April 1992

Markov Random Field Models: A Bayesian Approach to Computer Vision Problems

Gerda Kamberova
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-29.

Markov Random Field Models: A Bayesian Approach to Computer Vision Problems

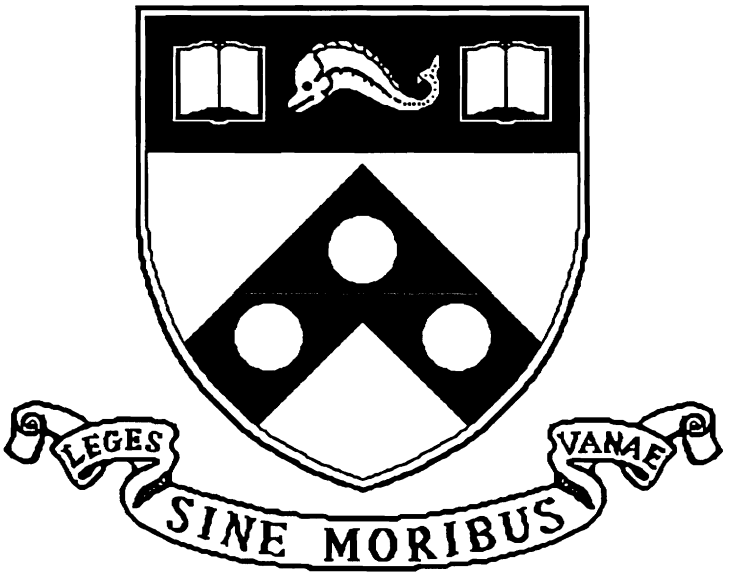
Abstract
The object of our study is the Bayesian approach in solving computer vision problems. We examine in particular: (i) applications of *Markov random field* (MRF) models to modeling spatial images; (ii) MRF based statistical methods for image restoration, segmentation, texture modeling and integration of different visual cues.

Comments
University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-92-29.

Markov Random Field Models: A Bayesian Approach To Computer Vision Problems

MS-CIS-92-29
GRASP LAB 310

Gerda L. Kamberova



University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

April 1992

Markov Random Field Models: a Bayesian Approach to Computer Vision Problems

Gerda L. Kamberova
Department of Computer and Information Science
University of Pennsylvania

Special Area Exam

Advisor: Max Mintz

Contents

List of Figures

List of Tables

1 Introduction

The object of our study is the Bayesian approach in solving computer vision problems. We examine in particular: (i) applications of *Markov random field* (MRF) models to modeling spatial images; (ii) MRF based statistical methods for image restoration, segmentation, texture modeling and integration of different visual cues.

The Bayesian method is a probabilistic framework which has four major components: (i) prior model which in low-level vision tasks describes prior information about an original image; (ii) degradation (sensor) model which models an observed degraded image as a result of applying some transformation to the original image; (iii) posterior model which relates the prior and degradation models, and given the observed data, for any allowable image, it represents the probability that this image has generated the observed data; (iv) loss function which is used for expressing costs of errors and preferences to particular estimators. The goal is, to find an optimal estimate with respect to the prior model, the degradation model, and the loss function, given the observed data. Finding the "best" estimate is an optimization problem which is usually intractable by traditional methods. This is where the MRF's have important applications. Geman and Geman [GG84] state and prove the main theorems on which the application of MRF's to image modeling and estimation is based. These theorems together with the "simulated annealing" technique [KGV83] provide computational methods for finding optimal image estimates.

Our survey is structured as follows.

In Section 2, we state the computer vision paradigm and give a brief overview of the major problems. In Section 3, we formulate the Bayesian approach and illustrate how it can be used in modeling spatial images and computer vision tasks. In Section 4, we introduce the Markov random field models and give some examples relevant to modeling spatial images. We discuss the difficulties associated with the MRF models and how these are overcome by exploiting the MRF-Gibbs equivalence. In Section 5, we consider the *maximum a posteriori* (MAP) estimate for MRF prior, and the restrictions imposed on the degradation transformation. In Section 6, we discuss the optimization problem related to the MAP problem, the Gibbs sampler (a procedure for taking samples from Gibbs-distributed random vectors), and Geman and Geman's results [GG84] which are the mathematical theory for investigating MRF's: (i) by sampling (Relaxation Theorem); (ii) by computing modes (Annealing Theorem); and (iii) by computing expectations (Ergodic Theorem). In Section 7, we illustrate the application of the theory presented in the previous sections to modeling images and solving some computer vision tasks. The examples are taken from [GG84], [GG86b], [Rip88], [Mar85], and [PT88]. They are related to restoration of images, texture modeling, texture segmentation and integration of visual cues from different low-level vision modules.

2 Computer Vision

The purpose of this section is to introduce the reader to computer vision. It is not intended to be complete, but only to give the necessary minimum background for the purpose of our presentation.

The ultimate goal of computer vision is to provide the computer systems with human-like vision: to give the machines ability to look for, see and understand (interpret) what has been seen [PT88]. The input is a digitized image usually on a square grid of pixels (each pixel measures the image in a small square). The measurement at each pixel may be a gray level (usually at most 256 levels of luminance) or a vector of gray levels (measuring luminance in different spectral bands). The resolution (the size of each pixel per scene) is often limited by hardware considerations in the sensors. In computer vision the resolution is also limited by real-time requirements. Most images available today are used in at most 512×512 sections. But these limits will increase with time. The output is a description of the scene observed. As Horn [Hor77] states, computer vision systems work toward symbolic description; in short: from 2-D images to 3-D symbolic description. In this, the computer vision systems differ from image processing systems which deal with conversion of images into new images, usually for human viewing. Marr [MN78] considers the process from 2-D images to 3-D symbolic descriptions as a complex hierarchical one consisting of three major levels (low, intermediate and high), each with its own representations and computational algorithms. These levels on the other hand are not isolated, but interconnected.

Here is the structure of the vision system as Marr views it. See figure 1 (page 3). The 2-D digitized image is processed at the lowest level so that important features are extracted, intensity variations in the image are made explicit (edges, homogeneous regions). A primal sketch which is a primitive description of local geometric relationships is obtained. This is a local camera-dependent image representation. Next, different low-level vision modules (texture, stereo, motion, color, "shape from shading") process the image in attempt to recover surface depth, orientation, local curvature, reflectance, texture, motion, color. The locally definable features are given 3-D interpretations: from 2-D edges to 3-D boundaries, and from 2-D regions to 3-D surfaces. The result from the low-level vision processing is the so called $2\frac{1}{2}$-D sketch. It is a viewer-dependent and viewer-centered local representation of an object (scene), and it is at the intermediate level in the vision hierarchy. Since the input data of the low-level vision modules are noisy and sparse, pre- or post-processing for image smoothing, or restoration may be necessary. At this stage a restoration and approximation by using MRF's may be applied. The intermediate-level vision algorithms integrate different visual cues (outputs from low-level vision modules). This stage of the vision processing is geared toward a global 3-D description of the object or the scene (not the image, not the surfaces). This description is at the high level of the

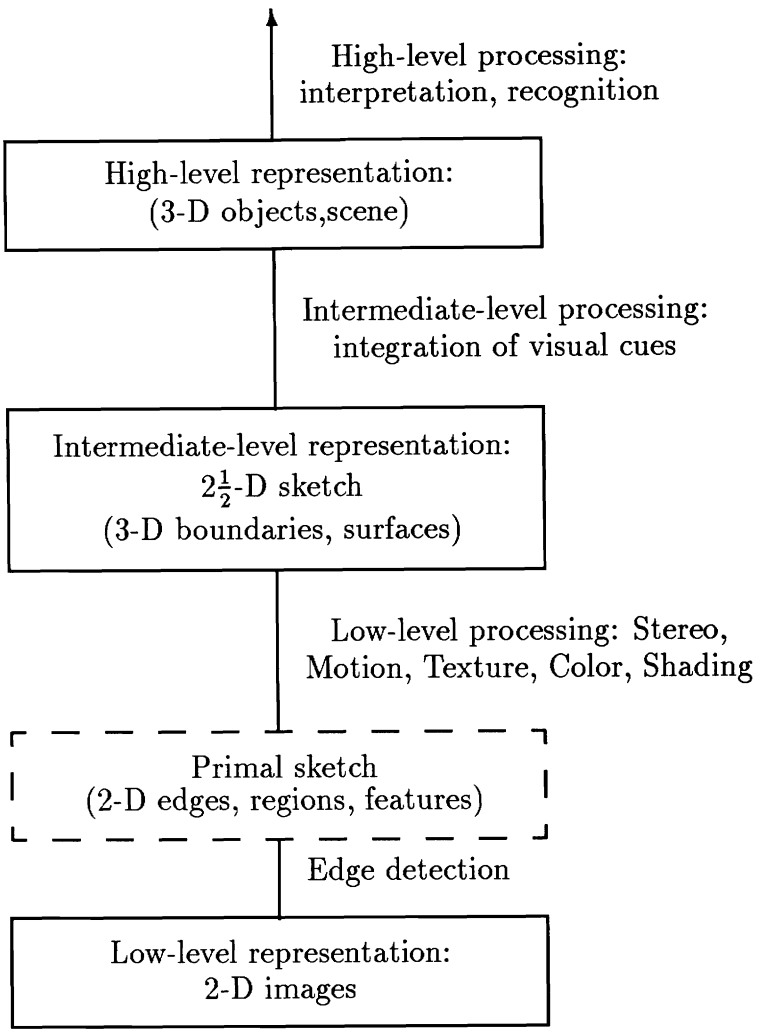


Figure 1: Vision Hierarchy: block-diagram.

visual hierarchy. It is a viewer-independent, object-centered global representation. Higher-level processing concerns the issues of interpretation, recognition, and knowledge representation. We do not study these questions in the present survey.

In his study of computer vision problems, Horn [Hor77] emphasizes the importance of understanding how images are formed, and what are the constraints imposed by the physical world. He considers vision tasks as "inverse problems" relative to optics; in optics the processing is from objects to images. In order to be able to deal with low-level vision tasks, explicit assumptions about the world being seen are made (for example, point light source, smooth surfaces, specific materials).

Poggio *at al.* [PTK85] observe that the problems arising in low-level vision are in general ill-posed (in a mathematical sense). A problem is well-posed when it has unique solution which depends continuously on the input data. Ill-posed problems fail to meet these conditions. The image data are inherently ambiguous and noisy, and based on this image data we try to recover unique surface properties. These data does not imply unique solution. Many low-level vision problems are ill-posed [PTK85]. The main idea, in solving ill-posed problems, is to restrict the class of admissible solutions. To make an ill-posed problem well-posed, regularization methods are used. One possibility is to reformulate the problem in terms of a variational principle, and then to use standard regularization methods to solve it [TA77].

The regularization of an ill-posed problem of finding z from the data y

$$\boldsymbol{A}z = y$$

requires a choice of a norm and a stabilizing functional $\|Pz\|$. Two methods that can be applied are:

- Find z which minimizes $\|Pz\|^2$, and satisfies $\|\boldsymbol{A}z - y\| \leq \varepsilon$, for any sufficiently small ε. "This method looks for a z which is sufficiently close to the data and is most regular, meaning minimizes the criterion $\|Pz\|^2$" [PTK85].

- Find z which minimizes
$$\|\boldsymbol{A}z - y\|^2 + \lambda\|Pz\|^2, \tag{2.1}$$
where $\lambda > 0$ is the so called regularization parameter. This parameter controls the degree of regularization of the solution and its closeness to the data.

The standard regularization methods impose constraints on an ill-posed problem by a variational principle such as (2.1) [PTK85]. This principle reflects the physical constraints about what represents a good solution: it has to be both close to the data, and regular (making $\|Pz\|$ minimal). Note that the standard regularization methods have to be applied after a careful analysis of the ill-posed nature of the problem. The

choice of the norm, the stabilizing functional, and the functional spaces involved, is dictated by both, mathematical and physical, considerations.

In this survey we present an alternative approach to regularizing ill-posed problems—Bayesian modeling.

3 Bayesian Approach

The Bayesian approach models the image formation and the sensor output as stochastic processes. It assumes prior knowledge expressed in terms of a probability distribution. A motivation for using the Bayesian approach in computer vision is that image problems do not occur in isolation [Rip88]. Often, many images, obtained under similar conditions, have to be processed. Previous experience, and physical considerations may impose some constraints on the model. For example, astronomers know a lot about atmospheric distortion [Rip88].

Remark 3.1 *Notational conventions.* For a square matrix $\boldsymbol{G}$, denote the transpose and the inverse matrices of $\boldsymbol{G}$ with $\boldsymbol{G}^t$ and $\boldsymbol{G}^{-1}$, respectively. Random variables are denoted by capital Roman letters, and their particular values by small Roman letters. Vectors are denoted by boldface type. The distributions of discrete random variables can be specified in terms of point mass functions (p.m.f.). For a discrete random variable Z, the p.m.f. is $p_Z(z)$

$$p_Z(z) = \mathrm{P}\,(Z = z),$$

where $(Z = z) \stackrel{\text{def}}{=} \{\omega : Z(\omega) = z\}$. Similarly for the joint and the conditional distributions of two random variables X and Z

$$\begin{aligned} p_{Z,X}(z,x) &= \mathrm{P}\,(Z = z, X = x), \\ p_{Z|X}(z \mid x) &= \mathrm{P}\,(Z = z \mid X = x). \end{aligned}$$

The distributions of continuous random variables can be specified in terms of density functions (d.f.). For a continuous random variable Z, the d.f. is $f_Z(z)$

$$f_Z(z)\,dz = \mathrm{P}\,(z < Z \leq z + dz),$$

where $(z < Z \leq z + dz) \stackrel{\text{def}}{=} \{\omega : Z(\omega) \in (z, z + dz]\}$. Similarly for the joint and the conditional distributions of two random variables X and Z

$$\begin{aligned} f_{Z,X}(z,x)\,dz\,dx &= \mathrm{P}\,(z < Z \leq z + dz,\ x < X \leq x + dx), \\ f_{Z|X}(z \mid x)\,dz &= \mathrm{P}\,(z < Z \leq z + dz \mid X = x). \end{aligned}$$

3.1 Definitions and Problem Statement

From a decision-theoretic point of view the formulation of the problem is as follows [Ber85]. We are given:

- A *parameter space* Ω (a nonempty set of possible values of the unknown parameter $\theta \in \Omega$).
- An *action space* $\mathcal{A}$ (a nonempty set of actions available to the statistician).
- A *loss function* $L : \Omega \times \mathcal{A} \rightarrow \mathrm{R}$ (a real-valued function defined on $\Omega \times \mathcal{A}$).

In a statistical decision problem we are given a triple $(\Omega, \mathcal{A}, L)$ coupled with an experiment involving an observable random variable (vector) $\boldsymbol{Z}$ whose probability distribution depends on the value of θ.

- The set $\mathcal{Z}$ of all possible realizations of the observable random variable (vector) $\boldsymbol{Z}$, is called the *sample space*. In computer vision applications $\mathcal{Z}$ is a subset of a finite dimensional Euclidean space. The cumulative distribution function (CDF) $F_Z(\boldsymbol{z};\theta)$ of $\boldsymbol{Z}$ depends on the true state of nature θ, and is called the *sampling distribution*.

A function $d : \mathcal{Z} \rightarrow \mathcal{A}$ which maps the sample space $\mathcal{Z}$ into the action space $\mathcal{A}$ is called a *decision rule* (procedure). Based of the outcome of the experiment, $\boldsymbol{Z} = \boldsymbol{z}$ the statistician chooses an action $d(\boldsymbol{z}) \in \mathcal{A}$. The loss, $L(\theta, d(\boldsymbol{Z}))$ is now a random quantity itself. The expected value, $\mathrm{E}_\theta L(\theta, d(\boldsymbol{Z}))$ is called the *risk function*

$$R(\theta, d) \stackrel{\text{def}}{=} \mathrm{E}_\theta L(\theta, d(\boldsymbol{Z})).$$

The expectation here is taken with respect to the sampling distribution for fixed θ. It represents the average loss to the statistician when θ is the true state of the unknown parameter, and the statistician uses a decision rule d.

There exist several principles by which the decision rules can be chosen. One such basic principle is *Bayes' Principle*. The Bayesian approach involves the notion of

- *prior probability distribution* π on the parameter space Ω.

We need two more quantities:

- the *Bayes risk* of a decision rule d with respect to π

$$r(\pi, d) \stackrel{\text{def}}{=} \mathrm{E}^\pi R(\Theta, d), \tag{3.1}$$

where Θ is a random variable over Ω with prior distribution π;

- the *posterior distribution*, $\pi(\theta \mid \boldsymbol{z})$, of the parameter θ given the observations. In the discrete case,

$$\pi(\theta \mid \boldsymbol{z}) = \frac{p_{\Theta,Z}(\theta, \boldsymbol{z})}{p_Z(\boldsymbol{z})}. \tag{3.2}$$

In the Bayesian setting, the parameter θ is treated as a random variable. Using Bayes' Principle, the statistician acts as if the parameter θ were a random variable Θ with known distribution.

A *Bayes decision rule* is a decision rule with smallest Bayes risk (3.1). Note that a Bayes rule may not be unique, or may not exist at all. When the Bayes rule does not exist, we obtain an *ε-Bayes rule* [Fer67].

Observation 3.1 [Fer67] The Bayes decision rule d minimizes the posterior conditional expected loss, given the observation(s). Hence, we choose $d(\boldsymbol{z}) = a$, where $a \in \mathcal{A}$ minimizes

$$\int_\Omega L(\theta, a)\, d\pi(\theta \mid \boldsymbol{z}).$$

In order for Observation 3.1 to be true, certain conditions have to be satisfied (namely, the conditions of Fubini's theorem [Roy68]). Since all the functions we use satisfy these conditions, we will not discuss this aspect.

Another way of using Bayesian analysis in inference problems is to look directly at the posterior distribution (3.2). The idea is that the posterior distribution, $\pi(\theta \mid \boldsymbol{z})$ contains all the available information about θ (both prior and sample information). Thus, any inference concerning θ should be based on this distribution. Ideally, the entire posterior distribution should be reported. But the most common technique is to give the maximum likelihood estimate obtained from the posterior distribution.

Definition 3.1 *The maximum a posteriori (MAP) estimate, $\hat{\theta}$, is the value of the parameter θ, which maximizes the posterior distribution $\pi(\theta|\boldsymbol{z})$. Thus, given $\boldsymbol{z} \in \mathcal{Z}$*

$$\pi(\hat{\theta} \mid \boldsymbol{z}) \geq \pi(\theta \mid \boldsymbol{z}), \text{ for all } \theta \in \Omega.$$

The MAP estimate is the "most likely" value of the parameter given the prior and the sample information. This is the estimate we discuss in our presentation.

In the Baysian approach to computer vision problems, the MAP estimate is widely used (the modes of the posterior distribution are considered to be good estimates). An example of an estimator different than the MAP is used in a tomography application [GM87]. There, the authors use the posterior mean in the reconstruction problem.

Marroquin [Mar85] explicitly discusses loss functions (error criteria) for computer vision applications, and the appropriate optimal estimators with respect to the prior and the degradation models, and the specific loss function. The loss function is an important component of the Bayes estimation, no less important than the prior model for example, and our view is that it should receive serious attention.

3.2 Computer Vision Applications—Example

Let us now show how a specific low-level vision task can be placed in a Bayesian setting. Consider the problem of image restoration ("cleaning").

An image consists of an $M \times M$ rectangular array of pixels. With each pixel, we associate a random variable which represents the measure of a certain attribute (intensity gray level, object label, texture label) at this pixel. Then, an image is regarded as the collection of the random variables, associated with all the pixels. Let $\boldsymbol{X} = \{X_{ij}\}$ represents an original unobservable image. An observed image $\boldsymbol{Z} = \{Z_{ij}\}$ is obtained from $\boldsymbol{X} = \{X_{ij}\}$ through some degradation transformation Ψ, $\boldsymbol{Z} = \Psi(\boldsymbol{X}, \boldsymbol{\epsilon})$, where $\boldsymbol{\epsilon}$ is a noise vector. In particular, we may use the model

$$\boldsymbol{Z} = H(\boldsymbol{X}) + \boldsymbol{\epsilon},$$

where H is a blurring transformation. The problem of *image restoration* is to infer $\boldsymbol{X}$ from $\boldsymbol{Z}$. Assume a prior probability model π for $\boldsymbol{X}$, $\pi(\boldsymbol{x}) = p_X(\boldsymbol{x})$. From Bayes' theorem

$$p_{X|Z}(\boldsymbol{x} \mid \boldsymbol{z}) = \frac{f_{Z|X}(\boldsymbol{z} \mid \boldsymbol{x})\, p_X(\boldsymbol{x})}{f_Z(\boldsymbol{z})}.$$

There is a practical difficulty here: the dimensions of the pixel arrays are huge, so it is impossible to evaluate directly $p_{X|Z}(\boldsymbol{x} \mid \boldsymbol{z})$, for each $\boldsymbol{x}$. Instead, we look for the MAP estimate. This is equivalent with finding an $\boldsymbol{x}$ which minimizes

$$-\log p_{X|Z}(\boldsymbol{x} \mid \boldsymbol{z}) = -\log f_{Z|X}(\boldsymbol{z} \mid \boldsymbol{x}) - \log p_X(\boldsymbol{x}) + \log f_Z(\boldsymbol{z}). \tag{3.3}$$

Assume that the noise $\boldsymbol{\epsilon}$ is independent of $\boldsymbol{X}$, and has Multivariate Normal distribution with covariance matrix Σ, and mean zero. Then, the sampling distribution is the conditional Normal distribution

$$f_{Z|X}(\boldsymbol{z} \mid \boldsymbol{x}) \propto \exp\{ -\frac{1}{2}(\boldsymbol{z} - H(\boldsymbol{x}))^t \Sigma^{-1} (\boldsymbol{z} - H(\boldsymbol{x})) \}.$$

Let the prior model for $\boldsymbol{X}$ be a Gibbs distribution, meaning

$$p_X(\boldsymbol{x}) \propto \exp\{-\lambda U(\boldsymbol{x})\},$$

where λ is a parameter of the distribution, and U is a function of a certain form which we will consider later. Then, (3.3) can be rewritten as

$$-\log p_{X|Z}(\boldsymbol{x} \mid \boldsymbol{z}) = \frac{1}{2}(\boldsymbol{z} - H(\boldsymbol{x}))^t \Sigma^{-1}(\boldsymbol{z} - H(\boldsymbol{x})) + \lambda U(\boldsymbol{x}) + \log f_Z(\boldsymbol{z}) \tag{3.4}$$

So the problem of maximizing the posterior p.m.f. $p_{X|Z}(\boldsymbol{x} \mid \boldsymbol{z})$ in this case reduces to the problem of minimizing (3.4). This is a difficult optimization problem.

Ripley [Rip88] draws a correspondence between the MAP estimate and the standard regularization technique. Denote the first summand in the left-hand side of (3.4) by $I(\boldsymbol{x})$. Then the MAP estimate has to minimize $I(\boldsymbol{x}) + \lambda U(\boldsymbol{x})$. We may think of I as measure of the 'infidelity' of the data $\boldsymbol{Z}$ to the true image $\boldsymbol{X}$. If the energy U is a measure of the 'roughness' of $\boldsymbol{X}$, it is clear that

$$\text{MAP } \textit{minimizes } (\text{ 'infidelity' } + \lambda \text{ 'roughness'}). \tag{3.5}$$

Note that (3.5) can be identified with a special case of a standard regularization (2.1).

In order to discuss an important issue, we look at the fundamental problem of image segmentation. The *segmentation* of an image denotes the division of the image into homogeneous subimages and the labeling of the pixels by their type. We observe $\boldsymbol{Z}$, consisting of image intensities, and infer $\boldsymbol{X}$, the map of labels assigned to each pixel. In image segmentation, the MAP estimation is justified by a simple decision theory problem [Rip88]. Let the images have size $M \times M$. Consider a parameter space

$$\Omega = \{\, \boldsymbol{x} = \{x_{ij}\}_{i,j=1}^{M} : x_{ij} \in \Lambda,\ 1 \le i, j \le M \,\},$$

where Λ is a finite set of labels. In this notation $\boldsymbol{x}$ takes the role of the parameter θ. The action space $\mathcal{A}$ may be the same as Ω. Let $L(\boldsymbol{a}, \boldsymbol{x})$ be a zero-one loss function with error tolerance zero:

$$L(\boldsymbol{a}, \boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{a} = \boldsymbol{x}, \\ 1, & \text{otherwise.} \end{cases}$$

Hence, the loss is zero for all correct maps, and one for the incorrect ones. From Observation 3.1, it follows that the MAP estimator is a Bayes rule.

Another measure of the error, commonly used in segmentation, is the number of misclassified pixels [Rip88], corresponding to a loss function equal to that number. In this case, Bayes rules maximize

$$G(\boldsymbol{x}) = \sum_{i,j=1}^{M} \mathrm{P}\,(X_{ij} = x_{ij} \mid \boldsymbol{Z}), \quad \boldsymbol{x} \in \Omega,\ \boldsymbol{x} = \{x_{ij}\}_{i,j=1}^{M},$$

which is equivalent to maximizing each of the posterior marginal distributions $\mathrm{P}(X_{ij} = x_{ij} \mid \boldsymbol{Z})$, $\{i,j\} \subset \{1,\ldots,M\}$. This estimation is known as *maximum a posteriori marginals* (MAM) [Mar85]. The posterior marginal distributions do not have a simple form. There are methods which seek to overcome this difficult problem. Marroquin considers a simulation-based method for finding MAM. Since $G(\boldsymbol{x})$ cannot be found analytically for computational reasons, farther simplifications, by approximations, $\mathrm{P}(X_{ij} = x_{ij} \mid \boldsymbol{Z}_N)$, of the exact computations, $\mathrm{P}(X_{ij} = x_{ij} \mid \boldsymbol{Z})$, are necessary. Here $\boldsymbol{Z}_N$ denotes the observations in a neighborhood of (i,j) [Rip88]. In our study we will not discuss this method. Our concern is the MAP estimate.

4 Markov Random Field Models

4.1 MRF—Definition and Specification

Definition 4.1 *Let $S = \{ s_1, s_2, \ldots, s_N \}$ be a lattice (a set of sites), and let $\mathcal{G} = \{ \mathcal{G}_s : s \in S \}$ be a family of subsets of S. The family $\mathcal{G}$ is a neighborhood system for S if:*

1. *for every s in S, $s \notin \mathcal{G}_s$;*
2. *for every s and r in S, $s \in \mathcal{G}_r$ if and only if $r \in \mathcal{G}_s$.*

For each $s \in S$, $\mathcal{G}_s$ is the *neighborhood of s*; the elements of $\mathcal{G}_s$ are the *neighbors of s*. The neighborhood relation is antireflexive and symmetric. It is not transitive.

Remark 4.1 The pair $\{S, \mathcal{G}\}$ is a graph in the usual sense: the vertices are the sites of S; two vertices are connected if they are neighbors in S with respect to $\mathcal{G}$. From here on, $\{S, \mathcal{G}\}$ is a graph as in Definition 4.1.

Definition 4.2 *Let $\{S, \mathcal{G}\}$ be a graph, $|S| = N$, and let Ω be a subset of the N-dimensional Euclidean space,*

$$\Omega = \{ \boldsymbol{\omega} = (x_{s_1}, x_{s_2}, \ldots, x_{s_N}) : x_{s_i} \in \Lambda_{s_i},\ |\Lambda_{s_i}| < \infty,\ i = 1,\ldots,N \}. \tag{4.1}$$

A random vector $\boldsymbol{X} = \{X_s\}_{s \in S}$, with range Ω, is a Markov random field (MRF) over S with respect to $\mathcal{G}$, if there exists a probability measure P on Ω such that:

1. *for every $\boldsymbol{\omega}$ in Ω*
$$\mathrm{P}(\boldsymbol{X} = \boldsymbol{\omega}) > 0, \tag{4.2}$$
2. *for every s in S*
$$\mathrm{P}(X_s = x_s \mid X_r = x_r\ \forall r \neq s) = \mathrm{P}(X_s = x_s \mid X_r = x_r\ \forall r \in \mathcal{G}_s), \tag{4.3}$$
for any $\boldsymbol{\omega} = (x_{s_1}, x_{s_2}, \ldots, x_{s_N})$ in Ω.

The set Ω is the *configuration space*, and any ω in it is a *configuration.* The *local characteristics* of the MRF $\boldsymbol{X}$ (4.3) determine uniquely the joint probability (4.2). For any site $s \in S$, the equality (4.3) states that the probability of X_s conditioned on the remaining random variables, $X_r \quad r \in S \quad r \neq s$, is equal to the probability of X_s conditioned only on the random variables X_r where r is a neighbor of s, $r \in \mathcal{G}_s$. Hence, $\boldsymbol{X}$ is characterized only by local interactions, the value at a site s, is dependent only on the values of its neighbors, not on the values over the whole lattice S.

Without loss of generality, assume that all the random variables X_s, $s \in S$ have a common finite range space Λ, $\Lambda_s = \Lambda$, $s \in S$.

Remark 4.2 Any random vector $\boldsymbol{X}$ with positive joint density is a MRF for neighborhoods large enough to represent the dependences. But MRF are useful for modeling spatial images when neighborhoods are small enough to ensure fast computations, and yet large enough to represent varieties of images [GG84].

Example 4.1 *One-dimensional Markov Random Chains* [GG84]. A one-dimensional Markov random chain $\boldsymbol{X}$ defined on $S = \{1, 2, \ldots, n\}$ with respect to the nearest neighbor system $\mathcal{G}_1 = \{2\}$, $\mathcal{G}_i = \{i-1, i+1\}, 1 < i < n-1$, $\mathcal{G}_n = \{n-1\}$, if started at equilibrium, is an example of a MRF.

Observation 4.1 The MRF's are defined by their local characteristics (4.3). These represent a very large family of functions, satisfying certain consistency conditions. "It is extremely difficult to spot the local characteristics" [GG84]. This is why the Definition 4.2 is not useful in practice, it does not give a formalism for constructing MRF's.

This problem is overcome by utilizing the MRF-Gibbs equivalence.

Definition 4.3 *Let $\mathcal{G}$ be a neighborhood system for S, and let* C *be a subset of S. The set* C *is a clique in $\{S, \mathcal{G}\}$ either if it is a one element set* ($|\mathrm{C}| = 1$), *or* $|\mathrm{C}| > 1$ *and any two of its elements are neighbors. We denote the set of all cliques in $\{S, \mathcal{G}\}$ by $\mathcal{C}$.*

Remark 4.3 In terms of graphs, a clique is a completely connected subgraph. Note that a clique is not necessarily maximal.

Example 4.2 Let $Z_m = \{(i,j)\}_{i,j=1}^{m}$ be an integer lattice. A *homogeneous neighborhood system* $\mathcal{F}_c$ on S is defined by

$$\mathcal{F}_c = \{\, \mathcal{F}_{ij} : \mathcal{F}_{ij} = \{(k,l) : 0 < (k-i)^2 + (l-j)^2 \leq c\},\ i,j = 1, \ldots, m \,\}. \qquad (4.4)$$

Figure 2 (page 12) illustrates the interior neighborhoods for homogeneous four-,eight- and twenty four-neighbor systems ($c = 1, 2, 8$). Figure 3 (page 12) shows the clique

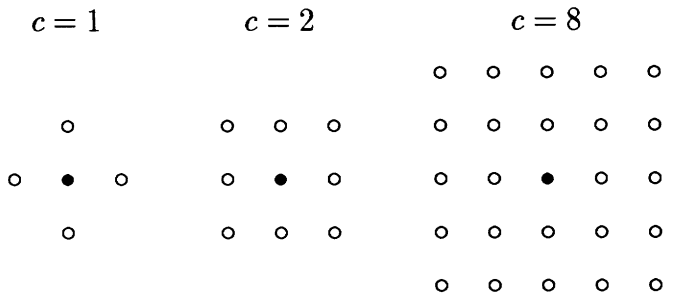


Figure 2: Interior neighborhoods for homogeneous neighborhood systems, $c = 1, 2, 8$.

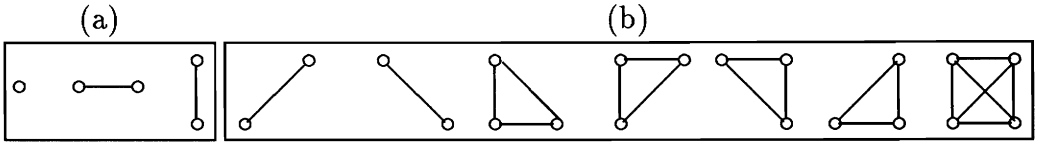


Figure 3: (a): The clique types for a four-neighbor system; (a) and (b): The clique types for an eight-neighbor system.

types: part (a) for $c = 1$, and parts (a) and (b) the clique types for $c = 2$. "Obviously, the number of clique types grows rapidly with c" [GG84].

Definition 4.4 *Let $\{S, \mathcal{G}\}$ be a graph, $|S| = N$, and let Ω be a configuration space (4.1). A Gibbs probability measure π on Ω relative to $\{S, \mathcal{G}\}$, is defined by*

$$\pi(\omega) = \frac{\exp\{-U(\omega)/T\}}{Z}, \tag{4.5}$$

such that

- *for any configuration $\omega = (x_{s_1}, x_{s_2}, \ldots, x_{s_N}) \in \Omega$*

$$U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega),$$

  where for every clique C in $\mathcal{C}$, $V_C(\omega)$ depends only on these components x_{s_i} of ω for which $s_i \in C$.

- *Z is a normalizing constant.*

- *T is a positive constant.*

Gibbs distributions are widely used in statistical physics. They describe equilibrium states of large-scale physical systems. The sites of the lattice are identified with the components of the system. The names of the functions and the constants in a Gibbs distribution come from statistical physics:

- U is the *energy* function. Low-energy configurations have high probabilities, and high-energy configurations have low probabilities.
- V_C, C in $\mathcal{C}$, are the *potentials*. They are real-valued functions, which represent contributions to the total energy from external fields ($|C| = 1$) and from interactions of the elements of the system ($|C| > 1$).
- Z is the partition function,

$$Z = \sum_{\omega \in \Omega} \exp\{ -U(\omega)/T \}.$$

- T is the *temperature*. It is a scale parameter for the distribution. By varying T, the height of the peaks of π changes. Let the energy function U is fixed. At very high temperature, the "density" π is almost uniform on Ω. Lowering the temperature, exaggerates the peaks of π, the difference between the probability of a minimal energy configuration and the probability of a configuration which does not have minimal energy, is increased. At very low temperature, π is almost uniform on the subset of Ω on which U attains its global minimum (π is zero outside this subset), a random sample from the corresponding Gibbs distribution is with high probability a minimal energy configuration. This is the principle of the technique used for finding MAP estimates. That technique is called *simulated annealing*, and it is discussed in Section 6.1.

Theorem 4.1 *Let $\{S, \mathcal{G}\}$ be a graph, Ω be a configuration space, and* P *be a probability measure on Ω. Let $\boldsymbol{X}$ be a random vector. The random vector $\boldsymbol{X}$ is a* MRF *on S with respect to $\mathcal{G}$ if and only if $\boldsymbol{X}$ has a Gibbs distribution over Ω with respect to $\{S, \mathcal{G}\}$,* $\mathrm{P}(\boldsymbol{X} = \omega) = \pi(\omega)$, *(4.5), $\omega \in \Omega$.*

"Explicit formulas exist for obtaining U from the local characteristics" [GG84]; for sketch of a proof see [Mar85]. For our purposes, more important is the other "direction" of the theorem. The local characteristics can be expressed in terms of the potentials (the energy). By applying Bayes rule and the definition of conditional probability, we see this as follows:

Let $\boldsymbol{X}$ has a Gibbs distribution,

$$\mathrm{P}(\boldsymbol{X} = \omega) = \frac{\exp\{ -U(\omega)/T \}}{Z}.$$

By the definition of conditional probability

$$\mathrm{P}(X_s = x_s \mid X_r = x_r \ \forall r \neq s) = \frac{\mathrm{P}(\boldsymbol{X} = \omega)}{\mathrm{P}(X_r = x_r \ \forall r \neq s)}, \tag{4.6}$$

where $\omega = (x_{s_1}, \ldots, x_{s_N})$, and the denominator denotes the joint probability distribution of all X_r's except X_s. Let x denote the s^{th} coordinate in ω. By the definition of marginal probability

$$\mathrm{P}\,(\,X_r = x_r\ \forall r \neq s\,) = \sum_{x \in \Lambda} \mathrm{P}\,(\,\boldsymbol{X} = \omega^s\,),$$

where ω^s denotes a configuration which is x at site s, and agrees with ω everywhere else. So by (4.5)

$$\mathrm{P}\,(\,X_r = x_r\ \forall r \neq s\,) = \sum_{x \in \Lambda} \frac{\exp\{\,-U(\omega^s)/T\,\}}{Z}.$$

Substitute in (4.6)

$$\mathrm{P}\,(\,X_s = x_s \mid X_r = x_r\ \forall r \neq s\,) = \frac{\exp\{\,-U(\omega)/T\,\}}{\sum_{x \in \Lambda} \exp\{-U(\omega^s)/T\}}.$$

Expand the energy, and cancel the common factors. Thus, obtain

$$\mathrm{P}\,(\,X_s = x_s \mid X_r = x_r\ \forall r \neq s\,) = \frac{\exp\{-\sum_{\mathrm{C}: s \in \mathrm{C}} V_{\mathrm{C}}(\omega)/T\}}{\sum_{y_s \in \Lambda_s} \exp\{-\sum_{\mathrm{C}: s \in \mathrm{C}} V_{\mathrm{C}}(\omega^s)/T\}}. \tag{4.7}$$

This is the local characteristic of a site s in terms of the potentials over the cliques which contain s. The computations in (4.7) are local and simple. They are the basis for the Gibbs sampler—an algorithm which we discuss in Section 6.2.

The advantage of specifying MRF in terms of Gibbs distributions (potentials) instead in terms of local characteristics is evident. To define a MRF prior for a class of images, it is sufficient to specify the spatial dependences in terms of potentials. The potentials in a Gibbs distribution reflect the spatial coherence of the image. For a given pixel, they show how the neighboring pixels taken one at a time, two at a time (and so on) change the probability that this pixel has a certain value. The potentials V_{C}, $\mathrm{C} \in \mathcal{C}$, are a family of real-valued functions, which do not satisfy any consistency conditions (note the difference with the local characteristics). Specifying the potentials is more practical than specifying the local characteristics directly.

4.2 MRF—Examples

Ising model

Let $Z_m = \{(i,j)\}_{i,j=1}^m$ be an integer lattice with the homogeneous four-neighbor system $\mathcal{F}_1$ (4.4), and let $\boldsymbol{X}$ be a MRF over a configuration space Ω

$$\Omega = \{\,\omega = \{x_{ij}\}_{i,j=1}^m : x_{ij} \in \Lambda,\ 1 \leq i,j \leq m\,\}.$$

A clique is either a single site, or a pair of adjacent sites. The MRF $\boldsymbol{X}$ has a Gibbs distribution with energy function U. The most general form of the energy U is

$$U(\boldsymbol{\omega}) = \sum V_{\{(i,j)\}}(x_{ij}) + \sum V_{\{(i,j),(i,j+1)\}}(x_{ij}, x_{i(j+1)}) + \sum V_{\{(i,j),(i+1,j)\}}(x_{ij}, x_{(i+1)j}), \tag{4.8}$$

where $\boldsymbol{\omega} = \{ x_{ij} \}_{i,j=1}^{m}$, and the first sum is over all sites in S, the second over all horizontal pair-cliques, and the third one—over all vertical pair-cliques [GG84]. The Ising model is a special case of (4.8) in which the MRF $\boldsymbol{X}$ is binary ($|\Lambda| = 2$), homogeneous (translationally invariant, strictly stationary), and isotropic (rotationally invariant) [Won71]. The Ising energy function U is

$$U(\boldsymbol{\omega}) = \alpha \sum x_{ij} + \beta \left(\sum x_{ij} x_{i(j+1)} + \sum x_{ij} x_{(i+1)j} \right), \tag{4.9}$$

where α and β are parameters. Without loss of generality we assume that the temperature $T = 1$. Substituting (4.9) in (4.7), we obtain the local characteristics for the Ising model. For $(i,j) \in Z_m$

$$\mathrm{P}\left(X_{ij} = x_{ij} \mid X_{kl} = x_{kl} \ \forall (k,l) \neq (i,j) \right) = \frac{\exp\{ -x_{ij}(\alpha + \beta v_{ij}) \}}{\sum_{x \in \Lambda} \exp\{ -x(\alpha + \beta v_{ij}) \}},$$

where $v_{ij} = x_{i(j-1)} + x_{(i-1)j} + x_{i(j+1)} + x_{(i+1)j}$. For $\Lambda = \{ -1, 1 \}$, it is easy to recognize that the Ising model favors images in which adjacent pixels have the same values. It represents adequately scenes consisting of homogeneous regions (of intensity, colors).

Strauss model

This is a model for unordered categories, such as maps of colors (labels) [Rip88]. Let $\boldsymbol{L}$ be a random field over an integer lattice $Z_m = \{ (i,j) \}_{i,j=1}^{m}$ with the homogeneous neighborhood system $\mathcal{F}_c$, and let Λ be a finite set of colors (labels) in which the components of $\boldsymbol{L}$ take values. The configuration space Ω consists of all maps which can be generated with the allowable colors, $\Omega = \Lambda^{m^2}$. For any configuration $\boldsymbol{l} \in \Omega$, the energy $U(\boldsymbol{l})$ represents the number of pairs of neighbors with different colors. Only potentials over the pair-cliques may be different from zero. For any pair-clique $\{ s, r \} \in \mathcal{C}$

$$V_{\{s,r\}}(\boldsymbol{l}) = \begin{cases} 1, & l_s \neq l_r; \\ 0, & \text{otherwise.} \end{cases}$$

Maps all of one color have lowest energy (zero), and, by the Gibbs model, are most probable. This probability decreases when the number of pair neighbors with different colors increases. The local characteristic at (i,j), derivable from (4.7), depends on the number of neighbors of (i,j) with color k

$$\mathrm{P}\left(L_{ij} = k \mid L_{nt} = l_{nt} \ \forall (n,t) \neq (i,j) \right) \propto \exp\{ \beta \, |\{ (n,t) : (n,t) \in \mathcal{F}_{ij}, \, l_{nt} = k \}| \} \tag{4.10}$$

map	U	No. maps	Description
a	0	2	the whole map has one color
b	2	8	one corner has color different from the rest of the map
c	3	24	one pixel along the edge has color different from the rest
d	3	16	two pixels next to each other at a corner have the same color and the reminder has the complimentary color
e	4	8	one edge has a color different from the rest of the map
f	8	8	one interior column or row has a color different than the rest
$\vdots$			

Table 1: Some energy values for the Strauss model on a 4×4 grid.

where $k \in \Lambda$. In the case of a four-neighbor system ($c = 1$), and only two colors ($|\Lambda| = 2$), the Strauss model reduces to the Ising model (4.9). But even with this simple model on a 4×4 grid, the number of possible maps is huge—2^{16}. Table 1 illustrates some of the energy values, and descriptions of corresponding maps [Rip88]. The pictures of the maps are shown on figure 4 (page 17).

4.3 MRF—Hierarchical model

The simple MRF models are not rich enough to express adequately complicated degree of prior knowledge. In some cases we would expect long straight boundaries which, for example, the Strauss model penalizes. Geman and Geman introduce the hierarchical MRF models [GG84]. The hierarchy reflects the type and the degree of prior knowledge about the image being studied. The image is modeled as a "multiple processes" MRF $\boldsymbol{X} = (\boldsymbol{X}^P, \boldsymbol{X}^E, \boldsymbol{X}^L, \ldots)$, each component of which is a MRF itself. The first process $\boldsymbol{X}^P$ is a process of image intensities. The rest of the processes model some geometrical (structural) attributes in the image. These correspond to edges [GG84], feature or texture labels [GG86b], etc. "They are part of the image model but not of the physical data." [GG86a]. In Section 7.2 we present the the hierarchical model from [GG86b], where images are modeled in two levels, by intensity process and texture label process. Here we present Geman and Geman's hierarchical model for the MAP image restoration problem from [GG84]. The images are modeled in two levels, by intensity process and edge (discontinuity) process.

The hierarchical MRF model [GG84]

The image $\boldsymbol{X}$ is regarded as a MRF composed of two processes,

$$\boldsymbol{X} = (\boldsymbol{F}, \boldsymbol{L}),$$

where the *intensity process* $\boldsymbol{F}$ is a MRF of observable pixel intensities, and the *line process* $\boldsymbol{L}$ is a MRF of unobservable edge elements. Here "observable" refers to the

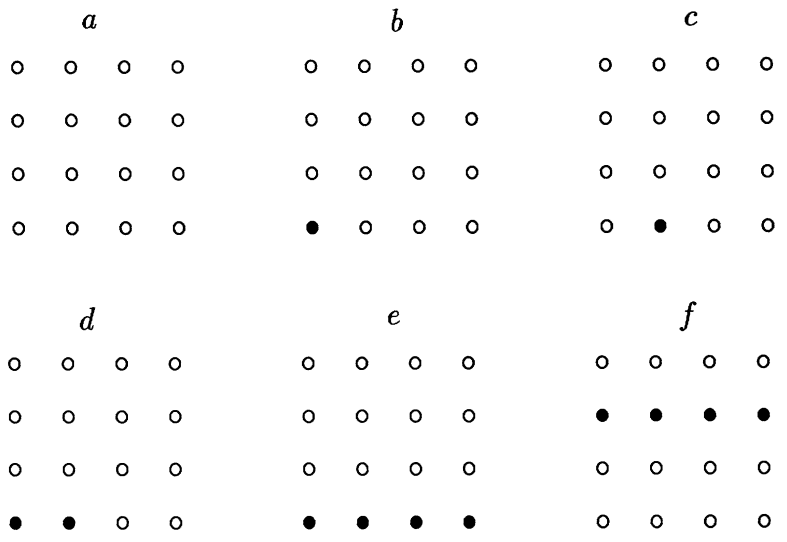


Figure 4: Maps described in Table 1.

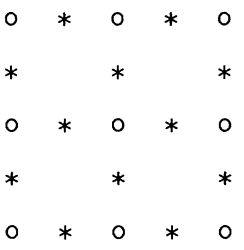


Figure 5: Pixel sites (o) and line sites (*).

fact that the pixel intensities are the values at the pixels, measured by the sensors; and "unobservable" refers to the fact that in the input image, there are no measurements about the geometrical structure of the intensity edges. Let Z_m be a square $m \times m$ integer lattice with a homogeneous four-neighbor system $\mathcal{F}_1$ (4.4). Consider a lattice D_m consisting of the "places" between any two adjacent pixels of Z_m. See figure 5 (page 17). The "line" neighborhood $L_{((i,j)(i+1,j))}$ of a "place" between pixels (i,j) and $(i+1,j)$ consists of the six places (Look at figure 6 (page 18))

$$\begin{aligned} L_{((i,j)(i+1,j))} = \{&((i,j-1),(i,j)),\ ((i,j),(i,j+1)),\ ((i,j-1),(i+1,j-1)),\\ &((i,j+1),(i+1,j+1)),\ ((i+1,j-1),(i+1,j)),\\ &((i+1,j),(i+1,j+1))\}, \end{aligned}$$

where $((i,j),(k,l))$ denotes the "place" between the adjacent pixels (i,j), and (k,l). This neighborhood is horizontal. A neighborhood $L_{((i,j)(i,j+1))}$ is similar but vertical.

Figure 6: Interior neighborhoods for vertical and horizontal line sites.

A line site in D_m has a horizontal or vertical neighborhood, depending on its position among the pixels of Z_m.

Denote the neighborhood system over D_m with $\mathcal{L}$.

The intensity process $\boldsymbol{F}$ is a MRF over the graph $\{Z_m, \mathcal{F}_1\}$; at each pixel (i,j), F_{ij} represent the measure of the intensity at that pixel. The line process $\boldsymbol{L}$ is a MRF over the graph $\{D_m, \mathcal{L}\}$; at each line site, the corresponding component of $\boldsymbol{L}$, represents the "measure" of an intensity edge at that line site. Each component of the line process represents a lack of an edge, or presence (and orientation) of an edge at a line site.

The MRF $\boldsymbol{X} = (\boldsymbol{F}, \boldsymbol{L})$ is specified over the graph $\{S, \mathcal{G}\}$. The lattice S is an union of the "pixel" and the "line" lattices

$$S = Z_m \cup D_m. \tag{4.11}$$

See figure 5 (page 17). The neighborhood system $\mathcal{G}$ is defined as follows: (i) the line neighbors of a line site in S are the same as in $\{D_m, \mathcal{L}\}$; (ii) the pixel neighbors of a pixel site in S are the same as in $\{Z_m, \mathcal{F}_1\}$; (iii) the line neighbors of a pixel site in S are the four nearest line elements (two horizontal and two vertical line sites, placed between the pixel site and its four nearest pixel neighbors under $\mathcal{F}_1$). By symmetry of the neighborhood relation, the interior neighborhoods for a pixel site, and a vertical and a horizontal line sites are inferred. See figure 7 (page 19). The line process $\boldsymbol{L}$ modifies the pixel neighborhoods in $\mathcal{F}_1$: if an edge is present at a line site, the potential over the pair-clique consisting of the pixels which this line site separates is zero; the two pixels separated by the edge element do not influence each others intensities, the bonding between them is broken. In this sense these pixels are not neighbors with respect to $\boldsymbol{X}$, and they will not influence each other intensities, in spite of the fact, that they are neighbors in $\{\mathcal{F}_1, Z_m\}$, and influence each other intensities with respect to $\boldsymbol{F}$. Figure 8 (page 19) illustrates this concept; pixels separated by an edge elements are not considered neighbors any more; there may be sharp difference

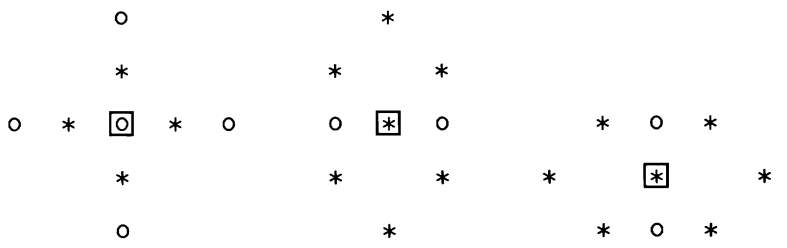


Figure 7: Interior pixel and line sites neighborhoods for the hierarchical MRF.

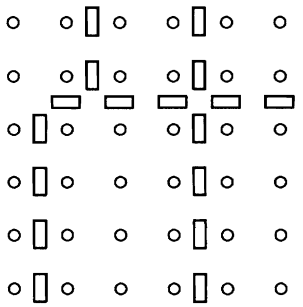


Figure 8: A grid of pixels (circles) with an edge process (bars).

between their intensities, and configurations which have this particular pattern will not be penalized (note the difference with the Strauss model, which penalizes the difference in intensity of the neighboring pixels).

The hierarchical model reflects the type and the degree of the prior knowledge about the class of images under study.

The MRF $\boldsymbol{X} = (\boldsymbol{F}, \boldsymbol{L})$ on $\{S, \mathcal{G}\}$ is defined by the following Gibbs distribution

$$\mathrm{P}\,(\,\boldsymbol{F} = \boldsymbol{f}, \boldsymbol{L} = \boldsymbol{l}\,) = \frac{\exp\{\,-U(\boldsymbol{f}, \boldsymbol{l})/T\,\}}{Z}, \tag{4.12}$$

where

$$U(\boldsymbol{f}, \boldsymbol{l}) = \sum_{C \in \mathcal{C}} V_C(\boldsymbol{f}, \boldsymbol{l}). \tag{4.13}$$

The configuration space consists of all pairs $\boldsymbol{\omega} = (\boldsymbol{f}, \boldsymbol{l})$, where the components of $\boldsymbol{f}$ are allowable intensity values (gray levels) and the components of $\boldsymbol{l}$ are encoded line states (a presence of an edge is coded with one, and a lack of an edge is coded with zero; if it is desirable to code possible orientations of an edge at a line site, more than two values are necessary).

5 The MAP problem

We return now, to the MAP problem. It has three major components.

1. Prior model
2. Degradation model
3. Posterior model

The MAP estimator is the one which maximizes the posterior distribution, recall Definition (3.1). The results are stated for the general case of a hierarchical MRF prior, $\boldsymbol{X}$. The MRF models were discussed in Section 4.

5.1 Prior model

The prior is a MRF $\boldsymbol{X} = (\boldsymbol{F}, \boldsymbol{L})$ over Ω with respect to $\{S, \mathcal{G}\}$ (discussed in Section 4.3).

Next, we focus on the degradation and the posterior models.

5.2 Degradation model

The intensity process $\boldsymbol{F}$ is subject to some degradation, but the line process $\boldsymbol{L}$ is preserved. So the observed process (complete degraded image) is

$$\begin{aligned} \boldsymbol{Z} &= (\boldsymbol{G}, \boldsymbol{L}), \text{ where} \\ \boldsymbol{G} &= \Psi(\, \phi(H(\boldsymbol{F})), \boldsymbol{N}\,). \end{aligned} \tag{5.1}$$

Here H denotes a blurring matrix (shift-invariant point spread function), ϕ is a non-linear transformation, and $\boldsymbol{N}$ is a noise process. Assume that

- $\boldsymbol{N}$ and $\boldsymbol{F}$ are independent, and $\boldsymbol{N}$ and $\boldsymbol{L}$ are independent.
- The transformation Ψ is invertible in the second argument when the first one is fixed: $\Psi^{-1}(\boldsymbol{a}, \cdot\,)$ exists for any $\boldsymbol{a} \in \phi(H(\Omega^F))$, where Ω^F denotes the configuration space of the intensity process $\boldsymbol{F}$. For example, Ψ may be addition or multiplication of its arguments. This ensures that the posterior distribution is a well-defined function which we can derive directly, and use in the inference problem.
- For computational purposes, the degradation function should preserve the approximate locality of $\boldsymbol{F}$, so that the neighborhood systems of the prior and degraded images are comparable.

These restrictions are fulfilled for a wide class of useful degradation models, including combinations of blur, additive or multiplicative noise, and a variety of nonlinear transformations.

5.3 Posterior model

For a MRF prior model, and for a degradation model which satisfy the assumptions stated in the previous subsection, the posterior distribution defines a MRF with a local neighborhood structure. The following theorem is proved in [GG84]. To be specific, let consider a Gaussian noise process $\boldsymbol{N}$ consisting of independent identically distributed (i.i.d.) Normal random variables $\mathcal{N}(\mu, \sigma^2)$.

Theorem 5.1 *For each fixed observation* $\boldsymbol{g}$, $\boldsymbol{g} \in \phi(H(\Omega^F))$, *the posterior distribution* $\mathrm{P}(\boldsymbol{F} = \boldsymbol{f}, \boldsymbol{L} = \boldsymbol{l} | \boldsymbol{G} = \boldsymbol{g})$, *is a Gibbs distribution over* $\{S, \mathcal{G}^{\mathrm{P}}\}$ *with energy function*

$$U^{\mathrm{P}}(\boldsymbol{f}, \boldsymbol{l}) = U(\boldsymbol{f}, \boldsymbol{l}) + \frac{\| \boldsymbol{\mu} - \Psi^{-1}(\phi(H(\boldsymbol{f})), \boldsymbol{g}) \|^2}{2\sigma^2}, \tag{5.2}$$

where $U(\boldsymbol{f}, \boldsymbol{l})$ *is the prior energy (4.13), and* $\boldsymbol{\mu} = (\mu, \ldots, \mu)^t$ *has dimension* $m^2 \times 1$.

The theorem is stated for Gaussian noise process, but the proof can be extended directly for a general noise process. Here we give a sketch of the proof [GG84]. In order to keep the notation simple, and the derivations general (for any noise distribution), we make the following agreement: for arbitrary continuous random variable Y, $\mathrm{P}(Y = y)$ denotes $\mathrm{P}(y < Y \leq y + dy)$.

From (3.2), the posterior distribution can be expressed as

$$\mathrm{P}(\boldsymbol{X} = \boldsymbol{\omega} \mid \boldsymbol{G} = \boldsymbol{g}) = \frac{\mathrm{P}(\boldsymbol{G} = \boldsymbol{g} \mid \boldsymbol{X} = \boldsymbol{\omega})\,\mathrm{P}(\boldsymbol{X} = \boldsymbol{\omega})}{\mathrm{P}(\boldsymbol{G} = \boldsymbol{g})}, \tag{5.3}$$

for $\boldsymbol{\omega} = (\boldsymbol{f}, \boldsymbol{l}) \in \Omega$ and $\boldsymbol{g} \in \phi(H(\Omega^F))$. Next, from the assumed degradation transformation (5.1) and the independence of the noise $\boldsymbol{N}$ from $\boldsymbol{X}$

$$\begin{aligned} \mathrm{P}(\boldsymbol{G} = \boldsymbol{g} \mid \boldsymbol{X} = \boldsymbol{\omega}) &= \mathrm{P}(\Psi(\phi(H(\boldsymbol{F})), \boldsymbol{N}) = \boldsymbol{g} \mid \boldsymbol{F} = \boldsymbol{f}, \boldsymbol{L} = \boldsymbol{l}) \\ &= \mathrm{P}(\boldsymbol{N} = \Psi^{-1}(\phi(H(\boldsymbol{f})), \boldsymbol{g}) \mid \boldsymbol{F} = \boldsymbol{f}, \boldsymbol{L} = \boldsymbol{l}) \\ &= \mathrm{P}(\boldsymbol{N} = \Psi^{-1}(\phi(H(\boldsymbol{f})), \boldsymbol{g})). \end{aligned}$$

Since any random process with a positive joint density can be represented as a MRF (Remark 4.2), we are able to express the sampling distribution in a Gibbs form. In the particular case of a Gaussian noise process,

$$\mathrm{P}(\boldsymbol{G} = \boldsymbol{g} \mid \boldsymbol{X} = \boldsymbol{\omega}) \propto \exp \frac{- \| \boldsymbol{\mu} - \Psi^{-1}(\phi(H(\boldsymbol{f})), \boldsymbol{g}) \|^2}{2\sigma^2}.$$

But the prior model $\boldsymbol{X}$ has the Gibbs distribution (4.12). Substituting in (5.3) and combining the constants

$$\mathrm{P}(\boldsymbol{X} = \boldsymbol{\omega} \mid \boldsymbol{G} = \boldsymbol{g}) = \frac{\exp\{-U^{\mathrm{P}}(\boldsymbol{\omega})\}}{Z^{\mathrm{P}}}, \tag{5.4}$$

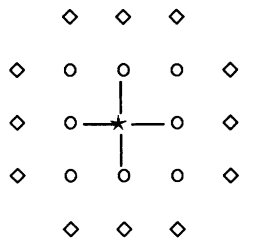


Figure 9: Posterior neighborhood.

where the posterior energy U^{P} is as in (5.2). The posterior neighborhood system, $\mathcal{G}^{\mathrm{P}}$ is obtained from the prior neighborhood system $\mathcal{G}$ by preserving the "line neighborhoods," but enlarging the pixel neighborhoods. The enlargement of the pixel neighborhood of a site s is done by including in a posterior neighborhood of s all the pixels which are not neighbors of s with respect to the prior model, but now, due to the blurring, affect the intensity at the pixel s. For illustration look at figure 9 (page 22). Let the neighborhood system for $\boldsymbol{F}$ be $\mathcal{F}_1$; the neighbors of $\star$ are the o's connected to it. Let the blurring transformation H be averaging over the eight nearest neighbors; each of the o's will influence the intensity of the $\star$ in the blurred image. By the Markov property, each o is statistically dependent on its four nearest neighbors (with respect to $\mathcal{F}_1$), and these dependences does not spread farther. Thus, the intensity at the pixel $\star$ in the degraded image is statistically dependent on all the pixels represented at figure 9 (page 22), o's, and $\diamond$'s.

Hence, the class of the MRF's has a nice feature—it describes both the prior and the posterior models. For computational reasons, MRF models with only short-range interactions (up to $10 - 20$ sites in a neighborhood [GG86a]) are suitable for image modeling. When the blurring function is "local," the prior and the posterior MRF's have similar neighborhood structures. In this case, the degradation transformation preserves the short-range of the interactions of the prior model, and the posterior model has relatively small neighborhoods.

The MAP estimate

The MAP estimate maximizes the posterior distribution (5.4) which is equivalent to maximizing

$$\log \mathrm{P}\left(\boldsymbol{X} = \boldsymbol{\omega} \mid \boldsymbol{G} = \boldsymbol{g}\right) = -U^{\mathrm{P}}(\boldsymbol{\omega}) + \text{constant}.$$

And subsequently, the MAP estimate has to minimize the posterior energy function $U^{\mathrm{P}}(\boldsymbol{\omega})$, $\boldsymbol{\omega} \in \Omega$.

6 Optimization Algorithm

The MRF's have practical representations in terms of Gibbs distributions. The MAP estimate for such a process minimizes the posterior energy function (5.2), which is an energy function of a Gibbs distribution. At this point, we are confronted by computational problems. For MAP estimate we have to search a huge number of configurations. Even for a binary image on a 64×64 lattice, with no line process, there are 2^{4096} possible configurations. Another computational difficulty, in case of a Gibbs distribution, is inherit from the partition function the computation of which is an intractable problem [GG84].

Computational methods are needed for (i) sampling from Gibbs distributions, (ii) minimizing Gibbs energy functions, and (iii) computing functions of Gibbs distributed processes.

As seen at the end of the previous section, the MAP estimate problem is reduced to the problem of minimizing Gibbs posterior energy function. Gibbs distributions describe the equilibrium states of large-scale discrete physical systems. This suggests an analogy to statistical physics. Because, for many physical systems the equilibrium states at very low temperatures have desirable properties, a fundamental question is, what is the state of the matter at these temperatures. At such temperatures the physical systems are close to ground states (the lowest energy states). A way, for exploring such states is trough lowering the temperature until a lowest energy state is reached. But just lowering the temperature is not enough; during that process, the system has to be kept in equilibrium. The cooling process is therefore very delicate. The *chemical annealing* is a method for obtaining low energy states of a material: first the substance is melted at high temperature (so the equilibrium is reached fast), then the temperature is lowered gradually; enough time is spent at low temperatures for the system to reach equilibrium states.

6.1 Simulated annealing

In analogy with the chemical annealing Kirkpatrick [KGV83] develops *"simulated annealing"*, for solving combinatorial optimization problems. The optimization problem is to obtain a minimum energy configuration for the Gibbs distribution

$$\pi(\omega) = \frac{\exp\{-U(\omega)/T\}}{Z}. \tag{6.1}$$

Note, that the temperature T is a scale parameter of π (6.1), so a configuration which maximizes π (minimizes the energy U) does not depend on T. Hence, if we obtain a configuration which has a minimal energy at some very low temperature, this same configuration is a solution of the optimization problem. In the simulated annealing, a

solution of the optimization problem is identified with a ground state of an imaginary physical system with energy U. A control parameter T (temperature) is introduced. For a fixed value of the temperature T, a simulation of a collection of the elements of the system in equilibrium, at that temperature, is performed. For that purpose, some algorithm for sampling from the Gibbs distribution (6.1) is emploied. Next, the temperature T is lowered, and at the new temperature the sampling algorithm is repeated and so on. At high temperatures the equilibrium states are easily reached, fewer iterations, through the steps of the sampling algorithm employed, are needed.

In his work [KGV83], Kirkpatrik employs the Metropolis' algorithm, to do the sampling at any fixed temperature. The *Metropolis' algorithm* provides a computational technique to determine the equilibrium properties, especially ensemble averages, time evolution, and low-temperature behavior, of very large systems of essentially identical, interacting components, such as molecules in a gas or atoms in binary alloys. Let Ω be the configuration space (all possible states of the system). If the system is in thermal equilibrium with its surroundings, the probability that it is in a certain configuration $\boldsymbol{\omega}$ is $\pi(\boldsymbol{\omega})$. The iteration scheme is as follows. Let $\boldsymbol{X}(t)$ denote the state of the system at time t. Given $\boldsymbol{X}(t-1)$ transfer to $\boldsymbol{X}(t)$, as follows. *Randomly* select a configuration $\boldsymbol{\omega}$, and compute the energy change $\Delta U = U(\boldsymbol{\omega}) - U(\boldsymbol{X}(t-1))$ and the quantity

$$q \stackrel{\text{def}}{=} \frac{\pi(\boldsymbol{\omega})}{\pi(\boldsymbol{X}(t-1))} = \exp\{ -\beta \Delta U \}.$$

If $q > 1$, set $\boldsymbol{X}(t) = \boldsymbol{\omega}$. If $q \leq 1$, set $\boldsymbol{X}(t) = \boldsymbol{\omega}$ with probability q, and set $\boldsymbol{X}(t) = \boldsymbol{X}(t-1)$ with probability $1 - q$.

When the temperature T is near zero, the imaginary physical system should converge to a state of minimal energy.

This is a *stochastic relaxation* algorithm. The essence of such an algorithm is that transfers from a state $\boldsymbol{X}(t-1)$ to a state $\boldsymbol{X}(t)$ are permitted even if the energy of the system will be increased (remember that the goal is to obtain minimal energy state). This is not the case with deterministic iterative algorithms, where only changes leading to decreasing of the energy are allowed. (Always go "down-hill"!) The latter scheme may lead the system to a state of locally minimal energy (local minimum), from which "back tracking" is the natural way to continue the search. But the latter is computationally very expensive. Stochastic relaxation algorithms avoid that situation, by allowing, occasionally, sequencing through states which will lead to increase of the energy. Unfortunately, it may take a very long time for such un algorithms, to find a state of minimal energy. Some scheme which will guarantee better convergence is needed.

The scheme for lowering the temperature, and the number of the rearrangements of the system, attempted at each fixed temperature, is the *annealing schedule.* The annealing schedule which Kirkpatrick suggests is determined by trial and error. This is not satisfactory. Geman and Geman [GG84] designed the *Gibbs sampler*, a stochastic relaxation scheme for image processing, which is a variation of the Metropolis' algorithm. The computations in the Gibbs sampler are simple and local. Geman and Geman also propose an annealing schedule which guarantees convergence. The dynamics of the chemical annealing are simulated by producing a Markov chain $\boldsymbol{X}(1), \boldsymbol{X}(2), \ldots,$ with Gibbs equilibrium distribution [GG84]. When this distribution is the posterior distribution (5.4) the simulated annealing algorithm obtains a solution of the MAP estimate problem.

6.2 The Gibbs sampler

Let $\boldsymbol{X}$ be a MRF over a graph $\{S, \mathcal{G}\}$ with Gibbs distribution

$$\pi(\boldsymbol{\omega}) = \frac{\exp\{-U(\boldsymbol{\omega})/T\}}{Z}, \quad T = \text{constant}. \tag{6.2}$$

The *Gibbs sampler* is a procedure for sampling from (6.2). Note that once the sampling problem is solved, the simulated annealing equipped with the sampling algorithm, approximates a solution of the MAP problem.

For a fixed temperature T, the transition scheme is as follow. Imagine a simple processor at every site $s \in S$. For each site $s \in S$, the processor at the site s is connected only to the processors at the neighbors of s. The size of the lattice S is very big, but the size of the neighborhoods is modest. All processors follow the same simple algorithm. The system evolves due to discrete changes in time. All sites of the lattice S are visited in some order. At each time step, the current configuration of the system undergoes a possible change only at one site. Hence, the states of the system in two consecutive time steps can differ in at most one coordinate. The Gibbs sampler algorithm is based on Geman and Geman's Relaxation theorem. This theorem states that, regardless of the initial configuration, and the sequence in which the sites are visited for replacement, provided that all sites are visited infinitely many times, the distribution of the sequence, $\{X(t)\}_{t\geq 1}$, produced by the Gibbs sampler, converges in distribution to the Gibbs distribution π (6.2).

Let $\{\, t : t \geq 0 \,\}$ denotes the discrete time sequence, $\{\, n_t \,\}_{t\geq 0}$ the sequence in which the sites are visited for update, and $\boldsymbol{X}(t)$ the state of the system at time t, $t \geq 0$. At time t the state of the system at site s is $X_s(t)$, $X_s(t) \in \Lambda$. The total configuration, $\boldsymbol{X}(t) = (X_{s_1}(t), \ldots, X_{s_N}(t))$, evolves due to state changes, such that, at time t, $X_s(t) = X_s(t-1)$, $s \neq n_t$, and only the state at site n_t is changed. At time t, the processor at site n_t computes the local characteristic $\pi(X_{n_t} = x_{n_t} | X_s = x_s : s \in$

$\mathcal{G}_{n_t}$), $\forall x_{n_t} \in \Lambda$; recall that this local characteristic is easily computable in terms of the potentials over the cliques which contain the site n_t. Next, a *random* sample is drawn with respect to the probability low of the local characteristic, and this sample replaces the state at the site n_t. The probability that at time t, the system is in a state $\boldsymbol{\omega} = (x_{s_1}, \ldots, x_{s_N})$ is

$$\mathrm{P}\,(\,\boldsymbol{X}(t) = \boldsymbol{\omega}\,) \;=\; \pi(\,X_{n_t} = x_{n_t} \mid X_s = x_s : s \in \mathcal{G}_{n_t}\,)\,\mathrm{P}\,(\,X_s(t-1) = x_s : s \neq n_t\,),$$

where for any s in S, $\mathcal{G}_s$ denotes the neighborhood of s.

Theorem 6.1 (Relaxation) *Let $\boldsymbol{X}$ is a MRF, over $\{S, \mathcal{G}\}$, represented by a Gibbs distribution π. Assume that the sequence $\{n_t,\, t \geq 1\}$ contains each $s \in S$ infinitely many times. Than for any starting configuration $\boldsymbol{\eta} \in \Omega$, and for any configuration $\boldsymbol{\omega} \in \Omega$*

$$\lim_{t \to \infty} \mathrm{P}\,(\,\boldsymbol{X}(t) = \boldsymbol{\omega} \mid \boldsymbol{X}(0) = \boldsymbol{\eta}\,) = \pi(\boldsymbol{\omega}),$$

where $\{\boldsymbol{X}(t)\}_{t=1}^{\infty}$ is the sequence produced by the Gibbs sampler.

Geman and Geman implement the Gibbs sampler in a raster scan fashion. In the raster version of the Gibbs sampler, the processor updates the states of the lattice sites in order. At each site s, the neighborhood relation and the values at the neighbors are loaded, then a sample is drawn from the local characteristic of s, and that sample replaces the state at site s. The time for one complete iteration grows linearly with the size N of the lattice S (a complete iteration is one full swept through S).

A higher degree of parallelism can be achieved if the lattice S is divided into sublattices. A processpor is assigned to each sublattice. Each procrssor runs a raster-scan version of the Gibbs sampler in a sublattice. Since the convergence of the algorithm does not depend on the initial state and the sequence in which the different sites are visited (Relaxation theorem), all the processors can run simultaneously. If the division of S is done with respect to the natural topology of the scene the communication time will be diminished. The time for a complete iteration depends on the longest time for iterating over a sublattice, so it will grow linearly with the maximal size of a sublattice. Due to hardware limitations, this was the best parallelism which could be achieved at the time the Gibbs sampler was designed.

With the development of massively parallel high-performance computers, higher degree of parallel implementation is achieved. Marroquin [Mar85] implemented the Gibbs sampler on the Connection Machine. A processor is assigned to each site. The number of steps necessary for a complete iteration is determined by the *chromatic number* of the graph with respect to which MRF is defined.

The *chromatic number* of a graph is the minimal number of colors needed to color the

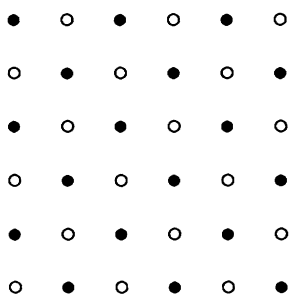


Figure 10: Scheme for parallel implementation of Gibbs sampler.

graph in such a way that no neighbors have the same color. This number is bounded below by the size of the largest clique. In a parallel implementation of the Gibbs sampler, we may update simultaneously all the states which have the same color. States which belong to a same clique cannot be updated simultaneously. The execution time for one complete iteration, will decrease with respect to the raster scan version, with a factor of N/H, where H is the chromatic number of the graph.

Figure 10 (page 27) shows an example of coloring the lattice for the Ising model (four-neighbor homogeneous neighborhood system) with two colors—black and white. A complete iteration may be done in two steps. In the first step all white pixels are updated, and in the second step all black pixels are updated.

6.3 Annealing scheme

An "artificial temperature" T is introduced into the posterior distribution. This temperature is lowered in a way which forces the system into a minimum energy state. The Gibbs sampler is incorporated with this process of lowering the temperature. The difference between that scheme and the Gibbs sampler is that the latter produces a sequence of configurations, at constant temperature.

First we introduce some notations. Let $T(t)$ denotes the temperature at time t, and $\pi^{T(t)}$ is the Gibbs measure corresponding to that temperature. The Gibbs sampler incorporated with the annealing scheme, generates a sequence $\{\boldsymbol{X}(t),\, t \geq 1\}$ such that

$$\mathrm{P}\,(\,\boldsymbol{X}(t) = \boldsymbol{\omega}\,) \;=\; \pi^{T(t)}(\,X_{n_t} = x_{n_t} \mid X_s(t) = x_s : r \in \mathcal{G}_{n_t}\,)\,\mathrm{P}\,(\,X_s(t-1) = x_s : s \neq n_t\,).$$

Theorem 6.2 (Annealing) *Assume that there exists an integer $\tau \geq N$ such that for every $t = 0, 1, 2, \ldots$, $S \subseteq \{n_{t+1}, n_{t+2}, \ldots, n_{t+\tau}\}$. Let $T(t)$ be a decreasing sequence of temperatures for which*

1. $\lim_{t\to\infty} T(t) = 0$,

2. *There exists* t_0 *such that*

$$T(t) \geq N\Delta / \log t, \quad for\ all \quad t \geq t_0,$$

where Δ *is the difference between the maximum and the minimum values of* $U(\boldsymbol{\omega}), \boldsymbol{\omega} \in \Omega$.

Then for any starting configuration $\boldsymbol{\eta} \in \Omega$ *and for every* $\boldsymbol{\omega} \in \Omega$,

$$\lim_{t \to \infty} \mathrm{P}\left(\boldsymbol{X}(t) = \boldsymbol{\omega} \mid \boldsymbol{X}(0) = \boldsymbol{\eta} \right) = \mathcal{U}(\boldsymbol{\omega}), \tag{6.3}$$

where $\mathcal{U}$ *is the uniform distribution over the subset of* Ω *on which* U *attains its minimum.*

This stochastic relaxation algorithm generates a Markov chain which converges in distribution to the uniform distribution, $\mathcal{U}$, over the minimal energy configurations [GG84]. A major practical weakness of the algorithm is the second condition—$N\Delta$ is too big. However, this is consistent with the physical experiments, in the sense that T must be lowered very slowly, particularly near the freezing point [GG84]. For their experiments Geman and Geman use the annealing scheme

$$T(k) \geq \frac{\Gamma}{\log(1+k)}, \ 1 \leq k \leq K, \tag{6.4}$$

where $T(k)$ is the temperature during the k^{th} iteration (one iteration is one complete sweep trough the lattice S), and K is the total number of iterations. The constant Γ is estimated by ad hoc methods. The bound for Γ has subsequently been improved by other authors. "The smallest constant which guarantees convergence of the annealing algorithm can be specified in terms of the energy function." [GG86b].

Remark 5.1 We remind the definition of *convergence with probability one.* Given a sequence of random variables $\{\boldsymbol{Z}_n\}_{n=1}^{\infty}$, and a constant c, $\{\boldsymbol{Z}_n\}_{n=1}^{\infty}$ converges to c with *probability one* if, for any positive numbers ε and δ, there exists a positive integer $N(\varepsilon, \delta)$ such that

$$\mathrm{P}\left(\bigcap_{n \geq N(\varepsilon,\delta)} \{\boldsymbol{\eta} \in \Omega : |\boldsymbol{Z}_n(\boldsymbol{\eta}) - c| < \varepsilon\} \right) \geq 1 - \delta.$$

The next theorem concerns the ergodicity of the sequence produced by the Gibbs sampler. We are interested in computing the average of some function, Y, with respect to a Gibbs distribution. This cannot be accomplished analytically (due to the partition function) [GG84]. In some cases Monte Carlo Methods are applied: R samples are taken based on an uniform distribution on Ω, and then the average of Y

with respect to $\pi(\boldsymbol{\omega})$ is approximated by the ergodic average (the sample mean of the values of Y at the sampled configurations). But this strategy is not practical when $\pi(\boldsymbol{\omega})$ is a Gibbs distribution because with high probability we sample low probability configurations (the exponential factor puts more mass over a small subset of Ω). Fortunately the following theorem holds [GG84].

Theorem 6.3 (Ergodicity) *Assume $\{\boldsymbol{X}(t)\}_{t=0}^{\infty}$ is the sequence generated by the Gibbs sampler at constant temperature T, and that there exists τ such that for all $t \geq 0$, $S \subset \{n_{t+1}, \ldots, n_{t+\tau}\}$. Then, for every real-valued function $Y : \Omega \rightarrow \mathrm{R}$, and for every starting configuration $\boldsymbol{\eta} \in \Omega$,*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} Y(\boldsymbol{X}(t)) = \int_{\Omega} Y(\boldsymbol{\omega})\, d\pi(\boldsymbol{\omega})$$

holds with probability one, namely,

$$\mathrm{P}\left(\lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} Y(\boldsymbol{X}(t)) = \int_{\Omega} Y(\boldsymbol{\omega})\, d\pi(\boldsymbol{\omega}) \right) = 1.$$

7 Applications

In Bayesian approach to computational vision, several major components have to be considered

1. Prior probability model for the original image. The prior models we present are MRF's.

2. Degradation model which reflects the transformation of the original image into the observed data. This may include a sensor model.

3. An estimator. The estimator we consider is the one which uses the MAP.

4. Efficient optimization algorithms. The challenges of the problems in computer vision are (i) the optimizations have to be done over huge configuration spaces, and (ii) the real-time restriction in present in most of the computer vision applications. The question of efficient algorithms is vital.

7.1 Restoration of Images

The image restoration problem is to infer original unobservable image $\boldsymbol{X}$ from observed degraded image $\boldsymbol{Z}$. The prior models are MRF's. They are specified by Gibbs distributions, the potentials of which model the spatial coherence in the original images.

Here we present experimental results in image restoration from [GG84]. The restored images are approximations of the MAP estimates obtained by using the Gibbs sampler and the annealing scheme based on the Annealing theorem 6.2 and annealing schedule (6.4).

For any of the illustrated experiments, the prior process $\boldsymbol{X}$ is either a simple MRF, consisting only of an intensity process $\boldsymbol{F}$, or a hierarchical MRF $(\boldsymbol{F}, \boldsymbol{L})$, consisting of an intensity process $\boldsymbol{F}$, and a line processes $\boldsymbol{L}$. The neighborhood system is homogeneous four- or eight-neighbor systems (4.4). For the intensity process, $\boldsymbol{F}$, only potentials over the pair-cliques may be different than zero. The particular prior models Geman and Geman use for illustration in [GG84] are both homogeneous and isotropic MRF. The mean of X_s is a non-zero constant for every site $s \in S$, and the correlations between any two random variables are preserved by rigid body motions on S [Won71].

Only the intensity process $\boldsymbol{F}$ may be degraded, the line process $\boldsymbol{L}$is preserved. The total degraded image is $\boldsymbol{Z} = (\boldsymbol{G}, \boldsymbol{L})$, where $\boldsymbol{G}$ is the observed degradation of the intensity process. The observed image $\boldsymbol{G}$ is obtain from $\boldsymbol{F}$ through degradation transformation which satisfies the restrictions specified in Section 5.2. In particular, the degradations are obtained by the following transformations, or their combinations:

- A blurring effect H modeled by a convolution over a small window $\boldsymbol{H} = \{h_{kl}\}_{k,l=-1}^{1}$:

$$h_{kl} = \begin{cases} 1/2, & k = 0, l = 0, \\ 1/16, & 0 < (k^2 + l^2) \leq 2. \end{cases} \tag{7.1}$$

  At each pixel $(i, j) \in Z_m$, the blurring effect $H(\boldsymbol{F})$ is

$$(\, H(\boldsymbol{F})\,)_{ij} = \sum_{k,l=-1}^{1} h_{k,l} F_{i+k,j+l}.$$

  "In this case the intensity at (i, j) is weighted equally with the average of the eight nearest neighbors" [GG84].

- Nonlinear transformation ϕ absent or $\phi(x) = \sqrt{x}$.

- Degradation map Ψ which is either addition or multiplication of its arguments (additive or multiplicative noise)

$$\begin{aligned} \boldsymbol{Z} &= (\boldsymbol{G}, \boldsymbol{L}), \text{ where} \\ \boldsymbol{G} &= \Psi(\, \phi(H(\boldsymbol{F})), \boldsymbol{N}\,). \end{aligned}$$

All restorations are approximations of MAP estimates, generated by a serial Gibbs sampler. The update of the states is done in a raster fashion.

The annealing schedule is

$$T(k) = \frac{\Gamma}{\log(1+k)}, \ 1 \leq k \leq K,$$

where $T(k)$ is the temperature during the k^{th} iteration, and K is the total number of iterations. The constant Γ is 3.0 or 4.0, estimated by trial and error.

Different signal-to-noise levels are examined. The signal to noise ratios are very low. As it can be seen in the Appendix the algorithm gives satisfactory results.

Two groups of experiments are conducted.

7.1.1 Computer-generated original image

The first class of experiments uses an original image which is sampled from a MRF over Z_{128} with the eight-neighbor system. There is no line process, so $\boldsymbol{X} = \boldsymbol{F}$. The pixel process has 5 intensity levels. The form of the potentials over the pair-cliques is

$$V_C(\boldsymbol{f}) = \begin{cases} -\frac{1}{3}, & \text{if the elements of C have the same intensity,} \\ \frac{1}{3}, & \text{otherwise.} \end{cases}$$

The original image is obtained using Gibbs sampler with two hundred iterations at temperature $T = 1$. Experiments are performed with two different degradation transformations.

Case 1.1 *Additive Gaussian noise.* See the Appendix, figure 13 (page 49). The degraded image in the first experiment, $\boldsymbol{Z} = \boldsymbol{G}$, is obtained by adding Gaussian noise with $\sigma = 1.5$ to the original image

$$\boldsymbol{G} = \boldsymbol{X} + \boldsymbol{N}.$$

Gibbs sampler with $K = 25$ and $K = 300$ iterations is run. From the results we conclude that the bigger the number of iterations, the better the restoration is.

Case 1.2 *Blur, nonlinear transformation and multiplicative Gaussian noise.* See the Appendix, figure 14 (page 50). The degraded image is obtained by blurring the computer generated image (convolving with $\boldsymbol{H}$ (7.1)), applying square root operation at every pixel, and multiplying ($\odot$) by Gaussian noise with $\mu = 1, \sigma = 0.1$

$$\boldsymbol{Z} = \boldsymbol{G} = \sqrt{\boldsymbol{H} * \boldsymbol{X}} \odot \boldsymbol{N}.$$

Again, the Gibbs sampler with $K = 25$ and $K = 300$ interactions is run. The increased complexity of the degradation transformation does not affect the quality of

the restored image. This is consistent with the Relaxation and Annealing theorems. As long as the locality is preserved in the degradation, and the degradation transformation Ψ is invertible with respect to the first argument, these theorems guarantee the convergence.

7.1.2 Hand-drawn original image

The second class of experiments uses a "hand-drawn" original image on Z_{64} with three intensity levels. It consists of overlapping rectangles with parallel sides. See the Appendix, figure 18 (page 52). The prior model for this image has to be designed. Several experiments are conducted.

Case 2.1 *Prior: no line process; Degradation: additive Gaussian noise.* See the Appendix, figure 18 (page 52). The prior for the original image is without presence of a line process, $\boldsymbol{X} = \boldsymbol{F}$. It does not account of the straight line edges. A Gaussian noise $\mu = 0, \sigma = 0.7$ is added to the original image $\boldsymbol{Z} = \boldsymbol{X} + \boldsymbol{N}$. Some big blobs which miss in the original image, are present in the restored image.

Case 2.2 *Prior: with line process; Degradation: additive Gaussian noise.* See the Appendix, figure 18 (page 52). For a comparison and an improvement a binary line process is adjointed to the original image model, $\boldsymbol{X} = (\boldsymbol{F}, \boldsymbol{L})$. The line process is as in section 4.3. The energy $U(\boldsymbol{f}, \boldsymbol{l})$ for the prior is modeled by $-U(\boldsymbol{f}|\boldsymbol{l}) - U(\boldsymbol{l})$. For $U(\boldsymbol{l})$, only potentials over cliques of size four may be nonzero. See the Appendix, figure 15 (page 51). A line process component models a lack (coded as 0) or a presense (coded as 1) of an edge at a line site. Depending on the site the edge may be horizontal or vertical. See the Appendix, figure 16 (page 51). There are six possible configurations for "line elements" in a clique. These configurations are unique up to rotations, and are specified by a presense (bar) or a lack (empty space) of an edge element at the line sites. We express our preferences to these configurations by assigning different values to the potentials indexed by the corresponding cliques. See the Appendix, figure 17 (page 51).

In the interaction term $U(\boldsymbol{f}|\boldsymbol{l})$ only potentials over pair-cliques may be non-zero. Over such a clique the potential $V_C(\boldsymbol{f}|\boldsymbol{l}) = 0$ if there is an edge between the pixels of the clique C. If there is no edge

$$V_C(\boldsymbol{f}|\boldsymbol{l}) = \begin{cases} 1 & \text{if the pixels of C have same intensity,} \\ -1 & \text{otherwise.} \end{cases}$$

Looking at the restored images obtained in Case 2.1 and 2.2, shown in the Appendix, figure 18 (page 52), it is clear that the restoration with a presence of a line process is much more suitable. This experiments also show how important the prior probabilistic mode in Bayes estimation is. This prior is task dependent. In each case careful

examination of the image is necessary. The result should be a neighborhood system, and a model of the interactions between the neighboring pixels in terms of potentials.

Case 2.3 *Prior: with line process; Degradation: blur, nonlinear transformation and multiplicative Gaussian noise.* See the Appendix, figure 19 (page 53). The prior model is as in Case 2.2, $\boldsymbol{X} = (\boldsymbol{F}, \boldsymbol{L})$. The line process is not subjected to any degradation. The degradation transformation for the pixel process is as in Case 1.2

$$\boldsymbol{Z} = (\boldsymbol{G}, \boldsymbol{L}),\ \boldsymbol{G} = \sqrt{\boldsymbol{H} * \boldsymbol{X}} \odot \boldsymbol{N}.$$

For the restoration $K = 1000$ iterations are done. Even with this nontrivial degradation model the result examined by eye is satisfactory.

7.2 Texture modeling

We present an application of MRF for texture modeling, and the use of these models for texture segmentation from [GG86b]. By texture we mean the spatial distribution of the surface markings. It can refer to both, statistical variations in intensity (micro-structure texture), and patterns of lines or shape tokens (macro-structure texture). Segmentation refers both to dividing an image into homogeneous regions (in intensity, color, texture), and labeling each pixel with a label identifying the region to which that pixel belongs. When texture is used in segmentation, the different regions are identified not by their intensities, but by their texture types.

Geman and Graffigne [GG86b] use hierarchically structured MRF to model images consisting of different natural textures. The prior model, for the texture modeling application [GG86b], consists of a *pixel process* and a *texture label process.* Simulated annealing with Gibbs Sampler is used for approximating the MAP estimate for segmenting a scene consisting of patches of natural textures (wood, plastic, carpet).

The prior model

Images are represented on an $M \times M$ pixel grid. The original image $\boldsymbol{X}$ is a MRF over a graph $\{S, \mathcal{G}\}$. The lattice is $S = S^P \cup S^L$, where S^P is the square lattice of pixel intensity sites, and S^L is the same square lattice as S^P, but it represents sites for texture labels. The random vector $\boldsymbol{X}$ has two components, a pixel process $\boldsymbol{X}^P = \{X_s^P\}_{s \in S^P}$, and a texture label process $\boldsymbol{X}^L = \{X_s^L\}_{s \in S^L}$, $\boldsymbol{X} = (\boldsymbol{X}^P, \boldsymbol{X}^L)$. The pixel process components may take any of sixteen gray levels, $x_s^P \in \{0, 1, \ldots, 15\}$. The texture label process components may take any of N given texture labels, $x_s^L \in \{1, 2, \ldots, N\}$, $s \in S^L$.

The neighborhood system for the pixel process is a nonstandard one. There is no uniform way for choosing it. There are only six types of cliques. From here, a

Figure 11: The six types of pair-cliques and the neighborhood structure for the pixel process, fixed texture type.

diamond-shaped pixel neighborhoods are inferred. See figure 11 (page 34). For different types of textures, the neighboring pixels have different degrees to which they tend to have similar grey levels. For fixed texture type $l \in \{1,\ldots,N\}$, the degrees of dependence of the intensities of the neighboring pixels, are modeled by six *texture dependent parameters* $\{\theta_i^{(l)}\}_{i=1}^6$. Roughly, for positive parameters neighbors tend to have similar intensities, and for negative parameters, the intensities tend to be different. For each texture type l, $l = 1,\ldots,N$, the texture dependent parameters $\theta_i^{(l)}$, $i = 1,\ldots,6$, are estimated from a single training sample.

For a given texture type $l \in \{1,2,\ldots,N\}$, the pixel intensity process $\boldsymbol{X}^P$ is a MRF specified by the Gibbs distribution

$$\mathrm{P}\,(\,\boldsymbol{X}^P = \boldsymbol{x}^P \mid X_s^L = l : s \in S^L\,) = \frac{\exp\{\,-U^{(l)}(\boldsymbol{x}^P)\,\}}{Z^{(l)}}, \tag{7.2}$$

where $Z^{(l)}$ is a normalizing constant. Only the six pair-clique types (figure 11 (page 34)) appear in the energy function

$$U^{(l)}(\boldsymbol{x}^P) \stackrel{\text{def}}{=} -\sum_{i=1}^{6} \sum_{\langle s,t\rangle_i} \theta_i^{(l)} \Phi(x_s^P - x_t^P). \tag{7.3}$$

For any index i of the outher sum, the summation in the inner sum is over all pair-clique parameters $\langle s,t\rangle_i$ of type i. The function Φ is defined as follows

$$\Phi(\Delta) \stackrel{\text{def}}{=} \left(1 + \frac{\Delta^2}{\delta^2}\right)^{-1}, \ \Delta \in \{-15,\ldots,0,\ldots,15\}, \tag{7.4}$$

where δ is some positive parameter. There is no general method for designing the function Φ. As Geman and McLure [GM87] explain, the exact form of Φ is not important. What matters are its qualitative features. For the particular problem, a function Φ sensitive to difference in intensities of neighboring pixels is needed. But the

energy also have to account for certain type of intensity changes (edges, boundaries). The authors in [GG86b], [GM87] examine different types of $\Phi(\Delta)$, mainly decreasing in $|\Delta|$. Hence the energy function (7.3) is increasing in $|\Delta|$, (if $\theta_i^{(l)} > 0, i = 1, \ldots, 6$). This is consistent with the heuristic that more likely are close intensity levels for the neighboring pixels. The natural choice for Φ is $\Phi(\Delta) = \Delta^{-2}$ [GM87]. But in this case the estimator results in *oversmoothing* of the original image. Under this choice of Φ some natural boundaries (edges) are unlikely. Instead, to account for the oversmoothing, a function Φ of the form (7.4) is considered. The scale parameter δ is estimated based on the range of the possible intensity values. Geman and McLure establish that the reconstruction is not sensitive to moderate changes of δ [GM87].

After the probability models for all texture types $l \in \{1, 2, \ldots, N\}$ are specified separately (7.2), a composite MRF $\boldsymbol{X}$ which couples pixel intensities and texture labels is specified. It has the joint Gibbs distribution

$$\mathrm{P}\left(\boldsymbol{X}^P = \boldsymbol{x}^P, \boldsymbol{X}^L = \boldsymbol{x}^L\right) = \frac{\exp\{-U_1(\boldsymbol{x}^P, \boldsymbol{x}^L) - U_2(\boldsymbol{x}^L)\}}{Z}. \tag{7.5}$$

The interactions between texture labels and intensities are given by $U_1(\boldsymbol{x}^P, \boldsymbol{x}^L)$. The heuristic is that each texture label at a pixel s, is influenced by the gray levels of the other pixels around it. The interactions are on three levels (i) local pixel-based interactions, (ii) local block-based interactions, and (iii) global interactions.

The *local pixel-based* interactions account for the influence from the gray levels at the pixels which are immediate neighbors of the site s

$$H(\boldsymbol{x}^P, l, s) \stackrel{\text{def}}{=} -\sum_{i=1}^{6} \theta_i^{(l)} \sum_{t:\langle s,t\rangle_i} \Phi(x_s^P - x_t^P).$$

The *local block-based* interactions account for the influence from local pixel-based interactions at the pixels in a 5×5 block of sites centered at s

$$Z(\boldsymbol{x}^P, l, s) \stackrel{\text{def}}{=} \frac{1}{a} \sum_{t \in B_s} H(\boldsymbol{x}^P, l, t).$$

Here B_s denotes the set of 5×5 sites centered at s, and a is a constant which accounts for the fact that with the use of block interactions some cliques will contribute more than once to the total energy $U^{(l)}(\boldsymbol{x}^P)$. The constant a adjust the sum of the block-based interactions such that it would be consistent with the energy function for homogeneous textures (7.3)

$$U^{(l)}(\boldsymbol{x}^P) = \sum_{s \in S} Z(\boldsymbol{x}^P, l, s).$$

Finally, the "interaction energy" models the *global* interactions between the block-based interactions at all sites $s \in S$

$$U_1(\boldsymbol{x}^P, \boldsymbol{x}^L) \stackrel{\text{def}}{=} \sum_{s \in S} Z(\boldsymbol{x}^P, x_s^L, s).$$

Note that this model is consistent with the homogeneous texture model (7.3), $X_s^L = l$, for all s in S.

Since the textures are expected to appear in patches, the Ising like model with four-neighbor homogeneous system is suitable for the specification of the texture potential energy U_2 in (7.5)

$$U_2(\boldsymbol{x}^L) = -\beta \sum_{\{s,r\}} 1_{x_s^L = x_r^L} + w(\boldsymbol{x}^L),$$

where the sum is taken over all horizontal and vertical pair-cliques, and w is a "bias correction term" [GG86b].

Parameter estimation

The parameters δ (for Φ) and β (for U_2) are determined by trial and error. The estimation of the pair-clique parameters $\theta_i^{(l)}, i = 1, \ldots, 6,\ l = 1, \ldots, N$, is systematic, trial and error methods are not feasible [GG86b]. These parameters are crucial for the performance of the model [GG86b]. They are estimated using a "training sample" $\tilde{\boldsymbol{x}}^P$ for each fixed texture l. Let us denote the pair-clique parameters for arbitrary fixed texture type with $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_6)$. The prior model for the pixel process and the homogeneous texture type depends on the parameter $\boldsymbol{\theta}$

$$\mathrm{P}\,(\boldsymbol{X}^P = \boldsymbol{x}^P; \boldsymbol{\theta}) = \frac{\exp\{-U(\boldsymbol{x}^P; \boldsymbol{\theta})\}}{Z(\boldsymbol{\theta})}. \tag{7.6}$$

We are given a single "training sample," it corresponds to $\boldsymbol{x}^P = \tilde{\boldsymbol{x}}^P$. Our goal is to estimate the parameter $\boldsymbol{\theta}$ in (7.6) for $\boldsymbol{x}^P = \tilde{\boldsymbol{x}}^P$.

One popular estimate is the *maximum likelihood estimate* (MLE). It is the value, $\hat{\boldsymbol{\theta}}$ which maximizes the likelihood function (7.6)

$$\mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}_P; \boldsymbol{\theta}) \leq \mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}^P; \hat{\boldsymbol{\theta}}).$$

Maximizing the likelihood function is equivalent to maximizing its logarithm $\log \mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}^P; \boldsymbol{\theta})$. The latter function is concave in $\boldsymbol{\theta}$ with a gradient

$$\nabla \log \mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}^P; \boldsymbol{\theta}) = \{\frac{\partial \log \mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}^P; \boldsymbol{\theta})}{\partial \theta_i}\}_{i=1}^6.$$

From (7.6) and (7.3) with $\theta_i^l = \theta_i, i = 1, \ldots, 6$

$$\begin{aligned}
\frac{\partial \log \mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}^P; \boldsymbol{\theta})}{\partial \theta_i} &= \frac{\partial \log(\exp\{-U(\tilde{\boldsymbol{x}}^P; \boldsymbol{\theta})\}/Z(\boldsymbol{\theta}))}{\partial \theta_i} \\
&= \frac{\partial\{-U(\tilde{\boldsymbol{x}}^P; \boldsymbol{\theta}) - \log Z(\boldsymbol{\theta})\}}{\partial \theta_i} \\
&= \frac{\partial\{\sum_{i=1}^{6} \sum_{\langle s,t\rangle_i} \theta_i \Phi(\tilde{x}_s^P - \tilde{x}_t^P)\}}{\partial \theta_i} - \frac{1}{Z(\boldsymbol{\theta})} \frac{\partial Z(\boldsymbol{\theta})}{\partial \theta_i} \\
&= \sum_{\langle s,t\rangle_i} \Phi(\tilde{x}_s^P - \tilde{x}_t^P) - \frac{1}{Z(\boldsymbol{\theta})} \frac{\partial \sum_{x^P \in \Omega} \exp\{-U(\boldsymbol{x}^P; \boldsymbol{\theta})\}}{\partial \theta_i} \\
&= \sum_{\langle s,t\rangle_i} \Phi(\tilde{x}_s^P - \tilde{x}_t^P) - \frac{1}{Z(\boldsymbol{\theta})} \sum_{x^P \in \Omega} \exp\{-U(\boldsymbol{x}^P; \boldsymbol{\theta})\} \frac{\partial\{-U(\boldsymbol{x}^P; \boldsymbol{\theta})\}}{\partial \theta_i} \\
&= \sum_{\langle s,t\rangle_i} \Phi(\tilde{x}_s^P - \tilde{x}_t^P) - \sum_{x^P \in \Omega} \frac{\exp\{-U(\boldsymbol{x}^P; \boldsymbol{\theta})\}}{Z(\boldsymbol{\theta})} \sum_{\langle s,t\rangle_i} \Phi(x_s^P - x_t^P).
\end{aligned}$$

But

$$\sum_{x^P \in \Omega} \frac{\exp\{-U(\boldsymbol{x}^P; \boldsymbol{\theta})\}}{Z(\boldsymbol{\theta})} \sum_{\langle s,t\rangle_i} \Phi(x_s^P - x_t^P) = \mathrm{E}_{\boldsymbol{\theta}} \sum_{\langle s,t\rangle_i} \Phi(X_s^P - X_t^P),$$

where the expectation $\mathrm{E}_{\boldsymbol{\theta}}$ is with respect to the prior (7.6) for fixed $\boldsymbol{\theta}$. Hence

$$\frac{\partial \log \mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}^P; \boldsymbol{\theta})}{\partial \theta_i} = \sum_{\langle s,t\rangle_i} \Phi(\tilde{x}_s^P - \tilde{x}_t^P) - \mathrm{E}_{\boldsymbol{\theta}} \sum_{\langle s,t\rangle_i} \Phi(X_s^P - X_t^P), \tag{7.7}$$

where the sums are taken over the pair-cliques of i^{th} type. To maximize $\log \mathrm{P}\,(\boldsymbol{X}^P = \tilde{\boldsymbol{x}}^P; \boldsymbol{\theta})$ we have to solve for $\boldsymbol{\theta}$ the system

$$\sum_{\langle s,t\rangle_i} \Phi(\tilde{x}_s^P - \tilde{x}_s^P) - \mathrm{E}_{\boldsymbol{\theta}} \sum_{\langle s,t\rangle_i} \Phi(X_s^P - X_t^P) = 0, \; i = 1, 2, \ldots, 6.$$

The expectation $\mathrm{E}_{\boldsymbol{\theta}}$ is with respect to the prior (7.6) for fixed $\boldsymbol{\theta}$. This expectation cannot be computed directly. One possible way is to approximate it using the Ergodic theorem (Theorem 5.3, Section 5.4).

But even this computation can be omitted. An efficient improvement for a homogeneous MRF is achieved by replacing the MLE with the *Maximum Pseudolikelihood Estimate* (MPLE) [Bes74]. The MPLE maximizes the pseudolikelihood function

$$\mathrm{P\,L}\,(\boldsymbol{x}^P; \boldsymbol{\theta}) \stackrel{\text{def}}{=} \prod_s \mathrm{P}\,(X_s^P = x_s^P \mid \{X_r^P = x_r^P : r \neq s\}; \boldsymbol{\theta}),$$

where the product is taken over all sites s interior with respect to the pixel neighborhood system. The analytical justification for using that estimate is the "consistency of the pseudolikelihood in the "large graph" limits"" which is established in [GG86b]. The issue is that the parameter estimation is based on a single sample. The consistency refers to limiting case when the size of the graph grows (not the number of the samples taken).

The advantage of using MPLE instead of MLE is that the expectations which participate in the gradient expression of $\log \mathrm{P}\,\mathrm{L}\,(\tilde{\boldsymbol{x}}^P;\boldsymbol{\theta})$ are directly computable. In this case there is no need of applying the stochastic relaxation method.

The Degradation model

We observe the clean pixel process with no noise or any other kind of degradation. The degradation model is the projection $\Psi : (\boldsymbol{X}^P, \boldsymbol{X}^L) \rightarrow \boldsymbol{X}^P$, and the observed image is $\boldsymbol{Z} = \boldsymbol{X}^P$.

The Posterior distribution

The posterior distribution is the same as the conditional distribution of the texture label process given the pixel process:

$$\begin{aligned} \mathrm{P}\,(\boldsymbol{X}^P, \boldsymbol{X}^L \mid \boldsymbol{Z} = \boldsymbol{x}^P) &= \frac{\mathrm{P}\,(\boldsymbol{Z} = \boldsymbol{x}^P \mid \boldsymbol{X}^P = \boldsymbol{x}^P, \boldsymbol{X}^L)\mathrm{P}\,(\boldsymbol{X}^P = \boldsymbol{x}^P, \boldsymbol{X}^L)}{\mathrm{P}\,(\boldsymbol{Z} = \boldsymbol{x}^P)} \\ &= \frac{\mathrm{P}\,(\boldsymbol{X}^P = \boldsymbol{x}^P, \boldsymbol{X}^L)}{\mathrm{P}\,(\boldsymbol{Z} = \boldsymbol{x}^P)} \\ &= \mathrm{P}\,(\boldsymbol{X}^L \mid \boldsymbol{X}^P = \boldsymbol{x}^P). \end{aligned} \tag{7.8}$$

The MAP estimate

We have to find $\boldsymbol{x}^L$ which will maximize (7.8). That is equivalent of maximizing (7.5), given the observation, $\boldsymbol{X}^P = \boldsymbol{x}^P$. Thus the MAP maximizes the prior energy $U_1(\boldsymbol{x}^P, \boldsymbol{x}^L) + U_2(\boldsymbol{x}^L)$, for fixed $\boldsymbol{x}^P$.

The computation of a minimal energy configuration is done by simulated annealing with Gibbs sampler. For the experiments considered, to achieve good approximations, about a hundred and fifty iterations are enough.

Experiments are done with four different types of natural textures: wood, plastic, carpet and cloth. There are scenes involving two and four different textures as well. No pre- or post-processing is done, and though the gray level histograms of the different texture types are similar very good segmentation results are obtained (examined

by eye). See the Appendix, figure 20 (page 54). A drawback of this application is the fact that the model is dedicated to fixed types of textures. For each type of texture the neighborhood systems and the parameters of the energy (excluding the texture type parameters $\{\theta_i^{(l)}\}$) have to be estimated "by hand." The method is good for segmentation but not for recognition, since the "texture synthesis" is not satisfactory [GG86b].

7.3 Integration of image cues

In this section we present the MIT vision machine project [PT88]. "The MIT vision machine is mostly a specialized software running on the Connection machine" [PT88]. Its goal is to explore the issue of integration of early vision modules (such as edge detection, motion, texture, color), and to develop parallel algorithms for use on the CM (to organize a real-time vision system on a massively parallel computer).

MRF models are used in some low-level vision modules (image restoration, stereo), but most important in the integration stage. The visual modules are coupled to each other and to the image data in a parallel fashion.

The input images are processed in parallel through independent algorithms corresponding to different visual cues.

1. Edges are extracted using: (i) zero-crossings of the Laplacian of the image filtered through an appropriate Gaussian; (ii) Canny's edge detector. These edges at coarse resolution are input to the MRF-based integration stage.

2. Stereo computes disparity from left and right images. The match used is feature-based.

3. From pair of images in a time sequence optical flow is estimated.

4. Texture module attributes density and orientation of textons. The *textons* are elongated blobs by variation of which different texture regions are distinguished. Textons include rectangles, ellipses, and line segments, with specific properties such as color, orientation, length, and width. The end-of lines and crossings of lines are also textons.

5. Spectral albedo of the surface is estimated independently of the effective illumination.

The measurements provided by the early vision modules are typically noisy and possibly spare. The approximation and restoration of the data is performed using MRF models. In each process known constraints are exploited.

Simultaneously discontinuities are found in each cue (intensity, motion, texture, color, stereo), and the prior knowledge about these discontinuities is utilized. Then each cue is coupled to the edges of brightness. The complete algorithm consists of finding various types of discontinuities. The output of the system is a set of labeled discontinuities of the surfaces around the viewer. "This discontinuities taken together represent a *cartoon* of the original scene which can be used later for recognition and navigation" [PT88].

- *The integration stage*

Intuitively it is clear that the evidence provided by multiple cues (texture, color, stereo, etc.) should provide more reliable information about the scene than any single cue alone. But the problem of intergating the different visual cues is not trivial, actually it is not obvious at all how the integration should be done. How strong should be the coupling? The coupling depends critically on the reflectance and the imaging models. What features should be integrated and how? The authors argue that the coupling of the image data and surface properties is more robust and qualitative at locations of brightness discontinuities, at edges. Discontinuities are often most important locations in a scene (object boundaries or object parts for example). The changes in surface properties usually produce large brightness changes in the image. This suggests an integration scheme in which the brightness edges guide the computation of the discontinuities in the physical properties of the surface.

The approach is to detect the discontinuities in each cue, simultaneously with the approximation of the image data. In each case a prior information about the type of the discontinuities is exploited. For example the discontinuities themselves may be expected to be continuous, non intersecting. The different cues are coupled through their discontinuities. Surface depth, orientation, motion, texture and color are each coupled to the edges of brightness data and to each other. The information from several cues is used simultaneously to help refine the initial estimation of discontinuities.

The integration stage exploits the hierarchically structured MRF model introduced by Geman and Geman [GG84] (see Example 4.5). For example, the prior depth model for a smooth surface may be a MRF $\boldsymbol{X} = \boldsymbol{F}$ on a square $M \times M$ lattice S with homogeneous four-neighbor system, and potentials on the pair-cliques $\mathrm{C} = \{i, j\}$,

$$V_{\mathrm{C}}(\boldsymbol{f}) = (f_i - f_j)^2, \quad \boldsymbol{f} = (f_1, f_2, \ldots, f_N), \quad \{i, j\} \subset S, \; N = M^2,$$

where $f_i \in \Lambda, i \leq N$, and Λ is a finite set of possible depth values. More prior information (about depth edges) may be embodied by the addition of a depth line process $\boldsymbol{L}$ to the pixel depth process $\boldsymbol{F}$, $\boldsymbol{X}$=($\boldsymbol{F}$,$\boldsymbol{L}$). The potentials are modified

$$V_{\mathrm{C}}(\boldsymbol{\omega}) = (f_i - f_j)^2(1 - l_{ij}) + \beta V_{\mathrm{C}}^{l}(l_{ij}), \; \boldsymbol{\omega} = (\boldsymbol{f}, \boldsymbol{l}), \tag{7.9}$$

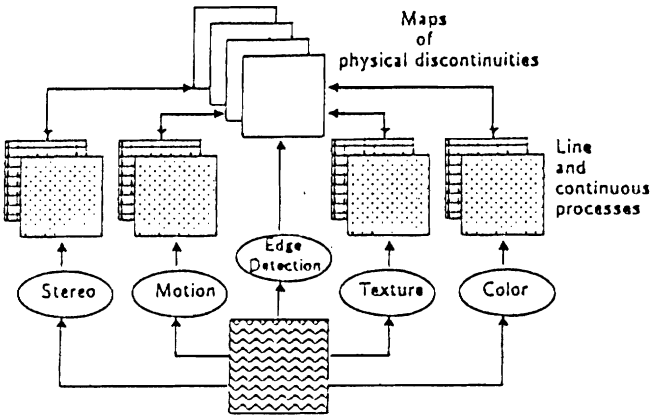


Figure 12: The MIT vision machine: block-diagram.

where $l_{ij} \in \{0, 1\}$ is the value of the component of the "depth line process" for the line site between pixel sites i and j. The line process is binary so only presence of horizontal and vertical depth edges is modeled. The direction of the edge is determined by the place of the line site. This process models the prior knowledge about the type of the depth edges expected. If there is no "depth edge element" between two "depth pixel" neighbors i and j, $(l_{ij} = 0)$, a smooth depth is most likely at these pixels (they should have similar depths). At the presence of a "depth edge," the potentials are modified in accordance with the depth discontinuity model V_C^l.

Using hierarchical MRF models even more complicated information than just depth edges (line process) can be incorporated in the energy function. The energy is extended to couple several of the early vision modules to brightness edges in the image. For the depth map example the coupling to the brightness edges may be done by replacing the term V_C^l in (7.9) by a function

$$g(e_{ij}, V_C^l(l_{ij})),$$

where e_{ij} represents a measure of a brightness edge between sites i and j. The function g modifies the probability of a depth edge in the presence of a brightness edge, for example

$$g(e_{ij}, V_C^l(l_{ij})) = -\log \mathrm{P}_{L_{ij}|E_{ij}}(L_{ij} = l_{ij} \mid E_{ij} = e_{ij}),$$

where $\mathbf{E}$ is the brightness edges process. This model for the depth discontinuities given the brightness discontinuities have to be chosen carefully. It should reflect the knowledge how the two types of discontinuity relate to each other. The heuristic is that the probability of a depth edge increases in the presence of a brightness edge.

We should note that the brightness edges activate different surface discontinuities (depth, color, texture, etc.) with different probabilities. These probabilities have to be estimated individually. The exact type of the function g depends on the particular

cue coupled with the edges of brightness. We do not know of any consistent method for choosing g. The surface discontinuities are coupled to the outputs of stereo, motion, color, texture using the form of potentials "similar" to (7.9) [PT88].

The result of the integration stage is a set of edges labeled in terms of physical discontinuities of the scene. They can be used for image recognition later. Initially only the discontinuities are used for recognition. But the information from the MRF's of the surfaces between the discontinuitied may be utilize in the recognition stage.

There are many open questions concerning the implementation of the "coupling" in the MIT vision machine project. The general idea is clear, but the details which are critical for the performance at the integration stage are not. We face all the problems concerning MRF modeling (neighborhood structure, type of energy, parameter estimation), but they expand in the integration application. This is because, in order to model the discontinuity dependences, we have to understand how they relate on practice. Which is very difficult, and that is not surprising. What we have to do is first to model every low-level vision output cue in terms of a MRF (this is difficult), second to model the relation between the particular type of discontinuity and the brightness discontinuity, next to model the feedback influence of the output of the integration to the separate low-level vision modules. But there is no easy way of performing the integration stage. On the other hand, though difficult, the MRF-based integration seems a consistent and an uniform way of performing the integration. There is a need fore more research in this direction, which hopefully will achieve a satisfactory results. The MIT vision machine system "will be improved at an incremental fashion" [PT88]. The authors expect to develop a deterministic algorithms which will eliminate the difficulties with MRF applications.

8 Conclusions

In this survey we introduce the Bayesian approach to solving some computer vision problems. This approach gives an uniform way of looking at the low-level vision tasks. It involves five major components: (i) a prior probability model; (ii) a sampling (degradation) model; (iii) a loss function; (iv) an optimal estimate with respect to (i)-(iii); (v) algorithms for computing (iv). To use Bayesian approach we have to study carefully the prior and the sensor (degradation) models. The optimal estimate should be relevant to a loss function suited to the specific problem. Mathematical simplicity should not be the primary concern in building the models. We consider the issue of an appropriate error criteria (loss function) very important. The loss function helps in understanding the nature of the problem. It is a mathematical interpretation of what a solution should be. We may use it to tune an algorithm to the particular problem under study. The choice of the loss function is task dependent and it is not

obvious. This is a hard problem. But, for any application, it is very useful the right error criteria to be established. In some cases of image reconstruction/segmentation, the loss is naturally associated with squared distance/misclassification rate. Special attention has to be paid to the possible loss functions for different low-level vision tasks.

Our presentation focuses on the MAP estimate. In this case the issue of the loss function is omitted. The MAP estimate is assumed to be a "good" estimate. It summarizes the prior and the sampling information. In some problems the MAP estimate comes naturally from a decision theory problem, but in the research we studied only Ripley [Rip88] explicitly stated this relation.

We introduce the MRF models and show how they can be used in modeling spatial images. The MRF-Gibbs equivalence provides a practical way for specifying MRF: by potentials. The computational algorithms exploit the type of the interactions in these models. The interactions are rich enough to represent a wide classes of images and at the same time they are local so the models ensure feasible computational algorithms. MRF models are naturally suited for distributed parallel computations. Under reasonable constraints, MRF give a uniform way of describing the prior and the posterior models. Another advantage, is that by Gibbs sampler we may simulate a sample and verify if the choice of a particular model is consistent with the prior information about the process under study. Subsequently, the model may be improved.

The hierarchical MRF models [GG84] can express complicated prior knowledge. The hierarchical structure of the prior model reflects the type and the degree of the prior information.

There is another useful feature of the hierarchy. First, note that for any MRF $\boldsymbol{X} = (X_{s_1}, X_{s_2}, \ldots, X_{s_N})$ the marginal $\boldsymbol{X}^{(r)} = \{X_s : s \neq r\}$ is a MRF for any fixed r. The neighborhood structure of the marginal is such that, two sites are neighbors with respect to it either if they are neighbors or each of them is a neighbor of r, with respect to the neighborhood structure of $\boldsymbol{X}$. Generalize this to the case when $\boldsymbol{X} = (\boldsymbol{F}, \boldsymbol{L})$ from section 4.3. The result is that the marginal of $\boldsymbol{F}$ is a MRF defined over a completely connected graph. This way, we we may incorporate long-range interactions in the model and still perform local computations.

A step is made toward using the hierarchical MRF models in the process of integrating the data from different low-level vision modules [PT88].

In case of MRF the MAP estimate reduces to minimizing the posterior energy function. The optimization algorithms exploit the annealing techniques and the results

obtained by Geman and Geman in [GG84].

The main problems in using MRF models relate to the following issues.

The neighborhood system. The neighborhood system with respect to which the MRF is specified has to be investigated carefully. Any random process with positive joint density may be specified in terms of a MRF. The useful models are these in which the neighborhoods are small enough to ensure fast computations, and at the same time can define a variety of images. There are no design methods for neighborhood specification. The homogeneous neighborhood systems may be a class to start with. But in many applications these simple models cannot capture complicated prior knowledge. In the latter cases, the choice of the neighborhood system is done by the researchers, using intuition, past experience, and trial and error methods.

The energy function. The type of the energy function is task dependent. A good understanding of the process and the local interactions is very important. There are no algorithms for constructing the energy function, just general principles which suggest its qualitative features [GM87]. Even these principles are valid only for a certain class of problems. However, it is true that in many practical problems, these general guidelines are enough for specifying a parametric family of suitable energy functions [GG84], [GG86b], [Mar85].

The parameter estimation. The models used in the Bayesian approach are directly related to the particular problem under study. But if it were necessary to start the application of the approach "from scratch" for any particular problem , it is worth asking ourselves "Does it pay?". Fortunately such models are described in terms of some free parameters. So in their general form they are applicable to variety of problems. We have to investigate the problem and choose a model based on some more general principles. The fine tuning of the model to the particular task is done by fixing the parameters. And the big question is: How? To illustrate the difficulties, lets look at some of the parameters from the experiments in Section 6.

The energy function usually has several free parameters. Representing more and more complicated prior knowledge usually implies increasing number of parameters in the prior model. Rigorous methods for estimating these parameters are critical for the performance of the algorithms. There is no even a general principle for estimating the different parameters for a single energy function used in a specific problem.

In Subsection 7.2 we showed how in case of texture modeling, some of parameters may be "learned" from training samples. Statistical methods (MLE, MPLE) may be used in the estimation process. These methods give an uniform way of estimating the texture parameters. But note that in the special

case of texture modeling the neighborhood structure is estimated by hand. The the parameter δ (7.4) on the other hand is loosely related to the range of the intensity values. It is tuned by trial and error.

The smoothing (regularization) parameter $1/T$ in the specification of the Gibbs distribution is estimated based on different heuristic. The choice of this parameter is very important, it is directly related to the "roughness" of the estimate. This parameter may be estimated utilizing preferences and understanding how a typical image should look [Rip88], or some statistical methods may be employed in its estimation [GM87].

Past experience may be very useful in the parameter estimation.

Recently increasing attention is given to the sensor models—a good understanding of the image formation (degradation) will help in the estimation process [Sze88].

A feature of the decision-theoretic approach which we did not discuss here is that this approach is natural for estimating the uncertainty in the obtained estimate, and subsequently improving it in a dynamic fashion [Sze88].

In applying Bayesian modeling important questions have to be answered: How do we establish prior and sampling models? What are the parameters of these models? What is the most appropriate loss function? Are the solutions robust? None of these questions has a simple solution. In summary, the research which relates to the application of the Bayesian modeling to low-level vision problems has to be done on three major levels. First, a variety of models suitable for different low-level vision problems has to be investigated. This include prior and degradation models, and error criteria. In case of MAP estimate the error criteria is not questioned but still the neighborhood structure and the energy function (up to some parameters) have to be specified. Second, rigorous parameter estimation methods have to be designed. This is a very difficult problem and it relates not only to computer vision applications but to any application of probabilistic models. At this point the integration of a fundamental research in statistics, statistical decision theory is necessary. Third, efficient computer algorithms have to be designed (some already exist) for estimation of the parameters of the models and for computing the estimators.

References

[Ber85] J.O. Berger. *Statistical decision theory and Bayesian analysis.* Springer-Verlag, 1985.

[Bes74] J. Besag. Spatial interactions and the statistical analysys of lattice systems (with discussion). *J. Roy. Statist. Soc. Ser. B*, 36:192–236, 1974.

[Fer67] T. Ferguson. *Mathematical Statistic: a decision theoretic approach.* Academic Press, Inc., 1967.

[GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[GG86a] D. Geman and S. Geman. Bayesian image analysis. In *NATO ASI Series, Vol. F20, Disordered Systems and Biological Organization*, Springer-Verlag, Berlin, 1986.

[GG86b] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *Proc. of the Int. Congress of Mathematicians*, pages 1496–1517, 1986.

[GM87] S. Geman and D. McLure. Statistical methods for tomographic image reconstruction. In *Proc. of the 46th Session of the ISI, Bulletin of the ISI*, 1987.

[Hor77] B. Horn. Understanding image intensities. *Artificial Intelligence*, 8(2):201–231, 1977.

[KGV83] S. Kirkpatrick, C.D.J. Gellat, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:495–502, 1983.

[Mar85] J.L. Marroquin. *Probabilistic solutions of inverse problems.* PhD thesis, Massachusetts Institute of Technology, 1985.

[PT88] T. Poggio and V. Torre. The MIT vision machine. In *Image understanding workshop*, pages 177–198, Morgan Kaufmann Publishers, Boston, 1988.

[PTK85] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(26):314–319, 1985.

[Rip88] B.D. Ripley. *Statistical inference for spatial processes.* Cambridge University Press, Cambridge, 1988.

[Roy68] H.L. Royden. *Real analysis.* Macmillan pub. co. Inc., New York, 1968.

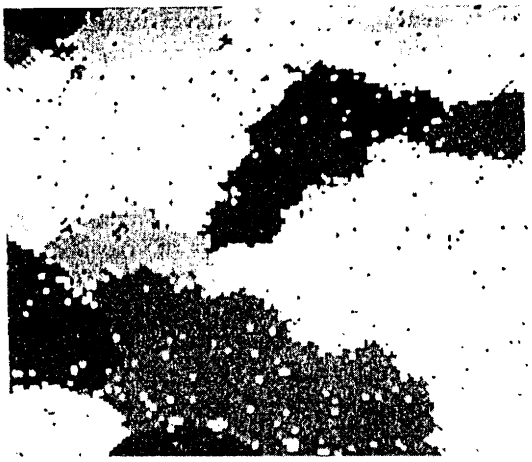
[Sze88] R. Szeliski. *Probabilistic solutions of inverse problems.* PhD thesis, Carnegeie Mellon University, 1988.

[TA77] A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems.* Winston, Washington DC, 1977.

[Won71] E. Wong. *Stochastic processes in information and dynamical systems.* McGraw-Hill Inc., 1971.

9 Appendix

Figure 13: Original image: Sample from MRF.



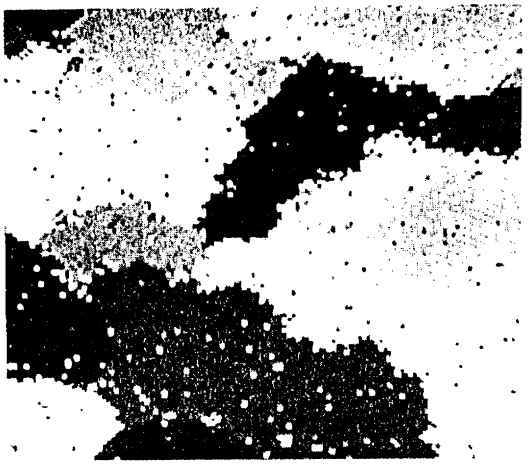
original image

degraded image: additive noise

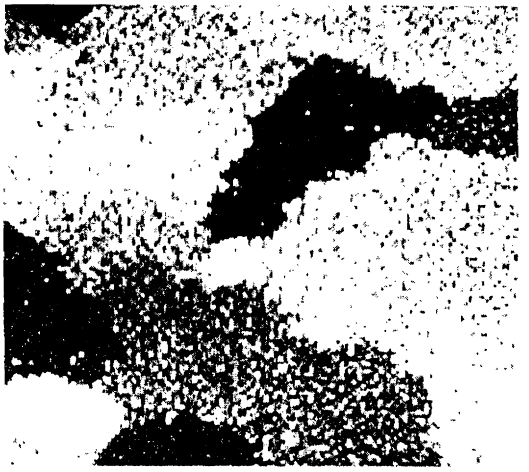
restoration: $K = 25$

restoration: $K = 300$

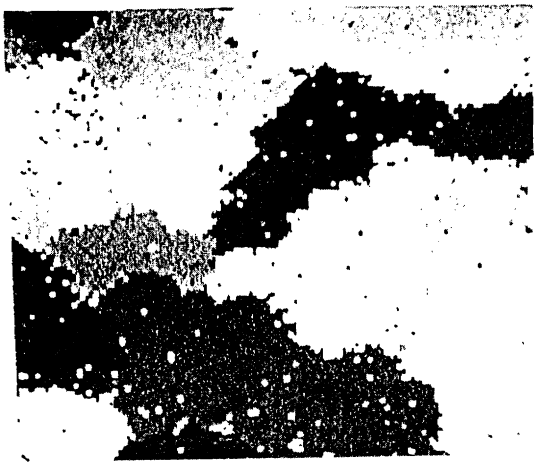
Figure 14: Original image: Sample from MRF.



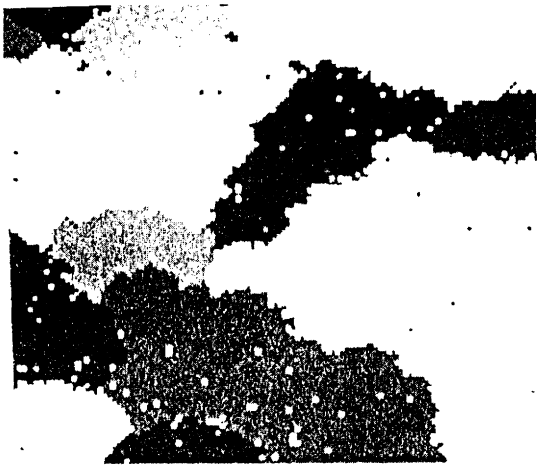
original image



degraded image: blur, nonlinear transformation, multiplicative noise



restoration: $K = 25$



restoration: $K = 300$

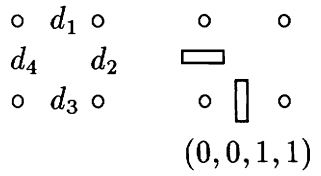


Figure 15: Original image: Hand-drawn. Line process' cliques.

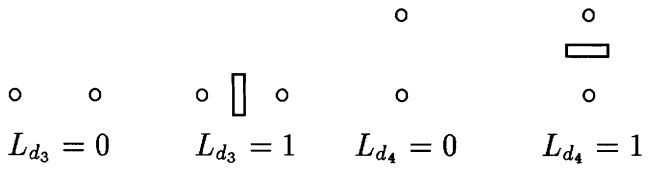


Figure 16: Original image: Hand-drawn. Type of the edge elements of the line process.

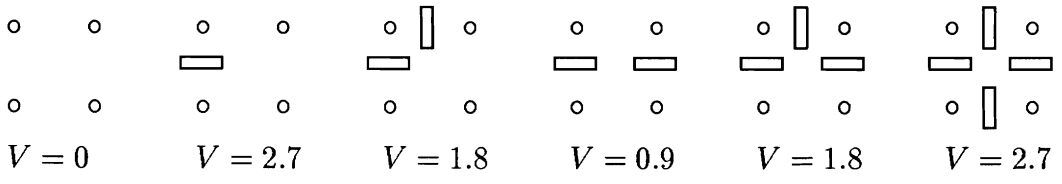
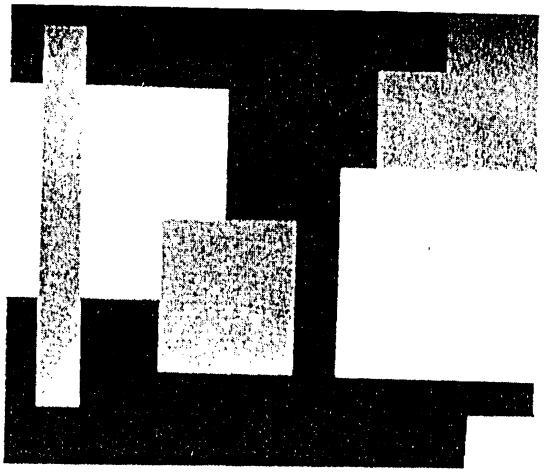


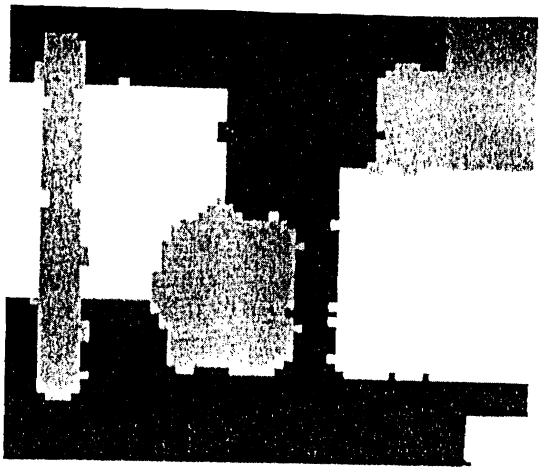
Figure 17: Original image: Hand-drawn. Potentials over the line process cliques.

Figure 18: Original image: Hand-drawn.

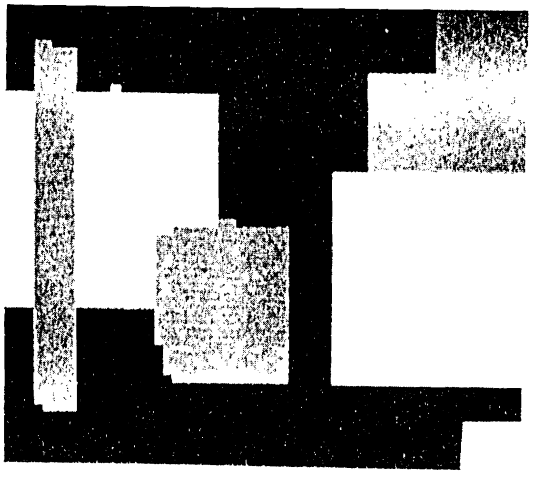


original image

degraded image: additive noise

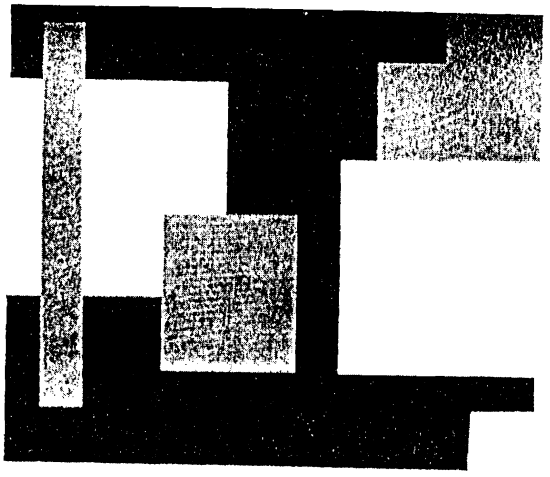


restoration: no line process, $K = 1000$

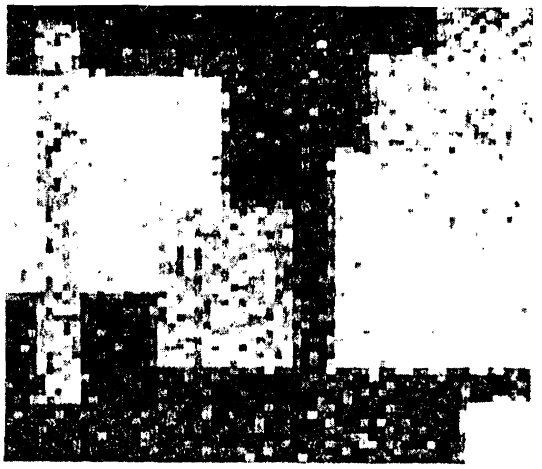


restoration: line process, $K = 1000$

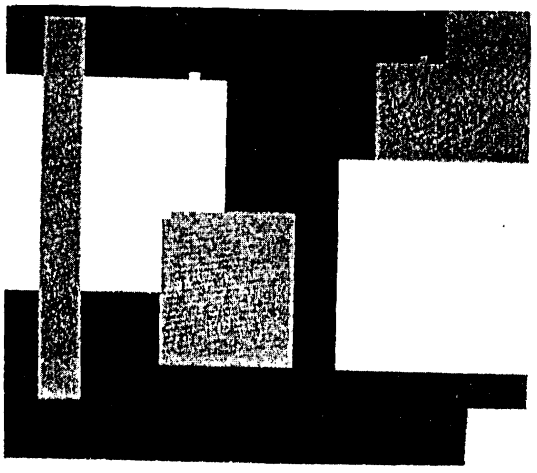
Figure 19: Original image: Hand-drawn.



original image

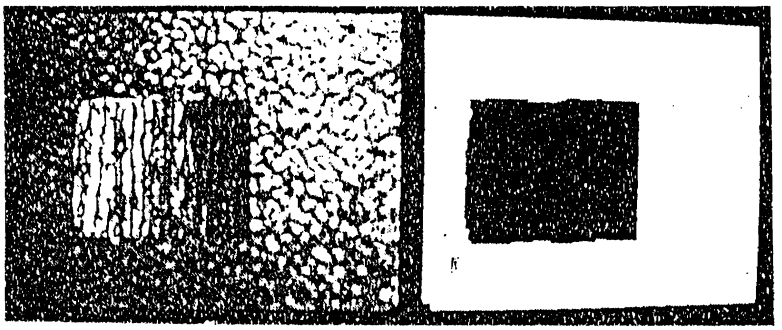


degraded image: blur, nonlinear transformation, multiplicative noise

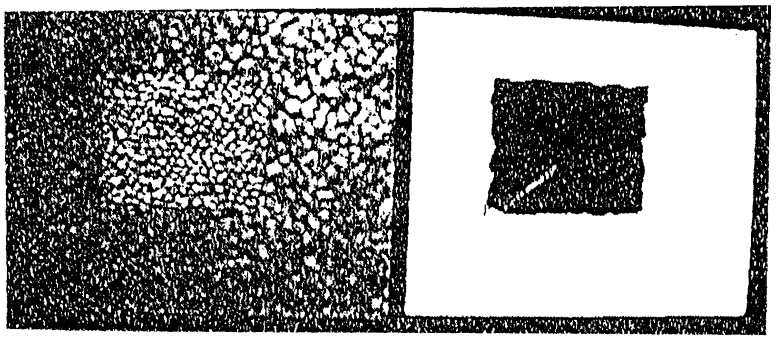


restoration: line process, $K = 1000$

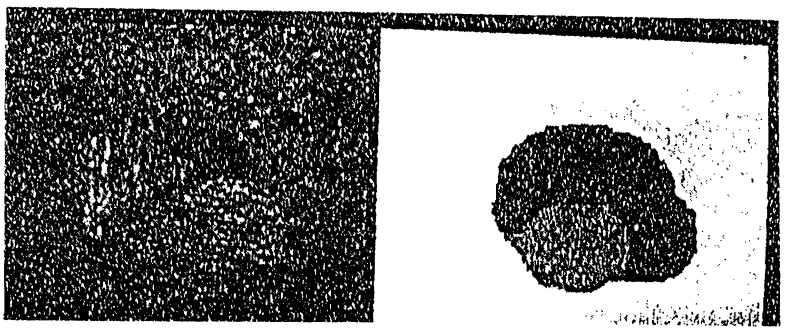
Figure 20: Texture segmentation.



wood on plastic background



carpet on plastic background



wood, plastic and cloth on plastic background