To compute $P(f \mid \mathcal{D})$, we assume that f is a first order MRF whose configuration is biased by $\mathcal{D}$:

$$P(f \mid \mathcal{D}) = \frac{1}{Z} \exp[-\frac{1}{T_0} \sum_{i,j} V(f_i, f_j) + \lambda \sum_i q_i f_i]$$

$P(\mathcal{D})$ can be computed from the prior probabilities for the occurance of each type of object, and from the prior distributions for the values of the corresponding parameters. Since the conditional distribution of the observations depends directly only on f, the posterior distribution will be:

$$P(f, \mathcal{D} \mid g) = \frac{P(g \mid f)P(f, \mathcal{D})}{P(g)}$$

where $P(g)$ is a constant. From this expression we can compute the optimal estimates for f and $\mathcal{D}$ using methods similar to the ones developed here.

We will now present a summary of our main results and a list of some interesting open technical questions.

1. Summary of our Main Results.

1.1. Optimal Bayesian Estimators.

Several researchers have used Bayes theory and MRF models for the restoration of piecewise uniform images. It has been implicitly assumed by all of them that the maximization of the posterior probability is the best possible performance criterion. We have shown that it is possible to choose other criteria that are better adapted to each particular problem, and have derived the corresponding optimal estimators, which not only improve substantially the quality of the results (particularly for low signal to noise ratios), but also lead to more efficient and better behaved computational schemes.

1.2. General Monte Carlo Algorithms.

We have shown that the optimal Bayesian estimators can be obtained from the observation of the equilibrium behavior of a MRF (which in physical terms correspond to a ferromagnet subject to a spatially varying external magnetic field). This behavior can be effectively simulated by Monte Carlo procedures which generate a regular Markov chain with an invariant Gibbs measure.

This method differs from "simulated annealing" (which has been used to approximate the MAP estimator) in that it is based on the collection of statistics of the evolution of the chain at a fixed temperature, while the latter attempts to find the ground state of the coupled system by slowly decreasing it. From a computational viewpoint, our method exhibits a faster and more consistent convergence behavior.

1.3. Parallel Implementations.

The implementation of this general Monte Carlo procedure in parallel hardware was discussed. We proved that the Gibbs sampler (but not the Metropolis or Heat Bath algorithms) will produce consistent results in this case.

1.4. Reconstruction of Piecewise Constant Funcions.

The problem of reconstructing a piecewise constant function from noisy (but dense) observations was formulated in probabilistic terms, and the form of the optimal estimators derived. For the one-dimensional case, we presented a deterministic algorithm with minimal complexity which computes (exactly) the MAP estimate of binary fields. For the two-dimensional case, we presented a

method for improving the computational efficiency of the "Simulated Annealing" scheme for approximating the MAP estimator, and derived a fast algorithm for approximating the optimal (MPM) Bayesian estimator.

We also presented a maximum likelihood procedure, which based on an analysis of the residual ("innovations") process permits the simultaneous estimation of the field and the parameters of the system. We applied this technique to the construction of a parameter–free algorithm for the reconstruction of arbitrary binary patterns.

1.5. Formation of Perceptual Clusters.

We suggested that the process of formation of perceptual clusters of certain dot patterns can be modeled in terms of the estimation of binary images corrupted by multiplicative noise, and illustrated the application of our estimation algorithm to this task.

1.6. Reconstruction of Piecewise Continuous Surfaces.

The problem of simultaneously detecting the discontinuities and recontructing a piecewise smooth surface from sparse observations was cast in the Bayesian framework. A model consisting of two coupled MRF's: one representing the depth and the other the boundaries between continuous regions, was adapted to our problem. Since the straightforward use of the general Monte Carlo algorithm for finding the optimal estimate is computationally unfeasible in this case, an approximation (which showed an excellent experimental performance with both synthetic and "real" data) was derived and implemented. We also developed, and heuristically justified a fast algorithm that produces results that are practically indistinguishable from the optimal ones. The implementation of these procedures in digital parallel hardware, as well as in hybrid and analog networks was also discussed.

1.7. Signal Matching.

We presented a class of problems that is characterized by the fact that the conditional probability distribution of the observations $P(g \mid f)$ is multimodal (as a function of f), which means that the solution remains ambiguous, even for arbitratily

high signal to noise ratios. We studied a prototype problem of this class: the signal matching problem (in particular, the reconstruction of depth from stereoscopic pairs of images), and showed that it is possible, in principle, to find the solution using the general estimation procedures that we have developed (although the computational cost will be high in the general case). We also presented a different scheme, which is based on the direct implementation of the local constraints (generated by the probabilistic model) in a highly distributed cooperative network of a particular form: a "Winner–Take–All" network, and showed that the state of this network will converge to the correct solution in a few iterations (in the high SNR case). The application of this technique to the reconstruction of the depth of real objects from stereoscopic photographs was discussed, and some modifications to the algorithm were introduced, which permitted us to produce results which compare favourably with those of other "state of the art" algorithms.

2. Open Technical Questions.

2.1. Stochastic Models.

We have shown throughout this work the richness and versatility of simple (first and second order) MRF models. It is clear,however, that there are classes of physical structures whose behavior cannot be adequately modeled by these processes (as a simple example, consider images formed by clusters of blobs of certain average size). There have been some attempts to model these and other "textured" patterns via a hierarchy of independent MRF's: one that represents the structure of the image, in terms of regions of uniform texture, and individual models for each textured regions. This representation, however, is not very convenient for estimation purposes. A more rigorous approach has been suggested by Grenander (1984), who has proposed the use of generalized Markovian fields to model complex patterns; these fields consist of several layers of "generators", which in the first layer correspond to grey levels, and in the succeeding ones, to features of increased complexity (lines, corners, etc.). It is not clear, however, how to use this approach to construct models of textured images; objects of different shapes, etc.

These considerations suggest the need for much more research in this area, which should include, perhaps, the use of probabilistic models that are not based on the Gibbs distribution.

2.2. Multiple Scale Representations.

It is the current view that the production of high-level (symbolic) descriptions of a scene should be mediated by the construction of numerical descriptions of the surfaces involved at different "scales". The parameters that describe a MRF play in some sense the role of scale parameters (see figure 1 of chapter 1; section 5 of appendix 4.B and section 6 of chapter 5); this identification, however, is not completely satisfactory. A good multiscale representation should feature not only a progressive blurring of detail, but the aggregation of substructures into larger units in a way that is not accomplished by the current algorithms.

2.3. Parameter Estimation.

Intimately linked with the previous questions, is the determination of the optimal set of parameters of a given model from noisy samples. The maximum likelihood method that we have presented here (see chapter 5) becomes computationally unfeasible as the complexity of the model (the dimensionality of the parameter space) increases; therefore, alternative procedures need to be derived (for instance, the use of time-varying algorithms, such as the one presented in section 6 of chapter 5 should be more rigorously investigated).

A related (and more difficult) question is the selection of the optimal model from a certain class given only the noisy observations. It is possible that the ideas of Rissanen (1978, 1981, 1983) about "minimum description length" schemes, and also of Akaike (1977) about generalized maximum likelihood methods could be useful in this connection, although the high computational complexity of the present problem might limit the applicability of these techniques.

2.4. Fast Algorithms.

The practical use of the general Monte Carlo estimation algorithms of chapter 3 is limited by the relatively large number of iterations needed for the convergence

of these systems. A very important question, then, is how to improve on the convergence time without sacrificing the power of these methods. The use of "multigrid" type strategies (Brandt, 1973; Terzopoulos, 1984), which in the present case may take the form of "block–spin"" algorithms, such as the one presented in chapter 4 (see also White, 1983) should be investigated.

Also in this connection, it should be interesting to find more rigorous justifications for the performance of the fast deterministic schemes that we have developed, based on heuristic considerations, in chapters 4 and 5, to see if it is possible to find some general principles that may guide the extension of these schemes to other, more general cases.

2.5. Analog Computers.

It would be interesting to actually construct prototypes of the hybrid and analog networks described in chapter 4 and 5, to assess the practicality and performance of such schemes. A more intriguing possibility is to exploit the isomorphism between the estimation process of a MRF from noisy data, and the equilibrium behavior of a ferromagnet with a coupled, spatially varying external field (see chapter 3), to construct very fast, special purpose "quantum" computers to perform the former task.

REFERENCES.

Abend K. "Compound Decision Procedures for Unknown Distributions and for Dependent States of Nature". in Pattern Recognition pp. 207-249 L. Kanal, ed. Thompson Book Co. Washington, D.C. (1968).

Akaike H. "On the Entropy Maximum Principle". Applications of Statistics. Ed. P.R. Krishnaia. North Holland, Amsterdam (1977) pp 27–41.

Baker H.H. and Binford T.O. "Depth from Edge and Intensity Based Stereo" Seventh International Joint Conference on Artificial Intelligence, August 1981, 631-636.

Besag J. "Spatial Interaction and the Statistical Analysis of Lattice Systems". J. Royal Stat. Soc. B 34 75-83 (1972).

Blake A. "The Least–Disturbance Principle and Weak Constraints" Patt. Rec. Let. 1 393-399 (1983).

Blake A. "Parallel Computation in Low–level Vision" Ph.D. Thesis. Univ. of Edinburgh (1985).

Blumenthal A.F., Davis L.S. and Rosenfeld A. "Detecting Natural Plateaus in One Dimensional Patterns". IEEE Trans. on Computers. Feb. 1977 p. 178–179.

Brandt A. "Multilevel Adaptive Technique for Fast Numerical Solution of Boundary Value Problems" Proc. Third Inter. Conf. Num. Meth. Fluid Mech. Paris (1972). Lecture Notes in Physics. Vol 18. Springer Verlag. Berlin.

Cannny J.F. "Finding Edges and Lines in Images". Massachusetts Institute of Technology. Artificial Intelligence Laboratory. TR-720 (1983).

Clark J.J. and Lawrence P.D. "Scale Space Analysis of Stereo Disparity Errors". Univ. of British Columbia. Faculty of Applied Science. T.R. No. 3. (1985).

Cohen F.S. and Cooper D.B. "Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields" Brown University Laboratory for Engineering Man/Machine Systems. Tech. Report LEMS-7 (1984).

Courant R. and Hilbert D. "Methods of Mathematical Physics". Vol I. Interscience, London (1953).

Cramer H. "Mathematical Methods of Statistics". Princeton Univ. Press. (1946)

Cross G.C. and Jain A.K. "Markov Random Field Texture Models". IEEE Trans. PAMI 5 25–39 (1983).

Davis L. "A Survey of Edge Detection Techniques". Computer Graphics and Image Processing. 4 (1975) 248-270.

Davis L. and Rosenfeld A. "Cooperating Processes for Low Level Vision: A Survey" Art. Intell. 17, 245 (1981)

Elliot H., Derin R., Christi R. and Geman D. "Application of the Gibbs Distribution to Image Segmentation". Univ. of Massachusetts Technical report (1983).

Fahlman S.E. Hinton G.E. and Sejnowski T.J. "Massively Parallel Architectures for AI: NETL, Thistle and Boltzmann Machines". Proc. Nat. Conf. on A.I. AAAI-83, Washington, D.C. (1983) 109-113.

Feldman J. "Memory and Change in Connection Networks" TR96 C.S. Dept. Univ. of Rochester. (1981)

Feller W. "An Introduction to Probability Theory and its Applications" Vol I. Wiley, New York (1950).

Gallager R.G. "Information Theory and Reliable Communication" J. Wiley (1968).

Geman S. and Geman D. "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images". IEEE Trans. Pattern Analysis and Machine Intelligence 6, 721–741 (1984).

Geman S. and Hwang C. "Diffusions for Global Optimization" preprint. Brown University (1984).

Gidas B. "Non-Stationary Markov Chains and the Convergence of the Annealing Algorithm". Unpublished Manuscript. Rudgers University (1984).

Good I.J. "Good Thinking. The Foundations of Probability and Its Applications". Univ. of Minnesota Press. (1983).

Grenander U. "Tutorial in Pattern Theory" Div. of Applied Math. Brown University (1984).

Grimson W.E.L. "From Images to Surfaces". MIT Press, Cambridge, Mass. (1981).

Grimson W.E.L. "A Computational Theory of Visual Surface Interpolation". Phil. Trans. R. Soc. London., B, 298, 395–427 (1982a).

Grimson W.E.L. "A Computer Implementation of a Theory of Stereo Vision". Phil. Trans. Royal Soc. Lond. B 298 (1982b).

Grimson W.E.L. "Computational Experiments with a Feature Based Stereo Algorithm" Artificial Intelligence Lab. Memo 762. MIT (1984)

Habibi A. "Two Dimensional Bayesian Estimation of Images" Proc. IEEE 60, 878–883 (1972).

Hammersley J.M. and Handscomb D.C. "Monte Carlo Methods". Chapman and Hall. London (1964).

Hansen A.R. and Elliot H. "Image Segmentation using Simple Markov Field Models" Comp. Graphics and Image Proc. 20, 101–132 (1982).

Hassner M. and Sklansky J. "The Use of Markov Random Fields as Models of Texture" Comput. Graphics and Image Proc. 12, 357–370 (1980).

Hastings W.K. "Monte Carlo Sampling Methods using Markov Chains and their Applications" Biometrika 57 97–109 (1970).

Hildebrand F.B. "Advanced Calculus for Applications". Prentice Hall (1976).

Hildreth E.C. "The Measurement of Visual Motion" MIT Press, Cambridge, Mass. (1984a).

Hildreth E.C. "Computation of the Velocity Field". Proc. R. Soc. London. B, 221, 189–220 (1984b)

Hillis D. "The Connection Machine". Ph.D. Thesis. MIT Dept of Elec. Eng. and Comp. Science. (1985).

Hinton G.E. and Sejnowski T. "Optimal Perceptual Inference" Proc. IEEE CVPR83 Washington, D.C. (1983)

Hinton G.E. Sejnowski T.J. and Ackley D.H. "Boltzmann Machines: Constraint Satisfaction Networks that Learn" Tech. Rep. CMU-CS-84-119 Carnegie-Mellon Univ. Dept. of Comp. Sc. (1984).

Hopfield J. "Neural Networks and Physical Systems with Emergent Computational Abilities" Proc. Nat. Ac. Sc. USA 79, 2554 (1982)

Hopfield J. and Tank D.W. "Neural Computation of Decisions in Optimization Problems" Preprint, 1985.

Hopfield J. Personal Communication (1985).

Horn B.K.P. "Determining Lightness from an Image." Comuter Graphics and Image Processing, 3, 11–299, (1974).

Horn B.K.P. and Schunck B.G. "Determining Optical Flow". Artificial Intelligence, 17, 185–203 (1981).

Hummel R. and Zucker S. "On the Foundations of Relaxation Labeling Processes". McGill Univ. Computer Vision and Graphics Laboratory, TR-80-7 (1980).

Ikeuchi K. and Horn B.K.P. "Numerical Shape from Shading and Occluding Boundaries". Artificial Intelligence, 17,141–184 (1981).

Ising E. "Beitrag sur Theorie des Ferromagnetismus" Zeit. fir Physik 31, 253–258 (1925).

Julesz B. "Binocular Depth Perception of Computer Generated Patterns" Bell Sys Tech J 39 (1960) 1125-1162.

Kailath T. "Lectures on Wiener and Kalman Filtering" Springer Verlag Wein–New York (1981).

Kashyap R.L. and Chellappa R. "Estimation and Choice of Neighbors in Spatial Interaction Models of Images". IEEE Trans. on Info. Theory 29 60-72 (1983).

Kemeny J.G. and Snell J.L. "Finite Markov Chains" Van Nostrand. New York (1960).

Kindermann R. and Snell J.L. "Markov Random Fields and their Applications". Vol 1. Amer. Math. Soc. (1980).

Kirkpatrick S. Gelatt C.D. and Vecchi M.P. "Optimization by Simulated Annealing". Science 220 (1983) 671–680.

Laughunn D.J. "Quadratic Binary Programming with Application to Capital–Budgeting Problems" Operations Research 18, 454-461 (1970).

Levy P.A. "A Special Problem of Brownian Motion and a General Theory of Gaussian Random Functions" Proc. Third Berkeley Sympos. Math. Stat. and Prob. Vol 2, (1956).

Levy B.C. and Tsitsiklis J.N. "An Inverse Scattering Procedure for the Linear Estimation of Two–Dimensional Isotropic Random Fields" Laboratory for Information and Decision Systems LIDS-P-1192 MIT (1983).

Luenberger D.G. "Introduction to Linear and Non–Linear Programming" Addison–Wesley. Reading, Mass. (1973).

Marr D. and Poggio T. "Cooperative Computation of Stereo Disparity" Science, 194, 283–287 (1976).

Marr D., Palm G. and Poggio T. "Analysis of a Cooperative Stereo Algorithm" Biol. Cyb. 28, 223 (1978)

Marr D. and Poggio T. "A Theory of Human Stereo Vision". Proc. Roy. Soc. Lond. B 204 (1979) 301–328.

Marroquin J. "Human Visual Perception of Structure" Sc. M. Thesis. MIT (1976).

Marroquin J. "Surface Reconstruction Preserving Discontinuities" AI Memo 792. MIT Artificial Intelligence Lab. and LIDS-R-1423, Lab. for Information and Decision Systems. MIT (1984).

McCoy B.M. and Wu T.T. "The Two–Dimensional Ising Model". Harvard Univ. Press (1973).

McLachlan D. "Statistical Mechanical Analogies" Prentice Hall (1968).

Mc Quarrie D. "Statistical Thermodynamics" Harper & Row (1973)

Metropolis N. et. al. "Equation of State Calculations by Fast Computing Machines". J. Phys. Chem. 21 6 (1953) 1087.

Nahi N.E. and Assefi T. "Bayesian Recursive Image Estimation" IEE Trans. on Computers 21, 734–738 (1972).

O'Callahan J. "Human Perception of Homogeneous Dot Patterns" Perception 3, 33 (1974).

Oster G.F., Perelson A. and Katchalsky A. "Network Thermodynamics" Nature, 234, 393–399 (1971).

Pavlidis T. "Waveform Segmentation through Functional Approximation" IEEE Trans. on Computers 22, 689–697 (1983).

Poggio T. and Koch C. "Analog Networks: A New Approach to Neural Computation" Artificial Intelligence Lab. Memo 783. MIT (1984).

Poggio T. and Torre V. "Ill–Posed Problems and Regularization Analysis in Early Vision". A.I. Memo 773 MIT (1984).

Poggio T., Voorhees H. and Yuille A. "Regularizing Edge Detection" Artificial Intelligence Lab. Memo 776. MIT (1984).

Pollard S.B., Mayhew J.E.W. and Frisby J.P. "Disparity Gradients and Stereo Correspondences". Preprint. Sheffield Univ. Dept. of Psychology. (1984).

Prazdny K. "Waveform Segmentation and Description Using Edge Preserving Smoothing" Computer Vision, Graphics and Image Proc. 23, 327–333 (1982)

Prazdny K. "Detection of Binocular Disparities". Preprint (1984).

Prekopa A., editor "Survey of Mathematical Programming" vol 2. North Holland (1976).

Preston C.J. "Gibbs States on Countable Sets". Camb. Univ. Press (1974).

Reif F. "Fundamentals of Statistical and Thermal Physics" McGraw–Hill (1965).

Rinooy Kan A.H.G., Boender C.G.E. and Timmer G. Th. "A Stochastic Approach to Global Optimization" preprint (1984).

Rissanen J. "Modeling by Shortest Data Description". Automatica. V14, pp 465–471 (1978).

Rissanen J. "A Universal Prior for Integers and Estimation by Minimum Description Length" The Annals of Statistics. V 11. No. 2, pp 416–431 (1983).

Taha H.A. "Integer Programming" Academic Press (1975).

Terzoupulos D. "Multiresolution Computation of Visible–Surface Representations". Ph. D. Thesis. Dept. of E.E. & C.S. MIT (1984).

Terzoupulos D. "Integrating Visual Information for Multiple Sources for the Cooperative Computation of Surface Shape" to appear in From Pixels to Predicates: Recent Advances in Computational and Robotic Vision, ed. A. Pentland, Ablex, (1985).

Tikhonov A.N. and Arsenin V.Y. "Solutions of Ill–Posed Problems". Winston & Sons. (1977).

Torre V. and Poggio T. "On Edge Detection". A.I. Memo 768 MIT (1983).

Ullman S. "Filling in the Gaps: The Shape of Subjective Contours and a Model for their Generation" Biological Cybernetics 25 1–6 (1976).

Ullman S. "The Interpretaion of Visual Motion" MIT Press (1979)

Vidyasagar M. "Nonlinear Systems Analysis" Prentice Hall (1978).

Wahba G. "Ill-posed Problems: Numerical and statistical methods for mildly, moderately and severely ill-posed problems with noisy data". Tech. Rep. 595 Univ. of Wisconsin, Madison (1980).

Wannier G.H. "Elements of Solid State Theory" Cambridge Univ. Press (1959).

White S. "Concepts of Scale in Simulated Annealing". Preprint (1984).

Wilson K.G. "The Renormalization Group: Critical Phenomena and the Kondo Problem" Rev. Mod. Phys. 47, 4 (1975).

Winston P. "Proposal to Darpa". MIT (1984).

Witkin A. "Scale Space Filtering", Proc. IJCAI, Karlsruhe (1983) 1019-1021.

Wolfram S. "Statistical Mechanics of Cellular Automata" Rev. Mod. Phys. 55, 601–644 (1983).

Wong E. "Two–Dimensional Random Fields and Representation of Images" SIAM J. Appl. Math. 16, 4, 756–770 (1968).

Wong E. and Zakai M. "On the Relation between Ordinary and Stochastic Differential Equations" Int. J. Eng. Sci. 3, 213-229 (1965).

Woods J.W. "Two–Dimensional Discrete Markovian Fields" IEEE Trans. Info. Theory 18, 232–240 (1972).

Yuille A. and Poggio T. "Fingerprints Theorems for Zero Crossings" A.I. Memo 730. MIT (1983a).

Yuille A. and Poggio T. "A Generalized Ordering Constraint for Stereo Correspondance". Artificial Intelligence Lab. Memo 777 MIT (1984).