Markov Random Fields in Image Segmentation.

Zoltan Kato, Josiane Zerubia

Markov Random Fields in Image Segmentation

Markov Random Fields in Image Segmentation

Zoltan Kato

Image Processing and Computer Graphics Dept.
University of Szeged
Szeged 6720
Hungary
kato@inf.u-szeged.hu

Josiane Zerubia

INRIA Sophia Antipolis-Mediterranee
Sophia Antipolis
06902 Cedex
France
Josiane.Zerubia@inria.fr

now
the essence of knowledge
Boston – Delft

Foundations and Trends® in Signal Processing

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is Z. Kato and J. Zerubia, Markov Random Fields in Image Segmentation, Foundations and Trends® in Signal Processing, vol 5, nos 1–2, pp 1–155, 2011

ISBN: 978-1-60198-588-0

Foundations and Trends® in Signal Processing

Volume 5 Issues 1–2, 2011

Editorial Board

Editorial Scope

Foundations and Trends® in Signal Processing will publish survey and tutorial articles on the foundations, algorithms, methods, and applications of signal processing including the following topics:

- Adaptive signal processing
- Audio signal processing
- Biological and biomedical signal processing
- Complexity in signal processing
- Digital and multirate signal processing
- Distributed and network signal processing
- Image and video processing
- Linear and nonlinear filtering
- Multidimensional signal processing
- Multimodal signal processing
- Multiresolution signal processing
- Nonlinear signal processing
- Randomized algorithms in signal processing
- Sensor and multiple source signal processing, source separation
- Signal decompositions, subband and transform methods, sparse representations
- Signal processing for communications
- Signal processing for security and forensic analysis, biometric signal processing
- Signal quantization, sampling, analog-to-digital conversion, coding and compression
- Signal reconstruction, digital-to-analog conversion, enhancement, decoding and inverse problems
- Speech/audio/image/video compression
- Speech and spoken language processing
- Statistical/machine learning
- Statistical signal processing
  - classification and detection
  - estimation and regression
  - tree-structured methods

Information for Librarians
Foundations and Trends® in Signal Processing, 2011, Volume 5, 4 issues. ISSN paper version 1932-8346. ISSN online version 1932-8354. Also available as a combined paper and online subscription.

Foundations and Trends® in
Signal Processing
Vol. 5, Nos. 1–2 (2011) 1–155

DOI: 10.1561/2000000035

now
the essence of knowledge

Markov Random Fields in Image Segmentation

Zoltan Kato[1] and Josiane Zerubia[2]

[1] *Image Processing and Computer Graphics Dept., University of Szeged, Arpad ter 2, Szeged, 6720, Hungary, kato@inf.u-szeged.hu*
[2] *INRIA Sophia Antipolis-Mediterranee, 2004 Route des Lucioles, Sophia Antipolis, 06902 Cedex, France, Josiane.Zerubia@inria.fr*

Abstract

This monograph gives an introduction to the fundamentals of Markovian modeling in image segmentation as well as a brief overview of recent advances in the field. Segmentation is considered in a common framework, called image labeling, where the problem is reduced to assigning labels to pixels. In a probabilistic approach, label dependencies are modeled by Markov random fields (MRF) and an optimal labeling is determined by Bayesian estimation, in particular maximum a posteriori (MAP) estimation. The main advantage of MRF models is that prior information can be imposed locally through clique potentials. The primary goal is to demonstrate the basic steps to construct an easily applicable MRF segmentation model and further develop its multiscale and hierarchical implementations as well as their combination in a multilayer model. MRF models usually yield a non-convex energy function. The minimization of this function is crucial in order to find the most likely segmentation according to the MRF model. Besides classical optimization algorithms, like simulated annealing or

deterministic relaxation, we also present recently introduced graph cut-based algorithms. We briefly discuss the possible parallelization techniques of simulated annealing, which allows efficient implementation on, e.g., GPU hardware without compromising convergence properties of the algorithms. While the main focus of this monograph is on generic model construction and related energy minimization methods, many sample applications are also presented to demonstrate the applicability of these models in real life problems such as remote sensing, biomedical imaging, change detection, and color- and motion-based segmentation. In real-life applications, parameter estimation is an important issue when implementing completely data-driven algorithms. Therefore some basic procedures, such as expectation-maximization, are also presented in the context of color image segmentation.

Note: A sample implementation of the most important segmentation algorithms is available in grey scale at http://dx.doi.org/10.1561/2000000035_demogray and in color at http://dx.doi.org/10.1561/2000000035_democolor.

Contents

Dedication

"To the memory of my mother" Zoltan Kato

"To the memory of my beloved sister Elise who passed away in August 2012" Josiane Zerubia

1

Introduction

An image processing system involves a sensing device (usually a camera) and computer algorithms to interpret the picture. The term *image* (more precisely, *monochrome image*) refers to a two-dimensional light intensity function whose value at any point is proportional to the brightness (*gray-level*) of the image at that point [70]. A *digital image* is a discretized image both in spatial coordinates and in brightness. It is usually represented as a two-dimensional matrix, the elements of such a digital array are called pixels. The digitized image is the starting point of any kind of computer analysis. In some applications, the sensing device may be more specific responding to other forms of light: infrared imaging, photon emission tomography, radar imaging [182], ultrasonic imaging, etc.

Many image processing tasks deal directly with raw pixel data involving image compression [2], restoration [35, 64, 91, 219, 220, 223], edge detection [65, 200, 219, 220, 223], segmentation [51, 52, 60, 61, 74, 83, 98, 115, 195, 196, 221], texture analysis [43, 66, 122], motion detection [90, 213], optical flow and motion analysis [87, 90, 167], etc. Most of these problems can be formulated in a general framework, called *image labeling*, where we associate a label to each pixel from a finite set. The meaning of this label depends on the problem that we

are trying to solve. For image restoration, it means a gray-level; for edge detection, it means the presence or the direction of an edge; for image segmentation, it means a region; etc. The problem here is how to choose a label for a pixel, which is *optimal* in a certain sense. Herein, we deal with a statistical approach of *labeling*. In real scenes, neighboring pixels usually have similar features (intensity, color, texture, etc). In a probabilistic framework, such regularities are well expressed mathematically by Markov random fields. In this survey, we will focus on the fundamental problem of image segmentation using Markovian models.

1.1 Image Segmentation

The primary goal of any segmentation algorithm is to divide the domain R of the input image into the disjoint parts R_i such that they belong to distinct objects in the scene. The solution of this problem sometimes requires high level knowledge about the shape and appearance of the objects under investigation [46, 123, 183, 202]. In many applications, however, such information is not available or impractical to use. Hence low-level features of the surface patches are used for the segmentation process [9, 141, 224]. Herein, we are interested in the latter approach. In either case, we have to summarize all relevant information in a model which is then adjusted to fit the image data.

One broadly used class of models is the so called *cartoon model*, which has been extensively studied from both probabilistic [64] and variational [19, 163, 169] viewpoints. The model assumes that the real world scene consists of a set of regions whose observed low-level features change slowly, but across the boundary between them, these features change abruptly. What we want to infer is a *cartoon* ω consisting of a simplified, abstract version of the input image $\mathcal{I}$: regions R_i have a constant value (called a *label* in our context) and the discontinuities between them form a curve Γ — the contour. The pair (ω,Γ) specifies a *segmentation*. Region based methods are mainly focused on ω while edge based methods try to determine Γ directly.

Taking the probabilistic approach, one usually wants to come up with a *probability measure* on the set Ω of all possible segmentations of $\mathcal{I}$ and then select the one with the highest probability. Note that

Ω is finite, although huge. A widely accepted standard, also motivated by the human visual system [121, 162], is to construct this probability measure in a Bayesian framework [37, 161, 214]: We shall assume that we have a set of observed (Y) and hidden (X) random variables. In our context, any observed value $y \in Y$ represents the low-level features used for partitioning the image, and the hidden entity $x \in X$ represents the segmentation itself. First, we have to quantify how well any occurrence of x fits y. This is expressed by the probability distribution $P(y|x)$ — the *imaging model*. Second, we define a set of properties that any segmentation x must possess regardless the image data. These are described by $P(x)$, the *prior*, which tells us how well any occurrence x satisfies these properties. Factoring these distributions and applying the Bayes theorem gives us the *posterior* distribution $P(x|y) \propto P(y|x)P(x)$. Note that the constant factor $1/P(y)$ has been dropped as we are only interested in $\widehat{x}$ which *maximizes* the posterior, that is, the maximum a posteriori (MAP) estimate of the hidden field X.

The models of the above distributions also depend on certain parameters that we denote by Θ. Supervised segmentation assumes that these parameters are either known or a set of joint realizations of the hidden field X and observations Y (called a *training set*) is available [64, 205]. This is known in statistics as the *complete data* problem which is generally easier to solve than the *incomplete case* [37]. Although the prior knowledge of the parameters is a strong assumption, supervised methods are still useful alternatives when working in a controlled environment. Many industrial applications, like quality inspection of agricultural products [166], fall into this category. In the unsupervised case, however, we know neither Θ nor X. This is called the *incomplete data* problem where both Θ and X have to be inferred from the only observable entity Y. Hence our MAP estimation problem becomes $(\widehat{x}, \widehat{\Theta}) = \arg\max_{x,\Theta} P(x, \Theta|y)$. *Expectation Maximization* (EM) [48] and its variants (Stochastic EM [33, 149], Gibbsian EM [36]), as well as *Iterated Conditional Expectation* (ICE) [30, 108] are widely used to solve such problems. It is important to note, however, that these methods calculate a local maximum [37].

Due to the difficulty of estimating the number of pixel classes (or clusters), unsupervised algorithms often suppose that this parameter

is *known a priori* [68, 77, 141, 145, 149]. When the number of pixel classes is also being estimated, the unsupervised segmentation problem may be treated as a *model selection* problem over a combined model space [102, 202, 203].

1.2 Markov Random Fields

In the early 20th century, mostly inspired by the Ising model [170], a new type of stochastic process appeared in the theory of probability, called *Markov random field* (MRF). MRFs rapidly became a broadly used tool in a variety of problems, not only in statistical mechanics. The use of MRFs in image processing became popular with the seminal paper of S. Geman and D. Geman [64] in 1984, but its first use in the domain dates to the early 70s [16, 215]. Here, we give a brief introduction to the theory of MRFs [39, 54, 57, 79, 125, 144, 160, 184, 214].

1.2.1 The Ising Model

Following Ising [10, 69, 170], we consider a sequence, $0,1,2,\ldots,n$ on the line. At each point, there is a small spin which is either *up* or *down* at any given moment (see Figure 1.1). Now, we define a probability measure on the set Ω of all possible configurations $\omega = (\omega_0,\omega_1,\ldots,\omega_n)$. In this context, each spin is a function

$$\delta_i(\omega) = \begin{cases} 1 & \text{if } \omega_i \text{ is } \textit{up} \\ -1 & \text{if } \omega_i \text{ is } \textit{down} \end{cases} \tag{1.1}$$

An *energy* $U(\omega)$ is assigned to each configuration:

$$U(\omega) = -J\sum_{i,j}\delta_i(\omega)\delta_j(\omega) - mH\sum_i \delta_i(\omega). \tag{1.2}$$

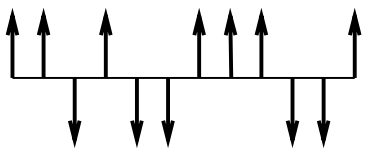


Fig. 1.1 One dimensional Ising model.

In the first sum, Ising made a simplifying assumption that only interactions of points with one unit apart need to be taken into account. This term represents the energy caused by the spin-interactions. The constant J is a property of the material. If $J > 0$, the interactions tend to keep neighboring spins in the same directions (*attractive case*). If $J < 0$, neighboring spins with opposite orientation are favored (*repulsive case*). The second term represents the influence of an external magnetic field of intensity H and $m > 0$ is a property of the material. The probability on Ω is then given by

$$P(\omega) = \frac{\exp\left(-\frac{1}{kT}U(\omega)\right)}{Z}, \tag{1.3}$$

where T is the temperature and k is a universal constant. The normalizing constant (also called *partition function*) Z is defined by

$$Z = \sum_{\omega \in \Omega} \exp\left(-\frac{1}{kT}U(\omega)\right). \tag{1.4}$$

The probability defined in Equation (1.3) is called a *Gibbs distribution*. One could extend the model to two dimensions in a natural way. The spins are arranged on a lattice, they are represented by two coordinates and a point have 4 neighbors unless it is on the boundary. In the two-dimensional case, the limiting measure P is unstable, there is a *phase transition*. As it is pointed out in [125], considering the *attractive* case and an external field h, the measure P_h converges to P^- if h goes to zero through negative values but it converges to $P^+ \neq P^-$ if h goes to zero through positive values. It has been shown, that there exists a *critical temperature* T_C and below this temperature phase transition always occurs. The temperature depends on the vertical (J_1) and horizontal (J_2) interaction parameters.

As a special example, we mention the *Cayley tree model* [125], originally proposed by Bethe [10] as an approximation to the Ising model. In this case, the points sit on a tree (see Figure 1.2). The root is called the 0th level. From the root, we have q branches ($q = 2$ in Figure 1.2). The $q = 1$ case simply gives a one-dimensional Markov chain. A configuration on a tree of n levels is an assignment of a label *up* or *down* to each point. We can define a similar energy function as for the Ising model.

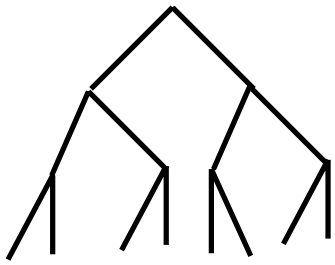


Fig. 1.2 Cayley tree model.

1.2.2 The Potts Model

Another important extension of the Ising model to more than two states per points is the Potts model [10, 195, 216]. The problem is to regard the Ising model as a system of interacting spins that can be either parallel or antiparallel. More generally, we consider a system of spins, each spin pointing one of the q equally spaced directions. These vectors are the linear combinations of q unit vectors pointing in the q symmetric directions of a hypertetrahedron in $q-1$ dimensions. For $q = 2, 3, 4$, examples are shown in Figure 1.3. The energy function of the Potts model can be written as

$$U(\omega) = \sum_{i,j} J(\Theta_{ij}), \tag{1.5}$$

where $J(\Theta)$ is 2π periodic and Θ_{ij} is the angle between two neighboring spins in i and j. The $q = 2$ case is equivalent to the Ising model.

1.2.3 Gibbs Distribution and MRFs

The most natural way to define MRFs [2, 64, 184] related to image models is to define them on a lattice. However, here we will define

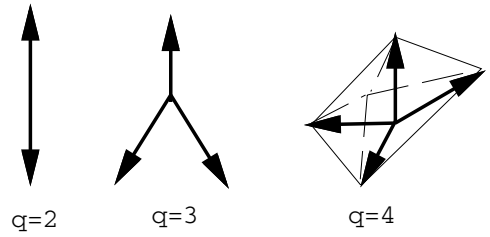


Fig. 1.3 The Potts model.

MRFs more generally on graphs. It will be useful in Section 2 for the study of hierarchical models. Let $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ be a graph where $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ is a set of vertices (or sites) and $\mathcal{E}$ is the set of edges.

Definition 1.1 (Neighbors). Two points s_i and s_j are neighbors if there is an edge $e_{ij} \in \mathcal{E}$ connecting them. The set of points which are neighbors of a site s (that is, the neighborhood of s) is denoted by $\mathcal{G}_s$.

Definition 1.2 (Neighborhood system). $\mathcal{G} = \{\mathcal{G}_s \mid s \in \mathcal{S}\}$ is a neighborhood system for $\mathcal{G}$ if

(1) $s \notin \mathcal{G}_s$
(2) $s \in \mathcal{G}_r \Leftrightarrow r \in \mathcal{G}_s$

To each site of the graph, we assign a label λ from a finite set of labels Λ. Such an assignment is called a configuration ω having some probability $P(\omega)$. The restriction to a subset $\mathcal{T} \subset \mathcal{S}$ is denoted by $\omega_{\mathcal{T}}$ and $\omega_s \in \Lambda$ denotes the label given to the site s. In the following, we are interested in the probabilities assigned to the set Ω of all possible configurations. First, let us define the *local characteristics* as the conditional probabilities $P(\omega_s \mid \omega_r, r \neq s)$.

Definition 1.3(Markov random field). $\mathcal{X}$ is a Markov random field (MRF) with respect to $\mathcal{G}$ if

(1) for all $\omega \in \Omega$: $P(\mathcal{X} = \omega) > 0$,
(2) for every $s \in \mathcal{S}$ and $\omega \in \Omega$:
$P(X_s = \omega_s \mid X_r = \omega_r, r \neq s) = P(X_s = \omega_s \mid X_r = \omega_r, r \in \mathcal{G}_s)$.

To continue our discussion about probabilities on Ω, the notion of *cliques* will be very useful.

Definition 1.4 (Clique). A subset $C \subseteq \mathcal{S}$ is a clique if every pair of distinct sites in C are neighbors. $\mathcal{C}$ denotes the set of cliques and $\deg(\mathcal{C}) = \max_{C \in \mathcal{C}} \mid C \mid$.

Using the above definition, we can define a *Gibbs measure* on Ω. Let V be a *potential* which assign a number $V_{\mathcal{T}}(\omega)$ to each subconfiguration $\omega_{\mathcal{T}}$. V defines an *energy* $U(\omega)$ on Ω by

$$U(\omega) = -\sum_{\mathcal{T}} V_{\mathcal{T}}(\omega). \tag{1.6}$$

Definition 1.5 (Gibbs distribution). A Gibbs distribution is a probability measure π on Ω with the following representation:

$$\pi(\omega) = \frac{1}{Z} \exp\left(-U(\omega)\right), \tag{1.7}$$

where Z is the normalizing constant (also called *partition function*):

$$Z = \sum_{\omega} \exp\left(-U(\omega)\right),$$

If $V_{\mathcal{T}}(\omega) = 0$ whenever $\mathcal{T}$ is not a clique then V is called a *nearest neighbor Gibbs potential*. In the following, we will focus on such potentials. The next famous theorem establish the equivalence between Gibbs measures and MRFs [16, 160].

Theorem 1.6 (Hammersley–Clifford). $\mathcal{X}$ is a MRF with respect to the neighborhood system $\mathcal{G}$ if and only if $\pi(\omega) = P(\mathcal{X} = \omega)$ is a Gibbs distribution with a nearest neighbor Gibbs potential V, that is

$$\pi(\omega) = \frac{1}{Z} \exp\left(-\sum_{C \in \mathcal{C}} V_C(\omega)\right) \tag{1.8}$$

The main benefit of this equivalence is that it provides us a simple way to specify MRFs, namely specifying potentials instead of local characteristics (see Definition 1.3), which is usually very difficult.

1.2.4 Spatial Lattice Schemes

In this section, we deal with a particular subclass of MRFs which are the most commonly used schemes in image processing. In this case,

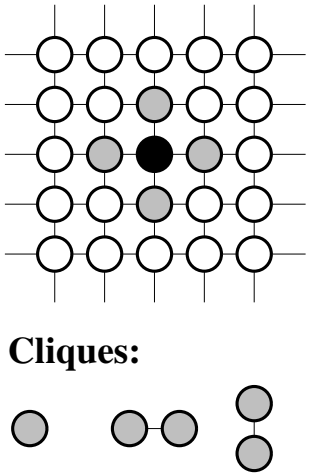


Fig. 1.4 First order neighborhood system.

we consider $\mathcal{S}$ as a lattice $\mathcal{L}$ so that $\forall s \in \mathcal{S} : s = (i,j)$ and define the so-called *nth order homogeneous neighborhood systems* as

$$\mathcal{G}^n = \{\mathcal{G}^n_{(i,j)} : (i,j) \in \mathcal{L}\}, \tag{1.9}$$

$$\mathcal{G}^n_{(i,j)} = \{(k,l) \in \mathcal{L} : (k-i)^2 + (l-j)^2 \leq n\}. \tag{1.10}$$

Obviously, sites near the boundary have fewer neighbors than interior ones (free boundary condition). Furthermore, $\mathcal{G}^0 \equiv \mathcal{S}$ and for all $n \geq 0 : \mathcal{G}^n \subset \mathcal{G}^{n+1}$. Figure 1.4 shows a first-order neighborhood corresponding to $n = 1$. The cliques are $\{(i,j)\}, \{(i,j),(i,j+1)\}, \{(i,j),(i+1,j)\}$. In practice, more than two order systems (cf. Figure 1.5) are rarely used since the energy function would be too complicated requiring a lot of computation. Although not as widespread as orthogonal lattice schemes, hexagonal lattices [45, 193] as well as MRFs on graphs [204] have also been studied in the literature.

1.3 Related Approaches

1.3.1 Weak Membrane Model

The *weak membrane model* was introduced in image reconstruction by A. Blake and A. Zisserman [19]. The problem is to reconstruct surfaces

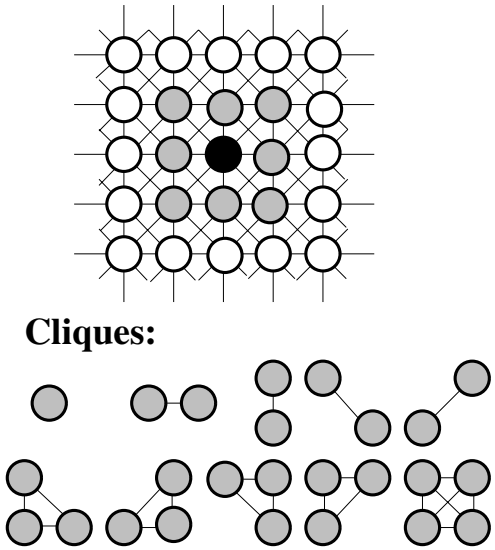


Fig. 1.5 Second order neighborhood system.

which are *continuous almost everywhere* or, in other words, continuous in patches. To reach a satisfactory formalization of this principle, they have used a membrane model: Imagine an elastic membrane which we are trying to fit to a surface. The edges will appear as tears in the membrane. Depending on how elastic is the membrane, there may be more or fewer edges. The membrane is described by an energy function (the elastic energy of the membrane) which has to be minimized in order to find an equilibrium state. The energy has three components:

D: A measure of faithfulness to the data:

$$D = \int (u - d)^2 dA, \tag{1.11}$$

where $u(x,y)$ represents the membrane and $d(x,y)$ represents the data.

S: A measure of how the function $u(x,y)$ is deformed:

$$S = \lambda^2 \int (\nabla u)^2 dA. \tag{1.12}$$

λ^2 is a measure of elasticity of the membrane.

P: The sum of penalties α levied for each break in the membrane:

$$P = \alpha Z, \tag{1.13}$$

where Z is a measure of the set of contours along which $u(x, y)$ is discontinuous (see [19] for more details).

The elastic energy of the membrane is then given by

$$E = D + S + P = \int (u - d)^2 dA + \lambda^2 \int (\nabla u)^2 dA + \alpha Z. \tag{1.14}$$

There is a strong relation between the *weak membrane model* and MRF models: An elastic system can also be considered from a probabilistic view-point. The link between the elastic energy E and probability P is

$$P \propto \exp\left(\frac{-E}{T}\right), \tag{1.15}$$

that is the Gibbs distribution. However, the *weak membrane model* operates with mechanical analogies, representing *a priori* knowledge from a mechanical point of view while MRF modelization is purely probabilistic.[1]

1.3.2 Snakes, Variational and Level Set Methods

Active Contours (snakes) are closed curves evolving toward the boundary of the object of interest. The curve evolution is governed by a boundary functional [101] which takes its minimum on the object contour. The main drawback of the parametric snake model is that it cannot handle topological changes easily. Nevertheless, they became quite popular because they make it relatively easy to enforce contour-smoothness; and starting from an appropriate initialization a local minimum of the associated energy function will give good results. One extension of the original model is gradient vector flow [217] snakes

[1] We notice that the weak membrane model has also been used in a Markovian context but originally, as proposed by Blake and Zisserman [19], it was a non-Markovian model.

which make the snake less sensitive to initialization and allow the contour to segment concave objects. Another extension is the so-called balloon force [44] which basically introduces an area minimizing term [31] into the snake energy.

Geodesic active contours [31] are curves of minimum length in the metric defined by a function u. The criterion to minimize is usually of the form $\int_\Gamma u(s)ds$. Most of the time, u is simply a function of the image gradient like $u = 1/(1 + |\nabla \mathcal{I}|)$. The contour evolution equation is as follows [31]:

$$\frac{\partial \Gamma}{\partial t} = (\kappa u \nabla u \cdot N)N, \tag{1.16}$$

where κ is the curvature and N is the inward normal of Γ.

Region based active contours are another class of boundary based methods where region descriptors (usually some kind of statistical features) are introduced into the energy in order to better characterize an object [169, 188, 224].

Variational approaches consider the segmentation as an optimal approximation of the original image $\mathcal{I}$ by a piecewise smooth function f having discontinuities across Γ. The classical Mumford–Shah energy functional [163] is then defined as

$$E(f,\Gamma) = \mu^2 \int\int_R (f - \mathcal{I})^2 dxdy + \int\int_{R-\Gamma} ||\nabla f||^2 dxdy + \nu|\Gamma|. \tag{1.17}$$

Clearly, the minimum is achieved when f approximates $\mathcal{I}$ (first term), f is smooth over each R_i (second term), and the boundaries Γ are as short as possible. Note that dropping any of the above three terms would result in $\inf E = 0$ with some trivial and (from a practical point of view) useless settings for f and Γ. The minimization of the above functional is far from trivial. Note also that in our context, $f = \omega$ is constant over each region R_i, hence the problem can be further simplified to a piecewise constant functional. A closely related model, proposed by Blake and Zisserman, is the so-called *weak membrane* model (see Section 1.3.1) which can be minimized via graduated non-convexity (GNC) [19].

More recently, the level set formulation [192] of the piecewise constant Mumford–Shah energy functional proposed by Chan and Vese [38] have become a popular framework for image segmentation. The contour Γ is represented as the zero level set of an embedding function (the level set function) $\phi : R \to \Re$ on the image domain R: $\phi(\Gamma) = 0$. The main advantage of this formulation is that it handles topological changes of the evolving contour. This makes the level set formalism well suited to the segmentation of multiple objects. The region based level set scheme for foreground–background segmentation consists in minimizing the following functional:

$$E_{CV}(c_1, c_2, \phi) = \int_R (\mathcal{I} - c_1)^2 H(\phi) dx + \int_R (\mathcal{I} - c_2)^2 (1 - H(\phi)) dx + \nu \int_R |\nabla H(\phi)| dx, \tag{1.18}$$

c_1 and c_2 are the means of the regions, where $\phi > 0$ (outside or background) and $\phi < 0$ (inside or foreground), and $H(.)$ is the Heaviside function. The last term measures the length of the zero crossing of ϕ (i.e., the contour). The Euler–Lagrange equation for this model is implemented by the following gradient descent:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\nu \text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - (\mathcal{I} - c_1)^2 + (\mathcal{I} - c_2)^2 \right]. \tag{1.19}$$

Unfortunately, even with the narrow band implementation [1], the level set approach has a rather high computational complexity. The fast marching method [192] has a lower complexity but it requires that the speed function doesn't change sign during evolution.

1.3.3 Conditional Random Fields

Conditional Random Fields (CRF) directly model the posterior distribution of $P(X|Y)$ as a Gibbs field [86, 135, 136, 206]. Unlike the generative image models commonly used in MRFs, CRFs can depend on arbitrary non-independent characteristics of the observation $Y = y$. Originally, CRFs were proposed for segmenting 1D text

sequences [140, 212], but it is straightforward to extend these consepts to 2D images.

Basically, a CRF is a random field globally conditioned on the observation Y. Following [140], we can formally define CRFs on graph:

Definition 1.7 (Conditional Random Field). Let $G = (V, E)$ be a graph such that the label field X is indexed by the vertices: $X = \{X_v\}_{v \in V}$ and neighboring elements $v \sim w$ of the field are connected by edges in G, i.e., $(v, w) \in E$. Then (Y, X) is a *conditional random field (CRF)* if the random variables X_v, when conditioned on Y, obey the Markov property with respect to the graph: $P(X_v|Y, X_w, w \neq v) = P(X_v|Y, X_w, w\tilde{v})$.

The simplest example of such a graph structure is a lattice where vertices correspond to pixels and neighboring lattice sites are connected by edges (see Section 1.2.4 for various neighborhood structures on lattices). Considering a first order neighborhood, the posterior distribution can be easily expressed using the Hammersley–Clifford theorem (see Theorem 1.6):

$$P(x|y) = \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, x|_e, y) + \sum_{v \in V,k} \mu_k g_k(v, x|_v, y))\right), \quad (1.20)$$

where x is a labeling of a given input image y and $x|_S$ is the set of components of x associated to the vertices in the subgraph S. Furthermore, the features f_k and g_k are assumed to be known and fixed, and the parameter values λ_k and μ_k are to be learned from training data [140]. As we can see from Equation (1.20), standard CRFs use two forms of feature functions, which can be interpreted in 2D as follows [86]:

- state feature function $g_k(s, x_s, y)$ of the label x_s at a site s and the observed image y,
- transition feature function $f_k(s, r, x_s, x_r, y)$ of the labels x_s and x_r at neighboring sites $s \sim r$ and the observed image y.

In image processing applications, state feature functions are usually defined as unary (also known as singleton) clique potentials based

on classifier responses (such as Ada-boost [194] or kernel SVMs [191]), while transition feature functions are defined as pairwise (also known as doubleton) potentials modeling the correlation between pairs of random variables. Recently, CRFs became popular in image segmentation [86, 206], especially CRFs coupled with graph cut energy minimization [128, 138, 208].

2

Markovian Segmentation Models

Early vision refers to a variety of digital image processing tasks dealing directly with massive amounts of pixel data. The goal of such a process is to transform the digitized image data into more meaningful tokens (edges, texture features, regions, etc).

The variety of early vision tasks has resulted in a variety of algorithms, sometimes dedicated to a single application and often tuned to a particular environment in which they are implemented [178, 179]. A general framework in image processing is image labeling where we want to associate to each pixel a label from a finite set. The meaning of this label depends on the problem that we are trying to solve. For image restoration, it means a gray-level; for edge detection, it means the presence or the direction of an edge; for image segmentation, it means a class (or region); etc. The problem here is how to choose a label for a pixel. There may be various responses. Our approach consists of building probabilistic image models and simply selecting the most likely labeling. To do this, we need to define some probability measure on the set of all possible labelings. In real scenes, neighboring pixels usually have similar intensities. In a probabilistic framework, such regularities are well expressed mathematically by Markov random fields (MRF) [178, 179]. Another reason for dealing with MRF models

is of course the *Hammersley–Clifford theorem* which allows us to define MRFs through clique-potentials. In the labeling problem, this leads us to the Bayesian formulation in which a prior distribution is needed on the labels. In general, we try to find the maximum a posteriori (MAP) estimate of the label field [182].

Unfortunately, finding such an estimate is a heavy computational problem. There are many heuristics to make the minimization easier such as multi-scale approaches. In the next sections, after introducing the classical monogrid MRF model, we present some multiscale MRF models proposed by a variety of authors. Finally, we discuss hierarchical MRF models. This model allows us to work with cliques with far apart sites for a reasonable price.

2.1 Bayesian Framework

MRF models in computer vision became popular with the seminal paper of S. Geman and D. Geman on image restoration [64]. Since then, the field has grown up rapidly addressing a variety of low-level[1] image tasks [2, 39, 144, 214]:

Compression: Find a new image as close as possible to the original one but described at a much smaller cost.

Restoration: Observing a degraded image, one wants to approximately recover the original one [197].

Edge Detection: Find smooth boundaries separating image regions.

Segmentation: Partition the image into homogeneous regions where homogeneity is measured in terms of gray-levels or texture characteristics.

Motion Detection: In a sequence of images, try to find a field of velocities linking one image to the next one.

We now turn to the mathematical formulation of a MRF image model. Let $\mathcal{R} = \{r_1, r_2, \ldots, r_M\}$ be a set of sites and $\mathcal{F} = \{F_r : r \in \mathcal{R}\}$ a set of image data (or observations) on these sites. The set of all possible observations $f = (f_{r_1}, f_{r_2}, \ldots, f_{r_M})$ is denoted by Φ. Furthermore, we

[1] *Low-level* is a traditional terminology for preliminary tasks to image understanding.

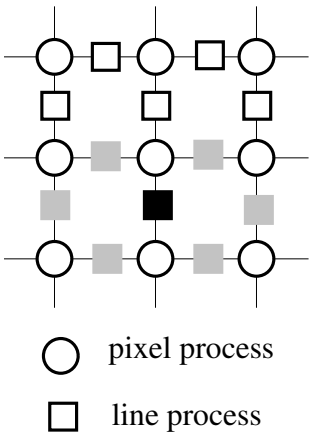


Fig. 2.1 Geman's image restoration model.

are given another set of sites $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$, each of these sites may take a label from $\Lambda = \{0, 1, \ldots, L-1\}$. The configuration space Ω is the set of all global discrete labeling $\omega = (\omega_{s_1}, \ldots, \omega_{s_N}), \omega_s \in \Lambda$. The two sets of sites $\mathcal{R}$ and $\mathcal{S}$ are not necessarily disjoint, they may have common parts (for example, Geman's image restoration model involving a line process [64], see Figure 2.1) or refer to a common set of sites. Our goal is to model the labels and observations with a joint random field $(\mathcal{X}, \mathcal{F}) \in \Omega \times \Phi$. The field $\mathcal{X} = \{X_s\}_{s \in \mathcal{S}}$ is called the *label field* and $\mathcal{F} = \{F_r\}_{r \in \mathcal{R}}$ is called the *observation field*.

2.1.1 Bayesian estimation

First, we construct a Bayesian estimator for the *label field*. Both the joint and conditional probabilities can be defined in terms of the *a priori* and *a posteriori distributions*:

$$P_{\mathcal{X},\mathcal{F}}(\omega, f) = P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) P_{\mathcal{X}}(\omega) \tag{2.1}$$

$$P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) = \frac{P_{\mathcal{X},\mathcal{F}}(\omega, f)}{P_{\mathcal{F}}(f)} = \frac{P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) P_{\mathcal{X}}(\omega)}{P_{\mathcal{F}}(f)}. \tag{2.2}$$

Since the realization of the observation field is known, $P(f)$ is constant and we can write:

$$P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) \propto P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) P_{\mathcal{X}}(\omega). \tag{2.3}$$

The estimator can be formulated as the following decision function δ [56]:

$$\delta : \Phi \longrightarrow \Omega \tag{2.4}$$

$$f \mapsto \delta(f) = \hat{\omega} \tag{2.5}$$

and the corresponding *Bayes risk* [56] is given by

$$r(P_X, \delta) = E\{R(\omega, \delta(f))\}, \tag{2.6}$$

where $R(\omega, \delta(f))$ is a cost function defined later. According to the *Bayesian decision rule* [56], our estimator must correspond to the minimum Bayes risk:

$$\hat{\omega} = \arg\min_{\omega' \in \Omega} \sum_{\omega \in \Omega} R(\omega, \omega') P_{\mathcal{X}|\mathcal{F}}(\omega \mid f). \tag{2.7}$$

We explain hereafter the three most commonly used Bayesian estimators [148].

2.1.1.1 Maximum A Posteriori (MAP)

The MAP estimator is the most frequently used estimator in image processing. Its cost function is defined by

$$R(\omega, \omega') = 1 - \delta(\omega', \omega), \tag{2.8}$$

where $\delta(\omega', \omega)$ is the Kronecker delta. Clearly, this function has the same cost for all configurations different from ω'. From Equations (2.7) and (2.8), the MAP estimator of the label field is given by

$$\hat{\omega}^{MAP} = \arg\max_{\omega \in \Omega} P_{\mathcal{X}|\mathcal{F}}(\omega | f). \tag{2.9}$$

This estimator provides for a given observation f, the modes of the posterior distribution, that is the most likely labelings given the observation f. Equation (2.9) is a combinatorial optimization problem which requires special algorithms such as Simulated Annealing (see Section 3).

2.1.1.2 Marginal A Posteriori Modes (MPM)

We define the cost function of the MPM estimator as

$$R(\omega,\omega') = \sum_{s\in\mathcal{S}}(1 - \delta(\omega'_s,\omega_s)). \tag{2.10}$$

Remark that the above function is related to the number of sites $s \in \mathcal{S}$ such that $\omega_s \neq \omega'_s$. The solution of Equation (2.7) is given by

$$\forall s \in \mathcal{S} : \hat{\omega}_s^{MPM} = \arg\max_{\omega_s\in\Lambda} P_{X_s|\mathcal{F}}(\omega_s \mid f), \tag{2.11}$$

which gives the configuration which maximizes at each site the a posteriori marginal $P_{X_s|\mathcal{F}}(. \mid f)$.

2.1.1.3 Mean Field (MF)

Here, we have the following cost function:

$$R(\omega,\omega') = \sum_{s\in\mathcal{S}}(\omega_s - \omega'_s)^2. \tag{2.12}$$

From Equations (2.7) and (2.12), we have

$$\forall s \in \mathcal{S} : \hat{\omega}_s^{MF} = \sum_{\omega\in\Omega} \omega_s P_{\mathcal{X}|\mathcal{F}}(\omega \mid f), \tag{2.13}$$

which is nothing else but the conditional expected value of $\mathcal{X}$ given $\mathcal{F} = f$ that is the *mean field* of $\mathcal{X}$.

2.1.2 Defining a Priori and a Posteriori Distributions

In the Bayesian framework, our knowledge about the "world" is represented by a priori probabilities. However, in practice, it is extremely difficult to define such probabilities globally, even if we focus on a specific area of image processing. But there are some well-defined properties if we are considering images *locally*: Usually, neighboring pixels have similar intensities, edges are smooth and often straight and textures have also well-defined local properties. It is then a better idea to represent our knowledge in terms of some local random variables. This kind of knowledge is best described by means of MRFs.

2.1.2.1 Prior Distribution

Let us suppose that $\mathcal{X}$ is a MRF with some neighborhood system $\mathcal{G}' = \{\mathcal{G}'_s : s \in \mathcal{S}\}$ and distribution

$$P(\mathcal{X} = \omega) = \frac{1}{Z} \exp\left(-U'(\omega)\right), \tag{2.14}$$

$$U'(\omega) = \sum_{C \in \mathcal{C}'} V'_C(\omega), \tag{2.15}$$

where $U'(\omega)$ is the energy function (see Section 1.2). The above equations give another good reason using MRF priors, namely their Gibbs representation through *clique-potentials*, which are more convenient than working directly with probabilities.

When human observers are interpreting images, they are not only taking into account direct observations like color or intensity, but also a priori knowledge about the world. Purely data driven methods cannot deal very well with high noise, cluttered background or occlusions. Hence the idea of incorporating some prior knowledge about the shape of the objects has been considered by many researchers. Early approaches for shape prior were quite generic, enforcing some kind of homogeneity and contour smoothness [19, 31, 44, 64, 101, 106]. For example, [64, 106] uses a Markovian smoothness prior (basically a Potts model [10]); [19, 64] uses a line process to control the formation of region boundaries; and active contour models [101] have been using elasticity, rigidity, contour length, balloon or area minimizing forces [31, 44] in order to favor smooth closed curves. In spite of their simplicity, these methods proved to be very efficient in dealing with noisy images.

More recently, there has been a great deal of work on statistical shape modeling [46, 76, 156]. These methods rely on a kind of template matching: The shape of the object under investigation is of known shape (template) and its allowed deformations are learned a priori [152, 155]. This knowledge is then summarized in a statistical model which is incorporated into a variational [41, 46, 186] or probabilistic [92, 156] model. These models often borrow ideas from mathematical pattern theory developed by Grenander [75]: The basic assumption is that the deformations are a result of some kind of transformations (usually affine) applied to the reference shape. The set of these shapes is called an

orbit. The modeling step involves the estimation and representation of the underlying transformations and the prior will penalize strong deviations from the orbit. Such models are useful when we have a clear idea on how the objects look like and the segmentation is driven by recognizing the object in the image data. A typical application is medical image processing where well-known objects (e.g., organs) has to be segmented. For example, in [41] a variational method with shape priors using an atlas has been proposed. The prior was restricted to a parametric deformation between the reference shape and the active contour.

For many applications, however, the assumption of a parametric deformation is too restrictive or impractical to use. An interesting approach is presented in [20] where basic geometrical constraints are modeled by long-range interactions in a Markovian framework. The method is applied to tree crown extraction from satellite images using a prior which favors circular objects.

2.1.2.2 Degraded Image Model and Posterior Distribution

The observations are related to the label process through a *degradation model* which establishes the relation between the *label field* $\mathcal{X}$ and the *observation process* $\mathcal{F}$. In image restoration [64] for example, what we observe is a blurred noisy image and we want to restore the original one. So, the label process represents gray-levels here. We are now considering a similar model but in a more general manner. Most of the problems result in the following function [171]:

$$\mathcal{F} = \Psi(H(\mathcal{X}), N), \tag{2.16}$$

or at the pixel level:

$$\forall r \in \mathcal{R} : F_r = \Psi(H_r(X_{\psi(r)}), N_r), \tag{2.17}$$

where $\Psi(a,b)$ is an invertible function in a. H_r is a local function defined on a small part $\psi(r)$ of $\mathcal{S}$ such that $\psi(r) \in \mathcal{S}, \mid \psi(r) \mid \ll \mid \mathcal{S} \mid$ and $\psi^{-1}(s) = \{r \in \mathcal{R} \mid s \in \psi(r)\}$. N is a random component (usually a Gaussian white noise but in tomography N_r are Poisson variables whose means are related to $\mathcal{X}$). In [64], for instance, H is a *blurring*

matrix and N is an additive Gaussian white noise. If we assume that the distribution of N is given by

$$P_N(.) = \prod_{r \in \mathcal{R}} P_{N_r}(.) \tag{2.18}$$

then we obtain

$$P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) = \prod_{r \in \mathcal{R}} P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r)). \tag{2.19}$$

The conditional distribution of the observation field $\mathcal{F}$ given $\mathcal{X}$ can be written as

$$P_{\mathcal{F}|\mathcal{X}}(f \mid \omega) = \exp\left(\sum_{r \in \mathcal{R}} -\ln(P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r)))\right), \tag{2.20}$$

assuming that $P_{N_r}(.) > 0$ at each site r in $\mathcal{R}$. Combining the above equation with Equations (2.3) and (2.14), the posterior distribution is of the following form:

$$P_{\mathcal{X}|\mathcal{F}}(\omega \mid f) \propto$$
$$\frac{1}{Z} \exp\left(\sum_{r \in \mathcal{R}} -\ln(P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r))) + \sum_{C \in \mathcal{C}'} V'_C(\omega)\right). \tag{2.21}$$

Notice that the posterior distribution is also a Gibbs distribution with the smallest neighborhood system $\mathcal{G}$ containing all the cliques in $\mathcal{C}'$ and the sets $\{\psi(r), r \in \mathcal{R}\}$:

$$\forall s \in \mathcal{S} : \mathcal{G}_s = \left(\bigcup_{r \in \psi^{-1}(s)} \psi(r) \setminus \{s\}\right) \cup \mathcal{G}'_s. \tag{2.22}$$

Let us denote the corresponding energy function by $U(\omega, f)$:

$$\begin{aligned} U(\omega, f) &= \sum_{r \in \mathcal{R}} -\ln(P_{N_r}(\Psi^{-1}(H_r(\omega_{\psi(r)}), f_r))) + \sum_{C \in \mathcal{C}'} V'_C(\omega) \\ &= \sum_{r \in \mathcal{R}} V_r(\omega_{\psi(r)}), f_r) + \sum_{C \in \mathcal{C}'} V'_C(\omega). \end{aligned} \tag{2.23}$$

In the following, we will be more specific about $V_r(\omega_{\psi(r)}), f_r)$ and suppose that it is of the form [64, 171]:

$$V_r(\omega_{\psi(r)}), f_r) = V_r(\omega_{\psi(r)}) + \sum_{s \in \psi(r)} V_{s,r}(\omega_s, f_r). \tag{2.24}$$

This restriction is less severe than it might be expected. As we will see, most of the nowadays used models have this kind of energy function. The above equation can be rewritten as

$$\begin{aligned} \sum_{r \in \mathcal{R}} V_r(\omega_{\psi(r)}), f_r) &= \sum_{r \in \mathcal{R}} V_r(\omega_{\psi(r)}) + \sum_{r \in \mathcal{R}} \sum_{s \in \psi(r)} V_{s,r}(\omega_s, f_r) \\ &= \sum_{r \in \mathcal{R}} V_r(\omega_{\psi(r)}) + \sum_{s \in \mathcal{S}} \underbrace{\sum_{r \in \psi^{-1}(r)} V_{s,r}(\omega_s, f_r)}_{V_s(\omega_s, f_{\psi^{-1}(s)})}. \end{aligned} \tag{2.25}$$

Finally, we have the following energy function associated with the posterior distribution of the label field $\mathcal{X}$:

$$U(\omega, f) = \sum_{s \in \mathcal{S}} V_s(\omega_s, f_{\psi^{-1}(s)}) + \sum_{C \in \mathcal{C}} V_C(\omega) \tag{2.26}$$

$$= U_1(\omega_s, f_{\psi^{-1}(s)}) + U_2(\omega). \tag{2.27}$$

where the clique-potentials $V_C(\omega)$ are defined as

$$V_C(\omega) = \begin{cases} V'_C(\omega) & \text{if } C \in \mathcal{C}' \text{ and } C \notin \{\psi(r), r \in \mathcal{R}\} \\ V_r(\omega_{\psi(r)}) & \text{if } C = \psi(r) \text{ and } \psi(r) \notin \mathcal{C}' \\ V'_C(\omega) + V_r(\omega_{\psi(r)}) & \text{if } C = \psi(r) \text{ and } \psi(r) \in \mathcal{C}'. \end{cases} \tag{2.28}$$

If we assume that the observed image $\mathcal{F}$ is affected at site s only by the pixel s itself then Equation (2.26) can be further simplified: $\psi(r)$ reduces to s and the neighborhood system of the posterior distribution is equivalent to the neighborhood of the prior distribution.

2.1.3 Some Examples of Markov Models

Herein, we present some classical Markov models applied to a various image processing tasks such as image restoration, texture segmentation,

edge detection, and motion analysis. Let us begin this discussion with the restoration model proposed by D. Geman and S. Geman in [64].

2.1.3.1 Image Restoration

We observe a blurred noisy image $\mathcal{F}$ and we want to restore the original one. The components of the degraded model in Equation (2.16) have the following meanings: N is supposed to be a white Gaussian noise with mean μ and variance σ^2. H is a shift-invariant blurring matrix. First of all, we define the lattices on which the label process and the observation process are defined. The *observation process* simply consists of the gray-level at each pixel of the given image. Thus $\mathcal{R}$ is a lattice, each site corresponds to a pixel. The *label process* is more sophisticated involving both *pixel sites* and *line sites*. $\mathcal{X}$ is then a "mixed" process, also called *compound MRF*, having two subprocesses: a *pixel process* and a *line process*. The lattice $\mathcal{S}$ contains $\mathcal{R}$ (*pixel sites*) and another lattice with sites between each vertical and horizontal pairs of pixels representing a possible location of edge elements (*line sites*, see Figure 2.1).

We now turn to the posterior distribution and its energy function. Let us denote the *line process* by $\mathcal{X}^l$ and the *pixel process* by $\mathcal{X}^p$. We assume that $\mathcal{X}^p$ is a MRF over a homogeneous neighborhood system (see Section 1.2.4) $\mathcal{G}$ on $\mathcal{R}$ and $\mathcal{X}^l$ is also a MRF over a neighborhood system shown in Figure 2.1. (the neighbors of the black site are the gray sites). $\mathcal{X}$ has a prior distribution of

$$P(\mathcal{X}^p = \omega^p, \mathcal{X}^l = \omega^l) = \frac{1}{Z}\exp(-U'(\omega^p,\omega^l)) = \frac{1}{Z}\exp\left(-\sum_{C\in\mathcal{C}} V_C(\omega)\right), \quad (2.29)$$

where $\omega = (\omega^p,\omega^l)$. ω^p takes values among the available (discrete) gray-levels and ω^l among the line states. If we choose $\mathcal{G}$ such that it is large enough to encompass the dependencies caused by the blurring H then the posterior distribution also defines a MRF with energy function

$$U(\omega^p,\omega^l) = U'(\omega^p,\omega^l) + \frac{\|\vec{\mu} - \Psi^{-1}(H(\omega^p), f)\|^2}{2\sigma^2}. \quad (2.30)$$

The optimal labeling $\hat{\omega}$ is found by the MAP estimate minimizing the above energy function. The restored image is then given by the *pixel process* $\hat{\omega}^p$.

2.1.3.2 Texture Segmentation

The observations consist of a set of various texture features describing spatial statistics of the image. These features are computed on local windows around each pixel including mean, variance, correlation, entropy, contrast, homogeneity, etc [43, 47, 50, 65, 66, 122]. Here, both the *observation process* and the *label process* are defined on the same lattice $\mathcal{S}$ with sites corresponding to image pixels. The terms U_1 and U_2 from Equation (2.27) are defined in the following way: The prior energy U_2 usually favors spatially homogeneous regions assigning lower potentials to homogeneous cliques. The term U_1 does not have such a "standard" definition. It has various form in the literature. In [122], it measures the distance, at a given point s, between the distribution of the texture features in a small block B_s centered at s and the one in the whole (candidate) region R_s to which we want to assign s. This technique, as claimed in [122], permits us to automatically determine the number of regions. The energy function is defined as

$$U_1(\omega, \vec{f}) = \sum_{s \in \mathcal{S}} V_s(B_s, R_s) \tag{2.31}$$

$$V_s(B_s, R_s) = \sum_{i=1}^{m} (2\Delta(d(\vec{f}^i_{R_s}, \vec{f}^i_{B_s}) > c^i) - 1), \tag{2.32}$$

where m is the number of considered features. $\vec{f}_{B_s}$ and $\vec{f}_{R_s}$ denote the set of feature vectors on block B_s and on the region R_s respectively. $d(a, b)$ stands for the *Kolmogorov–Smirnov distance* and c^i is a threshold given by statistical tables associated to the Kolmogorov limit distribution (see [122]). The function Δ returns 1 if its argument is true, 0 otherwise.

2.1.3.3 Edge Detection

MRF models for edge detection are often *compound Gauss–Markov random fields* (CGMRF) [97, 220]. The local characteristics of a CGMRF

are given by

$$P(f_s \mid f_r, r \in \mathcal{S}) = \frac{1}{\sqrt{2\pi}\varsigma} \exp\left(\frac{1}{2\varsigma^2}\left(f_s - \mu_m - \sum_{r \in \mathcal{G}_s} \vartheta_r (f_r - \mu_m)\right)^2\right), \tag{2.33}$$

where ς is the deviation, μ_m is the mean, and ϑ_r is the model parameter. The supporting graph is similar to the one reported in [64] (cf. Figure 2.1). The observations $\mathcal{F}$ are considered to be corrupted by an additive Gaussian noise with zero mean and variance σ^2. The label field is again a mixed process containing both a *pixel process* $\mathcal{X}^p$ and a *line process*. Assuming a first order neighborhood system (cf. Figure 1.4) and denoting the horizontal and vertical line process by $\mathcal{X}^h$ and $\mathcal{X}^v$ respectively, a possible form of the energy function is given by [220]:

$$\begin{aligned} U(\omega, f) = \frac{1}{2\sigma^2} \sum_{s=(i,j)\in\mathcal{S}} \Big(& (f_{i,j} - \omega^p_{i,j})^2 + \beta^2 (1 - 2(\vartheta_h + \vartheta_v)) f^2_{i,j} \\ & + \vartheta_h (\beta^2 (f_{i,j} - f_{i-1,j})^2 (1 - \omega^h_{i,j}) + \alpha \omega^h_{i,j}) \\ & + \vartheta_v (\beta^2 (f_{i,j} - f_{i,j+1})^2 (1 - \omega^v_{i,j}) + \alpha \omega^v_{i,j}) \Big) \end{aligned} \tag{2.34}$$

with $1 - 2(\vartheta_h + \vartheta_v) > 0$. ϑ_v and ϑ_h are the model parameters for the vertical and horizontal cliques. β^2 corresponds to a regularization term reflecting the confidence in the data. In [220], $\beta^2 = \sigma^2/\varsigma^2$ expressing that when $\mathcal{F}$ is very noisy, we have no confidence in the data (β^2 is high). This model is related to the weak membrane model presented in [19]. The estimation of the *line process* is done by a Mean Field approach (see Section 2.1.1).

2.1.3.4 Motion Analysis

In [173], a MRF model for motion detection is presented. The *observation process* is defined both on the image-lattice $\mathcal{S}$ and on a time axis t. The detection of moving objects relies on the analysis of the variation of the intensity distribution in time. At each pixel, we have a two-element observation vector:

$$\vec{f^1_s}(t) = \mid y_s(t) - y_s(t - dt) \mid, \tag{2.35}$$

where $y_s(t)$ stands for the intensity value at pixel s at time t. $\vec{f}^2$ is a logical map of temporal changes between time t and $t - dt$. It equals to 1 if a temporal change of the intensity is valid at site s and 0 otherwise (for more details, see [90]). The *label process* is binary valued ($X_s(t) = 1$ if s is on a mask of a mobile object at time t). The energy function U consists of three terms. Two of them related to the observations and labels simultaneously (taking the role of U_1 in Equation (2.27)). One of them is used to reconstruct the mask of a mobile object at a given time:

$$U_1^2(\omega, \vec{f}_s^2) = \sum_{s \in \mathcal{S}} V_1(\omega_s(t), \vec{f}_s^2(t), \vec{f}_s^2(t + dt)), \tag{2.36}$$

the other expresses consistency between the current labeling and the intensity variation:

$$U_1^1(\omega, \vec{f}_s^1) = \sum_{s \in \mathcal{S}} \left(\frac{1}{2\sigma^2} (\vec{f}_s^1(t) - \mu\omega_s(t))^2 + (\vec{f}_s^1(t + dt) - \mu\omega_s)^2 \right), \tag{2.37}$$

where μ and σ are model parameters. The third term of the energy function U corresponds to U_2 with potentials favoring homogeneous masks.

2.2 A Classical Monogrid Segmentation Model

Now we will show how to construct a simple Markovian image segmentation model. Our goal is to demonstrate the basic steps to construct an easily applicable MRF model for non-textured images and further develop its multi-scale and hierarchical implementations as well as their combination in a multilayer model.

Let us suppose that the observations consist of the gray-levels. A very general problem is to find the labeling $\hat{\omega}$ which maximizes the a posteriori probability $P(\omega \mid \mathcal{F})$. Note that $\hat{\omega}$ is nothing else than the *segmentation* of the input image $\mathcal{F}$. Obviously, the actual segmentation $\hat{\omega}$ is determined by the probability measure $P(\omega \mid \mathcal{F})$. In other words, our segmentation model is expressed by the posterior probability $P(\omega \mid \mathcal{F})$, and then the optimal segmentation is simply found as the most likely labeling according to the probability distribution $P(\omega \mid \mathcal{F})$. Using the

results reported in Section 2.1.1, $\hat{\omega}$ is simply the MAP estimate of the *label field*.

Therefore the main question is how to define $P(\omega \mid \mathcal{F})$. Bayes theorem tells us that

$$P(\omega \mid \mathcal{F}) = \frac{1}{P(\mathcal{F})} P(\mathcal{F} \mid \omega) P(\omega). \tag{2.38}$$

Actually $P(\mathcal{F})$ does not depend on the labeling ω and we make the assumption that

$$P(\mathcal{F} \mid \omega) = \prod_{s \in \mathcal{S}} P(f_s \mid \omega_s). \tag{2.39}$$

It is then easy to see that the global labeling, which we are trying to find, is given by:

$$\hat{\omega} = \arg\max_{\omega \in \Omega} \prod_{s \in \mathcal{S}} P(f_s \mid \omega_s) \prod_{C \in \mathcal{C}} \exp(-V_C(\omega_C)). \tag{2.40}$$

It is obvious from this expression that the a posteriori probability also derives from a MRF. The energies of cliques of order 1 (also called *singletons*) directly reflect the probabilistic modeling of labels without taking into account context, which would be used for labeling the pixels independently. Let us assume that $P(f_s \mid \omega_s)$ is Gaussian, the class $\lambda \in \Lambda = \{0, 1, \ldots, L-1\}$ is represented by its mean value μ_λ and its deviation σ_λ. Furthermore, we will adopt a *smoothing prior* which prefers homogeneous regions. We thus get the following energy function (using Equation (2.27)):

$$U_1(\omega, \mathcal{F}) = \sum_{s \in \mathcal{S}} \left(\ln(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) \tag{2.41}$$

$$\text{and } U_2(\omega) = \sum_{C \in \mathcal{C}} V_2(\omega_C) \tag{2.42}$$

$$\text{where } V_2(\omega_C) = V_{\{s,r\}}(\omega_s, \omega_r) = \begin{cases} -\beta & \text{if } \omega_s = \omega_r \\ +\beta & \text{if } \omega_s \neq \omega_r, \end{cases} \tag{2.43}$$

where $\beta > 0$ is a model parameter controlling the homogeneity of the regions. In fact, our prior terms corresponds to the Potts model in statistical physics (see Section 1.2.2). As β increases, the resulting regions

become more homogeneous. Clearly, we have $2L + 1$ parameters. They are denoted by the vector Θ:

$$\Theta = \begin{pmatrix} \vartheta_0 \\ \vartheta_1 \\ \vdots \\ \vartheta_{2L} \end{pmatrix} \equiv \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{L-1} \\ \sigma_0 \\ \vdots \\ \sigma_{L-1} \\ \beta \end{pmatrix}. \tag{2.44}$$

If the parameters are supposed to be known, we say that the segmentation process is *supervised*. If they are unknown (and hence they have to be estimated simultaneously during the segmentation), the segmentation process is called *unsupervised*. A simple unsupervised segmentation method will be discussed later in Section 5.1.

For supervised segmentation, we are given a set of training data (small sub-images), each of them representing a class (see Figure 2.2).

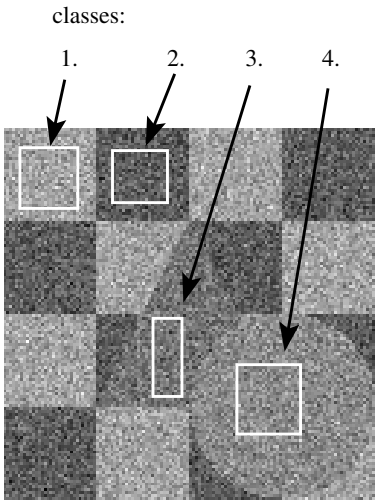


Fig. 2.2 Training sets on a synthetic image.

According to the law of large numbers, we can approximate the statistics of the classes (mean and variance) by the *empirical mean* and *empirical variance*:

$$\forall \lambda \in \Lambda : \qquad \mu_\lambda = \frac{1}{\mid S_\lambda \mid} \sum_{s \in S_\lambda} f_s, \tag{2.45}$$

$$\sigma_\lambda^2 = \frac{1}{\mid S_\lambda \mid} \sum_{s \in S_\lambda} (f_s - \mu_\lambda)^2, \tag{2.46}$$

where S_λ is the set of pixels included in the training set of class λ. The parameter β is initialized in an ad-hoc way (by trial and error). Typical values are between 0.5 and 1.

In Figure 2.3, we give an overview of a supervised segmentation process. We have two inputs: the image itself and the parameters Θ. They yield an energy function as defined in Equations (2.41)–(2.43). To find the MAP estimate, an algorithm is needed to minimize this energy function. In Section 3, we discuss a variety of such algorithms. Here, we have used the Gibbs Sampler [64] to get the minimum. The resulting image is just the labeling with minimum energy.

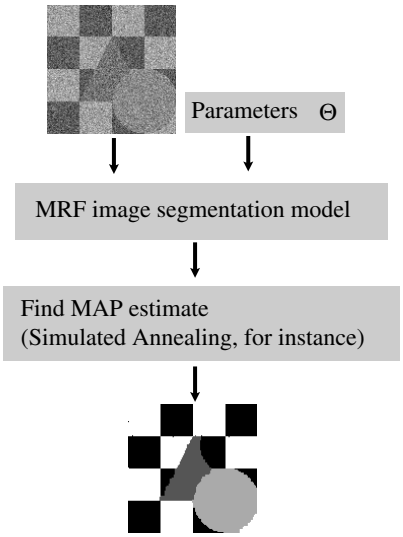


Fig. 2.3 Supervised image segmentation process.

2.3 Multigrid Approaches

Early vision processes deal with massive amounts of data. Thus such algorithms have two requirements in order to accomplish their tasks: they should be highly parallel to handle the data in a short time and they should provide a structure simplifying the extraction of higher level image features such as edges, regions, etc. Parallel multigrid (or pyramidal) schemes are one of the possible approaches satisfying these demands. An interesting book [99] on the subject presents various aspects of multiresolution image processing. Computation on image data with pyramids has importance not only in vision but also in the development of parallel computers (for example, see [99]). From a biological point of view, they support algorithms that share properties with the human vision (the human retina acquires visual information at multiple resolution at the same time).

Multigrid methods have a long existence in numerical analysis (e.g., partial differential equations). In image processing, they have also been used in various contexts from the mid 70s. Many vision problems have been formulated in terms of the optimization of a cost function. The multigrid approach of such an optimization is similar to the one used in numerical analysis (see [171] for a discussion about it). Here, we are interested in pyramidal methods applied to MRF image modeling [73]. We use the notion *pyramidal* to designate *multigrid* and *hierarchical* schemes. We are talking about *multigrid* methods, if the layers in the pyramid are not connected. In this case, the optimization algorithm is usually parallelizable only on the layers, but it is still sequential between layers. The layout of a *multigrid* model can be represented by a stack of smaller and smaller image lattices. If there is an inter-level communication, the model is called *hierarchical*. While the optimization algorithms associated with such models can be parallelized on the whole pyramid, the underlying MRF model becomes more complicated requiring more computation. The layout of the model is represented by a tree.

As we explained in Section 2.1, we usually have two processes in a MRF model: the *observation process* and the *label process*. In a multigrid scheme, we usually build scales with different resolution using the *label process* and keep the whole *observation process* [22, 23, 24, 67, 90, 142, 147, 173]. Herein, we briefly review some related techniques.

2.3.1 A Causal Hierarchical MRF Model

Bouman et al. [22, 23, 24] proposed an interesting multigrid model, which consists of a label-pyramid where each level is causally dependent on the coarser layer above it. Bouman also defines a new optimization criteria called *Sequential MAP* (SMAP) estimate. Let us briefly review this model.

First, we build a pyramid as shown in Figure 2.4. Each site at a coarse grid corresponds to a group of 2×2 sites at the grid below it. The fundamental assumption of the model is that the sequence of random fields from coarse to fine scale form a *Markov chain*. Denoting the label field at level n by $\mathcal{X}^n$, this relation may be stated as

$$P(\mathcal{X}^n = \omega^n \mid \mathcal{X}^l = \omega^l, l > n) = P(\mathcal{X}^n = \omega^n \mid \mathcal{X}^{n+1} = \omega^{n+1}). \quad (2.47)$$

The observation field $\mathcal{F}$ depends only on the labeling at the finest scale implying

$$P(\mathcal{F} = f \mid \mathcal{X}^n, n > 0) = P(\mathcal{F} = f \mid \mathcal{X}^0). \quad (2.48)$$

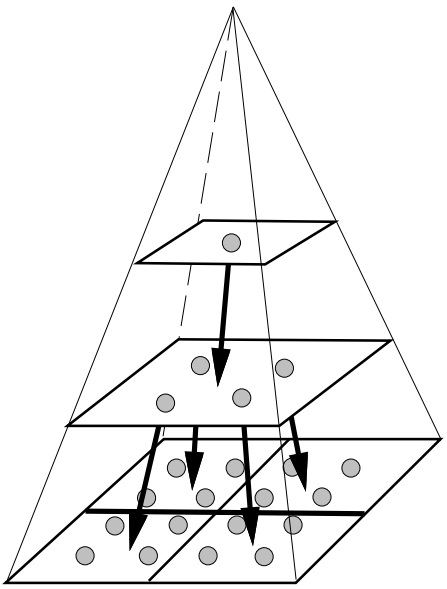


Fig. 2.4 A causal hierarchical model.

From the above equations, we can easily deduce the joint distribution of $\mathcal{F}$ and $\mathcal{X}$:

$$P(\mathcal{F} = f \mid \mathcal{X} = \omega) = P(f \mid \omega^0) \left(\prod_{n=0}^{M-1} P(\omega^n \mid \omega^{n+1}) \right) P(\omega^M). \quad (2.49)$$

where M is the coarsest scale.

As pointed out in [22], the conventional MAP estimate is not satisfying since its cost function (cf. Equation (2.8)) would assign equal cost to a single mislabeled pixel at $n = 0$ or to the mislabeling of hundreds of pixels at the coarsest scale. The proposed solution is to define a cost function related to the width of the largest grouping of mislabeled pixels. More precisely, let K be the coarsest scale containing a misclassified pixel. Obviously, the error at scale K will influence the labeling at finer scales leading to the misclassification of a group of pixels at the finest scale. The width of this group will be approximately 2^K. The new cost function is of the following form:

$$C_{\mathrm{SMAP}}(\mathcal{X},\omega) = \frac{1}{2} + \sum_{n=0}^{M} 2^{n-1} C_n(\mathcal{X},\omega) \quad (2.50)$$

$$\text{with } C_n(\mathcal{X},\omega) = 1 - \prod_{i=n}^{M} \delta(\mathcal{X}^i,\omega^i), \quad (2.51)$$

where δ is the Kronecker delta. The estimate $\hat{\omega}$ of the label field is obtained by minimizing the risk:

$$\hat{\omega} = \arg\min_{\omega} E\{C_{\mathrm{SMAP}}\} \quad (2.52)$$

$$= \arg\min_{\omega} \sum_{n=0}^{M} 2^{n-1}(1 - P(\omega^i, i \geq n \mid f)) \quad (2.53)$$

$$= \arg\max_{\omega} \sum_{n=0}^{M} 2^n P(\omega^i, i \geq n \mid f). \quad (2.54)$$

This estimate can be computed recursively since the fields $\mathcal{X}^n$ form a Markov chain. Denoting the estimate obtained at level i by $\hat{\omega}^i$, we

obtain the next procedure:

$$\hat{\omega}^M = \arg\max_{\omega^M} \ln(P(\omega^M \mid f)) \tag{2.55}$$

$$= \arg\max_{\omega^M} \ln(P(f \mid \omega^M)) \text{ (assuming that } \mathcal{X}^M \text{ is uniform)} \tag{2.56}$$

$$\hat{\omega}^n = \arg\max_{\omega^n} \ln(P(\omega^n \mid \hat{\omega}^{n+1}, f)) \tag{2.57}$$

$$= \arg\max_{\omega^n} (\ln(P(f \mid \omega^n)) + \ln(P(\omega^n \mid \hat{\omega}^{n+1}))), \quad n = M-1, \ldots, 0. \tag{2.58}$$

The procedure is initialized by the MAP estimate of the coarsest scale given the observed data. At finer scales, we are looking for the MAP estimate of $\mathcal{X}^n$, given the observations *and* the estimate $\hat{\omega}^{n+1}$ at the scale above it. Due to this structure, this estimator is called *sequential MAP* (SMAP).

Algorithm 1 (Causal Hierarchical Algorithm).

① Build a pyramid from the label field $\mathcal{X}$ by dividing the original grid into coarser scales. Each site corresponds to a block of 2×2 sites at the level below it.

② Find the optimal labeling at the coarsest scale using Equation (2.56) and set $n = M - 1$.

③ Find the MAP estimate of the label field at scale n given the observations f and the estimate ω^{n+1} at the coarser level above it using Equation (2.58).

④ Stop if $n = 0$, go to Step ③ with $n = n - 1$ otherwise.

The SMAP estimator has many advantages. The most important one is that it can be obtained with a single non-iterative pass in contrast to the MAP or MPM estimates (for more details, see [22]).

2.4 Multiscale MRF Models

Multiscale MRF models [190] became widespread in the '90s. Herein, we present a multiscale version of the monogrid segmentation model discussed in Section 2.2. A similar model has been proposed by Perez et al.

in [89, 90] for motion analysis using a second order neighborhood system shown in Figure 1.5 (see [171] for a more general description of the model). First, we give a general description of this model and then we study it in the case of a first order neighborhood system (see Figure 1.4) which is the most commonly used in image segmentation problems.

2.4.1 General Description

Let us suppose that $\mathcal{S} = \{s_1, \ldots, s_N\}$ is a $W \times H$ lattice, so that:

$$\mathcal{S} \equiv \mathcal{L} = \{(i,j) : 1 \leq i \leq W \text{ and } 1 \leq j \leq H\}, \tag{2.59}$$

and[2] $W = w^n$, $H = h^m$. Furthermore, we have some neighborhood system $\mathcal{G}$ on these sites. Let $\mathcal{X}$ be a MRF over $\mathcal{G}$ with an energy function U and potentials $\{V_C\}_{C \in \mathcal{C}}$. The following procedure will generate the multigrid MRF corresponding to $\mathcal{X}$:

(1) Let $\mathcal{B}^0 \equiv \mathcal{S}$ and $\Omega_0 \equiv \Omega$.
(2) For all $1 \leq i \leq M$ ($M = \inf(n,m)$), $\mathcal{S}$ is divided into blocks of size $w^i \times h^i$. These blocks will form the scale $\mathcal{B}^i = \{b_1^i, \ldots, b_{N_i}^i\}$ ($N_i = N/(wh)^i$).

The labels assigned to the sites of a block are supposed to be the same over the whole block. The common label of the block b_k^i is denoted by $\omega_k^i \in \Lambda$. This constraint yields a configuration space Ω_i which is a subset of the original set Ω. Obviously, for all $0 \leq i \leq M$: $\Omega_i \subset \Omega_{i-1} \subset \cdots \subset \Omega_0 \equiv \Omega$.

Now, let us consider the neighborhood system at scale i. It is clear, that b_k^i and b_l^i are neighbors if and only if there exist two neighbors $s \in \mathcal{S}$ and $r \in \mathcal{S}$ such that $s \in b_k^i$ and $r \in b_l^i$. This yields the same cliques as in $\mathcal{C}$. The cliques can be defined in the following way: Let $d = \deg(\mathcal{C})$. For all $1 \leq j \leq d$, the set of j blocks C_j^i at scale i is a clique of order j if there exists a clique $C \in \mathcal{C}$ (that is a clique at the finest scale) such

[2] This assumption introduces some restrictions on $\mathcal{L}$ but this is not crucial in practice since we work mostly on images where both W and H are a power of 2.

that:

$$1.\ \ C \subseteq \bigcup_{b_k^i \in C_j^i} b_k^i$$

$$2.\ \ \forall b_k^i \in C_j^i : C \cap b_k^i \neq \emptyset.$$

The set of cliques at scale i is denoted by $\mathcal{C}^i$ ($\mathcal{C}^0 \equiv \mathcal{C}$). The set of all cliques satisfying 1 and 2 for a given C_j^i is denoted by $\mathcal{D}_{C_j^i} \subseteq \mathcal{C}$.

Let us partition the original set $\mathcal{C}$ into the following disjoint subsets: For all $1 \leq j \leq d$, let $\mathcal{A}_j^i$ be the set of cliques $C \in \mathcal{C}$ for which there exists a clique C_j^i (that is, a clique of order j at the scale i) satisfying 1 and 2. Then, it turns out from the definition of $\mathcal{D}_{C_j^i}$ and $\mathcal{A}_j^i$, that

$$\mathcal{A}_j^i = \bigcup_{C_j^i \in \mathcal{C}^i} \mathcal{D}_{C_j^i}. \tag{2.60}$$

Using this partition, the energy function U can be decomposed in the following way:

$$\begin{aligned} U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega) &= \sum_{C \in \mathcal{A}_1^i} V_C(\omega) + \cdots + \sum_{C \in \mathcal{A}_d^i} V_C(\omega) \\ &= \sum_{C_1^i \in \mathcal{C}^i} \sum_{C \in \mathcal{D}_{C_1^i}} V_C(\omega) + \cdots + \sum_{C_d^i \in \mathcal{C}^i} \sum_{C \in \mathcal{D}_{C_d^i}} V_C(\omega). \end{aligned} \tag{2.61}$$

The main benefit of this decomposition is that the potentials at coarser scales can be derived by simple computation from the potentials at the finest scale. If we note the potential corresponding to a clique C_j^i of order j at the scale i by $V_{C_j^i}^{\mathcal{B}^i}$, we have the following family of potentials at scale $\mathcal{B}^i$:

$$V_{C_j^i}^{\mathcal{B}^i}(\omega) = \sum_{C \in \mathcal{D}_{C_j^i}} V_C(\omega). \tag{2.62}$$

If we examine our model, we see that there is some redundancy at coarser scales: we have the same label over the sites of a block. It seems then natural to associate a unique site to each block. These sites have the common state of the corresponding block and they form a coarser

grid $\mathcal{S}^i$ isomorphic to the corresponding scale $\mathcal{B}^i$. The coarser configuration space $\Xi_i = \{\xi_s^i : s \in \mathcal{S}^i, \xi_s^i \in \Lambda\}$ is isomorphic to Ω_i. Obviously, $\Xi_0 \equiv \Omega_0 \equiv \Omega$. The isomorphism Φ^i from $\mathcal{S}^i$ in $\mathcal{B}^i$ is just a projection of the coarser label field to the fine grid $\mathcal{S}^0 \equiv \mathcal{S}$:

$$\begin{aligned} \Phi^i &: \Xi_i \longrightarrow \Omega_i \\ &\xi^i \longmapsto \omega = \Phi^i(\xi^i). \end{aligned} \tag{2.63}$$

Φ^i keeps the same neighborhood structure on $\mathcal{S}^i$ as on $\mathcal{B}^i$ and the cliques on $\mathcal{S}^i$ inherit the potentials from the cliques defined on $\mathcal{B}^i$. These grids form a pyramid where level i contains the grid $\mathcal{S}^i$. The energy function of level i ($i = 0, \ldots, M$) is of the form:

$$U^i(\xi^i) = \sum_{C^i \in \mathcal{C}^i} V_{C^i}^i(\xi^i) \quad i = 0, \ldots, M \tag{2.64}$$

$$\text{where } V_{C^i}^i(\xi^i) = V_{C^i}^{\mathcal{B}^i}(\Phi^i(\xi^i)). \tag{2.65}$$

The multiscale algorithm essentially follows a top-down strategy (see Figure 2.5). First the highest layer of the pyramid is solved, then the next level is initialized by the result. The general formulation of the multiscale algorithm is the following:

Algorithm 2 (Multiscale MRF Algorithm).

① Let $\mathcal{B}^0 \equiv \mathcal{S}$, $\Omega_0 \equiv \Omega$ and divide $\mathcal{S}$ into blocks of size $w^i \times h^i$ ($1 \leq i \leq M$). Then associate a unique site to each block forming a coarse grid.

② Compute the clique-potentials at coarse grids using Equation (2.65).

③ Set $i = M$ and find the global minimum $\hat{\xi}^M$ of U^i in Equation (2.64).

④ Initialize the layer $i-1$ by a projection of $\hat{\xi}^i$ into $\mathcal{S}^{i-1} : \xi^{i-1} = (\Phi^{i-1})^{-1} \circ \Phi^i(\hat{\xi}^i)$, and find the minimum $\hat{\xi}^{i-1}$ of U^{i-1}.

⑤ Stop if $i = 1$, return to Step ④ with $i = i - 1$ otherwise.

The advantages of this algorithm are clear: each $\hat{\xi}^i$ gives a more or less good estimate of the final result. The estimate is better as i goes down to 0. For the higher values of i, the corresponding problem is simpler since the state space has only a few elements.

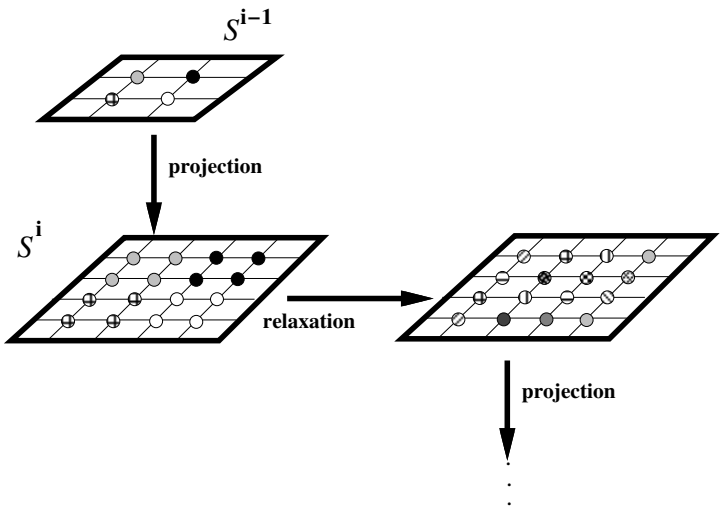


Fig. 2.5 Multiscale relaxation scheme.

This scheme is particularly well adapted to the deterministic relaxation methods which are more sensitive to the initial configuration than the stochastic[3] ones.

2.4.2 A Special Case

In the following, we will focus on a MRF with a first order neighborhood-system (see Figure 1.4) where the energy function is given by:

$$U(\omega,\mathcal{F}) = U_1(\omega,\mathcal{F}) + U_2(\omega). \tag{2.66}$$

U_1 (resp. U_2) denotes the energy of the first order (resp. second order) cliques. The notation $U_1(\omega,\mathcal{F})$ means that the first order potentials depend not only on the actual labeling but also on the given observations.

[3] Deterministic and stochastic relaxation algorithms will be discussed later in Section 3. Here, we only note that they are used to find a minimum of a non-convex energy function. Deterministic algorithms are usually faster than stochastic ones but they depend on the initial conditions. Stochastic algorithms find a global optimum starting from any initial configuration but they are much slower.

We follow the procedure described in Section 2.4.1 to generate a multigrid MRF model. Let $\mathcal{B}^i = \{b_1^i, \ldots, b_{N_i}^i\}$ denote the set of blocks and Ω_i the configuration-space at scale i ($\Omega_i \subset \Omega_{i-1} \subset \cdots \subset \Omega_0 = \Omega$). The label associated with block b_k^i is denoted by ω_k^i. We can define the same neighborhood structure on $\mathcal{B}^i$ as on $\mathcal{S}$:

$$b_k^i \text{ and } b_l^i \text{ are neighbors} \Longleftrightarrow \begin{cases} b_k^i \equiv b_l^i \text{ or} \\ \exists C \in \mathcal{C} \mid C \cap b_k^i \neq \emptyset \text{ and } C \cap b_l^i \neq \emptyset. \end{cases} \tag{2.67}$$

Now, let us partition the original set $\mathcal{C}$ into two disjoint subsets $\{\mathcal{C}_k^i\}$ and $\{\mathcal{C}_{k,l}^i\}$:

(1) cliques which are included in b_k^i (see Figure 2.6/(a)):

$$\mathcal{C}_k^i = \{C \in \mathcal{C} \mid C \subset b_k^i\} \tag{2.68}$$

(2) cliques which sit astride two neighboring blocks $\{b_k^i, b_l^i\}$ (see Figure 2.6/(b)):

$$\mathcal{C}_{k,l}^i = \{C \in \mathcal{C} \mid C \subset (b_k^i \cup b_l^i) \text{ and } C \cap b_k^i \neq \emptyset \text{ and } C \cap b_l^i \neq \emptyset\}. \tag{2.69}$$

It is obvious from this partition that our energy function (see Equation (2.66)) can be decomposed as:

$$\begin{aligned} U_1(\omega, \mathcal{F}) &= \sum_{s \in \mathcal{S}} V_1(\omega_s, f_s) \\ &= \sum_{b_k^i \in \mathcal{B}^i} \underbrace{\sum_{s \in b_k^i} V_1(\omega_s, f_s)}_{V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F})} = \sum_{b_k^i \in \mathcal{B}^i} V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F}) \end{aligned} \tag{2.70}$$

$$\begin{aligned} \text{and } U_2(\omega) &= \sum_{C \in \mathcal{C}} V_2(\omega_c) \\ &= \sum_{b_k^i \in \mathcal{B}^i} \underbrace{\sum_{C \in \mathcal{C}_k^i} V_2(\omega_c)}_{V_k^{\mathcal{B}^i}(\omega_k^i)} + \sum_{\{b_k, b_l\} \text{neighbors}} \underbrace{\sum_{C \in \mathcal{C}_{k,l}^i} V_2(\omega_c)}_{V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i)} \\ &= \sum_{b_k^i \in \mathcal{B}^i} V_k^{\mathcal{B}^i}(\omega_k^i) + \sum_{\{b_k, b_l\} \text{neighbors}} V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i). \end{aligned} \tag{2.71}$$

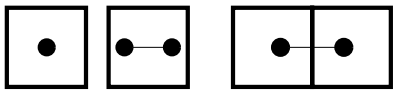


Fig. 2.6 The two subsets of $\mathcal{C}$ in the case of a first order neighborhood system. a: $\mathcal{C}_k^i$; b: $\mathcal{C}_{k,l}^i$.

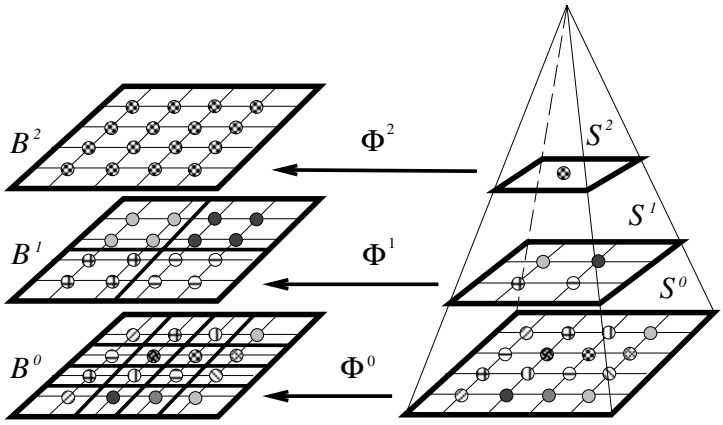


Fig. 2.7 The isomorphism Φ^i between $\mathcal{B}^i$ and $\mathcal{S}^i$.

Now, we can define our pyramid (cf. Figure 2.7) where level i contains the coarse grid $\mathcal{S}^i$ which is isomorphic to the scale $\mathcal{B}^i$. The coarse grid has a reduced configuration space $\Xi_i = \Lambda^{N_i}$.

The model on the grids $\mathcal{S}^i$ $(i = 0, \ldots, M)$ defines a set of consistent multiscale MRF models, whose energy functions are derived from Equations (2.70) and (2.71)

$$U^i(\xi^i, \mathcal{F}) = U_1^i(\xi^i, \mathcal{F}) + U_2^i(\xi^i) \tag{2.72}$$

$$= U_1(\Phi^i(\xi_i), \mathcal{F}) + U_2(\Phi^i(\xi_i)) \quad i = 0, \ldots, M \tag{2.73}$$

with

$$U_1^i(\xi^i, \mathcal{F}) = \sum_{k \in \mathcal{S}^i} (V_1^{\mathcal{B}^i}(\omega_k^i, \mathcal{F}) + V_k^{\mathcal{B}^i}(\omega_k^i)) = \sum_{k \in \mathcal{S}^i} V_1^i(\xi_k^i, \mathcal{F}) \tag{2.74}$$

and

$$U_2^i(\xi^i) = \sum_{\{k,l\} neighbors} V_{k,l}^{\mathcal{B}^i}(\omega_k^i, \omega_l^i) = \sum_{C^i \in \mathcal{C}^i} V_2^i(\xi_C^i), \tag{2.75}$$

where C^i is a second order clique corresponding to the definition in Equation (2.67) and $\mathcal{C}^i$ is the set of cliques on the grid $\mathcal{S}^i$.

2.4.3 Application to Image Segmentation

We can easily apply the equations obtained at the previous section to the image segmentation model presented in Section 2.2. For simplicity, the block size is supposed to be $n \times n$ (that is $w = h = n$). Then, we get [104, 107, 105]:

$$U_1^i(\xi^i, \mathcal{F}) = \sum_{s^i \in \mathcal{S}^i} V_1^i(\xi_{s^i}^i, \mathcal{F}), \tag{2.76}$$

where

$$\begin{aligned} V_1^i(\xi_{s^i}^i, \mathcal{F}) &= \sum_{s \in b_{s^i}^i} V_1(\omega_s, f_s) + \sum_{C \in \mathcal{C}_{s^i}^i} V_2(\omega_C) \\ &= \sum_{s \in b_{s^i}^i} \left(\log(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) - p^i\beta \end{aligned} \tag{2.77}$$

$$\tag{2.78}$$

and

$$U_2^i(\xi^i) = \sum_{C^i = \{r^i, s^i\} \in \mathcal{C}^i} V_2^i(\xi_{C^i}^i) \tag{2.79}$$

where

$$V_2^i(\xi_{C^i}^i) = \sum_{\{r,s\} \in \mathcal{D}_{C^i}} V_2(\omega_r, \omega_s) = \begin{cases} -q^i\beta & \text{if } \omega_r = \omega_s \\ +q^i\beta & \text{if } \omega_r \neq \omega_s \end{cases}. \tag{2.80}$$

The values of p^i and q^i depend on the chosen block size and the neighborhood structure. p^i is the number of cliques included in the same block at scale $\mathcal{B}^i$ and q^i is the number of cliques between two

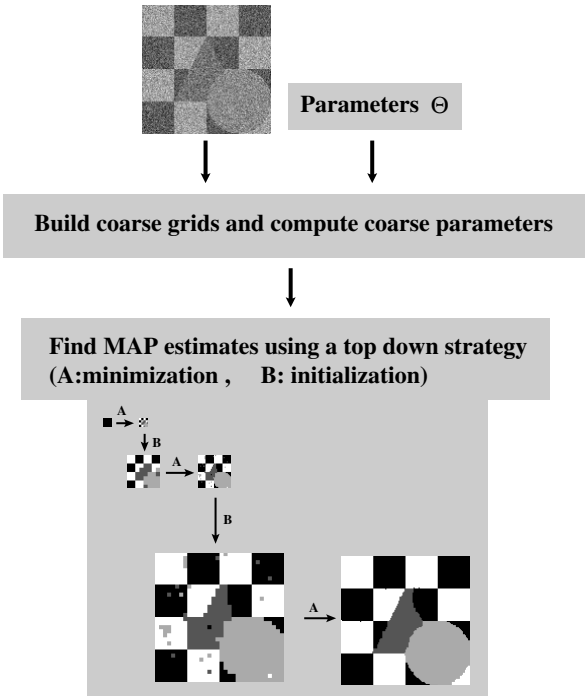


Fig. 2.8 Multiscale supervised image segmentation process.

neighboring blocks at scale $\mathcal{B}^i$. Considering blocks of $n \times n$ and a first order neighborhood system, we get:

$$p^i = 2n^i(n^i - 1) \tag{2.81}$$

$$q^i = n^i. \tag{2.82}$$

In Figure 2.8, we give an overview of a *supervised* multiscale segmentation process. As in the monogrid case, we have two inputs: the image and the monogrid parameters $\Theta \equiv \Theta^0$ defined in Equation (2.44). Then, we build the pyramid and compute the parameters $\Theta^i (i = 1, \ldots, M)$ at coarse grids obtaining $M + 1$ energy functions defined by Equations (2.77)–(2.80). Using a *top-down* strategy, we minimize these functions (using the Iterated Conditional Mode algorithm [17], for example) and we take the labeling at the finest level as the final segmentation.

2.5 Hierarchical Models

In this section, we present a hierarchical MRF model [103, 104, 106, 107]. The basic idea is to find a better way of communication between the levels than the initialization used for the multiscale model in Section 2.4. The model consists of introducing new interactions between two neighbor grids[4] in the pyramid. This scheme permits also the parallelization of the relaxation algorithm on the whole pyramid. First, we give a general description of the model, then we study a special case with a first order neighborhood system.

2.5.1 General Description

We consider hereafter the label pyramid and the whole observation field defined in the previous section. Let $\bar{\mathcal{S}} = \{\bar{s}_1, \ldots, \bar{s}_{\bar{N}}\}$ denote the sites of this pyramid. Obviously,

$$\begin{aligned} \bar{\mathcal{S}} &= \bigcup_{i=0}^{M} \mathcal{S}^i \\ \bar{N} &= \sum_{i=0}^{M} N_i. \end{aligned} \tag{2.83}$$

$\bar{\Omega}$ denotes the configuration-space of the pyramid:

$$\begin{aligned} \bar{\Omega} &= \Xi^0 \times \Xi^1 \times \cdots \times \Xi^M \\ &= \{\bar{\omega} \mid \bar{\omega} = (\xi^0, \xi^1, \ldots, \xi^M)\}. \end{aligned} \tag{2.84}$$

Let us define the following function Ψ between two neighbor levels, which assigns to a site of any level the corresponding block of sites at the level below it (that is its descendants). Ψ^{-1} assigns its ancestor to a site (see Figure 2.9):

$$\begin{aligned} \Psi : \quad & \mathcal{S}^i \longrightarrow \mathcal{S}^{i-1} \\ \Psi(\bar{s}) = \; & \{\bar{r} \mid \bar{s} \in \mathcal{S}^i \Rightarrow \bar{r} \in \mathcal{S}^{i-1} \text{ and } b^{i-1}_{\bar{r}} \subset b^i_{\bar{s}}\} \end{aligned} \tag{2.85}$$

[4] One can imagine interactions between more than two levels but these schemes are too complicated for practical use.

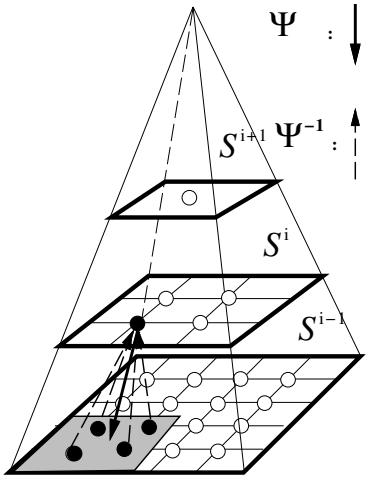


Fig. 2.9 The functions Ψ and Ψ^{-1}.

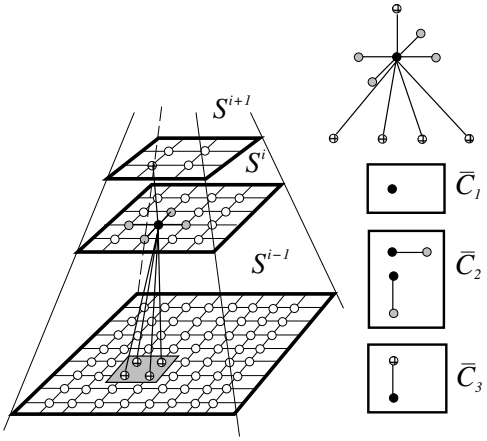


Fig. 2.10 The neighborhood system $\bar{\mathcal{G}}$ and the cliques $\bar{\mathcal{C}}_1$, $\bar{\mathcal{C}}_2$ and $\bar{\mathcal{C}}_3$.

Now, we can define on these sites the following neighborhood-system (see Figure 2.10):

$$\bar{\mathcal{G}} = (\bigcup_{i=0}^{M} \mathcal{G}_i) \cup \{\Psi^{-1}(\bar{s}) \cup \Psi(\bar{s}) \mid \bar{s} \in \bar{\mathcal{S}}\}, \tag{2.86}$$

where $\mathcal{G}_i$ is the neighborhood structure of the ith level, and we have the following cliques:

$$\bar{\mathcal{C}} = (\bigcup_{i=0}^{M} \mathcal{C}^i) \cup \mathcal{C}^*, \tag{2.87}$$

where $\mathcal{C}^*$ denotes the new cliques siting astride two neighbor grids. We can easily estimate the degree of the new cliques since it depends on the block size: Each site interacts with its ancestor (there is one) and its descendants (there are wh), thus:

$$\deg(\mathcal{C}^*) = \max_{C^* \in \mathcal{C}^*} \mid C^* \mid = wh + 2 \tag{2.88}$$

$$\text{and } \deg(\bar{\mathcal{C}}) = \deg(\mathcal{C}) + \deg(\mathcal{C}^*) - 1. \tag{2.89}$$

Furthermore, let $\bar{\mathcal{X}}$ be a MRF over $\bar{\mathcal{G}}$ with an energy function $\bar{U}$ and potentials $\{\bar{V}_{\bar{C}}\}_{\bar{C} \in \bar{\mathcal{C}}}$. The energy function is of the following form:

$$\begin{aligned}
\bar{U}(\bar{\omega}) &= \sum_{\bar{C} \in \bar{\mathcal{C}}} \bar{V}_{\bar{C}}(\bar{\omega}) \\
&= \sum_{i=0}^{M} \sum_{\bar{C} \in \mathcal{C}^i} V^i_{\bar{C}}(\bar{\omega}) + \sum_{\bar{C} \in \mathcal{C}^*} \bar{V}_{\bar{C}}(\bar{\omega}) \\
&= \sum_{i=0}^{M} \sum_{C^i \in \mathcal{C}^i} V^i_{C^i}(\xi^i) + \sum_{C^* \in \mathcal{C}^*} \bar{V}_{C^*}(\bar{\omega}) \\
&= \sum_{i=0}^{M} U^i(\xi^i) + U^*(\bar{\omega}).
\end{aligned} \tag{2.90}$$

It turns out from the above equation, that the energy function consists of two terms. The first one corresponds to the sum of the energy functions of the grids defined in the previous section and the second one $(U^*(\bar{\omega}))$ is the energy over the new cliques located between neighbor grids.

Since we have defined a MRF on the whole pyramid, the MAP estimate of the label field is obtained by minimizing the Hamiltonian defined in Equation (2.90). The algorithms are essentially the same as in

the monogrid case but in the parallelization, we can take benefit of the pyramidal structure and define a new annealing technique as we will see later in Section 3. The final result is the labeling obtained at the finest level as in the multiscale method. However an important difference is that the resulting labeling is not related to the MAP estimate of the finest level (without the pyramid). Intuitively, it can be seen as the MAP estimate of an MRF model (defined on the finest scale) with larger neighborhoods.

2.5.2 A Special Case

In this section, we study the model in the case of a first order neighborhood system. We will consider herein only first and second order cliques. Clique-potentials for the other cliques are supposed to be 0. The cliques can be partitioned into three disjoint subsets $\bar{\mathcal{C}}_1,\bar{\mathcal{C}}_2,\bar{\mathcal{C}}_3$ corresponding to first order cliques, second order cliques which are on the same level and second order cliques which sit astride two neighboring levels (see Figure 2.10). Using this partition, we can derive the following energy function:

$$\bar{U}(\bar{\omega},\mathcal{F}) = \bar{U}_1(\bar{\omega},\mathcal{F}) + \bar{U}_2(\bar{\omega}) \tag{2.91}$$

$$\begin{aligned}\bar{U}_1(\bar{\omega},\mathcal{F}) &= \sum_{\bar{s}\in\bar{\mathcal{S}}} \bar{V}_1(\bar{\omega}_{\bar{s}},\mathcal{F}) \\ &= \sum_{i=0}^{M}\sum_{s^i\in\mathcal{S}^i} V_1^i(\xi^i_{s^i},\mathcal{F}) = \sum_{i=0}^{M} U_1^i(\xi^i,\mathcal{F})\end{aligned} \tag{2.92}$$

$$\begin{aligned}\bar{U}_2(\bar{\omega}) &= \sum_{C\in\bar{\mathcal{C}}_2} \bar{V}_2(\bar{\omega}_C) + \sum_{C\in\bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) \\ &= \sum_{i=0}^{M}\sum_{C\in\mathcal{C}^i} V_2^i(\xi^i_C) + \sum_{C\in\bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C) \\ &= \sum_{i=0}^{M} U_2^i(\xi^i) + \sum_{C\in\bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_C).\end{aligned} \tag{2.93}$$

2.5.3 A Hierarchical Segmentation Model

Considering the segmentation model presented in Section 2.2, its hierarchical equivalent can be derived from Equations (2.92) and (2.93) [104, 106, 107]:

$$\bar{U}_1(\bar{\omega}, \mathcal{F}) = \sum_{i=0}^{M} \sum_{s^i \in \mathcal{S}^i} V_1^i(\xi^i, \mathcal{F}) \tag{2.94}$$

$$\text{and } \bar{U}_2(\bar{\omega}) = \sum_{i=0}^{M} \sum_{C^i \in \mathcal{C}^i} V_2^i(\xi_{C^i}^i) + \sum_{C \in \bar{\mathcal{C}}_3} \bar{V}_2(\bar{\omega}_c) \tag{2.95}$$

$$\text{where } \bar{V}_2(\bar{\omega}_c) = \bar{V}_{\{\bar{s},\bar{r}\}}(\bar{\omega}_{\bar{s}}, \bar{\omega}_{\bar{r}}) = \begin{cases} -\gamma & \text{if } \bar{\omega}_{\bar{s}} = \bar{\omega}_{\bar{r}} \\ +\gamma & \text{if } \bar{\omega}_{\bar{s}} \neq \bar{\omega}_{\bar{r}} \end{cases} \tag{2.96}$$

where V_1^i and V_2^i are defined in Equations (2.77) and (2.80). We have a new model parameter γ which favors similar classes between a site, its ancestor and its descendants. Thus, taking into account the new

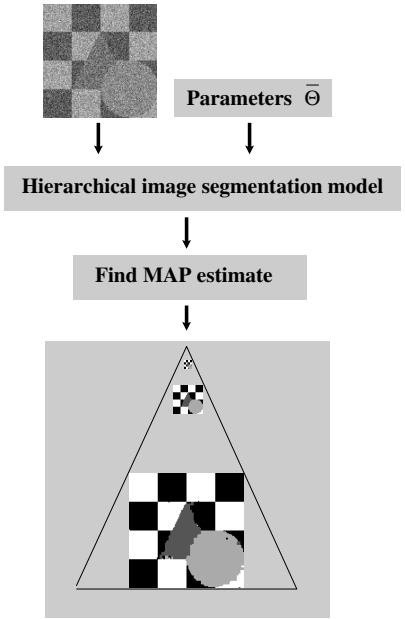


Fig. 2.11 Hierarchical supervised segmentation process.

parameter, Equation (2.44) can be rewritten as

$$\bar{\Theta} = \begin{pmatrix} \bar{\vartheta}_0 \\ \bar{\vartheta}_1 \\ \vdots \\ \bar{\vartheta}_{2L+1} \end{pmatrix} \equiv \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_{L-1} \\ \sigma_0 \\ \vdots \\ \sigma_{L-1} \\ \beta \\ \gamma \end{pmatrix}. \tag{2.97}$$

In Figure 2.11, we give an overview of a hierarchical image segmentation process. We have two inputs: the image itself and the parameters $\bar{\Theta}$. After building the label-pyramid, they give an energy function defined in Equations (2.94)–(2.96). To find the minimum of this function (that is, the MAP estimate of the label-pyramid), we essentially use the same algorithms as in the monogrid case (Iterated Conditional Mode in Figure 2.11). The resulting image is the labeling at lowest level of the pyramid.

3

Classical Energy Minimization

Bayesian methods coupled with Markovian models usually result in a non-convex energy function. In order to find an estimate, one has to minimize this function. Unfortunately, this is a very hard computational problem known as *combinatorial optimization*. For example, considering an image 16×16 with only two possible labels at each pixel, we obtain a configuration space of 2^{256} elements. It is then impossible to find the optimum by computing the possible values of the cost function. On the other hand, due to the non-convexity classical gradient descent methods cannot be used since they get stuck in a local minimum.

The idea of the solution comes again from statistical physics: In 1953, Metropolis et al. [153] proposed a Monte-Carlo simulation to find equilibrium states of thermodynamical systems. It was realized in the early 80's, independently by Černy [34] and Kirkpatrick et al. [126], that there is an analogy between minimizing the cost function of a combinatorial optimization problem and finding energy minima of thermodynamical systems by slowly cooling a solid until equilibrium is reached. They have substituted the energy function of the solid by the cost function and executed the Metropolis algorithm at a sequence of slowly

decreasing "temperature." The so defined combinatorial optimization algorithm was named *Simulated Annealing* (SA) [62, 137, 154, 176].

The research in the field has rapidly grown up resulting in a variety of contributions to the original SA. The most important is probably the *Gibbs Sampler* proposed by Geman and Geman in [64]. While SA algorithms find a global optimum with probability 1 [137], they require a large amount of computation. To avoid this drawback, two solutions have been proposed: One of them deals with the possible parallelization of SA algorithms [4]. Another solution is to use *deterministic* algorithms which are suboptimal but converge in a few iterations requiring less computing time [17, 115, 154].

3.1 Equilibrium State and the Metropolis Algorithm

For historical reasons, we begin this chapter by the original formulation of the Metropolis algorithm, as it was proposed by Metropolis et al. Following [153], we are given N particles on a square. If the positions of the particles are known, we can easily calculate the potential energy of the system:

$$E = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} V(\delta_{ij}), \tag{3.1}$$

where V is the potential between molecules and δ_{ij} is the distance between particles i and j. To calculate the equilibrium value of any quantity F, the following integral has to be computed [153]:

$$\bar{F} = \frac{\int F \exp\left(\frac{-E}{kT}\right) d^{2N}p d^{2N}q}{\int \exp\left(\frac{-E}{kT}\right) d^{2N}p d^{2N}q}, \tag{3.2}$$

where $d^{2N}p d^{2N}q$ is a volume element in the $4N$-dimensional phase space.

To compute this integral, the following procedure has been suggested: Place the N particles in any configuration in a lattice where each particle is represented by its coordinates (x, y). According to the following rules, we move each particles within a square of side 2α centered

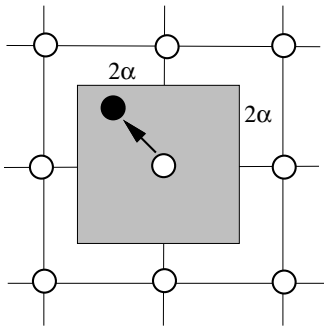


Fig. 3.1 Moving particles according to the Metropolis algorithm.

about its original position (see Figure 3.1):

$$x^{\text{new}} = x + \alpha\xi_1 \tag{3.3}$$

$$y^{\text{new}} = y + \alpha\xi_2, \tag{3.4}$$

where ξ_1 and ξ_2 are uniform random numbers in the range $[-1,1]$. Then, we compute the energy-change ΔE of the system caused by the move. If the move brings the system to a state of lower energy (i.e., if $\Delta E < 0$), then the particle is placed in its new position. If $\Delta E > 0$, we accept the new position with probability $\exp(-\Delta E/kT)$: Let ξ_3 be a random number in the range $[0,1]$. If $\xi_3 < \exp(-\Delta E/kT)$, we move the particle to $(x^{\text{new}}, y^{\text{new}})$. Otherwise, we keep its old position. The integral in Equation (3.2) is then approximated by

$$\bar{F} = \frac{1}{M}\sum_{i=1}^{N} F_i, \tag{3.5}$$

where F_i is the property F of the system after the *ith* move has been carried out. It has been proved in [153] that the above procedure generates configurations with probability $\exp(-E/kT)$.

3.2 Combinatorial Optimization and Simulated Annealing

The classic example of a combinatorial optimization problem is the *traveling salesman problem*:

Example 3.1. Given N cities and the distances D_{ij} between the cities i and j representing the cost of traveling. One has to plan the sales-man's *optimal* route which will pass through each city once and return finally to the starting point, minimizing the total length.

The above mentioned example belongs to the class of NP-complete problems. As it is well known, there is no method to find an exact solution of such a problem with a computing effort bounded by a power of N. SA is one of the heuristics proposed to solve the traveling salesman problem.

As we have said in the introduction, SA algorithm is based on the analogy between the simulation of the annealing of solids and the solving of combinatorial optimization problems. This is why the algorithm proposed by Černy [34] and Kirkpatrick et al. [126] became known as *Simulated Annealing*.

In physics, annealing means heating up a solid to a maximum value at which all particles randomly arrange themselves in the liquid phase then slowly cooling it down. In this way, all particles arrange themselves in the low energy state of a corresponding lattice. At each temperature T, the solid is allowed to reach a *thermal equilibrium* which is characterized by the *Boltzmann distribution*:

$$P(\omega) = \frac{1}{Z(T)} \exp\left(-\frac{U(\omega)}{kT}\right), \tag{3.6}$$

where $U(\omega)$ is the energy of the state, $Z(T)$ is the partition function depending on T and k is the Boltzmann constant. For a fixed T, the above equation is nothing else but a Gibbs distribution. Clearly, as the temperature decreases, the above distribution concentrates on the states with lower energy and when the temperature approaches zero, only the minimum states have a non-zero probability. Decreasing the temperature is crucial: It has been pointed out in [126] that if the cooling is too rapid and the system is not allowed to reach thermal equilibrium for each temperature, a global minimum cannot be reached.

For a fixed temperature, the evolution to thermal equilibrium is simulated by the *Metropolis algorithm* [153] presented in Section 3.1. Here, the algorithm is used to generate sequences of configurations of a combinatorial optimization problem. SA is then a sequence of *Metropolis algorithms* evaluated at a sequence of *decreasing* temperatures such that equilibrium is reached at each temperature.[1]

Now, it is time to give an exact definition of the SA algorithm in its original formulation: Let us denote by $\omega, \eta, \ldots$ the configurations of a combinatorial optimization problem (they correspond to the states of a solid) and let $U(\omega)$ denotes the cost (also called energy) of the configuration ω (it corresponds to the energy of the state ω in a thermodynamical system.). The elements of the configurations are indexed by $\mathcal{S} = \{s_1, \ldots, s_N\}$ and the common state space is denoted by $\Lambda = \{0, 1, \ldots, L-1\}$. The set of all possible configurations is denoted by Ω. Since $\forall s \in \mathcal{S} : \omega_s \in \Lambda$, $\Omega = \Lambda^N$.

Algorithm 3 (Simulated Annealing).

① Set $k = 0$ and initialize ω randomly. Choose a sufficiently high initial temperature $T = T_0$.

② Construct a trial perturbation η from the current configuration ω such that η differs only in one element from ω.

③ **(Metropolis criteria)** Compute $\Delta U = U(\eta) - U(\omega)$ and accept η if $\Delta U < 0$ else accept with probability $\exp(-\Delta U/T)$ (analogy with thermodynamics):

$$\omega = \begin{cases} \eta & \text{if } \Delta U \leq 0, \\ \eta & \text{if } \Delta U > 0 \text{ and } \xi < \exp(-\Delta U/T), \\ \omega & \text{otherwise} \end{cases} \tag{3.7}$$

where ξ is a uniform random number in $[0, 1)$.

④ Goto Step ② until equilibrium is reached.

⑤ Decrease the temperature: $T = T_{k+1}$ and goto Step ② with $k = k + 1$ until the system is *frozen*.

---

[1] This is the original formulation of the SA which is practically not used nowadays. We will discuss later more recent SA variants.

The above algorithm is also called *homogeneous annealing* since it is described by a sequence of homogeneous Markov chains. If the temperature is decreased after each transition, the algorithm is described by an inhomogeneous Markov chain thus it is referred to as *inhomogeneous annealing*. This is the most often used form of annealing. We obtain such an algorithm if we withdraw Step ④ from Algorithm 3.

3.2.1 Mathematical Model

The mathematical model of the Simulated Annealing was extensively studied by Aarts and van Laarhoven in [137]. We briefly describe this model:

The SA generates a sequence of configurations which constitutes a Markov chain. $P_{\omega,\eta}(k-1,k)$ is the probability that the configuration obtained after k transitions is η given the previous configuration ω. Furthermore, let $X(k)$ denote the state reached after the kth transition. The probability of this event is given by:

$$P(X(k)=\omega)=\sum_{\zeta} P(X(k-1)=\zeta)P_{\zeta,\omega}(k-1,k) \quad k=1,2,\ldots \tag{3.8}$$

If the transition probability $P_{\omega,\eta}(k-1,k)$ does not depend on k, the corresponding chain is homogeneous, otherwise it is inhomogeneous. The transition probabilities depend also on the temperature parameter T. Thus, if T is kept constant, the chain will be homogeneous and the transition matrix $P=P(T)$ can be written as:

$$P_{\omega,\eta}(T)=\begin{cases} G_{\omega,\eta}(T)A_{\omega,\eta}(T) & \forall \eta \neq \omega \\ 1-\sum_{\zeta} G_{\omega,\zeta}(T)A_{\omega,\zeta}(T) & \eta=\omega, \end{cases} \tag{3.9}$$

where $G_{\omega,\eta}(T)$ is the *generation probability* of generating η from ω and $A_{\omega,\eta}(T)$ is the *acceptance probability* of configuration η, once it has been generated from ω. It is clear from the definition in Equation (3.9) that $P(T)$ is a stochastic matrix:

$$\forall \omega : \sum_{\zeta} P_{\omega,\zeta}(T)=1 \tag{3.10}$$

In the original formulation of the algorithm, $G_{\omega,\eta}(T)$ is given by the uniform distribution on the configurations η which differs from ω only

for one component. $A(T)$ is given by the Metropolis criterion:

$$A_{\omega,\eta}(T) = \min(1, \exp(-(U(\eta) - U(\omega))/T)), \tag{3.11}$$

where $U(\omega)$ is the cost or energy function.

3.2.1.1 More on Cooling Schedules

In [137] it has been shown that SA converges with probability one to a globally optimal configuration if certain conditions hold for the temperature schedule. Namely, that T_k goes towards 0 not faster than $\Gamma/\ln(k)$ for some constant Γ independent of k (inhomogeneous annealing). For the homogeneous annealing, there are other conditions [137], but we will focus on *inhomogeneous annealing* since it is the most commonly used schedule.

D. Geman and S. Geman were the first researchers to obtain such conditions for the inhomogeneous annealing [64]. They proved in [64], that SA converges if the temperature is decreased according to the following rule:

$$T_k \geq \frac{\Gamma}{\ln(k)} \tag{3.12}$$

with

$$\Gamma > \max_{\omega \in \Omega} U(\omega) - \min_{\omega \in \Omega} U(\omega). \tag{3.13}$$

Later on, this condition has been refined by B. Hajek in [80, 81] to derive a necessary and sufficient condition. While Geman's condition says that Γ in Equation (3.12) must be larger than the difference between the maximum and minimum energy, Hajek's condition claims that it is sufficient if Γ is larger than the largest depth of any jump in the energy. In Figure 3.2, we compare the meaning of Geman's and Hajek's conditions.

Although both results are very important from a theoretical point of view, it is clear that in practice we can never compute Γ from Equation (3.13) as the exact shape of the energy function is unknown. Furthermore, the above theoretical schedule in Equation (3.12), being logarithmic, would be too slow for practical applications. For these

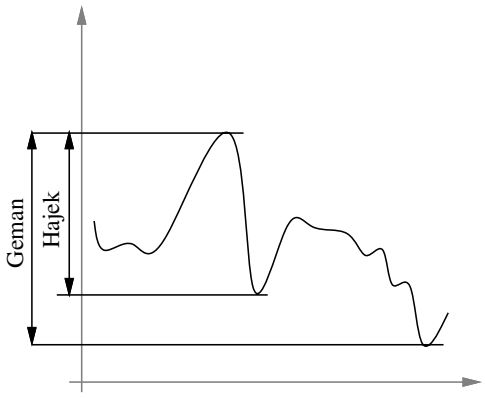


Fig. 3.2 Geman's and Hajek's condition.

reasons, the theoretical schedule have to be approximated. Due to this approximation, however, the algorithm is no longer guaranteed to find a global optimum, but following the recommendations given below, the obtained results should not be far from it.

Initial Temperature The initial temperature T_0 must be so high that virtually all transitions are accepted. It is extremely difficult to determine such a value since it is related to the maximum and minimum values of the energy function to be minimized [64]. There are some heuristics to get a reasonably estimate of the initial temperature [126, 137] but usually one set T_0 to a relatively low value resulting in a faster execution of the algorithm. In [64] for example, $T_0 = 4$ has been suggested, and we have used the same value throughout the simulations presented in this survey.

Final Temperature Obviously, $\lim_{k\to\infty} T_k = 0$ can only be approximated in a finite number of values for T_k. Thus, we need a stopping criteria determining the final value of the temperature. We can simply fix the number of values T_k or terminate the execution of the algorithm if the last few configurations obtained by SA have nearly the same energy (i.e., ΔU is less than a certain threshold).

Cooling Schedule The most important point is a decreasing rule of the temperature. Logarithmic rules (cf. Figure 3.3) are usually too

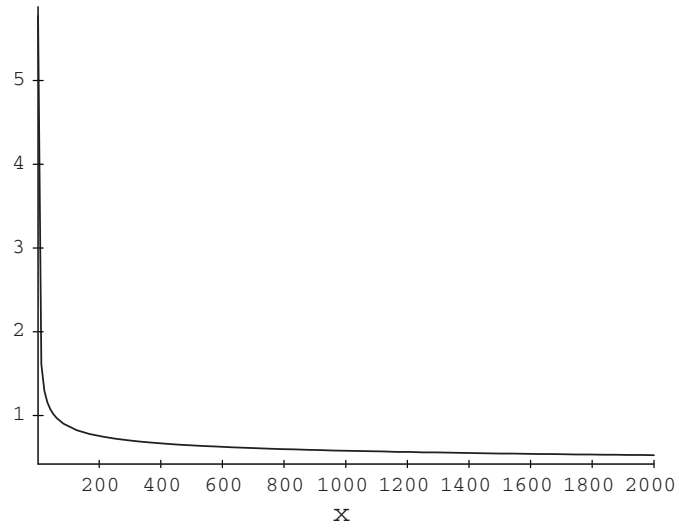


Fig. 3.3 Logarithmic cooling schedule $(4/\ln(k))$.

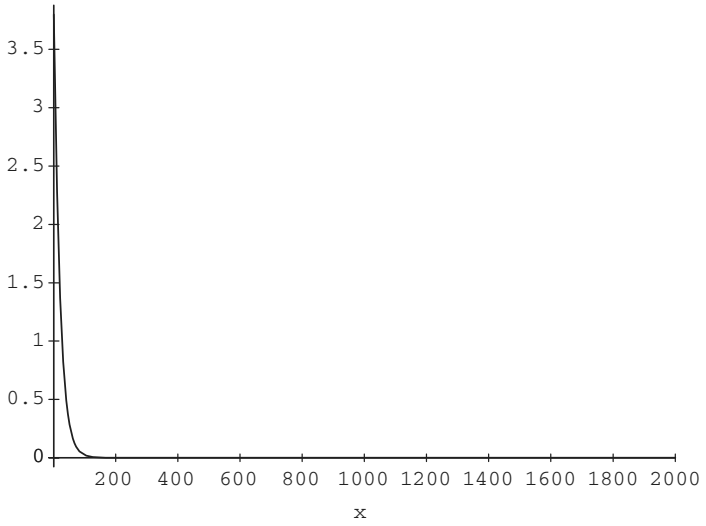


Fig. 3.4 Exponential cooling schedule $(0.95^k \cdot 4)$.

slow for practical use. Instead, exponential rules (cf. Figure 3.4) are the most frequently used:

$$T_{k+1} = c \cdot T_k, \quad k = 0, 1, 2, \ldots \tag{3.14}$$

where $c < 1$ is a constant close to 1. This rule was first proposed in [126] with $c = 0.95$ and it became widely used by others. Note, however, that the choice of c is largely determined by the landscape of the actual energy function (which is unknown, of course). Therefore higher values, such as $c = 0.98 - 0.999$ are recommended for more complex energy functions.

3.2.1.2 More on Generation Matrices

In some cases, there may be better ways of generating configurations than the uniform distribution. Hereafter, we discuss some improvements in the generation mechanism.

In image processing, the most convenient implementation is a raster scan where the pixels of the image are visited for update in order $(0,0),(0,1),\ldots,(1,0),(1,1)\ldots$ (see Figure 3.5). At each site, a new state is chosen with a uniform probability among the possible states different from the current one.

To speed up the algorithm, one can use synchronous update (that is, updating each pixel at the same time) but convergence can no longer be guaranteed in this case. A good compromise is a partially synchronous scheme: update only conditionally independent pixels at the same time. These pixels form a so-called *coding set* [17] (see Figure 3.6 for an example). The number of coding sets depends on the order of the MRF

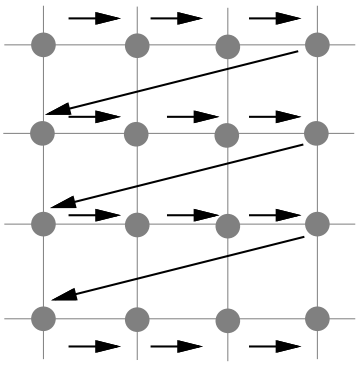


Fig. 3.5 Example of a raster scan generation mechanism.

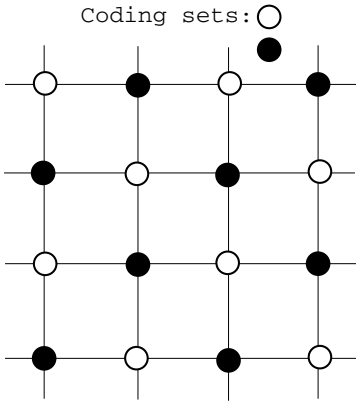


Fig. 3.6 Coding sets in the case of a first order MRF.

used as an image model. These techniques are mostly used in parallel implementation but one can also use them on a sequential machine.

A more interesting improvement is to use a non-uniform generation mechanism. For example, at high temperatures, it would be helpful to bias the generation of configurations to favor *large* transitions [137]. Large transitions often help the system to reach equilibrium faster because a single transition of length l can be implemented at a less computational cost than l transitions of length 1.

For *rejection-less methods* [137], the acceptance matrix is incorporated in the generation matrix and all transitions are accepted once they are generated. Considering the Metropolis criterion, the probability of generating ω from η is given by

$$G_{\omega,\eta} = \frac{\min(1, -(U(\eta) - U(\omega)))}{\sum_{\xi} \min(1, -(U(\xi) - U(\omega)))}, \tag{3.15}$$

where ξ differs only in one element from ω. One can prove that the algorithm also converges toward a global optimum. The problem is that after *each* transition, we have to compute *all* $G_{\omega,\eta}$'s.

In [42], the updating order depends on a *local stability measure*. At each step, only the least stable site is changed. The stability of a site is measured by the energy-loss or energy-gain which would be caused by changing the current state of the site. The larger the negative value of this measure, the more confidence we have to change the state of the

site. Thus, the sites with strong evidence in favor of a label are visited early. This method is called *highest confidence first* (HCF) [39, 42].

3.2.1.3 More on Acceptance Matrixes — The Gibbs Sampler

A more elaborated acceptance matrix has been proposed by D. Geman and S. Geman in [64]. Coupled with an inhomogeneous annealing schedule, it became the most popular SA algorithm known as *Gibbs Sampler*:

Algorithm 4 (Gibbs Sampler).

① Set $k = 0$, assign an arbitrary initial configuration ω and let $T = T_0$ be a sufficiently high initial temperature.

② For each configuration which differs at most in one element from the current configuration ω (they are denoted by $\mathcal{N}_\omega$), compute the energy $U(\eta)$ ($\eta \in \mathcal{N}_\omega$).

③ **(Gibbs Sampler)** From the configurations in $\mathcal{N}_\omega$, a sample η is drawn such that η is accepted with probability

$$\frac{\exp(-U(\eta))}{\sum_{\zeta \in \mathcal{N}_\omega} \exp(-U(\zeta))} \tag{3.16}$$

as the new configuration.

④ Decrease the temperature: $T = T_{k+1}$ and goto Step ② with $k = k + 1$ until the system is *frozen*.

In the case of a two state system such as the Ising model, the Gibbs Sampler is equivalent to the Metropolis algorithm. Notice that the generation matrix is simply $G_{\omega,\eta} = 1$ for all ω and $\eta \in \mathcal{N}_\omega$, 0 otherwise.

3.3 Clustered Sampling via Generalized Swendsen–Wang Method

As we have seen in Section 3.2, the core part of Simulated Annealing is essentially a Markov Chain Monte Carlo sampler which generates likely configurations (i.e., segmentations in our case) from the huge space of possible labelings Ω. It is clear that efficient sampling is crucial in order

to achieve convergence within a reasonably low number of iterations. Till now, we have seen two such samplers, the *Metropolis–Hastings* sampler [85, 153] in Section 3.1 and the *Gibbs sampler* [64] in Section 3.2.1.3. The common limitation of these classical MCMC samplers is that they only change the label of a single site within one step of the algorithms. Considering a simple Ising model (see Section 1.2.1), at high temperature, spins are basically independent and hence configurations are easily sampled in a point-wise way. At low temperature, however, spins are typically clustered into homegenous regions making any flip of a single spin almost impossible. Therefore any point-wise sampling algorithm would suffer from an exponential slow-down in generating independent configurations. In order to avoid this limitation of classical MCMC sampling, a more sophisticated method, called *clustered sampling* [7, 198] is adopted: at each step, instead of changing the label of a single site, the labels of a whole cluster of sites is swaped. For that purpose, however, the original MRF model has to be reformulated as a graph partitioning problem. In its original formulation, the algorithm is known as the *Swendsen–Wang* algorithm [198] which has been used for Monte Carlo simulation of the Ising model. The algorithm has also been succesfully applied to sample from a Potts model [157]. Recently, the *Swendsen–Wang* sampler has been generalized by Barbu and Zhu [7] to sampling arbitrary probabilities. Herein, we will overview this generalized version of the algorithm. We remark, however, that clustered sampling may not be a universal solution for all type of MRF models, for example, the Chien-model discussed in [53] is one such example.

3.3.1 Graph Representation of an MRF Model

First of all, let us formulate our MRF model as an adjacency graph $\mathcal{G} = \{\langle \mathcal{V}, \mathcal{E} \rangle\}$, where $\mathcal{V} = \{s | s \in \mathcal{S}\}$ denotes the set of nodes, which are the pixel sites from the original MRF model, and $\mathcal{E}$ is a set of edges connecting neighboring nodes. Clearly, for any nodes s and $r \in \mathcal{V}$, there is an edge $e = \langle s, r \rangle \in \mathcal{E}$ connecting them *if and only if* the corresponding sites s and r are neighbors in the MRF model. Furthermore, between any nodes s and r, there may be at most one edge. It is also important to note that although an MRF model is usually defined on a lattice, the

corresponding adjacency graph $\mathcal{G}$ may not be represented on a lattice due to, for example, higher order interactions.

An n-partition of the set of vertices $\mathcal{V}$ is denoted by

$$\mathcal{P}_n = (\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n) \text{ , with} \tag{3.17}$$

$$\forall i \neq j : \mathcal{V}_i \cap \mathcal{V}_j = \emptyset \text{ , and} \tag{3.18}$$

$$\bigcup_{i=1}^{n} \mathcal{V}_i = \mathcal{V}. \tag{3.19}$$

In our context, such a partitioning is equalivalent to a labeling ω of the sites $s \in \mathcal{S}$. Although the graph representation can be constructed for an arbitrary number of labels, for an easier understanding of the basic idea, let us limit the discussion to the binary case (basically an Ising-type model). Hence the number of such *label partitions* is fixed to $n = 2$ as each node s may only take a label from $\{+1, -1\}$. We will denote these sets of vertices as $\mathcal{V}^+$ and $\mathcal{V}^-$. In summary, the following equivalence holds:

$$\mathcal{P}_2 = (\mathcal{V}^+, \mathcal{V}^-) \equiv \omega, \quad \text{where} \tag{3.20}$$

$$\mathcal{V}^+ = \{s | \omega_s = +1\} \tag{3.21}$$

$$\mathcal{V}^- = \{s | \omega_s = -1\}. \tag{3.22}$$

The role of the graph edges $\mathcal{E}$ is to introduce spatial constraints on such partitionings. Thus the problem becomes *partitioning the graph* $\mathcal{G}$ into subgraphs $\mathcal{G}^\ell = \langle \mathcal{V}^\ell, \mathcal{E}^\ell \rangle, \ell \in \{+, -\}$ so that each subgraph is a *full subgraph* of $\mathcal{G}$, that is, it keeps all the edges connecting two vertices within $\mathcal{V}^\ell$:

$$\mathcal{E}^\ell = \{e = \langle s, r \rangle \in \mathcal{E} | s, r \in \mathcal{V}^\ell\} \tag{3.23}$$

Note that a subgraph is not necessarily connected. For example, in Figure 3.7 black regions in the left image denote sites labeled by -1 hence all such regions belong to $\mathcal{G}^- = \langle \mathcal{V}^-, \mathcal{E}^- \rangle$ although they are not connected. Therefore, the graph can be further partitioned based on *connected components*, each component having the label of either $+1$ or -1.

The edges between two arbitrary sets $\mathcal{V}_i$ and $\mathcal{V}_j$ $(i \neq j)$ define a *cut*

$$\mathbb{C}_{i,j} = \{e = \langle s, r \rangle \in \mathcal{E} | s \in \mathcal{V}_i \text{ and } r \in \mathcal{V}_j\}. \tag{3.24}$$

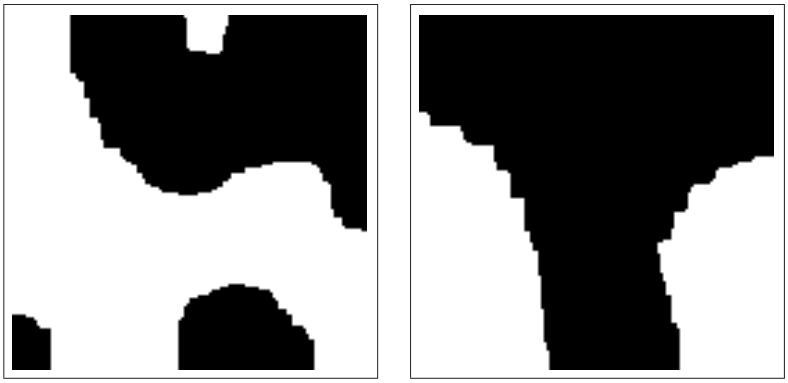


Fig. 3.7 Samples generated by the *Gibbs Sampler* from an Ising-type model. Black corresponds to state -1 and white corresponds to state $+1$.

It is clear that any partitioning $\mathcal{P}_n$ divides the graph edges $\mathcal{E}$ into cuts and subgraph edges:

$$\mathcal{E} = \left(\bigcup_{k=1}^{n} \mathcal{E}_k \right) \cup \left(\bigcup_{i \neq j} \mathbb{C}_{i,j} \right). \tag{3.25}$$

3.3.2 Generalized Swendsen–Wang Method

The Swendsen–Wang algorithm (SW) [198] generates partitionings of the adjacency graph $\mathcal{G}$, each of these graph partitionings is equivalent to a sample ω drawn from Ω according to the probability density of the MRF model. At each time step t, we thus have a partitioning

$$\mathcal{P}_2^t = (\mathcal{V}^{+,t}, \mathcal{V}^{-,t}) \tag{3.26}$$

and edges are connecting vertices that are in the same state (that is, their labels are equal):

$$\mathcal{E}^t = \{e = \langle s, r \rangle | \omega_s = \omega_r\}. \tag{3.27}$$

This defines a sparse graph which consists of a number of n disjoint *connected components* (subgraphs) g_i $(i = 1, 2 \ldots, n)$, such that the set of nodes $\mathcal{V}_i^t$ of each subgraph g_i is either $\mathcal{V}_i^t \subseteq \mathcal{V}^{+,t}$ or $\mathcal{V}_i^t \subseteq \mathcal{V}^{-,t}$. For example, in Figure 3.7, we have 5 subgraphs in the left image, of which

two belongs to $\mathcal{V}^{+,t}$ and three to $\mathcal{V}^{-,t}$. Thus we have

$$\mathcal{G}^t = \bigcup_{i=1}^{n} g_i, \text{ with} \tag{3.28}$$

$$\mathcal{V} = \bigcup_{i=1}^{n} \mathcal{V}_i^t, \text{ and} \tag{3.29}$$

$$\mathcal{E}^t = \bigcup_{i=1}^{n} \mathcal{E}_i^t \tag{3.30}$$

These connected components define an n-partition $\bar{\mathcal{P}}_2^t = (\mathcal{V}_1^t, \mathcal{V}_2^t, \ldots, \mathcal{V}_n^t)$. SW will turn on/off edges within one of the subgraphs $\mathcal{G}^{+,t}$ or $\mathcal{G}^{-,t}$ For that purpose, we will adopt the *SWC-2* algorithm from [7]. One time step of the algorithm works as follows:

Algorithm 5 (Swendsen–Wang Graph Cuts).

① Given the current partition $\mathcal{P}_2$, select a *seed vertex* s. Assume $s \in \mathcal{V}^\ell, \ell \in \{+,-\}$. Let $\mathcal{V}_0 = \{s\}$, $\mathcal{E}_0 = \emptyset$, and $\mathbb{C}_{0,\ell} = \emptyset$.

② For any edge $e = \langle s, r \rangle$, where $s \in \mathcal{V}_0$ and $r \in \mathcal{V}^\ell$, turn e on with probability $P_{\text{on}}(e)$, else turn e off. If e is on then set $\mathcal{V}_0 = \{r\} \cup \mathcal{V}_0$ and $\mathcal{E}_0 = \{e\} \cup \mathcal{E}_0$, otherwise set $\mathbb{C}_{0,\ell} = \{e\} \cup \mathbb{C}_{0,\ell}$.

③ Propose to assign $\mathcal{V}_0$ a new label $\ell' \in \{+,-\}, \ell \neq \ell'$ with probability $P(\omega_{\mathcal{V}_0} = \ell' | \mathcal{V}_0, \mathcal{P}_2)$.

④ Accept the move with probability $\mathcal{A}(\mathcal{P}_2 \to \mathcal{P}_2')$.

Note that the above algorithm generates a subgraph $g_0 = \langle \mathcal{V}_0, \mathcal{E}_0 \rangle$ such that $\exists i (i = 1, 2, \ldots, n) : g_0 \subseteq g_i$. Furthermore, the Markov chain states $\mathcal{P}_2$ and $\mathcal{P}_2'$ differ only in the label assigned to $\mathcal{V}_0$ (ℓ in $\mathcal{P}_2$ and ℓ' in $\mathcal{P}_2'$).

According to the Metropolis–Hastings method [85, 153], the acceptance probability is

$$\mathcal{A}(\mathcal{P}_2 \to \mathcal{P}_2') = \min\left(1, \frac{P(\mathcal{P}_2' \to \mathcal{P}_2) P(\mathcal{P}_2')}{P(\mathcal{P}_2 \to \mathcal{P}_2') P(\mathcal{P}_2)}\right). \tag{3.31}$$

The move $\mathcal{P}_2 \leftrightarrow \mathcal{P}_2'$ between the two states $\mathcal{P}_2$ and $\mathcal{P}_2'$ of the Markov chain is realized by selecting *and* flipping $\mathcal{V}_0$. We thus have

$$\frac{P(\mathcal{P}_2' \to \mathcal{P}_2)}{P(\mathcal{P}_2 \to \mathcal{P}_2')} = \frac{P(\mathcal{V}_0|\mathcal{P}_2')}{P(\mathcal{V}_0|\mathcal{P}_2)} \cdot \frac{P(\omega_{\mathcal{V}_0} = \ell|\mathcal{V}_0, \mathcal{P}_2')}{P(\omega_{\mathcal{V}_0} = \ell'|\mathcal{V}_0, \mathcal{P}_2)}. \tag{3.32}$$

The probability ratio selecting $\mathcal{V}_0$ in the moves $\mathcal{P}_2 \leftrightarrow \mathcal{P}_2'$ depends only on the cuts between $\mathcal{V}_0$ and the rest of the graph [7]. Assuming $\mathcal{V}_0 \subseteq \mathcal{V}^\ell$ in $\mathcal{P}_2$ and $\mathcal{V}_0 \subseteq \mathcal{V}^{\ell'}$ in $\mathcal{P}_2'$, and $P_{\text{off}}(e)$ and $P_{\text{on}}(e)$ are the probabilities of switching off or on a particular edge e, we get

$$\frac{P(\mathcal{V}_0|\mathcal{P}_2)}{P(\mathcal{V}_0|\mathcal{P}_2')} = \frac{\prod_{e \in \mathbb{C}_{0,\ell}} P_{\text{off}}(e)}{\prod_{e \in \mathbb{C}_{0,\ell'}} P_{\text{off}}(e)}. \tag{3.33}$$

Clearly, the probabilities P_{off} and P_{on} play an important role in the proposal step of the SW sampler. Obviously, they must be related to the local probability of the MRF model which is determined by the potentials of the interacting sites. For example, assuming $e = \langle s, r \rangle \in \mathcal{E}$, we may define

$$P_{\text{on}}(e) = \exp(-U_e), \text{ where} \tag{3.34}$$

$$U_e = \sum_{C \in \mathcal{C}: s \in C \text{ or } r \in C} V_C(\omega_C); \text{ and} \tag{3.35}$$

$$P_{\text{off}}(e) = 1 - P_{\text{on}}(e) \tag{3.36}$$

3.4 Multi-Temperature Annealing

Another interesting question related to Simulated Annealing is the temperature schedule itself. We have seen in Section 3.2, that convergence is only guaranteed when a proper schedule is adopted. In classical SA, it is assumed, that we have a *global* temperature. What about a *locally* changing temperature, when each clique may have its own temperature? Under what conditions would such an algorithm converge? In this section, we present a Multi-Temperature Annealing (MTA) schedule [106, 222], which allows different temeperatures at different cliques while convergence is still guaranteed with probability 1. More generally, we have the following problem:

Let $\mathcal{S} = \{s_1, \ldots, s_N\}$ be a set of sites, $\mathcal{G}$ some neighborhood system with cliques $\mathcal{C}$ and X a MRF over these sites with energy function U.

We define an annealing scheme where the temperature T depends on the iteration k and on the cliques C. Let $\oslash$ denotes the following operation:

$$U(\omega) \oslash T(k,C) = \sum_{C \in \mathcal{C}} \frac{V_C(\omega)}{T(k,C)} \tag{3.37}$$

$$P(X = \omega) = \pi_{T(k,C)}(\omega) = \frac{\exp(-U(\omega) \oslash T(k,C))}{Z}. \tag{3.38}$$

Let us suppose that the sites are visited for updating in the order $\{n_1, n_2, \ldots\} \subset \mathcal{S}$. The resulting stochastic process is denoted by $\{X(k), k = 0, 1, 2, \ldots\}$, where $X(0)$ is the initial configuration. $X(k)$ is an inhomogeneous Markov chain with transition matrix:

$$P_{\omega,\eta}(k-1,k) = \begin{cases} G_{\omega,\eta}(T(k,C))A_{\omega,\eta}(T(k,C)) & \forall \eta \neq \omega \\ 1 - \sum_{\zeta \neq \omega} G_{\omega,\zeta}(T(k,C))A_{\omega,\zeta}(T(k,C)) & \eta = \omega. \end{cases} \tag{3.39}$$

Considering the Gibbs sampler, the generation matrix $G_{\omega,\eta}(T(k,C))$ and acceptation matrix $A_{\omega,\eta}(T(k,C))$ are given by:

$$\begin{aligned} G_{\omega,\eta}(T(k,C)) &= G_{\omega,\eta}(k) = \begin{cases} 1, & \text{if } \eta = \omega|_{\omega_{n_k} = \lambda} \text{ for some } \lambda \in \Lambda \\ 0, & \text{otherwise} \end{cases} \\ A_{\omega,\eta}(T(k,C)) &= \pi_{T(k,C)}(X_{n_k} = \omega_{n_k} \mid X_s = \omega_s, s \neq n_k). \end{aligned} \tag{3.40}$$

Notice that the acceptance is governed by the *local* characteristics. $\pi_{T(k,C)}(X_{n_k} = \omega_{n_k} \mid X_s = \omega_s, s \neq n_k)$ has a slightly different meaning than $\pi_{T(k,C)}(\omega)$ in Equation (3.38):

$$\pi_{T(k,C)}(X_s = \omega_s \mid X_r = \omega_r, s \neq r) = \frac{1}{Z_s} \exp\left(-\frac{\sum_{C \in \mathcal{C}: s \in C} V_C(\omega)}{T(k,C)}\right) \tag{3.41}$$

with

$$Z_s = \sum_{\lambda \in \Lambda} \exp\left(-\frac{\sum_{C \in \mathcal{C}: s \in C} V_C(\omega|_{\omega_s = \lambda})}{T(k,C)}\right) \tag{3.42}$$

The transition matrix at time k is then of the following form:

$$P_{\omega,\eta}(k) = \begin{cases} \pi_{T(k,C)}(X_{n_k} = \eta_{n_k} \mid X_s = \eta_s, s \neq n_k), & \text{if } \eta = \omega|_{\omega_{n_k}=\lambda} \\ & \text{for some } \lambda \in \Lambda \\ 0, & \text{otherwise} \end{cases} \tag{3.43}$$

Let Ω_{opt} be the set of globally optimal configurations:

$$\Omega_{opt} = \{\omega \in \Omega : U(\omega) = \min_{\eta \in \Omega} U(\eta)\}. \tag{3.44}$$

Let π_0 be the uniform distribution on Ω_{opt}, and define:

$$U^{sup} = \max_{\omega \in \Omega} U(\omega), \tag{3.45}$$

$$U^{inf} = \min_{\omega \in \Omega} U(\omega), \tag{3.46}$$

$$\text{and } \Delta = U^{sup} - U^{inf}. \tag{3.47}$$

Let us examine the decomposition of $U(\omega) \oslash T(k,C)$ defined in Equation (3.37). Let $\omega' \in \Omega_{\text{opt}}$ be a *globally* optimal configuration ($U(\omega') = U^{inf}$). Furthermore, let $\omega \in \Omega \setminus \Omega_{\text{opt}}$ be any other non-optimal configuration. Obviously, $U(\omega) - U(\omega') > 0$. In the case of a classical annealing, dividing by a constant temperature does not change this relation ($\forall k$: $(U(\omega) - U(\omega'))/T_k$ is still positive). But it is not necessarily true that $(U(\omega) - U(\omega')) \oslash T(k,C)$ is also positive! Because choosing sufficiently small temperatures for the cliques where ω'_C is *locally* not optimal (strengthening the cliques where $V_C(\omega) - V_C(\omega') < 0$) and choosing sufficiently high temperatures for the cliques where ω'_C is *locally* optimal (weakening the cliques where $V_C(\omega) - V_C(\omega') \geq 0$), we obtain $(U(\omega) - U(\omega')) \oslash T(k,C) < 0$, meaning that ω' is no longer *globally* optimal with respect to $U \oslash T(k,C)$.

Thus, we have to impose further conditions on the temperature to assure the convergence toward the *global* optimum of U. First, let us examine the decomposition over the cliques of $U(\omega) - U(\eta)$ for arbitrary ω and η, $\omega \neq \eta$:

$$U(\omega) - U(\eta) = \sum_{C \in \mathcal{C}} (V_C(\omega) - V_C(\eta)). \tag{3.48}$$

Indeed, there may be negative and positive members in the decomposition. According to this fact, we have the following subsums:

$$\sum_{C\in\mathcal{C}}(V_C(\omega)-V_C(\eta)) = \underbrace{\sum_{C\in\mathcal{C}:(V_C(\omega)-V_C(\eta))<0}(V_C(\omega)-V_C(\eta))}_{\Sigma^-(\omega,\eta)} + \underbrace{\sum_{C\in\mathcal{C}:(V_C(\omega)-V_C(\eta))\geq 0}(V_C(\omega)-V_C(\eta))}_{\Sigma^+(\omega,\eta)}. \quad (3.49)$$

Now, let us examine Δ defined in Equation (3.47). If we want to decompose Δ as defined above, we have to choose some configuration ω' with a maximum energy (i.e., $U(\omega')=U^{\sup}$) and another configuration ω'' with a minimum energy (i.e. $U(\omega'')=U^{\inf}$). Obviously, there may be more than one decomposition depending on the number of globally optimal configurations ($\mid \Omega_{\text{opt}} \mid$) and the number of configurations with maximal global energy ($\mid \Omega_{\sup} \mid$). Thus, the decomposition of Δ for a given (ω',ω'') is of the following form:

$$\Delta = \Sigma^-(\omega',\omega') + \Sigma^+(\omega',\omega''). \quad (3.50)$$

Furthermore, let us define Σ^+_Δ as:

$$\Sigma^+_\Delta = \min_{\substack{\omega'\in\Omega_{\sup}\\ \omega''\in\Omega_{\text{opt}}}} \Sigma^+(\omega',\omega''). \quad (3.51)$$

Obviously $\Delta \leq \Sigma^+_\Delta$. The following theorem gives an annealing schedule, basically the same as in [64]. *However, the temperature here is a function of k and $C\in\mathcal{C}$.*

Theorem 3.2 (Multi-Temperature Annealing). Assume that there exists an integer $\kappa \geq N$ such that for every $k=0,1,2,\ldots$, $\mathcal{S}\subseteq\{n_{k+1},n_{k+2},\ldots,n_{k+\kappa}\}$. For all $C\in\mathcal{C}$, let $T(k,C)$ be any decreasing sequence of temperatures in k for which

(1) $\lim_{k\to\infty}T(k,C)=0$.
Let us denote respectively by $T_k^{\inf}$ and $T_k^{\sup}$ the maximum and minimum of the temperature function at k ($\forall C\in\mathcal{C}$: $T_k^{\inf}\leq T(k,C)\leq T_k^{\sup}$).

(2) For all $k \geq k_0$, for some integer $k_0 \geq 2$: $T_k^{\inf} \geq N\Sigma_\Delta^+ / \ln(k)$.
(3) If $\Sigma^-(\omega,\omega') \neq 0$ for some $\omega \in \Omega \setminus \Omega_{\text{opt}}$, $\omega' \in \Omega_{\text{opt}}$ then a further condition must be imposed:
For all k: $\frac{T_k^{\sup} - T_k^{\inf}}{T_k^{\inf}} \leq R$ with

$$R = \min_{\substack{\omega \in \Omega \setminus \Omega_{\text{opt}} \\ \omega' \in \Omega_{\text{opt}} \\ \Sigma^-(\omega,\omega') \neq 0}} \frac{U(\omega) - U^{\inf}}{\mid \Sigma^-(\omega,\omega') \mid}. \tag{3.52}$$

Then for any starting configuration $\eta \in \Omega$ and for every $\omega \in \Omega$:

$$\lim_{k \to \infty} P(X(k) = \omega \mid X(0) = \eta) = \pi_0(\omega). \tag{3.53}$$

The proof of this theorem appears in [106].

Remarks:

1 In practice, we cannot determine R and Σ_Δ^+, as we cannot compute Δ either.

2 Considering Σ_Δ^+ in condition 2, we have the same problem as in the case of a classical annealing. The only difference is that in a classical annealing, we have Δ instead of Σ_Δ^+. Consequently, the same solutions may be used: an exponential schedule with a sufficiently high initial temperature.

3 The factor R is more interesting. We propose herein two possibilities which can be used for practical implementations of the method: Either we choose a sufficiently small interval $[T_0^{\inf}, T_0^{\sup}]$ and suppose that it satisfies the condition 3 (we have used this technique in the simulations), or we use a more strict but easily verifiable condition [222] instead of condition 3, namely:

$$\lim_{k \to \infty} \frac{T_k^{\sup} - T_k^{\inf}}{T_k^{\inf}} = 0. \tag{3.54}$$

4 What happens if $\Sigma^-(\omega,\omega')$ is zero for all ω and ω' in condition 3 and thus R is not defined? This is the best case because it means that all *globally* optimal configurations are also *locally* optimal. That is we have no restriction on the interval $[T_k^{\inf}, T_k^{\sup}]$, thus any *local* temperature schedule satisfying conditions 1–2 is good.

3.4.1 Application to Hierarchical Markov Models

Hierarchical models usually require much more communication per pixel than monogrid ones. This is why classical annealing schemes are too slow even on a parallel machine to minimize the energy associated with such a model. However, taking benefit of the pyramidal structure of the model, we can define a MTA scheme, which consists of associating higher temperatures to higher levels, in order to be less sensitive to local minima at coarser grids (see Figure 3.8). For the cliques siting between two levels, we use either the temperature of the lower level or the higher level (but once chosen, we always keep the same level throughout the algorithm) Figure 3.9.

3.4.2 Comparison of MTA and Inhomogeneous Annealing

In Figure 3.11, we compare the inhomogeneous and MTA schedules on a noisy synthetic image using the Gibbs sampler. The energy function of the hierarchical segmentation model is defined in Section 2.5.3. In both cases, the parameters were strictly the same, the only difference is the

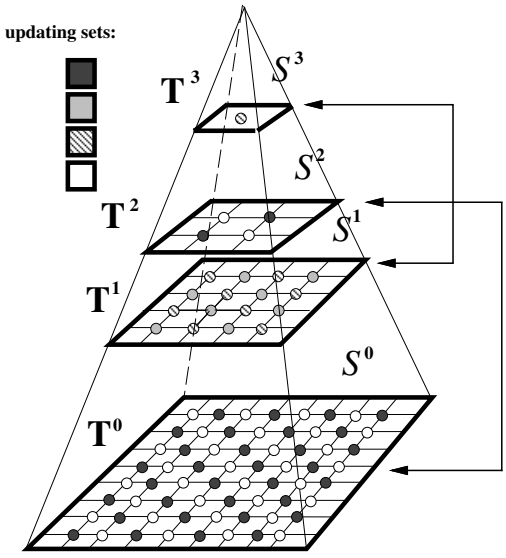


Fig. 3.8 Relaxation scheme on the pyramid $T^0 < T^1 < T^2 < T^3$.

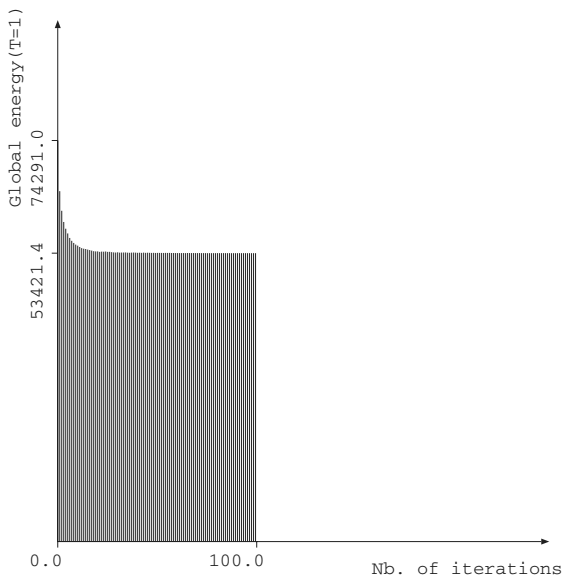


Fig. 3.9 Energy decrease with the MTA schedule.

applied schedule: the pyramid contains 4 levels and the initial temperatures were respectively 4 (at the highest level), 3, 2, and 1 (at the lowest level) for MTA and 4 at each level for inhomogeneous annealing. The potential β equals to 0.7 and γ equals to 0.1. In Figure 3.10 (resp. 3.9), we show the global energy (computed at a fixed temperature) versus the number of iterations of the inhomogeneous (resp. MTA) schedule. Both reach practically the same minimum (53415.4 for the inhomogeneous and 53421.4 for the MTA), however the inhomogeneous schedule requires 238 iterations but the MTA schedule requires only 100 iterations for the convergence.

3.5 Deterministic Relaxation

SA algorithms reach a global minimum but they require a large amount of computation. On the other hand, a global optimum is obtained only theoretically. In practice, we always implement an *approximation* of the SA and the convergence toward the global optimum is no longer guaranteed.

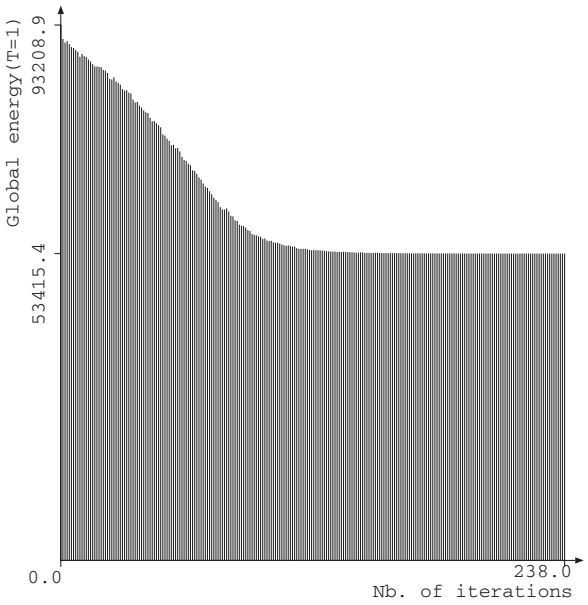


Fig. 3.10 Energy decrease with the inhomogeneous annealing schedule.

Noisy image (SNR= 3dB) Inhomogeneous schedule MTA schedule

Fig. 3.11 Results of the Gibbs sampler on a synthetic image with inhomogeneous and MTA schedules.

To speed up the convergence, many authors propose *deterministic* algorithms [15, 17, 19, 116, 117]. While the essence of every *stochastic* relaxation is that transitions with energy *increase* are permitted under certain conditions, *deterministic* relaxation allows only transitions with energy *decrease*.

Let us begin the discussion with the most popular deterministic algorithm: the *Iterated Conditional Modes* (ICM) [17].

3.5.1 Iterated Conditional Modes (ICM)

If we have a reasonably good initial configuration then an extremely rapid convergence can be obtained by the ICM method proposed by Besag in [17]. The quality of the final result strongly depends on the initialization since ICM realizes only a descent in the nearest energy-valley. Of course, the obtained minimum is only *local* but convergence toward this minimum is obtained usually in a few iterations (less than 10 in our experiments).

Algorithm 6 (ICM).

① Start with a "good" initial configuration ω^0 and set $k = 0$.

② For each configuration which differs at most in one element from the current configuration ω^k (they are denoted by $\mathcal{N}_{\omega^k}$), compute the energy $U(\eta)$ $(\eta \in \mathcal{N}_{\omega^k})$.

③ From the configurations in $\mathcal{N}_{\omega^k}$, select the one which has a minimal energy:

$$\omega^{k+1} = \arg \min_{\eta \in \mathcal{N}_{\omega^k}} U(\eta). \tag{3.55}$$

④ Goto Step ② with $k = k + 1$ until convergence is obtained (for example, the energy change is less than a certain threshold).

Notice that in the ICM algorithm there is no temperature parameter and thus there is no annealing. On the other hand, Step ③ is nothing else but the acceptance rule of the Gibbs Sampler (see Algorithm 4) at $T = 0$. Thus, the ICM algorithm corresponds to a purely deterministic "frozen" Gibbs Sampler.

The initialization method depends on the problem which we are trying to solve. For image labeling, one normally adopts the conventional *maximum likelihood* estimator which ignores spatial dependence of one pixel on the others. Other examples can be found in [17]. In practice, some heuristic combination of different methods is adopted,

for example, doing 10 to 30 iterations of ICM followed by a few iterations of HCM [42]. Another common strategy for multiresolution MRF models is the method of *embedded spaces* (sometimes also called multiscale methods), which has been explored for, for example, motion detection [172] and segmentation [105, 106].

3.5.2 Graduated Non-Convexity (GNC)

The idea behind GNC [19] is to approximate the *non-convex* energy function U by a new function $U^\star$ which is *convex* and hence can only have one minimum. In the simplest case, this minimum may also be the *global minimum* of the original function U. Generally, we need a *sequence* of functions U^p $(0 \leq p \leq 1)$ such that $U^0 = U$ and $U^1 = U^\star$. In between, U^p changes in a continuous way, between U and $U^\star$. The algorithm itself consists of minimizing the sequence U^p $(p = 1 \rightarrow p = 0)$ using the result of the previous optimization as the starting point for the next one. The main inconvenient of the method is that there is no exact formula to know how to choose a convex approximation. It depends on the original function U. Some examples can be found in [19].

Algorithm 7 (GNC).

① Define a convex approximation $U^\star$ of U. Set up a sequence of approximations U^{p_i}, $\forall i = 1 \ldots P:\ 0 \leq p_i \leq 1$ such that $U^0 = U$ and $U^1 = U^\star$. Initialize $i = 1$.

② Find the minimum $\hat{\omega}^i$ of U^{p_i} (by a direct descent or gradient descent method, for example).

③ Goto Step ② with $\hat{\omega}^i$ as the initial configuration and $i = i + 1$ until $i < P$.

3.5.3 Deterministic Pseudo Annealing (DPA)

DPA is a GNC-like algorithm proposed by M. Berthod et al. in [15]. It is also related to relaxation labeling algorithms [55, 93]. The basic idea is to extend the probability of a discrete labeling of pixels in an image to a merit function defined on continuous labelings which is a

polynomial with non-negative coefficients. Under certain constraints, the only extrema of this function is a discrete labeling. DPA consists of changing these constraints to convexify the merit function, find its global maximum, and then track down the solution until the original constraints are restored yielding an optimal discrete labeling of the original problem.

Let us consider the energy function of a discrete labeling ω:

$$U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega). \tag{3.56}$$

To obtain optimal labeling, we have to minimize this function, or equivalently, maximize the negative energy:

$$-U(\omega) = \sum_{C \in \mathcal{C}} -V_C(\omega) = \sum_{C \in \mathcal{C}} W_C(\omega). \tag{3.57}$$

It is possible to shift all W_C's so that they all become positive, without changing the solution. Now, this combinatorial optimization problem is transformed into a maximization problem in a compact subset of $\Re^{NL}$ (N is the number of sites and L is the number of elements in the common state space). Following [15], we define the following real function:

$$f(X) = \sum_{C \in \mathcal{C}} \sum_{\omega \in \Omega_C} W_C(\omega) \prod_{i=1}^{\deg(C)} x_{C_i, \omega_{C_i}}, \tag{3.58}$$

where C_i denotes the ith site of clique C and Ω_C denotes the set of all possible labelings of the sites of clique C. Indeed, f is a polynomial in $x_{i,j}$s, its degree is the maximum degree of the cliques ($\deg(\mathcal{C})$). If f is restricted to a compact subset $\mathcal{P}^{NL}$ of $\Re^{NL}$:

$$\forall i, j: \quad x_{i,j} \geq 0 \text{ and } \forall i: \quad \sum_{j=1}^{L} x_{i,j} = 1. \tag{3.59}$$

The maximum of f on $\mathcal{P}^{NL}$ is on the border:

$$\forall i, \exists j: \quad x_{i,j} = 1 \text{ and } \forall k \neq j: \quad x_{i,k} = 0. \tag{3.60}$$

Thus any maxima on $\mathcal{P}^{NL}$ directly yields a discrete labeling. To find the global maximum on $\mathcal{P}^{NL}$, DPA proceeds in the following

way: Maximize f on a subset $\mathcal{Q}^{NL,d}$:

$$\forall i, \exists j : \quad x_{i,j} = 1 \text{ and } \forall k \neq j : \quad x_{i,k} = 0 \tag{3.61}$$

on which it is concave and track down the maximum by slowly restoring the original subset $\mathcal{P}^{NL}$. As claimed in [15], one can prove that f has a unique maximum on $\mathcal{Q}^{NL,d}$. Maximization is performed by the *iterative power method* [6].

Algorithm 8 (DPA).

① Set $d = 2$ and initialize X by some X_0.

② Find $\hat{X}$ which maximizes f on $\mathcal{Q}^{NL,d}$ using the iterative power method:

$$X_{n+1} = (\nabla f(X_n))^{\frac{1}{d-1}} \quad n = 0, 1, 2, \ldots \tag{3.62}$$

③ Decrease d by some quantity and project $\hat{X}$ on the new $\mathcal{Q}^{NL,d}$. Goto Step ② with X_0 equals to the projection of $\hat{X}$ until $d > 1$.

④ For each site i, select the label j for which $x_{i,j} = 1$.

This iterative decrease of d can be compared, up to a point to a cooling schedule, or better to a Graduated Non-Convexity strategy [19].

Geometrically, Step ② simply means that at each iteration, we select on the pseudo-sphere of degree d the point where the normal is parallel to the gradient of f. This cannot be applied when $d = 1$, as claimed in [15], the procedure must stop for some d slightly larger than 1. Theoretical study of the convergence of the algorithm can be found in [6] while experimental studies showing that on real problems a very good solution is reached are presented in [15, 107].

3.5.4 Game Strategy Annealing (GSA)

Based on the game theory, Liu–Yu proposes a relaxation scheme called *Game Strategy Approach* [146, 218]. Herein, we give a slightly modified version of the original algorithm [218]. The algorithm is essentially a Metropolis algorithm with deterministic acceptance rule and modified generation mechanism depending on *local energies*.[2]

[2] The local energy can be problem-dependent. For image processing, it may be the sum of potentials over the cliques containing a site s.

Algorithm 9 (GSA).

① Choose an initial configuration ω^0 and set $k = 0$.

② For each element ω_s^k ($s \in \mathcal{S}$) of ω^k, select a state $\omega'_s \in \Lambda$ such that

$$\omega'_s = \arg \min_{\lambda \in \Lambda \setminus \{\omega_s^k\}} U_s(\lambda), \tag{3.63}$$

where $U_s(\lambda)$ is the local energy in s when s is in state λ. That is, we select at each site the state with minimal energy, *locally*.

③ Accept ω'_s at s as the new state with probability α if $U_s(\omega'_s) < U_s(\omega_s^k)$. More precisely:

$$\omega_s^{k+1} = \begin{cases} \omega'_s & \text{if } U_s(\omega'_s) < U_s(\omega_s^k) \\ & \text{and } \alpha \leq \exp(-U(\omega^k) - U(\omega^k|_{\omega_s^k = \omega'_s})) \\ \omega_s^k & \text{otherwise} \end{cases} \tag{3.64}$$

where α is constant chosen at the beginning of the algorithm.

④ Goto Step ② with $k = k + 1$ until convergence is obtained.

Notice that in the algorithm, there is no temperature parameter thus there is no annealing. However, we can use an annealing schedule in Equation (3.64). According to our experience, annealing may speed up the convergence of GSA (see Section 3.7).

3.5.5 Modified Metropolis Dynamics (MMD)

Here, we present a pseudo-stochastic variation of the Metropolis dynamics [116, 117, 133, 158]. At high temperature, the behavior of the algorithm is similar to the stochastic techniques. However, if the temperature is less than a certain threshold, it becomes deterministic. The "length" of the "pseudo-stochastic" phase is controlled by a constant threshold used in the modified dynamics. The difference between the Metropolis dynamics and our approach is the choice of ξ in Step ③ of Algorithm 3. For the original method, ξ is chosen randomly at each iteration, however for our algorithm, ξ is a constant threshold, say $\alpha \in (0, 1)$, chosen at the beginning of the algorithm. This simply means

that the jump to η is allowed if this does not increase *excessively* the energy. The threshold α controls this increasing of energy.

Algorithm 10 (MMD).

① Pick up randomly an initial configuration ω^0, with $k=0$ and $T=T_0$.

② Using a uniform distribution, pick up a global state η which differs only in one element from ω^k.

③ **(Modified Metropolis Dynamics)** Compute $\Delta U = U(\eta) - U(\omega)$ and accept η according to the following rule:

$$\omega^{k+1} = \begin{cases} \eta & \text{if } \Delta U \leq 0, \\ \eta & \text{if } \Delta U > 0 \text{ and } \ln(\alpha) \leq \left(-\frac{\Delta U}{T}\right), \\ \omega^k & \text{otherwise} \end{cases} \tag{3.65}$$

where α is a constant threshold ($\alpha \in (0,1)$), chosen at the beginning of the algorithm.

④ Decrease the temperature $T = T_{k+1}$ and goto Step ② until convergence is obtained (ΔU less than a certain threshold, for example).

The MMD algorithm is much faster than the original Metropolis as we will see in Section 3.7. Because for MMD, in Step ②, we have only to compute $\Delta U/T$ and compare it to $\ln(\alpha)$ which is a constant computed at the beginning of the algorithm. However for the original Metropolis dynamics, we have to compute $\exp(-\Delta U/T)$ at each iteration since it is compared to a random value which is not constant. The initialization is not as crucial as for the ICM algorithm because the *pseudo-stochastic phase* results in a good initialization for the *deterministic phase* of the MMD. There is no explicit formula to get the threshold α. In practice, α is determined by an ad-hoc way depending on the landscape of the energy function. If it is smooth enough, a shorter "stochastic" phase is sufficient thus α is chosen nearly equal to one.

The following theorem provides a more precise characterization of these phases.

Theorem 3.3 (MMD). For any $\alpha \in (0,1)$, there exists a temperature threshold

$$T_\alpha = -\frac{\Delta U_{min}}{\ln(\alpha)} \tag{3.66}$$

$$\text{where } \Delta U_{min} = \min_{\substack{\omega,\eta \in \Omega \\ U(\omega) \neq U(\eta)}} \mid U(\omega) - U(\eta) \mid \tag{3.67}$$

such that if $T_k < T_\alpha$ then only configurations with lower energy will be accepted and thus the algorithm converges towards a local minimum.

If $T_k = \Gamma / \ln(k)$ then $T_k < T_\alpha$ if and only if $k > K_\alpha$ where K_α is a threshold given by:

$$K_\alpha = \exp\left(-\frac{\Gamma \cdot \ln(\alpha)}{\Delta U_{\min}}\right). \tag{3.68}$$

In other words, after K_α iterations, the MMD algorithm enters the *deterministic phase* accepting only configurations with an energy decrease.

3.6 Parallelization Techniques

In the previous section, we have discussed a variety of deterministic algorithms proposed by different authors as an alternative to minimize non-convex functions. In this section, we deal with parallelization techniques adapted for stochastic as well as for deterministic methods. These general methods are useful for GPU implementations of MRF segmentation algorithms.

3.6.1 Data Parallelism

As a natural parallelization method in a lot of image processing problem, we have already mentioned the coding scheme [17] in Section 3.2.1.2. It consists of constructing *coding sets* such that pixels belonging to the same set are conditionally independent, thus they can be updated at the same time (see Figure 3.6 for an example). The main advantage of this technique is that it does not violate the convergence.

Another interesting method has been proposed by Azencott in [3]: At each iteration k, each site belongs to the active set A_k with probability τ ($\tau \in (0,1]$ is fixed). The updating is then carried out simultaneously at the active sites in A_k. By convention, $\tau = 0$ denotes the *sequential* algorithm and $\tau = 1$ corresponds to the fully parallel scheme where *all* sites are updated at the same time. For $\tau \in (0,1)$, Trouvé has proved in [201] that using an appropriate cooling schedule, the generated Markov chain converges:

$$\lim_{k\to\infty} P(X_k = \omega) = \pi_\tau(\omega). \tag{3.69}$$

The main result of Trouvé is that partially parallelized algorithms are *equivalent* with respect to their limiting distribution:

$$\forall \tau : 0 < \tau < 1 : \quad \pi_\tau \equiv \Pi. \tag{3.70}$$

However, it is *not* shown whether $\Pi \equiv \pi_0$ (π_0 denotes the stationary distribution of the sequential annealing which is known to coincide with the uniform distribution over the optimal configurations) but the experimental study in [59] seems to confirm this equivalence. On the other hand, many examples can be constructed where π_1 is *not* equivalent to π_0.

The parallelization schemes described here assume that the configuration space can be partitioned (in image processing, it is usually true). On the other hand, the optimization algorithm itself is still sequential since transitions are carried out one after the other which is typically a sequential process. We have only changed the generation mechanism. In the followings, we will study parallel implementations where Markov chains are generated simultaneously.

3.6.2 Parallel Simulated Annealing

Essentially, there are two approaches [4, 137]. The first one, called *systolic algorithm*, aims at generating multiple Markov chains with possible interactions between them. In the other approach, called *clustered algorithm*, all processors are used to generate cooperatively the same Markov chain.

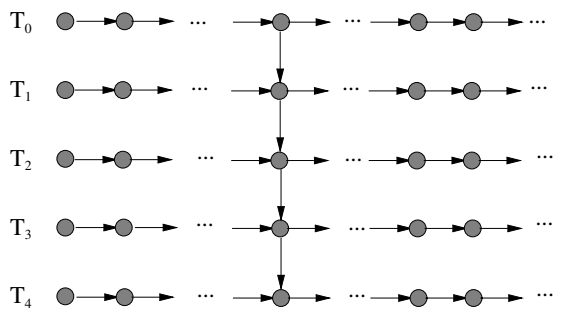


Fig. 3.12 Systolic parallelization scheme.

3.6.2.1 Systolic Algorithm

The basic idea is to generate n Markov chains simultaneously [4, 71, 137] (cf. Figure 3.12). Obviously, if the chains are independent and the temperature T is fixed (in Figure 3.12, $T_0 = T_1 = \cdots = T_4$), they generate the *same* Markov chain. According to the convergence results presented in [137], after an infinite number of iterations we obtain n realizations of the same chain. One possibility is to select the optimum among the n realizations. In [72], it has been shown that if the stationary distribution of the sequential homogeneous algorithm is denoted by $q(T)$ then, using n processors, the stationary distribution of the above defined parallel homogeneous algorithm is given by $q(T/n)$. Since the convergence rate of SA increases exponentially when T goes to 0, it is obvious that using n processors may considerably speed up the convergence.

A more general scheme is to allow communications between the processors at regular time intervals. The interactions may consist of selecting the best configuration among the n configurations or one can use a probabilistic rule (the acceptance rule of the Gibbs Sampler in Equation (3.16), for example). Surprisingly, this scheme is *asymptotically less efficient* than the independent scheme if each processor is executing *the same* annealing algorithm. As claimed in [3], performing interactions between n processors executing the same annealing is only a waste of computing time. It might as well be replaced by a single interaction at the end to select the best final configuration. The mathematical study

of this method is provided in [5, 71] and the experimental results can be found in [71, 72]. We mention here a conjecture from [72]:

Conjecture 3.6.1 (Graffigne). Consider n processors generating the same homogeneous Markov chain at temperature T interacting after $l, 2l, 3l \ldots$ iterations. The resulting stochastic process is a *nonhomogeneous* Markov chain $X_k = (X_k^1, X_k^2, \ldots, X_k^n)$. If l is sufficiently large to allow reaching each configuration in maximum l transitions then the asymptotic law of $X^n(T)$ is close to $q(T/n)$. This means that there is no need to add extra interactions before the last one.

Finally, let us discuss a more interesting scheme. The approach is exactly the same as before but the temperature now varies for each processor. The processors generate different Markov chains at different temperatures. Usually, the first processor is set to a high temperature, the last processor is set to a temperature close to 0 and the other processors are set to intermediate temperatures uniformly distributed between the highest and lowest ones (cf. Figure 3.12 with $T_0 > T_1 > \cdots > T_4$). The behavior of the algorithm is as follows: The first processor, at a high temperature, will randomly explore the energy landscape with large moves. The lower temperatures allows to investigate a selected energy-valley and the last processor, with temperature close to zero, will find the local minima. The convergence of the algorithm has not been proved but the experimental results presented in [72] are very promising.

We have used the idea of performing relaxation at different temperatures in our Multi-Temperature Annealing schedule (see Section 3.4) but, in our case, the convergence has been proved [106].

3.6.2.2 Clustered Algorithm

Here, all processors are used to carry out a transition of the same Markov chain (see Figure 3.13). Then the new configuration will be selected among the n configurations according to a deterministic or probabilistic rule as in the previous section. In [32], a detailed mathematical study is provided confirming the convergence of the method.

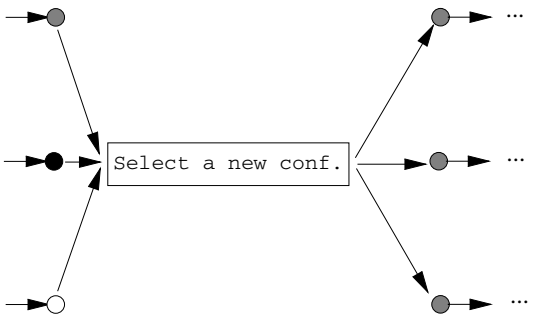


Fig. 3.13 Clustered parallelization scheme.

3.6.3 Parallel Multiscale Algorithms

Herein, we present some parallelization schemes for the multiscale relaxation algorithm described in Algorithm 2. The most simple method is to use a *data parallel* relaxation algorithm (see Section 3.6.1) at each level. In this case, the convergence of SA toward a global minimum is guaranteed. However, the levels are handled sequentially. Full parallelization is not a trivial task since the algorithm is intrinsically sequential (see Figure 2.5): we need the result of the coarser level to initialize the level below it. Many heuristics have been proposed to introduce additional parallelism in the pyramid. The common problem of these methods is that convergence is no longer guaranteed. Nevertheless, experimental results seem to give a reasonably support of the convergence.

In [88, 150] a parallel inter-level strategy has been proposed, similar to Graffigne's parallelized Markov chain approach [71, 72]. The algorithm consists of running a (possibly data parallel) relaxation algorithm at each level of the pyramid with different initial temperatures (the highest temperature is assigned to the coarsest grid). In regular time intervals, the coarse grids transmit a small block of labels (an *interaction block*) to the level below it. The block is accepted at the finer level if its energy is lower. The energies can be compared directly (after projection of the interaction block into the finer grid) due to the consistent definition of the energy functions at higher levels (they are all related to the energy function of the finest level as explained in Section 2.4).

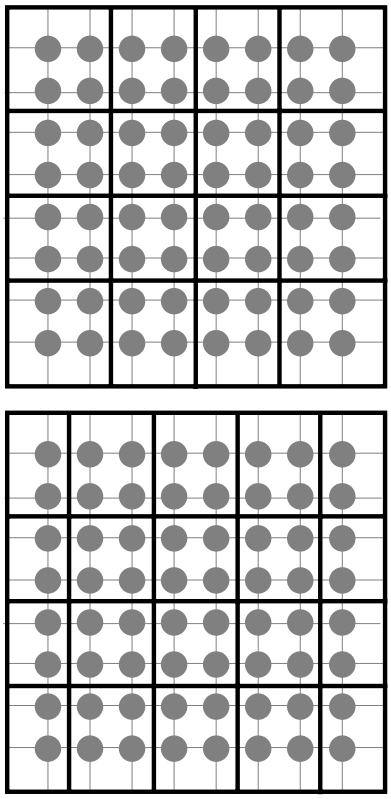


Fig. 3.14 Multiple partitioning in a multiscale model.

Another scheme has been proposed in [151]. Considering the partition of the original grid $\mathcal{S}$ reported in Algorithm 2, we may notice that the block partitioning is not unique at a given resolution. As shown in Figure 3.14, partitions can be obtained by considering successive shifts of the initial block partition along the horizontal and vertical directions (we have exactly $(wh)^l$ different partitionings at level l, as pointed out in [151]). Taking benefit of these different partitionings, we can define a parallel scheme: For each partitioning at a given level, we associate a relaxation algorithm. Thus we obtain $(wh)^l$ algorithm running in parallel. At the convergence, the configuration of lowest energy is selected among the $(wh)^l$ results.

For the ICM algorithm [17] (see Algorithm 6), a *multi-initialization* method has been proposed in [151]. It consists of running multiple ICM algorithm at coarser levels with different initial configurations. The next level is initialized with the configuration of minimal energy.

3.7 Experimental Results

The goal of this simulation is to evaluate the performances of the algorithms described in this chapter, in particular on image segmentation problems. We remark that in all cases, the execution has been stopped when the energy change ΔU was less than 0.1% of the current value of U. A demo implementation of these algorithms using a monogrid MRF segmentation model can be found at http://dx.doi.org/10.1561/2000000035_demogray (in gray scale) and http://dx.doi.org/10.1561/2000000035_democolor (in color). In general, stochastic schemes are better regarding the achieved minimum and deterministic algorithms are better regarding the computer time, but the final segmentation quality is also influenced by the MRF model choice.

3.7.1 MRF Models

First, we compare the Gibbs sampler [64] and Iterated Conditional Mode [17, 97] using three models for each algorithm (monogrid, multiscale, and hierarchical). In all cases, the execution has been stopped when the energy change ΔU was less than 0.1% of the current value of U.

In Figure 3.15, we studied the geometrical sensitivity of the models. While the Gibbs sampler gives nearly the same result in all cases, ICM is more sensitive to initial conditions. The multiscale model gives better result than the monogrid one but the fine details are lost in the triangle and the circle. These forms have a different structure from the block structure of the model, the initialization was wrong in these regions and the ICM was not able to correct these errors. In the hierarchical case, we have a real time communication between the levels which is able to give results close to the ones obtained with the Gibbs sampler. Of course, this model requires more computing time than the other ones.

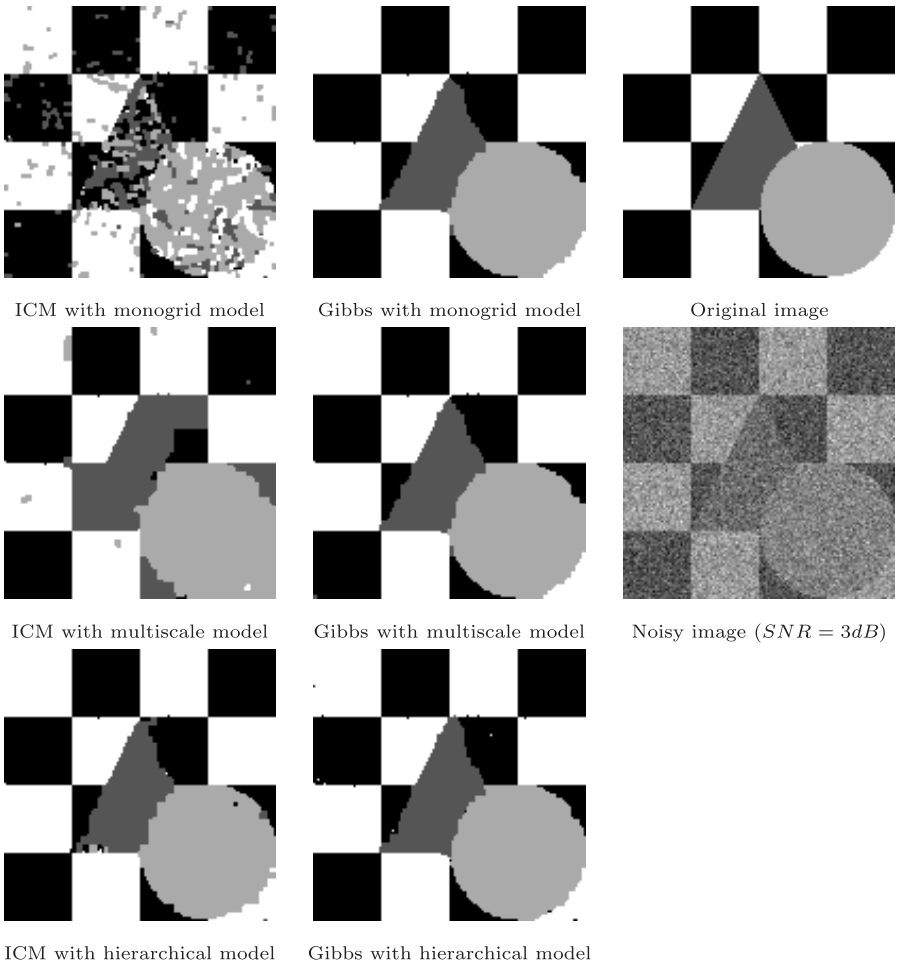


Fig. 3.15 Results obtained by various MRF models on the "triangle" image with 4 classes.

In Figures 3.16 and 3.17, we show some real images of size 256×256: a SPOT image with 4 classes and a microscopic image with 3 classes.

3.7.2 Stochastic and Deterministic Relaxation Algorithms

Herein, we present tests on a variety of images using the algorithms described in this chapter. For the simulations, we have used a first order

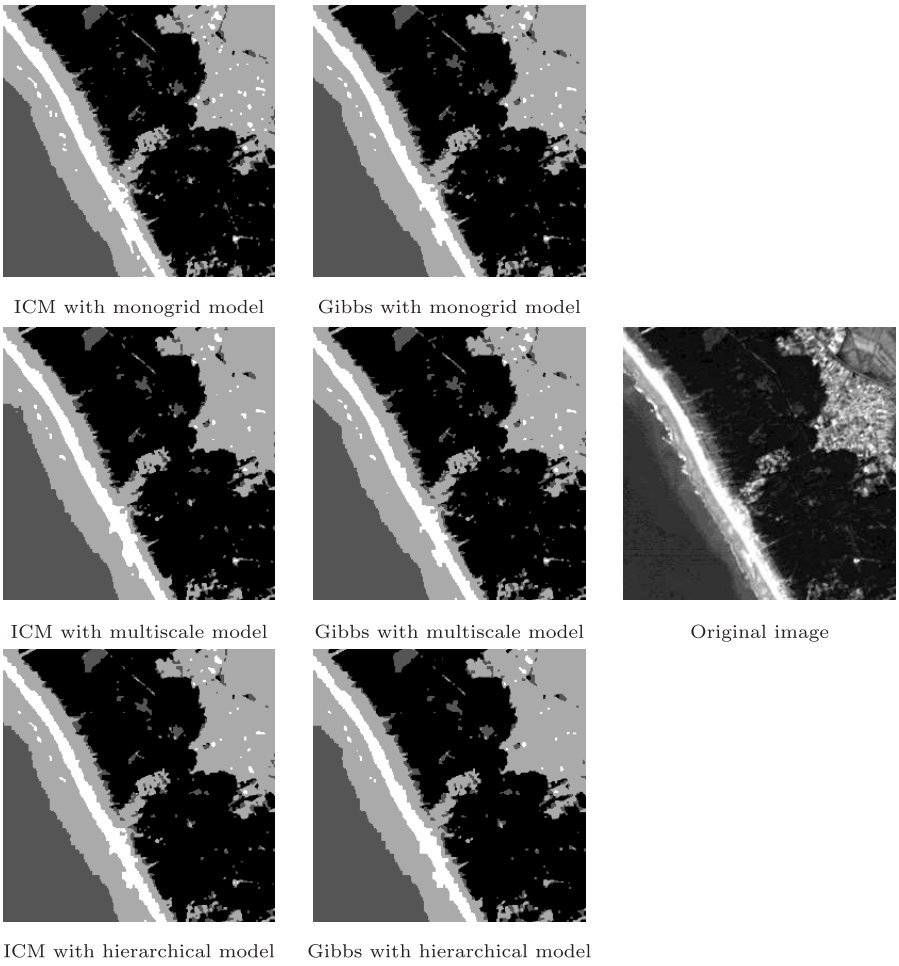


Fig. 3.16 Results obtained by various MRF models on the "SPOT" image with 4 classes.

MRF image model with the following energy function (see Section 2.2 for more details):

$$U(\omega, f) = \sum_{s \in \mathcal{S}} \left(\ln(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \right) + \sum_{\{s,r\} \in \mathcal{C}} \beta \delta(\omega_s, \omega_r) \tag{3.71}$$

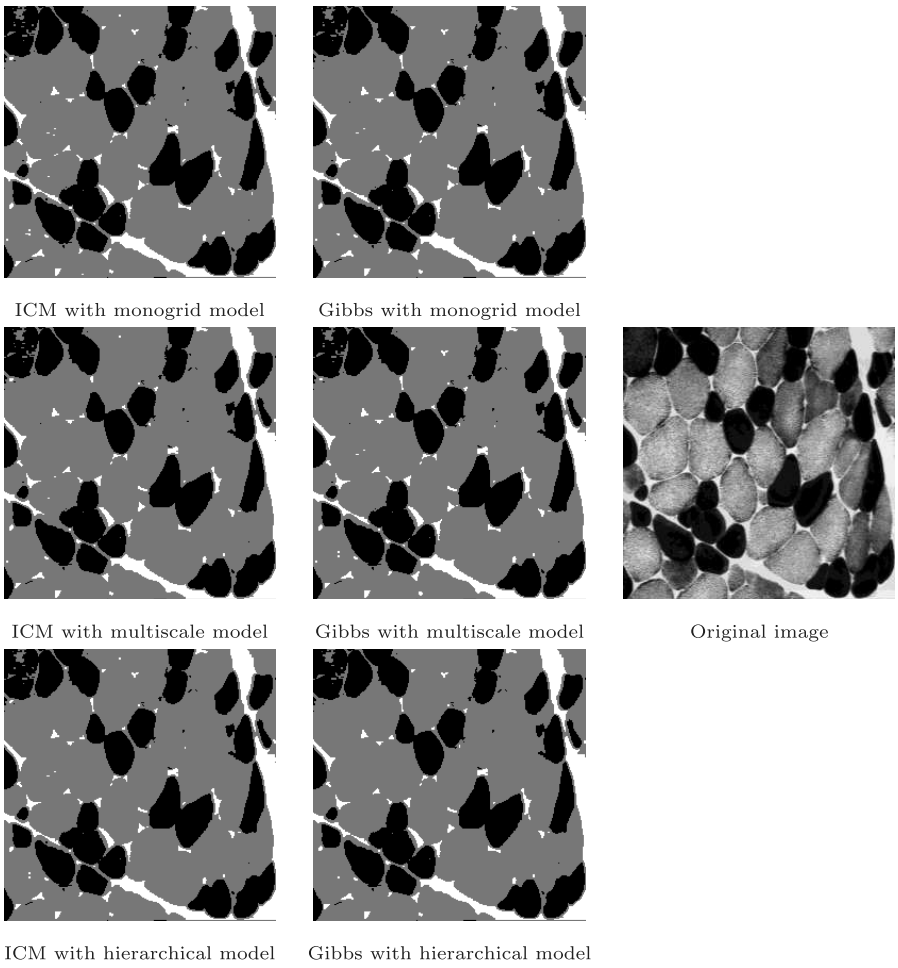


Fig. 3.17 Results obtained by various MRF models on a microscopic image ("muscle") with 3 classes.

where

$$\delta(\omega_s, \omega_r) = \begin{cases} -1 & \text{if } \omega_s = \omega_r \\ +1 & \text{if } \omega_s \neq \omega_r \end{cases} \tag{3.72}$$

and β is a model parameter controlling the homogeneity of the regions. Each class $\lambda \in \Lambda$ is represented by its mean value μ_λ and its deviation

σ_λ. $\omega_s \in \Lambda$ denotes the label attributed to the pixel $s \in \mathcal{S}$ and f_s stands for the gray-level value at pixel s. The model parameters are supposed to be known.

The initial temperature for the algorithms using annealing (that is Gibbs Sampler, Metropolis, MMD, GSA) was $T_0 = 4$ and the schedule is given by $T_{k+1} = 0.95 \cdot T_k$. ICM and DPA was initialized by using only the Gaussian term of the energy function (this means the *maximum likelihood* estimate of the labels). As for the other methods, *random* initial values were assigned to the labels. Since ICM is very sensitive to the initial conditions, better results could have been obtained with

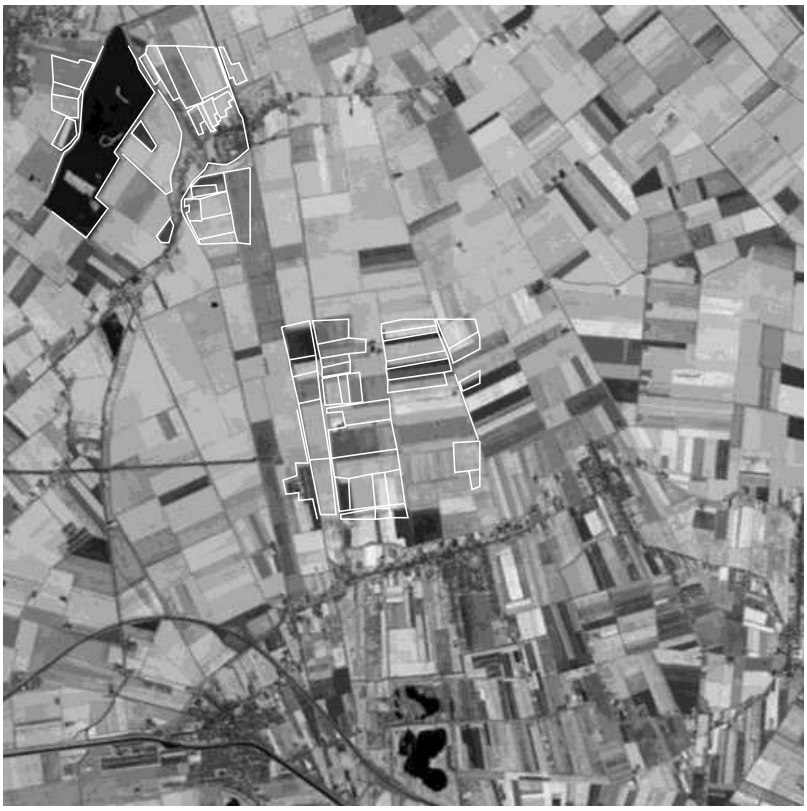


Fig. 3.18 Original SPOT image "holland" (©CNES).

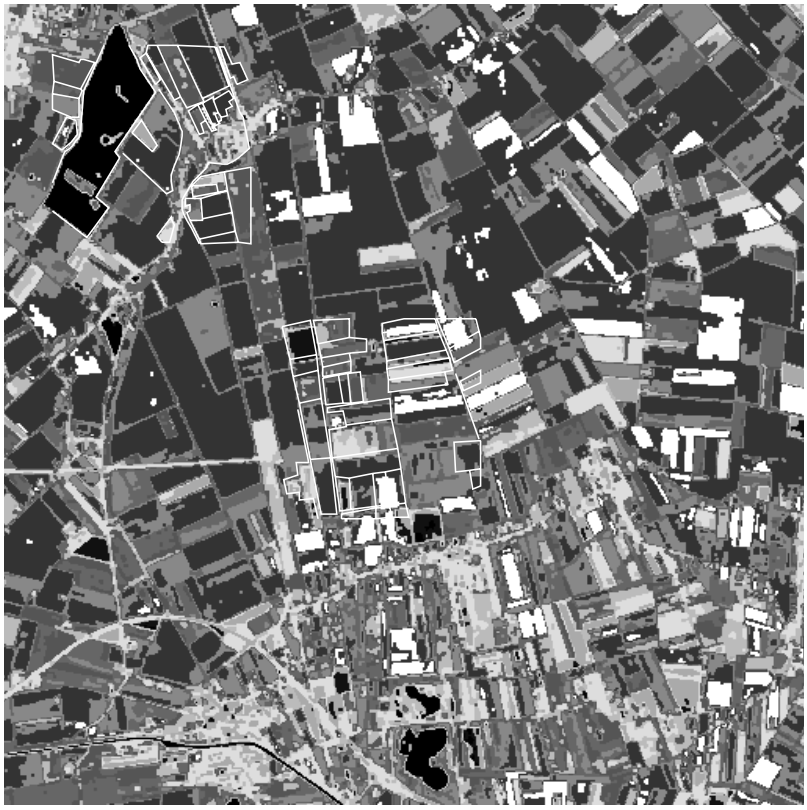


Fig. 3.19 Monogrid segmentation result with 10 classes (ICM).

another initialization. Nevertheless the DPA and ICM algorithms have been initialized with the same data for the simulation (Figures 3.18–3.21).

The obtained results are presented in Figures 3.22 and 3.23. As these results show, stochastic methods give the lowest energy values but they are slower than deterministic methods. ICM is the fastest but the reached minimum is much higher than for the other methods (as mentioned earlier, another initialization may lead to a better result, but more elaborated initialization usually increases the computer time). DPA, MMD, and GSA seem to be a good compromise

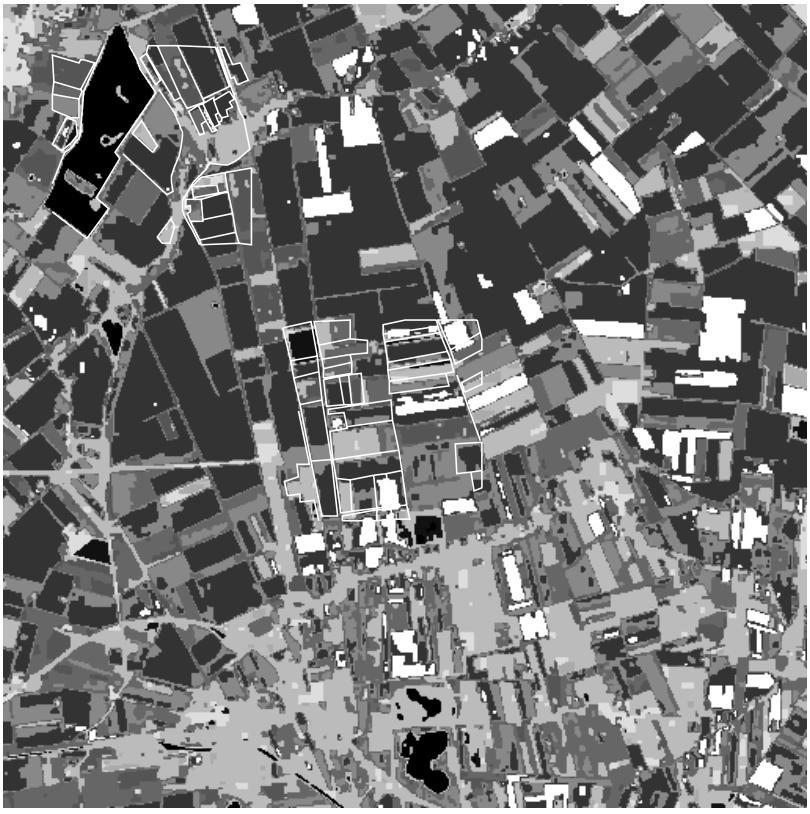


Fig. 3.20 Multiscale segmentation result with 10 classes (ICM).

between quality and execution time. Sometimes, the results obtained by these algorithms are very close to the ones of stochastic methods. Another advantage is that they are far less dependent on the initialization than ICM.

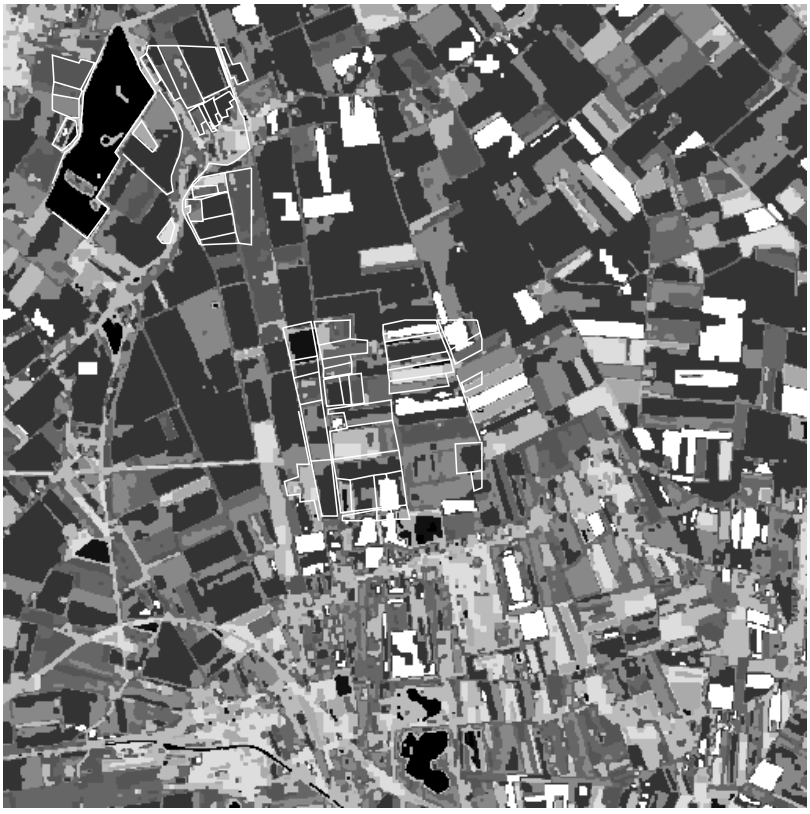


Fig. 3.21 Hierarchical segmentation result with 10 classes (ICM).

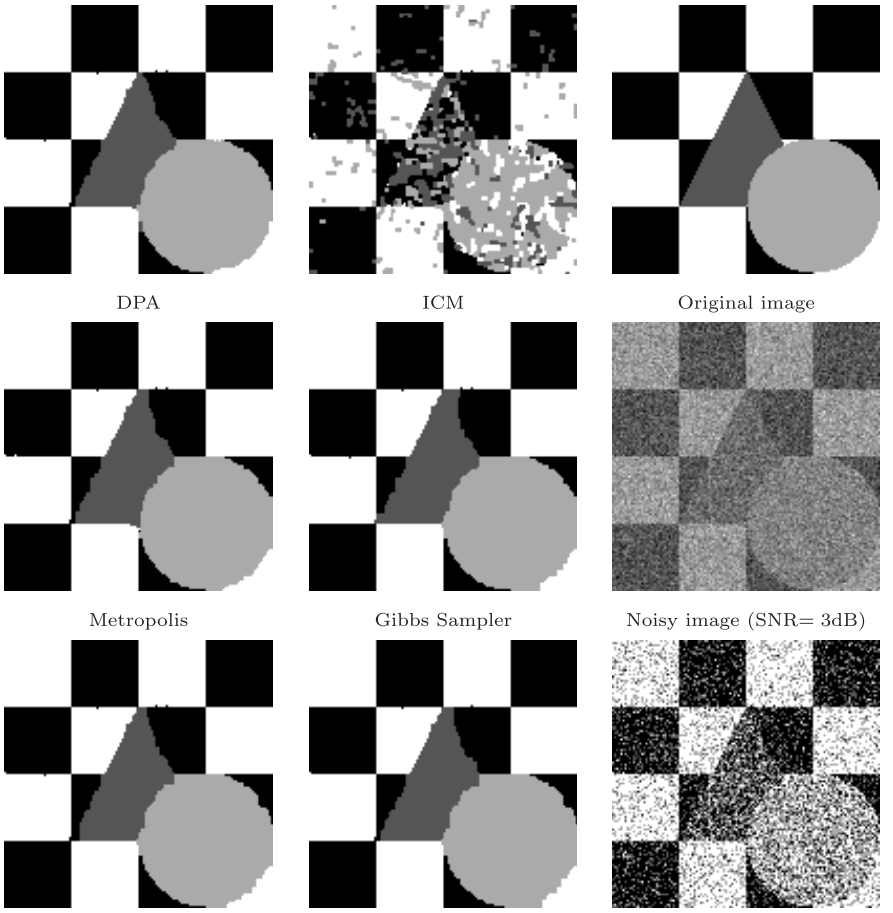


Fig. 3.22 Results obtained by various energy minimization algorithms on the "triangle" image with 4 classes.

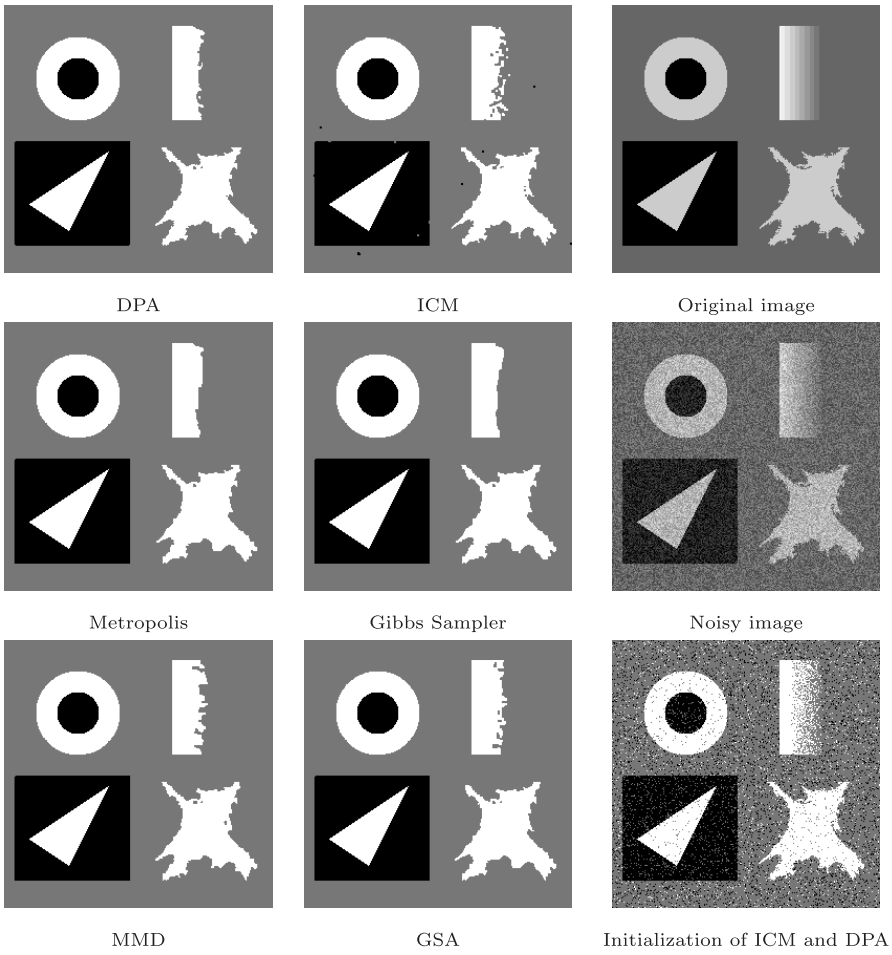


Fig. 3.23 Results obtained by various energy minimization algorithms on the "bruit" image with 3 classes.

4

Graph Cut

Markov random fields provide a powerful tool to construct segmentation models of degraded images yielding an energy minimization problem. Unfortunately, the exact minimization of a general energy function is NP-hard, requiring iterative algorithms [64, 137, 154], which is a major obstacle for adopting MRF models in interactive segmentation. Recently, combinatorial graph cut algorithms have been succesfully applied to a wide range of energy functions in image processing [25, 26, 27, 28, 132, 130, 131, 185, 199, 207]. It has been shown that a certain class of energy functions could be exactly minimized by graph cuts in *polynomial time*. This discovery revolutioned the use of Markov random fields not only in image processing and computer vision but also in computer graphics. Herein, we will overview the now classical graph cut algorithms and show an example MRF segmentation model constructed with a graph-representable energy function.

4.1 Exact MAP of Binary MRFs via Standard Maxflow/Mincut

Herein, we will focus on a special subclass of MRF models with binary labels $\omega_s \in \{0,1\}$ and the following energy function:

$$U(\omega) = \sum_s V_s(\omega_s) + \sum_{s \sim r} V_{s,r}(\omega_s, \omega_r), \tag{4.1}$$

where $V_s(\omega_s)$ and $V_{s,r}(\omega_s,\omega_r)$ are *singleton* and *doubleton* potentials, respectively (see Section 2.2).

Now let us consider the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nonnegative edge weights and two special vertices (called *terminals*): the source $\vec{\boldsymbol{s}}$ and the sink $\vec{\boldsymbol{t}}$. An *s-t*-cut on $\mathcal{G}$ corresponds to a binary partitioning S,T of the vertices such that $\vec{\boldsymbol{s}} \in S$ and $\vec{\boldsymbol{t}} \in T$, which can be described by the binary variables $\omega_s, s \in \mathcal{S}$. The cost of a cut $c(S,T)$ is the sum of edge costs $c(u,v)$ going from S to T:

$$c(S,T) = \sum_{\{(u,v) \in \mathcal{E} : u \in S, v \in T\}} c(u,v). \tag{4.2}$$

In order to minimize U via graph cuts, we have to create a graph *representing* U such that a minimum cut on the graph also minimizes the energy U.

Since a cut can also be described by the binary variables x_i corresponding to the vertices in $\mathcal{G}$ (excluding the two special ones) such that $x_i = 0$ if $v_i \in S$ and $x_i = 1$ when $v_i \in T$, the energy represented by $\mathcal{G}$ can be regarded as a function of N binary variables $E(x_1, x_2, \ldots, x_N)$. Of course, the configuration minimizing E will not change if we add a constant to E. Now the minimum of E (and the corresponding configurations of the binary variables x_i) can efficiently be obtained by computing the minimum *s-t*-cut on $\mathcal{G}$. Following [132], we can summarize the graph construction as follows:

Definition 4.1. A function $E(x_1, x_2, \ldots, x_N)$ of N binary variables is called *graph-representable*, if there exists a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with terminals $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{t}}$ and a subset of vertices $\mathcal{V}_0 = \{v_1, \ldots, v_N\} \subset \mathcal{V} - \{\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}}\}$ such that for any configuration $x_1 \ldots, x_N$, the energy $E(x_1, \ldots, x_N)$ is equal to a constant plus the minimum *s-t*-cut in which $v_i \in S$ if $x_i = 0$

and $v_i \in T$ otherwise ($1 \leq i \leq N$). If this constant is zero then E is *exactly represented* by $\mathcal{G}$ and subset $\mathcal{V}_0$.

The following lemma is an obvious consequence of the above definition [132]:

Lemma 4.2. If E is graph representable by $\mathcal{G}$ and subset $\mathcal{V}_0$, then it is possible to find the exact minimum of E in polynomial time by computing the minimum s-t-cut on $\mathcal{G}$.

The main result of graph representable energy functions has been summarized in [132]:

Theorem 4.3. Let $E(x_1, \ldots, x_N)$ be a function of N binary variables:

$$E(x_1, \ldots, x_N) = \sum_i E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j). \tag{4.3}$$

Then E is graph-representable if and only if each pairwise term $E^{i,j}$ satisfies the inequality

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0). \tag{4.4}$$

Functions satisfying Equation (4.4) are called *regular* in [132], but they are often called *submodular* functions. Although Equation (4.4) is a sufficient but not necessary condition for submodularity, the term *submodular* has been widely accepted. Based on the above theorem, we will now show how to construct a graph for E. First of all, we refer to the additivity theorem [132], which states that the sum of two graph-representable functions is also graph representable. Therefore, following [132], we restrict our presentation to the construction of a graph for each term in Equation (4.3). The whole graph is then obtained by merging these components. Each nonterminal vertex v_i corresponds to a binary variable x_i, thus the graph has $N + 2$ vertices. Furthermore, one or more edges will be added for each term of E as follows:

First, consider an unary term E^i. If $E^i(0) < E^i(1)$, then the edge $(\vec{s}, v_i)$ is added with the weight $E^i(1) - E^i(0)$. Otherwise, the edge

$(v_i, \vec{\boldsymbol{t}})$ is added with the weight $-(E^i(1) - E^i(0)) = E^i(0) - E^i(1)$. This guarantees positive edge weights and yields to a representation of E^i with different constants: $E^i(0)$ in the former case and $E^i(1)$ in the latter one.

Let us now consider a pairwise term $E^{i,j}$, which depends on x_i and x_j. Denoting the possible energy values by $A = E^{i,j}(0,0)$, $B = E^{i,j}(0,1)$, $C = E^{i,j}(1,0)$, and $D = E^{i,j}(1,1)$, we can rewrite these values according to the following table:

A	B
C	D

$= A +$

0	0
C-A	C-A

$+$

0	D-C
0	D-C

$+$

0	B+C-A-D
0	0

The first term A is constant, hence it is not represented in the graph. The next two terms depends on one single variable (x_i and x_j respectively), hence they are treated in the same way as the unary terms. Finally the last term is represented by an edge (v_i, v_j) with weight $B + C - A - D$. Note that this is a nonnegative weight due to the *submodular* property of Equation (4.4).

4.2 Solving Multilabel and Higher Order MRFs via GraphCut

There are many important classes of MRF models which do not satisfy all conditions discussed in the previous section. Herein, we will briefly overview the current graph based solutions for such models.

4.2.1 Multilabel MRF Models

The generalization of graph cut solutions to more than two labels has been studied by many researchers (see for example [29, 94]). In [94], it is shown that a first oder MRF energy with multiple labels can be *exactly* minimized when labels are linearly ordered and the prior term is convex. The method maps the problem into a minimum-cut problem for a directed graph, for which a globally optimal solution can be found in polynomial time. The convexity of the prior function in the energy is shown to be necessary and sufficient for the applicability of the method. The graph construction involves auxiliary vertices representing various label assignments of each site in the MRF model, and edges also include

so called contraint edges to guarantee that exactly one label is assigned to each site.

In [29], an approximate solution is proposed for a more general class of prior terms, which includes the Potts model and many important discontinuity preserving energy functions. While it is NP-hard to compute the exact minimum of these energy functions, the iterative graph cuts proposed in [29] achieve approximate solutions to this NP hard minimization problem with guaranteed optimality bounds. These algorithms approximately minimize the energy for an arbitrary finite set of labels under two fairly general classes of interaction potential: metric and semi-metric. A potential V is called semi-metric over the set of labels Λ, if for any pair of labels $\alpha, \beta \in \Lambda$, it satisfies two properties: $V(\alpha, \beta) = V(\beta, \alpha) \geq 0$ and $V(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$. If V also satisfies the triangle inequality $V(\alpha, \beta) \leq V(\alpha, \gamma) + V(\gamma, \beta)$ for any $\alpha, \beta \in \Lambda$, then V is called a metric. The first algorithm proposed in [29] is based on α–β-swap moves and works for any semi-metric: for a pair of labels α, β, this move exchanges the labels between an arbitrary set of pixels labeled α and another arbitrary set labeled β. The iterative algorithm stops when there is no remaining swap move that decreases the energy. The second algorithm is based on α-expansion: for a label α, this move assigns the label α to an arbitrary set of pixels. Note that this algorithm requires the smoothness term to be a metric. The iterative algorithm stops when there is no expansion move that decreases the energy. Although both method find a local minimum of the energy function, it is shown in [29] that α-expansion moves produce a solution within a known factor of the global minimum.

4.2.2 Higher Order and Non-submodular Cliques

In [132], it has been shown that, by extending the notion of submodularity to higher order cliques, an exact minimum can be found for energy functions of binary labels with cliques up to order 3 for submodular clique potentials. Unfortunately, the extension of the graph construction proposed in [132] to higher order cliques is far from trivial, while in many areas of computer vision higher order cliques are necessary (for example, modeling textures or natural image statistics). Another

issue is the restriction of graph cut algorithms to submodular energy functions.

Recent advances in minimizing non-submodular energy functions via graph cut [131] are usually based on the innovation of Boros, Hammer, and their co-authors [21, 82]. A standard algorithm for minimizing first order non-submodular functions is variously called QPBO [131] or roof-duality [185]. QPBO returns a partial solution by assigning either 0, 1, or -1 to each site, where -1 means an unlabeled value while sites assigned 0 or 1 are guaranteed to get the same label as it would be in a globally minimum labeling. There are various techniques to iteratively reduce the set of unlabeled sites [131] and eventually obtain a fully labeled solution.

Recently, there have been some promising attempts [95, 127] to minimize general higher order energies via graph cut. In [95], higher-order energies of binary variables are reduced to first-order ones and minimized via QPBO. The reduction algorithm can also be used for higher-order multilabel problems where the optimization is performed by a combination of fusion-move and QPBO algorithms [95]. In [127], Kohli et al. characterized a class of higher order clique potentials for which the optimal expansion and swap moves can be computed in polynomial time. They introduced the $\mathcal{P}^n$ Potts model family of clique potentials and showed that the optimal moves for it can be solved using graph cuts.

4.3 An Example: Interactive Segmentation of Fluorescent Microscopic Images

Image segmentation in biomedical imaging is aiming to find boundaries of various biological structures such as cells, chromosomes, genes, proteins and other sub-cellular components [40, 177, 187]. Due to the highly complex structures, semi-automatic (or interactive) methods allowing for a minimal user interaction are preferable as the identification of foreground regions requires expert knowledge. Classical solutions, for example, *Cellprofiler* [100], adopt either global or adaptive thresholding followed by a watershed method for separating adjacent regions. Fluorescence microscopy is a low light imaging technique

broadly used in live cell experiments. Segmentation of such images require sophisticated methods as this imaging technique is producing noisy, blurred and low contrast images.

Herein, we present an interactive segmentation algorithm in which a user indicates (for example, by free-hand painting) an initial set of pixels as foreground and background [143]. The method uses this input, along with a set of gradient vectors, to initialize a MRF. The optimal foreground/background assignment is then obtained via graph cut [27]. The minimal cost for the underlying MRF can be found in real time thus allowing interactive adjustments by adding additional patches of foreground or background. The method efficiently uses the full gradient information (that is, both magnitude and direction) in the MRF model without compromising the ability to find an exact solution via graph cut. This method has been validated on both synthetic and real microscopic images. Comparative results with classical MRF models confirmed the increased segmentation accuracy of this approach.

4.3.1 MRF Segmentation Model

Herein, we consider 8-neighborhood cliques on the image lattice $\mathcal{S}$, giving rise to cliques up to order 4. However, only pairwise interactions are considered in order to ensure that the Gibbs energy can be minimized via standard max-flow/min-cut [27, 132].

In our case (see Figure 4.3), the background/foreground gray-level distributions can be easily modeled as Gaussians with parameters $(\mu_\lambda, \sigma_\lambda), \lambda \in \{0,1\}$. In order to ensure object coherence, $P(\omega)$ is usually chosen to be the *Ising* prior consisting of pairwise clique potentials

$$\forall (s,r) \in \mathcal{C} : \beta\delta(\omega_s, \omega_r), \quad \beta > 0 \tag{4.5}$$

with $\delta(\omega_s, \omega_r) = -1$ for homogeneous and $+1$ for inhomogeneous arguments. Indeed, this prior will enforce homogeneity *everywhere*. A more efficient prior would be to encourage coherence only where intensity gradient is low. The idea of taking into account intensity edges has appeared as early as in [64], while recently, in the context of graph cut, a contrast-sensitive Gaussian mixture MRF model has been proposed in [18]. However, [64] defines a separate *line process* with higher order

interactions which are difficult to handle in a graph-cut framework. On the other hand, [18] uses a so called *contrast* term in the data likelihood, which is related to the squared intensity difference between interacting pixel pairs but ignores gradient direction.

In contrast to previous approaches, we propose to exploit the full gradient information (magnitude and direction) while keeping the ability to find an exact MAP solution via standard max-flow/ min-cut. Obviously, the prior cannot depend on the data, hence we have to include the additional gradient terms in our data likelihood. Given the gradient vector field $\nabla\mathcal{F}$ with normalized magnitudes $|\nabla\mathcal{F}(s)| \in [0,1]$ and quantized edge directions $\vartheta(s) \in \{0°, 45°, 90°, 135°\}$ perpendicular to the gradient direction, we define the gradient strength $M(s,r)$ and edge direction $\Theta(s,r)$ for all doubletons $(s,r) \in \mathcal{C}$ as

$$\begin{aligned} M(s,r) &= \min\{M_{\max}, \\ &\quad -\min\{\log(1-|\nabla\mathcal{F}(s)|), \log(1-|\nabla\mathcal{F}(r)|)\}\} \\ \Theta(s,r) &= \begin{cases} \vartheta(s) & \text{if } |\nabla\mathcal{F}(s)| > |\nabla\mathcal{F}(r)| \\ \vartheta(r) & \text{otherwise} \end{cases} \end{aligned} \tag{4.6}$$

where $M_{\max}$ is the maximum allowed value for $M(s,r)$ (that is, we clip $M(s,r)$ at $M_{\max}$). Furthermore, we define an indicator function

$$F(s,r) = H((\mu_{\omega_s} - \mu_{\omega_r})(f_s + f_j - f_r - f_i)), \tag{4.7}$$

where H is the Heaviside function and the location of sites j and i is shown in Figure 4.1. Clearly, F will return 0 whenever the labels ω_s and

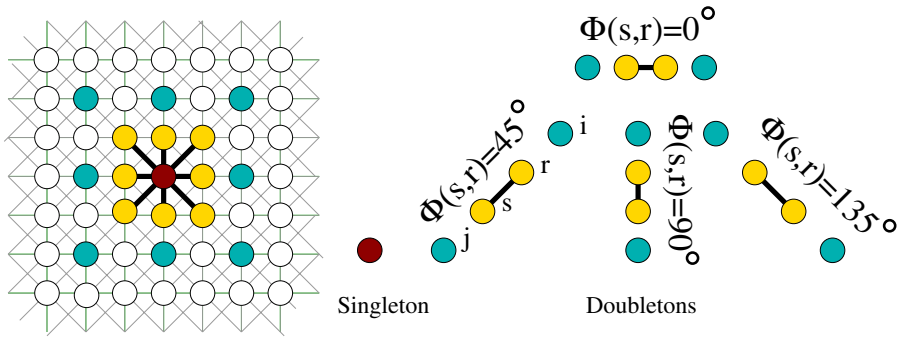


Fig. 4.1 Neighborhood and cliques.

ω_r are on the wrong side of the contour, because in such situations the difference in gray-level values $f_s + f_j$ and $f_r + f_i$ will have an opposite sign than that of the corresponding mean values. This function allows us to enforce object coherence around contours. The new doubleton potential added to the likelihood is then defined as

$$\begin{aligned} G(s,r) &= (1 - F(s,r))\mathcal{M} \\ &-F(s,r)H(\delta(\Theta(s,r),\Phi(s,r)))M(s,r), \end{aligned} \tag{4.8}$$

where $\mathcal{M} \gg M_{\max}$ corresponds to a large constant penalty preventing wrong label assignments around object boundaries. Otherwise, the energy is decreased by $M(s,r)$ whenever the edge direction $\Theta(s,r)$ doesn't match with the clique direction $\Phi(s,r)$ (see Figure 4.1), meaning that there is an intensity edge passing between s and r. The data likelihood MRF energy is then composed of singleton and doubleton potentials as follows:

$$\begin{aligned} U(\mathcal{F},\omega) = &\sum_{s\in\mathcal{S}} \log(\sqrt{2\pi}\sigma_{\omega_s}) + \frac{(f_s - \mu_{\omega_s})^2}{2\sigma_{\omega_s}^2} \\ &+\alpha \sum_{(s,r)\in\mathcal{C}} H(\delta(\omega_s,\omega_r))G(s,r), \end{aligned} \tag{4.9}$$

where $\alpha > 0$ is a weight factor between singleton and doubleton data terms. Putting together Equations (4.5) and (4.9), the Gibbs energy to be minimized can be written as

$$\widehat{\omega} = \arg\min_{\omega} \left(U(\mathcal{F},\omega) + \beta \sum_{(s,r)\in\mathcal{C}} \delta(\omega_s,\omega_r) \right). \tag{4.10}$$

4.3.2 Exact MAP Solution Via Graph Cut

Herein, we will show that the Gibbs energy of Equation (4.10) can be represented by a graph $\mathcal{G}$ and hence an exact MAP solution is found in polynomial time by computing the minimum *s-t-cut* on $\mathcal{G}$ [132]. The vertices include the terminals $\vec{s}$ (*source*) and $\vec{t}$ (*sink*) as well as sites $\mathcal{S}$. Since our model uses pairwise interactions and binary labels, it can be naturally translated into a graph representation where, in addition to

edges corresponding to doubletons, edges connecting vertices from $\mathcal{S}$ with the terminals $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{t}}$ are also defined (see [132] for details). A cut on $\mathcal{G}$ corresponds to a binary partitioning S,T of the vertices such that $\vec{\boldsymbol{s}} \in S$ and $\vec{\boldsymbol{t}} \in T$, which can be described by the binary variables $\omega_s, s \in \mathcal{S}$. Each cut has also a cost corresponding to the sum of edge weights that go from S to T, thus the energy represented by $\mathcal{G}$ can be seen as a function $E(\omega)$ equal to the cost of the cut defined by ω. In our case, $E(\omega)$ is as follows:

$$E(\omega) = \sum_{s \in \mathcal{S}} E_s(\omega_s) + \sum_{(s,r) \in \mathcal{C}} E_{s,r}(\omega_s, \omega_r), \tag{4.11}$$

where E_s corresponds to the Gaussian term from Equation (4.9), while $E_{s,r}$ includes both the Ising prior and the gradient term of Equation (4.9):

$$E_{s,r}(\omega_s, \omega_r) = \beta\delta(\omega_s, \omega_r) + \alpha H(\delta(\omega_s, \omega_r))G(s,r).$$

The main theoretical result of [132] states that a necessary and sufficient condition for graph-representability of E is the following *submodularity* condition:

$$E_{s,r}(0,0) + E_{s,r}(1,1) \leq E_{s,r}(0,1) + E_{s,r}(1,0). \tag{4.12}$$

It is easily seen that the left-hand side is always -2β for all (s,r), as the gradient term vanishes. On the right-hand side, we have a constant 2β from the Ising term, $\alpha\mathcal{M}$ from one of the inhomogeneous label configurations and either 0 or $-\alpha M(s,r)$ from the other depending on the edge direction. Thus for all $(s,r) \in \mathcal{C}$, we have

$$E_{s,r}(0,1) + E_{s,r}(1,0) \geq 2\beta + \alpha(\mathcal{M} - M_{\max})$$

since, according to Equation (4.6), $M_{\max} \geq M(s,r)$ always holds. Therefore submodularity is satisfied for $\beta, \alpha > 0$ if

$$-4\frac{\beta}{\alpha} \leq \mathcal{M} - M_{\max},$$

which is always true as we have chosen $\mathcal{M} \gg M_{\max}$.

4.3.3 Experimental Results

In the experiments, $\nabla\mathcal{F}$ was provided by a Sobel operator followed by non-maxima suppression (see Figure 4.3 for a typical gradient image) and we set $M_{\max} = 10^3$ and $\mathcal{M} = 10^6$. Gaussian parameters were learned from user selected input regions (see Figure 4.4), while the parameters α and β were set to their optimal value by trial and error. The MAP segmentation was then obtained by the max-flow implementation of Kolmogorov (http://pub.ist.ac.at/~vnk/software.html) [27]. We have also compared results obtained by two classical MRF models: The first one uses an Ising prior (equivalent to removing the gradient term by setting $\alpha = 0$); and the second uses a MRF model where the gradient term is replaced by the *boundary term* from [26], which penalizes discontinuities inversely proportional to differences in pixel intensity.

For quantitative evaluation, a set of synthetic images of size 140×140 has been generated from four binary images by Gaussian smoothing with $\sigma' = \{1, 2, 3, 4\}$ and adding Gaussian white noise ranging from $-15\,\text{dB}$ to $10\,\text{dB}$ (see Figure 4.2). The segmentation error is calculated

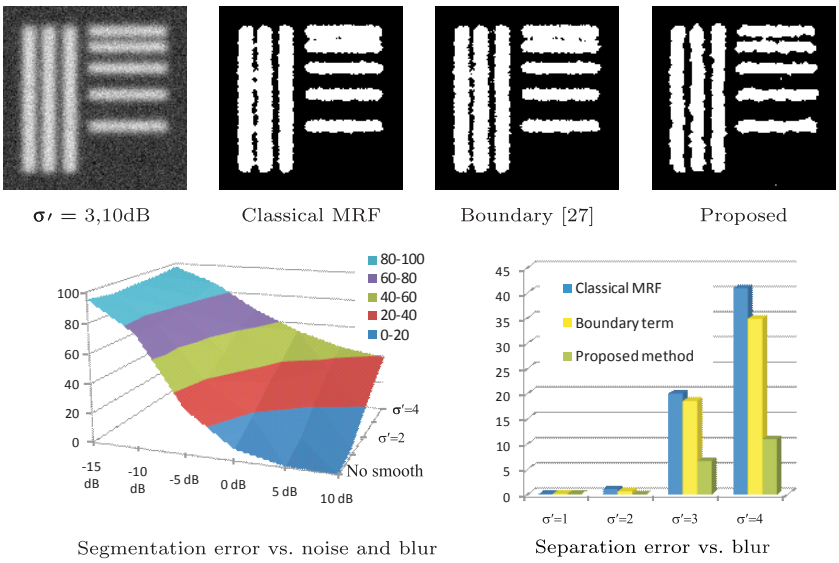


Fig. 4.2 Results on synthetic images.

as the percentage of misclassified pixels. Figure 4.2 shows the average error w.r.t. blur and noise. Obviously, error is linearly increasing with σ' as blurred regions become bigger. On the other hand, this method is quite robust down to an SNR of 0 dB, but becomes quickly unstable below it. We have also evaluated the separation accuracy of the method on noisy blurred images and found that even for moderate smoothing, it outperforms both classical MRF models. Figure 4.2 shows the separation error computed as the percentage of the false foreground pixels in gap areas w.r.t. the total number of pixels of the gap areas.

4.3.3.1 Application in TIRF Microscopy

The proposed approach has also been validated on images taken in Total Internal Reflection Fluorescence (TIRF) microscopy mode, which is an elegant optical technique that provides for the excitation of fluorophores in an extremely thin axial region (optical section) [58]. Images in Figures 4.3 and 4.5 were taken by a CytoScout fluorescent microscope

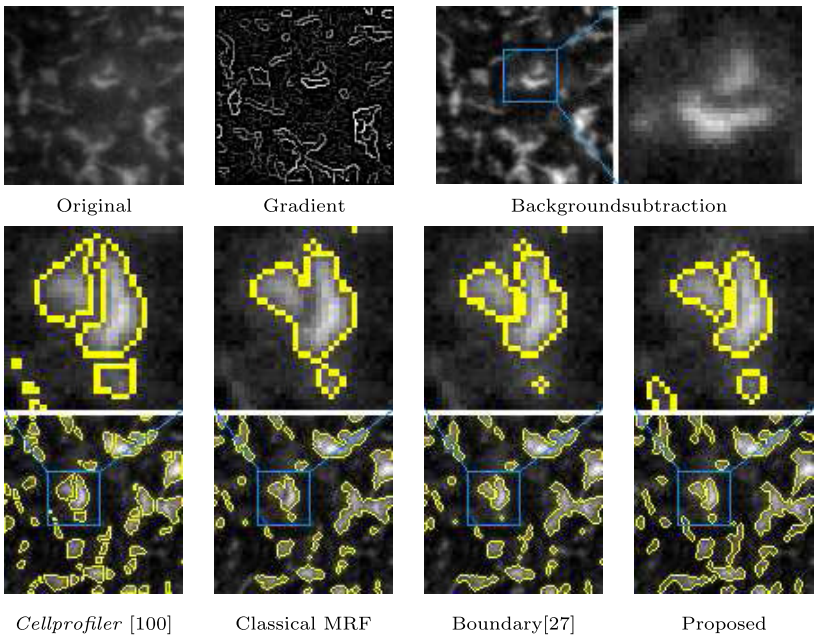
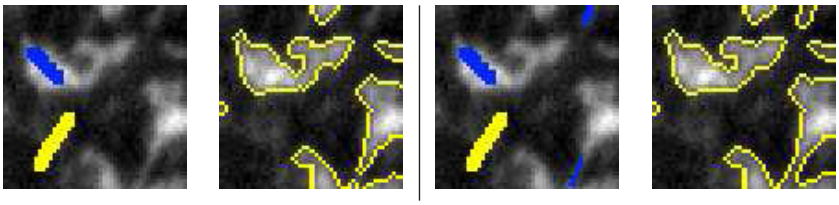


Fig. 4.3 Comparison on a TIRF image.



User interaction and segmentation | After further user interaction

Fig. 4.4 The effect of user interaction.

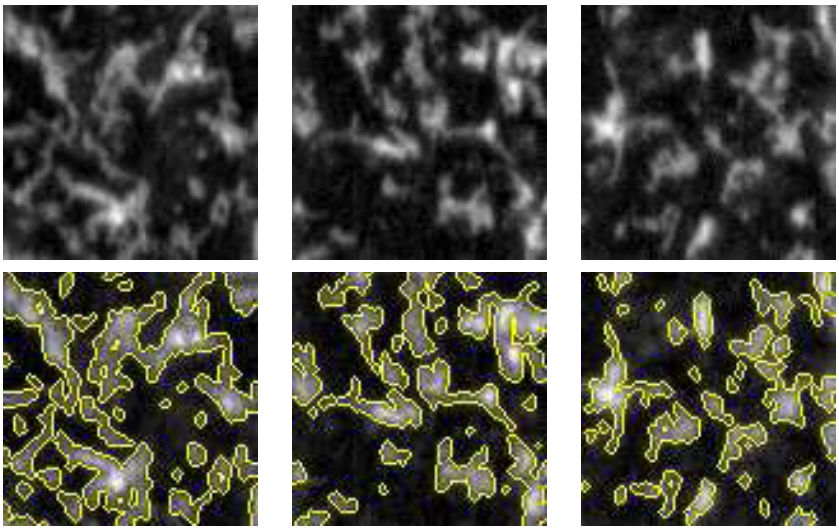


Fig. 4.5 Results on TIRF images.

system using the 488-nm argon-ion laser line for the excitation of fluorescein. They show the plasma membrane of B16 mouse melanoma cells labeled with the fluorescence cholesterol analogue fPEG-Chol which specifically recognizes cholesterol-rich membrane domains [164]. Higher intensity regions indicate cholesterol-rich membrane rafts, which are signaling platforms in the plane of biological membranes playing important roles in many cellular functions.

The quantitative analysis of these sub-cellular structures requires an accurate segmentation. Due to the rather low contrast, a standard background subtraction preprocessing step has been applied before segmentation (see Figure 4.3). The user interaction consists in simple

mouse operated brush strokes of blue (object) or yellow (for background) as shown in Figure 4.4. Based on these samples, the foreground/background Gaussian parameters are computed and an initial segmentation is created. If the segmentation is not accurate, then the user may brush part of a wrongly labeled area. In addition to updating the Gaussian parameters, it is also possible to constrain these marked regions either to be firm foreground or firm background, then a new segmentation is generated.

In Figure 4.3, we compare results obtained by *Cellprofiler* [100] and classical MRF models. Each method's parameters have been manually fine-tuned to get the best result. Notice that *Cellprofiler* tends to produce rather "blocky" boundaries, while the classical MRF model misses some foreground regions as well as merges nearby regions due to the lack of gradient information. Although the classical MRF model with boundary term [26] achieves slightly better separation, our method clearly provides the most accurate segmentation. We remark that the same watershed-based postprocessing step used in *Cellprofiler* can also be applied in our method to further cut larger regions into smaller patches. Additional results can be seen in Figure 4.5. These segmentation results have been validated by expert biologists who found them accurate and relevant. The runtime was consistently below 0.07 sec on TIRF images of size 100×100.

5

Parameter Estimation and Sample Applications

5.1 Unsupervised Image Segmentation

Herein, we give two examples on how to estimate model parameters for MRF segmentation models. First we give a general overview of the parameter estimation problem for the hierarchical MRF model presented in Section 2.5. Second, we show how to estimate Gaussian parameters via the EM algorithm [48, 181] for the segmentation of color textured images using a monogrid MRF model similar to the one presented in Section 2.2.

5.1.1 Parameter Estimation

In real life applications, model parameters are usually unknown, one has to estimate [8] them from the observable image. Here we develop an algorithm for hierarchical Markovian models [108, 118, 119, 120]. Our approach is similar in spirit to Iterative Conditional Estimation [149, 175] as well as to the EM algorithm [48, 181]: we recursively look at the Maximum a Posteriori (MAP) estimate of the label field given the estimated parameters then we look at the Maximum Likelihood (ML) estimate of the parameters given a tentative labeling obtained in the

previous step. The only parameter supposed to be known is the number of labels, all the other parameters are estimated.

When both the model parameters Θ and ω are unknown, the estimation problem becomes [63, 120, 141]

$$(\widehat{\omega}, \widehat{\Theta}) = \arg\max_{\omega,\Theta} P(\omega, \mathcal{F} \mid \Theta). \tag{5.1}$$

The pair $(\widehat{\omega}, \widehat{\Theta})$ is the global maximum of the joint probability $P(\omega, \mathcal{F} \mid \Theta)$. If we regard Θ as a random variable, the above maximization is an ordinary MAP estimation in the following way [63]: Let us suppose, that Θ is restricted to a finite volume domain $\mathcal{D}_\Theta$ and suppose that Θ is uniform on $\mathcal{D}_\Theta$ (that is $P(\Theta)$ is constant). Then, we get [63, 120]:

$$\begin{aligned}
\arg\max_{\omega,\Theta} P(\omega, \Theta \mid \mathcal{F}) &= \arg\max_{\omega,\Theta} \frac{P(\omega, \mathcal{F} \mid \Theta) P(\Theta)}{P(\mathcal{F})} \\
&= \arg\max_{\omega,\Theta} \frac{P(\omega, \mathcal{F} \mid \Theta)}{\int_{\mathcal{D}_\Theta} \sum_{\omega \in \Omega} P(\omega, \mathcal{F} \mid \Theta) d\Theta} \qquad (5.2) \\
&= \arg\max_{\omega,\Theta} P(\omega, \mathcal{F} \mid \Theta). \qquad (5.3)
\end{aligned}$$

However, this maximization is very difficult, having no direct solution. Even Simulated Annealing (SA) is not implementable because the local characteristics with respect to the parameters Θ cannot be computed from $P(\omega, \mathcal{F} \mid \Theta)$. One possible solution is to adopt the following criterion instead [63, 120, 141]:

$$\widehat{\omega} = \arg\max_{\omega} P(\omega, \mathcal{F} \mid \widehat{\Theta}) \tag{5.4}$$

$$\widehat{\Theta} = \arg\max_{\Theta} P(\widehat{\omega}, \mathcal{F} \mid \Theta). \tag{5.5}$$

Clearly, Equation (5.4) is equivalent to Equation (5.1) for $\Theta = \widehat{\Theta}$ and Equation (5.5) is equivalent to Equation (5.1) with $\omega = \widehat{\omega}$. Furthermore, Equation (5.4) is equivalent to the MAP estimate of ω in the case of known parameters:

$$\begin{aligned}
\arg\max_{\omega} P(\omega, \mathcal{F} \mid \widehat{\Theta}) &= \arg\max_{\omega} P(\omega \mid \mathcal{F}, \widehat{\Theta}) P(\mathcal{F} \mid \widehat{\Theta}) \\
&= \arg\max_{\omega} P(\omega \mid \mathcal{F}, \widehat{\Theta}). \qquad (5.6)
\end{aligned}$$

Hence in the following we will concentrate on Equation (5.5) which gives the ML estimate of the parameters. Considering the hierarchical MRF segmentation model (see Figure 2.10), we have the following logarithmic likelihood function [108, 118, 119, 120]:

$$\sum_{i=0}^{M}\sum_{s^i\in\mathcal{S}^i}\sum_{s\in b^i_{s^i}}\left(-\ln(\sqrt{2\pi}\sigma_{\widehat{\omega}_s})-\frac{(f_s-\mu_{\widehat{\omega}_s})^2}{2\sigma^2_{\widehat{\omega}_s}}\right)$$
$$-\beta\underbrace{\sum_{i=0}^{M}q^i\sum_{C^i\in\mathcal{C}^i}\delta(\widehat{\omega}_{C^i})}_{N^{ih}(\widehat{\omega})}-\gamma\underbrace{\sum_{C\in\bar{\mathcal{C}}_3}\delta(\widehat{\omega}_C)}_{\bar{N}^{ih}(\widehat{\omega})}-\ln(Z(\beta,\gamma)), \tag{5.7}$$

where q^i is the number of cliques between two neighboring blocks at scale $\mathcal{B}^i$, $N^{ih}(\widehat{\omega})$ denotes the number of inhomogeneous cliques siting at the same scale and $\bar{N}^{ih}(\widehat{\omega})$ denotes the number of inhomogeneous cliques siting astride two neighboring levels in the pyramid. Considering the first term, we get

$$\sum_{i=0}^{M}\sum_{s^i\in\mathcal{S}^i}\sum_{s\in b^i_{s^i}}\left(-\ln(\sqrt{2\pi}\sigma_{\widehat{\omega}_s})-\frac{(f_s-\mu_{\widehat{\omega}_s})^2}{2\sigma^2_{\widehat{\omega}_s}}\right)$$
$$=\sum_{\lambda\in\Lambda}\sum_{i=0}^{M}\sum_{s^i\in\mathcal{S}^i_\lambda}\sum_{s\in b^i_{s^i}}\left(-\ln(\sqrt{2\pi}\sigma_\lambda)-\frac{(f_s-\mu_\lambda)^2}{2\sigma^2_\lambda}\right), \tag{5.8}$$

where $\mathcal{S}^i_\lambda$ is the set of sites at level i where $\widehat{\omega}_{s^i}=\lambda$. Differentiating with respect to μ_λ and σ_λ, we get a closed form solution for the ML estimates of the Gaussian parameters:

$$\forall\lambda\in\Lambda:\quad \mu_\lambda=\frac{1}{\sum_{i=0}^{M}\mid\mathcal{S}^i_\lambda\mid}\sum_{i=0}^{M}\sum_{s^i\in\mathcal{S}^i_\lambda}\sum_{s\in b^i_{s^i}}f_s,$$
$$\sigma^2_\lambda=\frac{1}{\sum_{i=0}^{M}\mid\mathcal{S}^i_\lambda\mid}\sum_{i=0}^{M}\sum_{s^i\in\mathcal{S}^i_\lambda}\sum_{s\in b^i_{s^i}}(f_s-\mu_\lambda)^2. \tag{5.9}$$

Notice that a gray-level value f_s may be considered several times. More precisely, f_s is considered m-times in the above sum for a given λ if

there are m scales where $\widehat{\omega}$ assigns the label λ to the site s. m can also be seen as a weight. Obviously, the more s has been labeled by λ at different levels, the more is probable that s belongs to class λ and hence its gray-level value f_s characterizes better the class λ. The derivates of the logarithmic likelihood function with respect to β and γ are given by:

$$\frac{\partial}{\partial\beta}\Big(-\beta N^{ih}(\widehat{\omega}) - \ln(Z(\beta,\gamma))\Big) = -N^{ih}(\widehat{\omega}) - \frac{\partial}{\partial\beta}\ln(Z(\beta,\gamma))$$
$$\frac{\partial}{\partial\gamma}\Big(-\gamma \bar{N}^{ih}(\widehat{\omega}) - \ln(Z(\beta,\gamma))\Big) = -\bar{N}^{ih}(\widehat{\omega}) - \frac{\partial}{\partial\gamma}\ln(Z(\beta,\gamma)).$$

From which, we get

$$N^{ih}(\widehat{\omega}) = \frac{\sum_{\omega\in\Omega} N^{ih}(\omega)\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))} \tag{5.10}$$

$$\bar{N}^{ih}(\widehat{\omega}) = \frac{\sum_{\omega\in\Omega} \bar{N}^{ih}(\omega)\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))}{\sum_{\omega\in\Omega}\exp(-\beta N^{ih}(\omega) - \gamma\bar{N}^{ih}(\omega))}. \tag{5.11}$$

The solution of the above equations can be approximated using the following algorithm.

Algorithm 11 (Hyperparameter Estimation).

① Set $k = 0$ and initialize $\widehat{\beta}^0$ and $\widehat{\gamma}^0$. Furthermore, let $N^{ih}(\widehat{\omega})$ denote the number of inhomogeneous cliques at the same scale and $\bar{N}^{ih}(\widehat{\omega})$ denotes the number of inhomogeneous cliques between levels.

② Using SA at a fixed temperature T, generate a new labeling η sampling from

$$P(\mathcal{X} = \omega) = \frac{\exp\Big(-\frac{\widehat{\beta}^k}{T}\sum_{i=0}^{M}\sum_{\{s,r\}\in\mathcal{C}^i}\delta(\omega_s,\omega_r)\Big)}{Z(\widehat{\beta}^k,\widehat{\gamma}^k)} + \frac{\exp\Big(-\frac{\widehat{\gamma}^k}{T}\sum_{\{s,r\}\in\bar{\mathcal{C}}}\delta(\omega_s,\omega_r)\Big)}{Z(\widehat{\beta}^k,\widehat{\gamma}^k)}. \tag{5.12}$$

Compute the number of inhomogeneous cliques $N^{ih}(\eta)$ and $\bar{N}^{ih}(\eta)$ in η.

③ If $N^{ih}(\eta) \approx N^{ih}(\widehat{\omega})$ *and* $\bar{N}^{ih}(\eta) \approx \bar{N}^{ih}(\widehat{\omega})$ then stop, else $k = k + 1$. If $N^{ih}(\eta) < N^{ih}(\widehat{\omega})$ then decrease $\widehat{\beta}^k$, if $N^{ih}(\eta) > N^{ih}(\widehat{\omega})$ then increase $\widehat{\beta}^k$. $\widehat{\gamma}^k$ is obtained in the same way. Continue Step ② with $(\widehat{\beta}^k, \widehat{\gamma}^k)$.

This algorithm completes the computation of the ML estimate of the parameters given $\widehat{\omega}$. The unsupervised segmentation is then carried out using *Adaptive Simulated Annealing* [63, 120], which is an iterative algorithm generating tentative labelings based on current parameter estimates (that is, solving Equation (5.4)) then updating the parameter values to their ML estimate based on the current labeling (that is, solving Equation (5.5) by making use of Equation (5.9) and Algorithm 11). In fact, it is a classical Simulated Annealing with an additional step to reestimate model parameters during segmentation. The convergence of ASA has been proven in [141].

The algorithm has been tested on several synthetic and real images [108, 119, 120]. In summary, the presented unsupervised algorithm provide results comparable to those obtained by supervised segmentations, but of course at the price of higher computing time.

5.1.2 Unsupervised Segmentation of Color Textured Images

A monogrid MRF model for color textured images is proposed in [109, 111]. It uses a nearest neighborhood system (see Figure 1.4) with pixel classes represented by multivariate Gaussian distributions. This kind of modeling corresponds well to our features: Texture feature images (extracted by Gabor filters [96]) are constructed in such a way that similar textures map to similar intensities. Hence pixels with a given texture will be assigned a well determined value with some variance. Furthermore, pixels with similar color map to their average color [174, 189]. Putting these feature distributions into one multivariate Normal mixture, the modes will correspond to clusters of pixels which are homogeneous in both color and texture properties. Therefore regions will be formed where both features are homogeneous while boundaries will be present where there is a discontinuity in either color or texture. Applying these ideas, the *image process* $\mathcal{F}$ can be formalized

as follows: $P(\vec{\boldsymbol{f}}_s \mid \omega_s)$ follows a Gaussian distribution $N(\vec{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$, each pixel class $\lambda \in \Lambda = \{1, 2, \ldots, L\}$ is represented by its mean vector $\vec{\boldsymbol{\mu}}_\lambda$ and covariance matrix $\boldsymbol{\Sigma}_\lambda$. The whole posterior can now be expressed as a first order MRF by including the contribution of the likelihood term via the singletons (pixel sites $s \in \mathcal{S}$). Indeed, the singleton energies directly reflect the probabilistic modeling of labels without context, while doubleton clique potentials express relationship between neighboring pixel labels. Thus the energy function of the so defined MRF image segmentation model has the following form:

$$\sum_{s \in \mathcal{S}} \left(\ln(\sqrt{(2\pi)^n \mid \boldsymbol{\Sigma}_{\omega_s} \mid}) + \frac{1}{2} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s}) \boldsymbol{\Sigma}_{\omega_s}^{-1} (\vec{\boldsymbol{f}}_s - \vec{\boldsymbol{\mu}}_{\omega_s})^T \right) + \beta \sum_{\{s,r\} \in \mathcal{C}} \delta(\omega_s, \omega_r), \tag{5.13}$$

where $\beta > 0$ is a weighting parameter controlling the importance of the prior. As β increases, the resulting regions become more homogeneous.

The segmentation model has the following parameters:

(1) The weight β of the prior term,
(2) the number of pixel classes L,
(3) the mean vector $\vec{\boldsymbol{\mu}}_\lambda$ and covariance matrix $\boldsymbol{\Sigma}_\lambda$ of each class $\lambda \in \Lambda$.

While L strongly depends on the input image data, β is largely independent of it. Experimental evidence suggests that the model is not sensitive to a particular setting of β [109, 111]. We found that setting $\beta \geq 2.0$ gives satisfactory and stable segmentations. Unlike the first two parameters, the mean and covariance of the Gaussians must be computed directly from the input image. A solution to this problem [111] adopts the *EM algorithm* [78] to compute the maximum likelihood estimates of the parameters of a mixture density. Basically, we will fit a Gaussian mixture of L components to the histogram of the image features. The observations consist of the histogram data $\vec{\boldsymbol{d}}_i (i = 1, \ldots, D)$ of the feature images. D denotes the number of histogram points and the dimension of a data point equals to the dimension of the combined color-texture feature space. Assuming there are L classes, we want to

estimate the mean values $\vec{\boldsymbol{\mu}}_\lambda$ and covariance matrices $\boldsymbol{\Sigma}_\lambda$ for each pixel class $\lambda \in \Lambda$.

The *EM algorithm* aims at finding parameter values which maximize the normalized log-likelihood function:

$$\mathcal{L} = \frac{1}{D} \sum_{i=1}^{D} \log \left(\sum_{\lambda \in \Lambda} P(\vec{\boldsymbol{d}}_i \mid \lambda) P(\lambda) \right). \tag{5.14}$$

The underlying model is that the *complete data* includes not only the observable $\vec{\boldsymbol{d}}_i$ but also the *hidden data* labels $\vec{\boldsymbol{\ell}}_i$ specifying which Gaussian process generated the data $\vec{\boldsymbol{d}}_i$. Actually, $\vec{\boldsymbol{\ell}}_i$ is also a vector of dimension L and $\vec{\boldsymbol{\ell}}_i^{\lambda} = 1$ if $\vec{\boldsymbol{d}}_i$ belongs to class λ and 0 otherwise. The idea is that if labels were known, the estimation of model parameters would be equivalent to the supervised case. Hence the following algorithm is alternating two steps: The estimation of a tentative labeling of the data followed by updating the parameter values based on the tentatively labeled data.

Algorithm 12 (EM for Gaussian mixture identification).

① **[Estimation]** Replace $\vec{\boldsymbol{\ell}}_i$ with its conditional expectation based on the current parameter estimates. Since the labels may only take values 0 or 1, the expectation is basically equivalent to the posterior probability:

$$P(\lambda \mid \vec{\boldsymbol{d}}_i) = \frac{P(\vec{\boldsymbol{d}}_i \mid \lambda) P(\lambda)}{\sum_{\lambda \in \Lambda} P(\vec{\boldsymbol{d}}_i \mid \lambda) P(\lambda)}, \tag{5.15}$$

where $P(\lambda)$ denotes the component weight.

② **[Maximization]** Then, using the current expectation of the labels $\vec{\boldsymbol{\ell}}_i$ as the current labeling of the data, the estimation of the parameters is simple:

$$P(\lambda) = \frac{K_\lambda}{D} \tag{5.16}$$

$$\vec{\boldsymbol{\mu}}_\lambda = \frac{1}{K_\lambda} \sum_{i=1}^{D} P(\lambda \mid \vec{\boldsymbol{d}}_i) \vec{\boldsymbol{d}}_i \tag{5.17}$$

$$\boldsymbol{\Sigma}_\lambda = \frac{1}{K_\lambda} \sum_{i=1}^{D} P(\lambda \mid \vec{\boldsymbol{d}}_i) (\vec{\boldsymbol{d}}_i - \vec{\boldsymbol{\mu}}_\lambda)^T (\vec{\boldsymbol{d}}_i - \vec{\boldsymbol{\mu}}_\lambda), \tag{5.18}$$

where $K_\lambda = \sum_{i=1}^{D} P(\lambda \mid \vec{\boldsymbol{d}}_i)$. Basically the posteriors $P(\lambda \mid \vec{\boldsymbol{d}}_i)$ are used as a weight of the data vectors. They express the contribution of a particular data point $\vec{\boldsymbol{d}}_i$ to the class λ.

③ Go to Step ① until convergence. Each iteration is guaranteed to increase the likelihood of the estimates. The algorithm is stopped when the change of the log-likelihood $\mathcal{L}$ is less than a predetermined threshold (our test cases used 10^{-7}).

The algorithm has been tested on a variety of color images. We compared segmentation results using color-only, texture-only and combined (color+texture) features [109, 111] and found in all test-cases that segmentation based purely on texture gives fuzzy boundaries but usually homogeneous regions, whereas segmentation based on color is more sensitive to local variations but provides sharp boundaries. As for the combined features, the advantages of both color and texture based segmentation have been preserved: we obtained sharp boundaries and homogeneous regions. Results has also been compared to those obtained by the JSEG algorithm [49], a recent unsupervised method for color textured image segmentation. The method in [111] clearly outperforms JSEG (see Figure 5.1) but JSEG's advantage is that we do not have to specify the image dependent parameter L.

5.2 Classification of Synthetic Aperture Radar Images

Synthetic aperture radar (SAR) is known to be unaffected by sun-illumination, and almost not influenced by atmospheric conditions [168]. Recent improvements of spaceborne SAR currently enable acquisitions of Very High Resolution (VHR) data (up to metric resolution) with a very short revisit time (up to 12 hours). The acquisitions may be either single- or multi-polarized, thus highlighting different aspects of a same ground area. In this respect, SAR imagery offers a huge potential for risk management by, for example, allowing land-use or land-cover mapping or detection of areas damaged by a disaster event. In the framework of the assessment of environmental risk, we address here the problem of classifying SAR images of urban areas, a

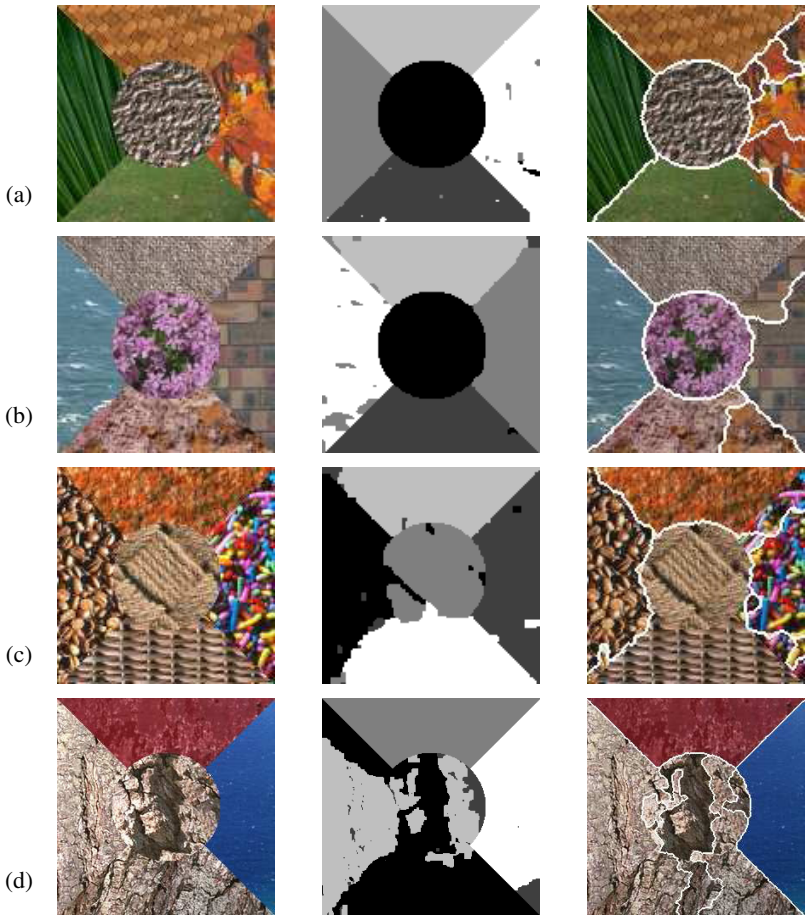


Fig. 5.1 Unsupervised segmentation results on color textured images, each with 5 classes [111].

specifically interesting typology given the fact that it is strategic and critical for population risks.

5.2.1 CoDSEM-MRF Method Overview

The classification considered here is developed in a supervised context and consists of three steps: SAR amplitude probability density function (PDF) estimation, MRF modeling and classification [211].

First, the method models the statistics of SAR amplitude PDF using a dictionary-based stochastic expectation maximization (DSEM) approach, that is, the statistics of SAR amplitudes are considered as mixtures of K parametric components that are automatically drawn from a dictionary of SAR-specific PDFs [134, 159]. For each class m considered for classification, $m \in [1; M]$, the mixture PDFs $p_m(z|\omega_m)$ are estimated following a finite mixture model for the independent distribution of gray-levels

$$p_m(z|\omega_m) = \sum_{i=1}^{K} P_{mi} p_{mi}(z|\vartheta_{mi}), \tag{5.19}$$

where z is a gray-level, $z \in [0; Z-1]$, and ω_m is the mth class. P_{mi} are mixing proportions, such that for a given m, $\sum_{i=1}^{K} P_{mi} = 1$ with $0 \leq P_{mi} \leq 1$. The various mixture PDF models $p_{mi}(z|\vartheta_{mi})$ are selected in a predefined dictionary (See Appendix A). When separately applied to each class considered in the VHR SAR image, DSEM represents a natural model for heterogeneous scenarios, leading to a mixture estimate where distinct components may be interpreted as the contributions of different ground materials present in each relevant class. This makes the algorithm robust with respect to possibly complicated shapes of class histograms.

We take into account an additional knowledge by extracting a textural feature from the original SAR image using the gray-level co-occurrence matrix [84] (GLCM) method. In fact, well-chosen textural features often turn out to be discriminant with respect to urban areas. The flexibility of DSEM, granted by its essentially nonparametric formulation, makes it feasible to estimate the marginal PDFs of both the amplitude and the textural features. Thanks to Sklar's theorem [165], copula functions allow a joint bivariate PDF to be modeled, given the related marginal PDFs (See Appendix B). This algorithm will be refered as CoDSEM (for copula-DSEM).

To proceed to contextual image classification, we combine the resulting joint PDF estimates with a Markov random field approach. For each class $m \in [1; M]$ we can define a local characteristic

$$p(x_s = \omega_m | x^{(s)}) = \frac{\exp(-H(x_s = \omega_m | x^{(s)}))}{\sum_{j=1}^{M} \exp(-H(x_s = \omega_j | x^{(s)}))}, \tag{5.20}$$

where s is the current site ($s \in S$), x_s the corresponding class label, $x^{(s)}$ the configuration outside the site s such that $x^{(s)} = \{x_t, t \neq s, t \sim s\}$ and $t \sim s$ means that t and s are neighboring pixels. We deal here with an anisotropic second order neighborhood; only cliques C of size 2 are considered. The MRF energy function H, given the conditional PDFs (5.19) and no prior information about the proportions of classes on the testing image, is expressed with only one parameter $\beta > 0$:

$$H(\omega_m|z,\beta) = \sum_{s' \in S} \left[-\log p_m(z|\omega_m) - \beta \sum_{s:\{s,s'\} \in C} \delta_{x_s = x'_s} \right] \tag{5.21}$$

with

$$\delta_{x_s = x_{s'}} = \begin{cases} 1, & \text{if } x_s = x_{s'} \\ 0, & \text{otherwise} \end{cases}.$$

With the knowledge of a training map, we can easily estimate β, by maximizing the pseudo-likelihood PL, defined with the local characteristics of Equation (5.20):

$$\log PL(x|\beta) = \log \left[\prod_{s \in S} p(x_s = \omega_m | x^{(s)}, \beta) \right]. \tag{5.22}$$

To optimize this function, a simulated annealing algorithm turns out to be effective. Preliminary experiments showed that for a correct β estimation, the training ground truth must be exhaustive, or at least sufficiently representative of class-transition areas. This is consistent with the role of this parameter in tuning the probability of spatial class transitions.

In order to generate the output classification map, the energy function H (Equations (5.21)) is minimized by the modified Metropolis dynamics (MMD) algorithm that is usually an effective tradeoff between accuracy and computational burden.

This classification algorithm will be refered as the CoDSEM-MRF algorithm.

5.2.2 Modified CoDSEM-MPM Method

A multiscale extension of the previous classifier has also been considered [209]. The graph taken into account is a quad-tree, and the data are hierarchically decomposed via wavelet transforms. The labels on the hierarchical graph are estimated using a marginal posterior mode (MPM) criterion. The likelihoods for each class are determined using the previously described CoDSEM algorithm. We slightly modified the basic classification by MPM criterion estimation introduced by Laferte et al. [139] by including an update of the prior estimation, as shown Figure 5.2 so as to increase the robustness against speckle. This method will be refered as modified CoDSEM-MPM.

5.2.3 SAR Classification Results

The results obtained with the proposed methods (CoDSEM combined to spatial or hierarchical MRFs) are illustrated with a COSMO-SkyMed acquisition over the Port-au-Prince quay in Haiti (©ISA). It is a single-pol image of size 920×820 pixels for which the characteristics are: HH polarization, StripMap acquisition mode (2.5 m ground resolution), geocoded (see Figure 5.3). Three main classes (urban, vegetation, and wet soil/water) are present in the considered images. We use manually annotated ground truths for training and test sets with spatially disjoint training and test fields. A small proportion of pixels (around 6%) is selected as learning samples. Given this small percentage, the β MRF parameter has to be obtained by trial and error for β values chosen in $[0.6; 2.2]$, leading to $\beta \approx 1.3$. The results are assessed both qualitatively (see Figure 5.4) and quantitatively (accuracy results, Table 5.1).

The results obtained for both CoDSEM-MRF and modified CoDSEM-MPM are very similar and the experiments pointed out high accuracy on the test images. The main difference lies in the visual maps, on which we can notice the smoothing effects of the first method. The main water misclassifications are due to the huge cross artifact, related to intrisic properties of SAR acquisitions.

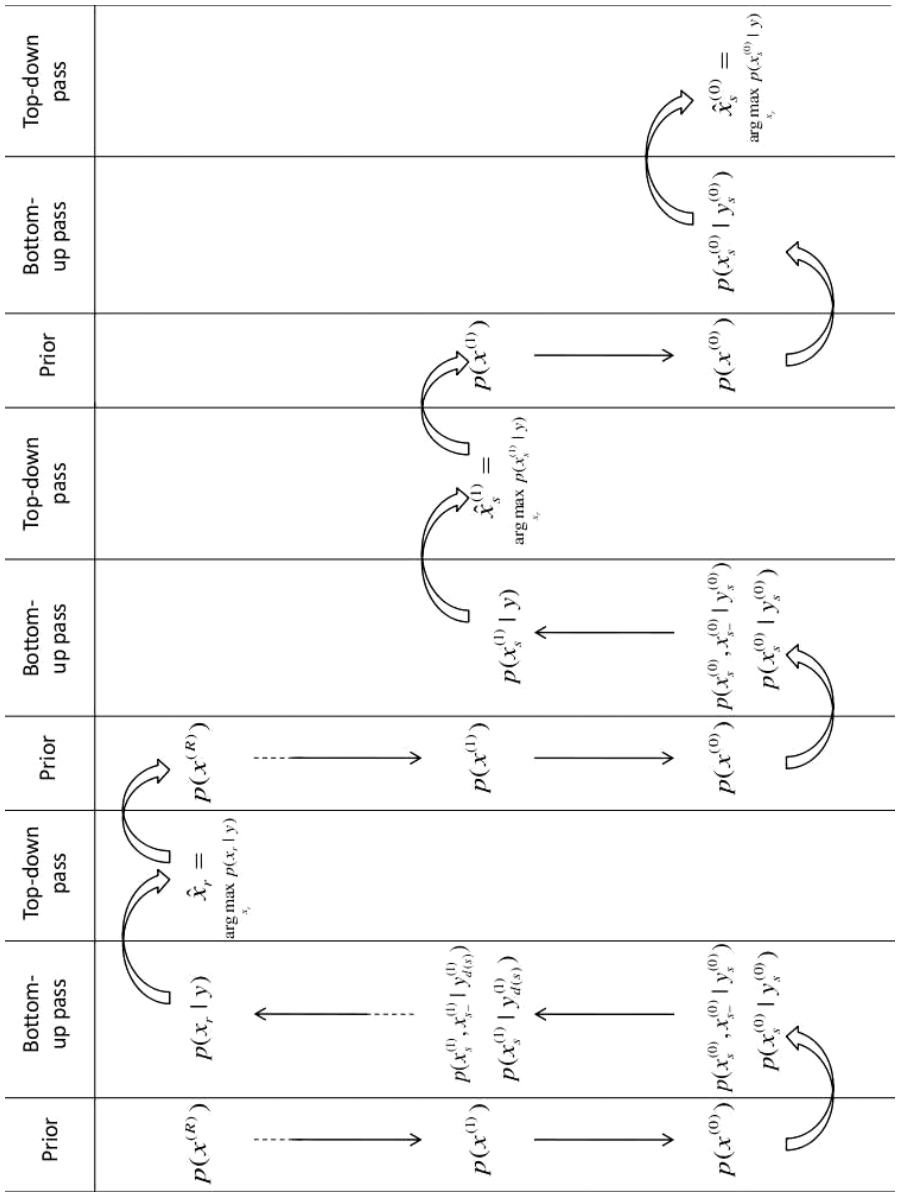
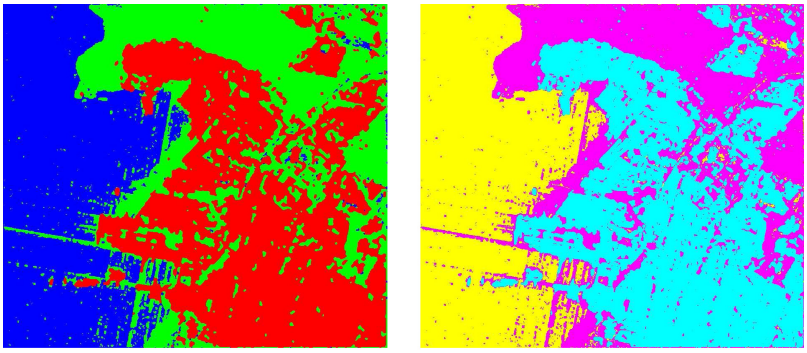


Fig. 5.2 Modified MPM estimation algorithm on the quad-tree. In this representation, the decomposition is done along $R = 2$ levels.

Fig. 5.3 Original SAR image (COSMO-SkyMed,©ASI).



(a) (b)

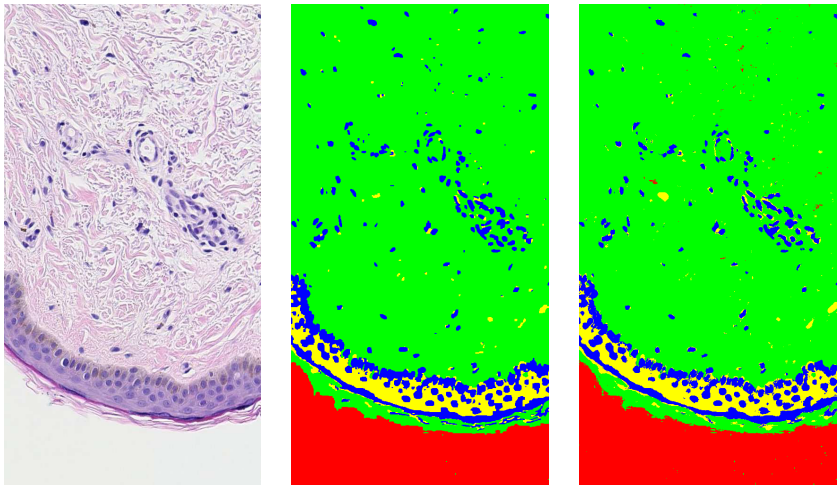
Fig. 5.4 (a): Classification map obtained by CoDSEM-MRF; (b): Classification map obtained by modified CoDSEM-MPM. In green: vegetation class; In blue: water class; In red: urban area.

5.2.4 Further Application in Histology

To show the applicability of the model presented in this section, the same algorithms (CoDSEM-MRF and CoDSEM-MPM) were applied

Table 5.1. Accuracy for each of the 3 classes and overall results for the test areas of the Port-au-Prince quay.

	Port-au-Prince quay			
	water %	urban %	vegetation %	**overall %**
CoDSEM-MRF	97.59	99.03	99.28	**98.63**
Modified CoDSEM-MPM	97.65	98.99	98.96	**98.53**



Original CoDSEM-MRF CoDSEM-MPM

Fig. 5.5 Classification results on a color RGB histological image of the skin of size 550 × 1020 (©Galderma).

to a mono-resolution color RGB histological image of the skin provided by *Galderma* of size 550 × 1020 pixels (see Figure 5.5) [210]. The R, G, and B channels are considered as the input features and the class-conditional marginal probability density functions are modeled by using Gaussian mixtures. The image shown in Figure 5.5 is classified into 4 classes that were interpreted by a dermatological expert as the cytoplasm (in yellow in the classification maps), the nuclei (in blue) and the background (in red). The green class gathers the dermis matrix, the collagen, and the stratum corneum keratin. Each of these classes is modeled by resorting to multivariate copulas, that merge the maginal PDFs into a joint class-conditional PDF. The multiresolution decompositions (required for the hierarchical MRF-based algorithm)

were obtained by a Haar wavelet transform on $R = 2$ levels. Note that no textural features are used for the classification.

5.3 Multilayer MRF Models

The human visual system is not treating different features sequentially. Instead, multiple cues are perceived simultaneously and then they are integrated by our visual system in order to explain the observations. Therefore different image features has to be handled in a parallel fashion. We have developed such a model in a Markovian framework and successfully applied it to color-texture [113, 114] and color-motion segmentation [12, 14, 110, 112]. Herein, we present the MRF image segmentation model which aims at combining color and motion features for video object segmentation [110, 112]. The model has a multi-layer structure (see Figure 5.6): Each feature has its own layer, called *feature layer*, where an MRF model is defined using only the corresponding feature. A special layer is assigned to the combined MRF model. This layer

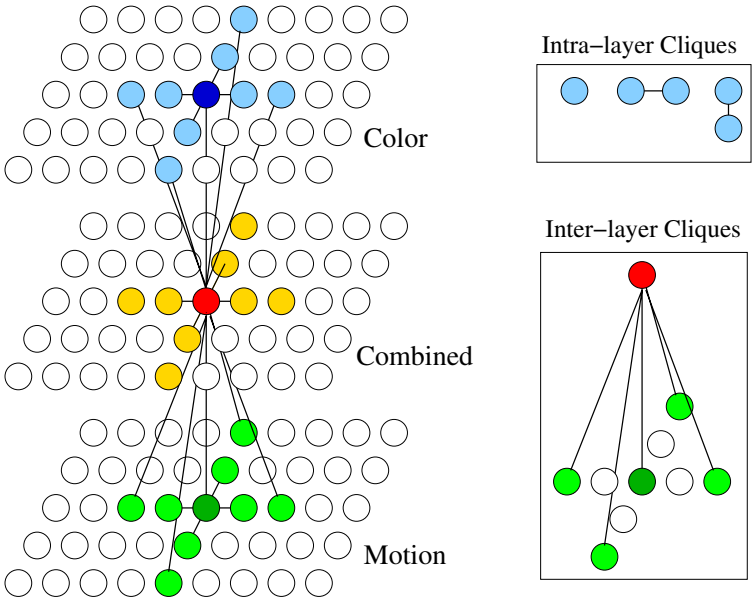


Fig. 5.6 Multi-layer MRF model [110, 112].

interacts with each feature layer and provides the segmentation based on the combination of different features. Unlike previous methods [129], our approach does not assume motion boundaries being part of spatial ones. The uniqueness of the proposed method is the ability to detect boundaries that are visible only in the motion feature as well as those visible only in the color one.

Perceptually uniform color values and precomputed optical flow data are used as features for the segmentation. The proposed model consists of 3 layers. At each layer, we use a first order neighborhood system and extra inter-layer cliques (Figure 5.6). The image features are represented by multivariate Gaussian distributions. For example, on the color layer, the observed image $\mathcal{F}^c = \{\vec{\boldsymbol{f}}_s^c | s \in \mathcal{S}^c\}$ consists of three spectral component values (L*u*v*) at each pixel s denoted by the vector $\vec{\boldsymbol{f}}_s^c$. The class label assigned to a site s on the color layer is denoted by ψ_s. The energy function $U(\psi, \mathcal{F}^c)$ of the so defined MRF layer has the following form:

$$\sum_{s \in \mathcal{S}^c} \mathcal{G}^c(\vec{\boldsymbol{f}}_s^c, \psi_s) + \beta \sum_{\{s,r\} \in \mathcal{C}} \delta(\psi_s, \psi_r) + \sum_{s \in \mathcal{S}^c} V^c(\psi_s, \eta_s^c),$$

where $\mathcal{G}^c(\vec{\boldsymbol{f}}_s^c, \psi_s)$ denotes the Gaussian energy term. The last term $(V^c(\psi_s, \eta_s^c))$ is the inter-layer clique potential. The motion layer adopts a similar energy function with some obvious substitutions (that is for simplicity, we assume a translational motion model here — for a more elaborate model see [112]).

The combined layer only uses the motion and color features indirectly, through inter-layer cliques. A label consists of a pair of color and motion labels such that $\eta = \langle \eta^c, \eta^m \rangle$, where $\eta^c \in \Lambda^c$ and $\eta^m \in \Lambda^m$. The set of labels is denoted by $\Lambda^x = \Lambda^c \times \Lambda^m$ and the number of classes $L^x = L^c L^m$. Obviously, not all of these labels are valid for a given image. Therefore the combined layer model also estimates the number of classes and chooses those pairs of motion and color labels which are actually present in a given image. The energy function $U(\eta)$ is of the following form:

$$\sum_{s \in \mathcal{S}^x} (V_s(\eta_s) + V^c(\psi_s, \eta_s^c) + V^m(\phi_s, \eta_s^m)) + \alpha \sum_{\{s,r\} \in \mathcal{C}} \delta(\eta_s, \eta_r),$$

where $V_s(\eta_s)$ denotes singleton energies, $V^c(\psi_s, \eta_s^c)$ (resp. $V^m(\phi_s, \eta_s^m)$) denotes inter-layer clique potentials. The last term corresponds to second order intra-layer cliques which ensures homogeneity of the combined layer. α has the same role as β in the color layer model and $\delta(\eta_s, \eta_r) = -1$ if $\eta_s = \eta_r$, 0 if $\eta_s \neq \eta_r$ and 1 if $\eta_s^c = \eta_r^c$ *and* $\eta_s^m \neq \eta_r^m$ or $\eta_s^c \neq \eta_r^c$ *and* $\eta_s^m = \eta_r^m$. The idea is that region boundaries present at both color and motion layers are preferred over edges that are found only at one of the feature layers. At any site s, we have 5 inter-layer interactions between two layers: Site s interacts with the corresponding site on the other layer as well as with the 4 neighboring sites two steps away (see Figure 5.6). This potential is based on the difference of the first order potentials at the corresponding feature layers. Clearly, the difference is 0 if and only if both the feature layer and the combined layer has the same label. If the labels are different then it is proportional to the energy difference between the two labels. Finally, the singleton energy controls the number of classes at the combined layer by penalizing small classes.

The proposed algorithm has been tested on real video sequences [110, 112]. We also compare the results to motion only and color only segmentation (basically a monogrid model similar to the one defined for the feature layers but without inter-layer cliques). The mean vectors and covariance matrices were computed over representative regions selected by the user. The number of motion and color classes is known a priori but classes on the combined layer are estimated during the segmentation process. Figure 5.7 shows some segmentation results. Note that the head of the men on this image can only be separated from the background using motion features. Clearly, the multilayer model provides significantly better results compared to color only and motion only segmentations. See Figure 5.8 to compare the performance of the proposed method with the one from [124] on the *Mother and Daughter* standard sequence. Some of the contours are lost by [124] (the sofa, for example) while our method successfully identifies region boundaries. In particular, our method is able to separate the hand of the mother from the face of the daughter in spite of their similar color. This demonstrates again that the proposed method is quite powerful in combining motion and color features in order to detect boundaries

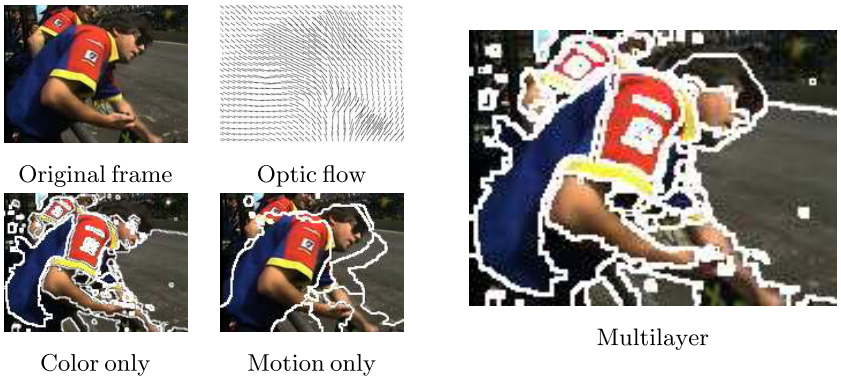


Fig. 5.7 Segmentation results [110, 112].

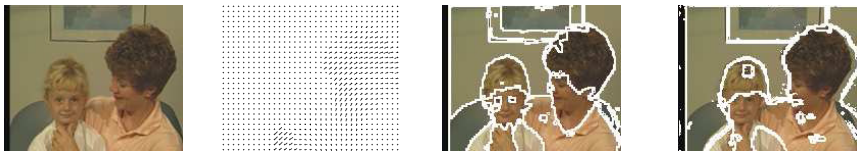


Fig. 5.8 Comparison of the segmentation results obtained by the proposed method [110, 112] and those produced by the algorithm of Khan & Shah [124].

visible only in one of the features. We can also handle occlusion and more complex motions using a modified multilayer model presented in [112]. The model has also been successfully applied to color-textured image segmentation [113, 114] as well as to change detection in aerial images [12, 13].

5.3.1 Application: Change Detection in Aerial Images

Herein, we present the application of multilayer modeling for automatic change detection on airborne images taken with moving cameras[14]. Essentially, we want to extract the accurate silhouettes of moving objects or object-groups in images taken by moving airborne vehicles in consecutive moments. This problem is solved in two steps: first a coarse (but robust) image registration is performed for camera motion compensation, then the aligned input image pair is segmented into moving (foreground) objects and background. Main challenges are

camera motion, noise and the parallax artifacts caused by the static objects having considerable height (buildings, trees, walls, etc.) from the difference image.

Denote by X_1 and X_2 the two consecutive frames of the image sequence above the same pixel lattice S. The gray-value of a given pixel $s \in S$ is $x_1(s)$ in the first image and $x_2(s)$ in the second one.

The first step is to remove camera motion by estimating the optimal similarity transform between the images. For that purpose, we will use the Fourier shift-theorem based method of [180], which yields the registered second frame, $X_2^\dagger$. The pixel values of $X_2^\dagger$ are denoted by $\{x_2^\dagger(s)\}$.

The final goal is to perform a binary segmentation of the images into foreground (fg) and background (bg) classes, which is solved by a three-layer MRF model. For the segmentation, two type of features are extracted from the aligned image pair (see Figure 5.9). The first feature is the gray-level difference of the corresponding pixels in the registered images: $d(s) = x_2^\dagger(s) - x_1(s)$. We validate this feature through experiments (Figure 5.9c): if we plot the histogram of $d(s)$ values corresponding to manually marked background points, then we can observe that a Gaussian approximation is reasonable: $P(d(s)|\text{bg}) = N(d(s), \mu, \sigma)$. On the other hand, any $d(s)$ value may occur in the foreground, hence the foreground class is modeled by a uniform density: $P(d(s)|\text{fg}) = 1/(b_d - a_d)$, if $d(s) \in [a_d, b_d]$, 0 otherwise. Next, we demonstrate the limitations of this feature. After supervised estimation of the distribution parameters, we derive D image in Figure 5.9d as the maximum likelihood estimate: the label of s is $\text{argmax}_{\psi \in \{\text{fg,bg}\}} P(d(s)|\psi)$. We can observe here that the registration and parallax errors cannot be filtered out using only $d(.)$, since their $d(s)$ values appear as outliers with respect to the previously defined Gaussian distribution.

From another point of view, assuming the presence of errors of a few pixels, we can usually find an $o_s = [o_x, o_y]$ offset vector, for which the rectangular neighborhood of s in X_1 and the same shaped neighborhood of $s + o_s$ in $X_2^\dagger$ is strongly correlated. In Figure 5.9e/f, we plot the correlation values over the search window of the offset o_s around two given pixels (marked with the beginning of the arrows in Figure 5.9d). The upper pixel corresponds to a parallax error in the background, while the lower one is part of a real object displacement.

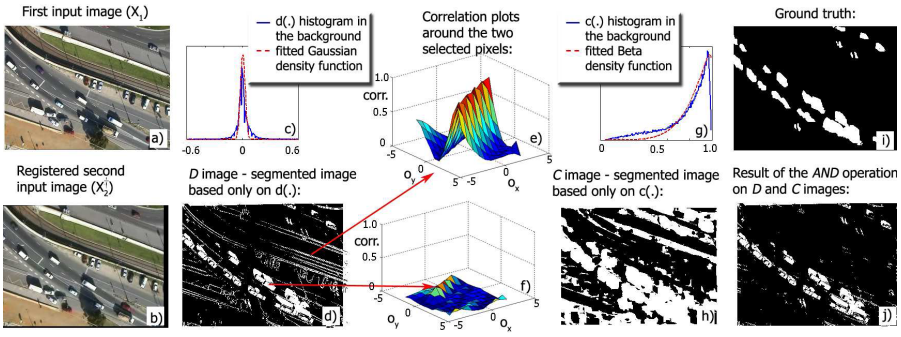


Fig. 5.9 Feature selection. Notations are in the text of Section 5.3.1.

The correlation plot has high peak only in the upper case. We use $c(s)$, the maxima in the local correlation function around pixel s as second feature. By examining the histogram of $c(s)$ values in the background (Figure 5.9g), we find that it can be approximated with a beta density function: $P(c(s)|\text{bg}) = B(c(s), \alpha, \beta)$. As for the foreground class we will use a uniform probability $P(c(s)|\text{fg})$ with a_c and b_c parameters. We see in Figure 5.9h (C image) that the $c(.)$ descriptor causes also poor result in itself. Even so, if we consider D and C as a Boolean lattice, where "true" corresponds to the foreground label, the logical AND operation on D and C improves the results significantly (Figure 5.9j). We note that this classification is still quite noisy, therefore the two features have to be fused in a sophisticated way. For that purpose, a three-layer MRF model is constructed on a graph $\mathcal{G}$ whose structure is shown in Figure 5.10. The sites of $\mathcal{G}$ are arranged into three layers: S^d, S^c, and S^*, each layer having the same size as the image lattice S. We assign to each pixel $s \in S$ a unique site in each layer: for example, s^d is the site corresponding to pixel s on the layer S^d. We denote $s^c \in S^c$ and $s^* \in S^*$ similarly. The segmentation is obtained by assigning a label $\omega(.)$ to all sites of $\mathcal{G}$ from the label-set: $L \triangleq \{\text{fg}, \text{bg}\}$. The labeling of S^d/S^c corresponds to the segmentation based on the $d(.)/c(.)$ feature, respectively; while the labels at the S^* layer present the final change mask. A global labeling of $\mathcal{G}$ is $\underline{\omega} = \left\{\omega(s^i) | s \in S, i \in \{d, c, *\}\right\}$.

A first order neighborhood on $\mathcal{G}$ (see Figure 5.10) is defined to ensure the smoothness of the segmentation: edges are put within each layer

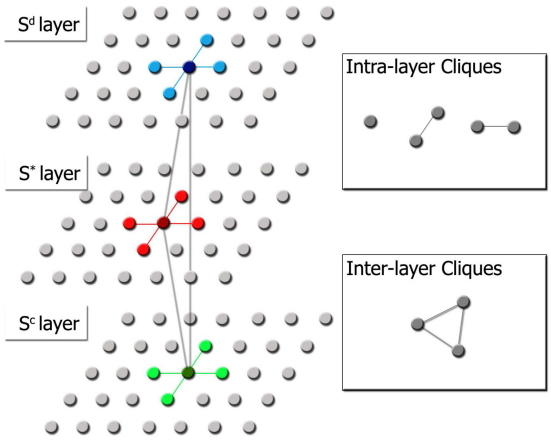


Fig. 5.10 Structure of the three-layer MRF model.

between site pairs corresponding to neighboring pixels of the image lattice S. Furthermore, sites corresponding to the same pixel must interact in order to proceed the fusion of the two different segmentations' labels in the S^* layer. Therefore interlayer edges are introduced between sites s^i and s^j: $\forall s \in S$; $i,j \in \{d,c,*\}$, $i \neq j$, yielding a neighborhood graph with doubleton "intra-layer" cliques (their set is $\mathcal{C}_2$) and "inter-layer" cliques ($\mathcal{C}_3$) of site-triples. The singleton cliques ($\mathcal{C}_1$) will link the model and the local observations. Hence, the set of cliques is $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$.

The next step is to define the corresponding clique potentials V_C, which completes the definition of the MRF model. As we stated previously, the labels in the S^d and S^c layers are directly influenced by the $d(.)$ and $c(.)$ values, respectively, while the labels at S^* have no direct links with these measurements. For this reason, the singleton potentials are of the following form:

$$V_{\{s^d\}}\left(\omega(s^d)\right) = -\log P(d(s)|\omega(s^d)) \tag{5.23}$$

$$V_{\{s^c\}}\left(\omega(s^c)\right) = -\log P(c(s)|\omega(s^c)) \tag{5.24}$$

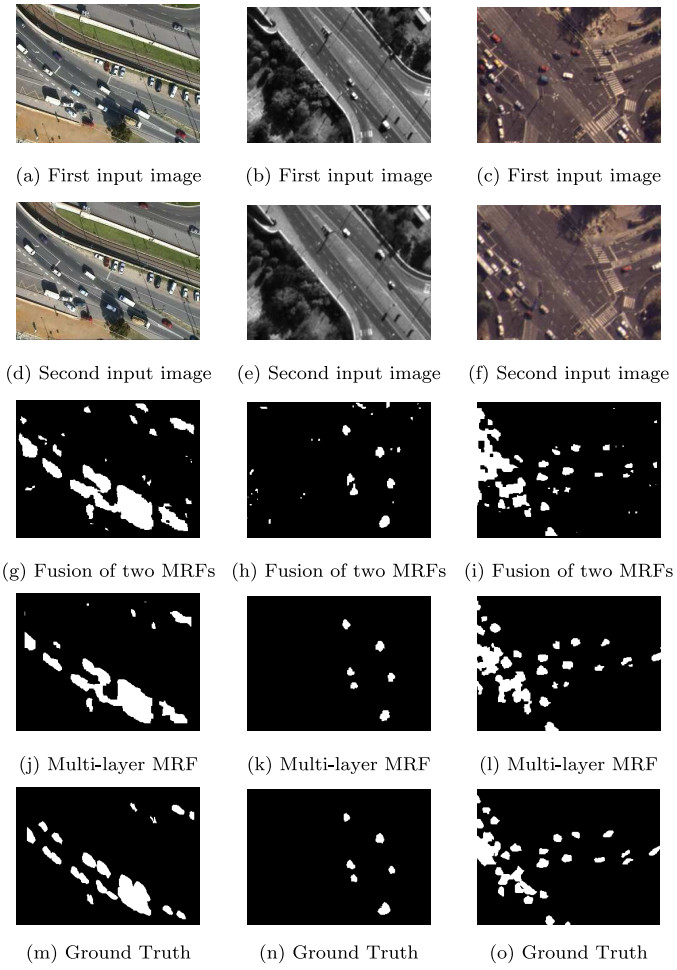
$$V_{\{s^*\}}\left(\omega(s^*)\right) = 0. \tag{5.25}$$

$$\tag{5.26}$$

The doubletons $C_2 = \{s^i, r^i\} \in \mathcal{C}_2,\ i \in \{d, c, *\}$ will ensure smooth regions, hence they take the usual form:

$$V_{C_2}(\underline{\omega}_{C_2}) = \begin{cases} -\delta^i & \text{if } \omega(s^i) = \omega(r^i) \\ +\delta^i & \text{if } \omega(s^i) \neq \omega(r^i), \end{cases} \tag{5.27}$$

where $\delta^i > 0$ is a constant. Finally, for the interlayer cliques, observe that a pixel is likely generated by the background process if and only if in the S^d and S^c layers at least one corresponding site has



(a) First input image (b) First input image (c) First input image

(d) Second input image (e) Second input image (f) Second input image

(g) Fusion of two MRFs (h) Fusion of two MRFs (i) Fusion of two MRFs

(j) Multi-layer MRF (k) Multi-layer MRF (l) Multi-layer MRF

(m) Ground Truth (n) Ground Truth (o) Ground Truth

Fig. 5.11 Experimental results.

the label "bg." Therefore the I_{bg} indicator function is defined for $i \in \{d, c, *\}$ as:

$$I_{\text{bg}}(s^i) = \begin{cases} 1 & \text{if } \omega(s^i) = \text{bg} \\ 0 & \text{otherwise.} \end{cases} \tag{5.28}$$

With this notation the potential of an inter-layer clique $C_3 = \{s^d, s^c, s^*\}$ is with $\rho > 0$:

$$V_{C_3}(\underline{\omega}_{C_3}) = \begin{cases} -\rho & \text{if } I_{\text{bg}}(s^*) = \max\left(I_{\text{bg}}(s^d), I_{\text{bg}}(s^c)\right) \\ +\rho & \text{otherwise.} \end{cases} \tag{5.29}$$

The MAP labeling is then obtained as the minimum energy configuration $\widehat{\underline{\omega}}$, which is obtained via simulated annealing:

$$\begin{aligned} \widehat{\underline{\omega}} = \text{argmin}_{\underline{\omega} \in \Omega} & - \sum_{s \in S} \log P(d(s)|\omega(s^d)) - \sum_{s \in S} \log P(c(s)|\omega(s^c)) \\ & + \sum_{C_2 \in \mathcal{C}_2} V_{C_2}\left(\underline{\omega}_{C_2}\right) + \sum_{C_3 \in \mathcal{C}_3} V_{C_3}\left(\underline{\omega}_{C_3}\right). \end{aligned} \tag{5.30}$$

The final segmentation is taken as the labeling of the S^* layer. In Figure 5.11, we show some results obtained on three pairs of aerial images. For each pair, we show the ground truth change masks obtained by manual segmentation, the multi-layer MRF results and a simple fusion obtained as a logical AND operation on the change masks of two monolayer segmentations based on each features. The increased precision of the multi-layer model is clearly visible. Another application of multilayer modeling can be found in [11].

6

Conclusion

In this monograph, we have reviewed the two main steps of statistical image image segmentation: modeling and energy minimization. We have considered various segmentation tasks in a common framework, called image labeling, where the problem is reduced to assigning labels to pixels.

Taking a probabilistic approach, labels were modeled using Markov random fields (MRF) and the inference is performed via Bayesian estimation, in particular maximum a posteriori (MAP) estimation. The advantage of MRF modeling is that prior information can be "coded" *locally* through clique potentials. Another advantage is that the local behavior of MRFs permits to develop highly parallel algorithms. Unfortunately, even with massively parallel algorithms, finding the MAP estimate is computationally demanding due to the non-convexity of the energy function. To eliminate this drawback, many authors propose multigrid pyramidal MRF schemes. The advantage of such an approach is that at coarser resolution, the configuration space is considerably smaller and thus the optimization problem becomes easier. Using a top-down relaxation strategy in the pyramid, computing time can be considerably reduced and the quality of final results are increased with

respect to monogrid schemes. Hierarchical models allow propagation of local interactions more efficiently, giving better estimates (in particular for fast deterministic relaxation algorithms, such as ICM). Intuitively, such models incorporate implicit long range interactions in the form of interconnected short range interactions. Multi-layer models are quite efficient in fusing different image segmentation cues, while keeping a relatively low complexity of the final MRF model.

All the MRF models considered in this monograph result in a non-convex energy function. Classical energy minimization techniques can be divided into two main categories: *stochastic* or *deterministic* algorithms. The former category — in theory — is guaranteed to find *global* minima with probability 1 but it requires a large amount of computing time. Deterministic methods aim at finding a reasonably good approximation of the minimum. The obtained result is always a *local* minimum, but deterministic methods are less time-consuming than stochastic algorithms. Parallelization is another possible way to speed up optimization algorithms. The convergence of some parallelization schemes has been proved (especially for such algorithms where the conditions of the convergence of sequential algorithms have not been violated.). Modern energy minimization is based on graph cut. The main advantage of these algorithms is low computational complexity and the guarantee to find an exact minima. However, these algorithms work only for a restricted class of energy functions: only binary labels and pairwise, attractive interactions are allowed. Nevertheless, many image segmentation task can be formulated under these constraints. Variations of the basic graph cut algorithm of Kolmogorov and Boykov have been recently developed for more complex MRF models.

Acknowledgments

The authors would like to thank INRIA and University of Szeged for partial financial support of their research works, ERCIM for the attribution of two ERCIM Alain Bensoussan Postdoctoral Fellowships, DGA in France for partial funding of a PhD, MAE, and MESR in France, and the NIH in Hungary for financial support for exchanges between France and Hungary through the PHC Balaton Program. Zoltan Kato was supported by the grants CNK80370 of the National Innovation Office (NIH) & the Hungarian Scientific Research Fund (OTKA); and the TÁMOP-4.2.1/B-09/1/KONV-2010-0005 of the European Union and European Regional Development Fund.

The authors greatfully thank the various contributions of the following colleagues: Csaba Benedek, Marc Berthod, Rama Chellappa, Xavier Descombes, Imre Gombos, Christine Graffigne, Ian Jermyn, Vladimir Krylov, John Chung Mong Lee, Milan Lesko, Robin Moris, Gabriele Moser, Antal Nagy, Ting Chuen Pong, Florimond Ployette, Sebastiano B. Serpico, Guo Qiang Song, Tamas Sziranyi, Zsolt Torok, Laszlo Vigh, Aurelie Voisin.

We thank Bob Gray for reading the early draft of the manuscript and giving many suggestions to improve its content and presentation.

The COSMO-SkyMed image is provided by the Italian Space Agency (ASI), the SPOT satellite images are provided by the French Space Agency (CNES), the histological image is provided by Galderma, and other test images are courtesy of GdR ISIS-CNRS.

References

[1] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," *Journal of Computational Physics*, vol. 118, pp. 269–277, 1995.

[2] R. Azencott, "Markov fields and image analysis," in *Proceedings of Association Francaise pour la Cybernetique Economique et Technique*, Antibes, France, 1987.

[3] R. Azencott, "Parallel simulated annealing: An overview of basic techniques," in *Simulated Annealing: Parallelization Techniques*, (R. Azencott, ed.), pp. 37–46, New York, NY: John Wiley & Sons, 1992.

[4] R. Azencott, ed., *Parallel Simulated Annealing. Parallelization Techniques.* New York, NY: John Wiley & Sons, 1992.

[5] R. Azencott and C. Graffigne, "Parallel annealing by periodically interacting multiple searches: Acceleration rates," in *Simulated Annealing: Parallelization Techniques*, (R. Azencott, ed.), pp. 81–90, John Wiley & Sons: New York, NY, 1992.

[6] L. Baratchart, M. Berthod, and L. Pottier, "Optimization of positive generalized polynomials under l^p constraints," Technical Report RR-2750, INRIA, December 1995.

[7] A. Barbu and S.-C. Zhu, "Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1239–1253, 2005.

[8] Y. Bard, *Nonlinear Parameter Estimation.* New York, NY: Academic Press, 1974.

[9] S. A. Barker and P. J. W. Rayner, "Unsupervised image segmentation using Markov random field models," *Pattern Recognition*, vol. 33, pp. 587–602, April 2000.

[10] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*. London: Academic Press, 1990.

[11] C. Benedek and T. Sziranyi, "Change detection in optical aerial images by a multi-layer conditional mixed markov model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 3416–3430, October 2009.

[12] C. Benedek, T. Sziranyi, Z. Kato, and J. Zerubia, "A multi-layer MRF model for object-motion detection in unregistered airborne image-pairs," in *Proceedings of International Conference on Image Processing*, pp. 141–144, San Antonio, Texas, USA, September 2007.

[13] C. Benedek, T. Sziranyi, Z. Kato, and J. Zerubia, "A three-layer MRF model for object motion detection in airborne images," Research Report 6208, INRIA, France, June 2007.

[14] C. Benedek, T. Sziranyi, Z. Kato, and J. Zerubia, "Detection of object motion regions in aerial image pairs with a multilayer Markovian model," *IEEE Transactions on Image Processing*, vol. 18, pp. 2303–2315, October 2009.

[15] M. Berthod, Z. Kato, and J. Zerubia, "DPA: A deterministic approach to the MAP," *IEEE Transactions on Image Processing*, vol. 4, pp. 1312–1314, September 1995.

[16] J. Besag, "Spatial interaction and the statistical analysis of lattice systems (with discussion)," *Journal of the Royal Statistical Society, Series B*, vol. 36, no. 2, pp. 192–236, 1974.

[17] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, Series B*, vol. 48, no. 3, pp. 259–302, 1986.

[18] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *Proceedings of European Conference on Computer Vision*, pp. 428–441, 2004.

[19] A. Blake and A. Zisserman, *Visual Reconstruction*. USA: MIT Press, 1987.

[20] T. Blaskovics, Z. Kato, and I. Jermyn, "A Markov random field model for extracting near-circular shapes," in *Proceedings of International Conference on Image Processing*, pp. 1073–1076, Cairo, Egypt, November 2009.

[21] E. Boros, P. L. Hammer, and X. Sun, "Network flows and minimization of quadratic pseudo-boolean functions," Research Report RRR 17-1991, RUTCOR, May 1991.

[22] C. Bouman, "A multiscale image model for Bayesian image segmentation," Technical Report TR-EE 91-53, Purdue University, 1991.

[23] C. Bouman and B. Liu, "Multiple resolution segmentation of texture images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 99–113, 1991.

[24] C. Bouman and M. Shapiro, "Multispectral image segmentation using a multiscale model," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 565–568, San Francisco, California, USA, March 1992.

[25] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-D image segmentation," *International Journal of Computer Vision*, vol. 70, pp. 109–131, February 2006.

[26] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-D images," in *Proceedings of International Conference on Computer Vision*, pp. 105–112, July 2001.

[27] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/ max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1124–1137, September 2004.

[28] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222–1239, November 2001.

[29] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222–1239, November 2001.

[30] B. Braathen, W. Pieczynski, and P. Masson, "Global and Local Methods of Unsupervised Bayesian Segmentation of Images," *Machine Graphics and Vision*, vol. 2, no. 1, pp. 39–52, 1993.

[31] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[32] O. Catoni and A. Trouvé, "Parallel annealing by multiple trials," in *Simulated Annealing: Parallelization Techniques*, (R. Azencott, ed.), pp. 129–144, New York, NY: John Wiley & Sons, 1992.

[33] G. Celeux and J. Diebolt, "The SEM algorithm: A probabilisitic teacher algorithm derived from the EM algorithm for the mixture problem," *Computational Statistics Quaterly*, vol. 2, pp. 73–82, 1985.

[34] V. Černy, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, pp. 41–51, January 1985.

[35] B. Chalmond, "Image restoration using an estimated Markov model," *Signal Processing*, vol. 15, pp. 115–129, September 1988.

[36] B. Chalmond, "An iterative Gibbsian technique for reconstruction of M-ary images," *Pattern Recognition*, vol. 22, no. 6, pp. 747–762, 1989.

[37] B. Chalmond, *Modeling and Inverse Problems in Image Analysis*. New York, NY: Springer, 2003.

[38] T. Chan and L. Vese, "An active contour model without edges," in *Proceedings of International Conference on Scale-Space Theories in Computer Vision*, pp. 141–151, 1999.

[39] R. Chellappa and A. K. Jain, eds., *Markov Random Fields: Theory and Applications*. Academic Press, 1993.

[40] C. Chen, H. Li, X. Zhou, and S. T. C. Wong, "Constraint factor graph cut-based active contour method for automated cellular image segmentation in RNAi screening," *Journal of Microscopy*, vol. 230, pp. 177–191, May 2008.

[41] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gopinath, R. Briggs, and E. Geiser, "Using prior shapes in geometric active contours in a variational framework," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 315–328, 2002.

[42] P. Chou and C. Brown, "The theory and practice of Bayesian image labeling," *International Journal of Computer Vision*, vol. 4, no. 3, pp. 185–210, 1990.

[43] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markov random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 195–219, March 1987.

[44] L. Cohen, "On active contour models and balloons," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 53, pp. 211–218, March 1991.

[45] L. Condat, D. V. de Ville, and T. Blu, "Hexagonal versus orthogonal lattices: A new comparison using approximation theory," in *Proceedings of International Conference on Image Processing*, pp. 1116–1119, Genoa, Italy, September 2005.

[46] D. Cremers, F. Tischhauser, J. Weickert, and C. Schnorr, "Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 295–313, 2002.

[47] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 25–39, January 1983.

[48] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[49] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 800–810, August 2001.

[50] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 39–55, January 1987.

[51] H. Derin, H. Elliott, R. Cristi, and D. Geman, "Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 707–720, November 1984.

[52] H. Derin and C. S. Won, "A parallel segmentation algorithm using relaxation with varying neighborhoods and its mapping to array procesors," *Computer Vision, Graphics and Image Processing*, vol. 40, pp. 54–78, October 1987.

[53] X. Descombes, R. Morris, J. Zerubia, and M. Berthod, "Maximum likelihood estimation of Markov random field paremeters using Markov chain Monte Carlo algorithms," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 133–148, Springer 1997.

[54] P. L. Dobruschin, "The description of a random field by means of conditional probabilities and constructions of its regularity," *Theory of Probability and its Applications*, vol. XIII, no. 2, pp. 197–224, 1968.

[55] O. Faugeras and M. Berthod, "Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 412–423, 1981.

[56] T. S. Ferguson, *Mathematical Statistics. A Decision Theoretic Approach. Probability and Mathematical Statistics*. New York, NY: Academic Press, 1967.

[57] P. Fieguth, *Statistical Image Processing and Multidimensional Modeling*. New York, NY: Springer, 2011.

[58] K. Fish, "Total internal reflection fluorescence (TIRF) microscopy," *Current Protocols in Cytometry*, vol. Chapter 12, 2009.

[59] I. Gaudron and A. Trouvé, "Massive parallelization of simulated annealing: An experimental and theoretical approach for spin-glass models," in *Simulated Annealing: Parallelization Techniques*, (R. Azencott, ed.), pp. 163–186, New York, NY: John Wiley & Sons, 1992.

[60] D. Geiger and F. Girosi, "Parallel and deterministic algorithms for MRFs: Surface reconstruction and integration," in *Proceedings of European Conference on Computer Vision*, pp. 89–98, Antibes, France, 1990.

[61] D. Geiger and A. Yuille, "A common framework for image segmentation," Technical Report 89-7, Harvard Robotics Lab, 1989.

[62] S. B. Gelfand and S. K. Mitter, "On sampling methods and annealing algorithms," in *Markov Random Fields*, (R. Chellappa, ed.), pp. 499–515, Boston, MA: Academic Press, 1993.

[63] D. Geman, "Bayesian image analysis by adaptive annealing," in *Proceedings of International Geoscience and Remote Sensing Symposium*, pp. 269–277, Amherst, USA, October 1985.

[64] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[65] S. Geman, D. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, pp. 609–628, 1990.

[66] S. Geman and C. Graffigne, "Markov random field image models and their application to computer vision," Research Report, Brown University, 1986.

[67] B. Gidas, "A renormalization group approach to image processing problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 164–180, February 1989.

[68] N. Giordana and W. Pieczynski, "Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 465–475, May 1997.

[69] J.-F. Giovannelli, "Estimation of the ising field parameter thanks to the exact partition function," in *Proceedings of International Conference on Image Processing*, pp. 1441–1444, Hong Kong, China, September 2010.

[70] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River, NJ: Prentice Hall, 2008.

[71] C. Graffigne, "A parallel simulated annealing algorithm," Research Report, CNRS, Université Paris-Sud, 1984.

[72] C. Graffigne, "Parallel annealing by periodically interacting multiple searches: An experimental study," in *Simulated Annealing: Parallelization Techniques*, (R. Azencott, ed.), pp. 47–80, New York, NY: John Wiley & Sons, 1992.

[73] C. Graffigne, F. Heitz, P. Pérez, F. Prêteux, M. Sigelle, and J. Zerubia, "Hierarchical Markov random field models applied to image analysis: a review," in *Proceedings of Conference on Neural, Morphological, and Stochastic Methods in Image and Signal Processing*, San Diego, USA, July 1995.

[74] C. Graffigne, J. Zerubia, and B. Chalmond, *Analyse d'images : filtrage et segmentation*, ch. Segmentation région: approches statistiques. Masson, 1995.

[75] U. Grenander, *General Pattern Theory*. New York, NY: Oxford University Press, 1993.

[76] U. Grenander and M. Miller, "Representations of knowledge in complex systems," *Journal of the Royal Statistical Society, Series B*, vol. 56, pp. 549–603, 1994.

[77] L. Gupta and T. Sortrakul, "A Gaussian-mixture-based image segmentation algorithm," *Pattern Recognition*, vol. 31, no. 3, pp. 315–325, 1998.

[78] M. R. Gupta and Y. Chen, "Theory and use of the em algorithm," *Foundations and Trends in Signal Processing*, vol. 4, no. 3, pp. 223–296, 2010.

[79] X. Guyon, *Champs aléatoires sur réseaux: modélisations, statistique et applications*. Masson, 1992.

[80] B. Hajek, "A tutorial survey of theory and applications of simulated annealing," in *Proceedings of International Conference on Decision and Control*, pp. 755–760, Lauderdale, FL, USA, December 1985.

[81] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operations Research*, vol. 13, pp. 311–329, May 1988.

[82] P. L. Hammer, P. Hansen, and B. Simeone, "Roof duality, complementation and persistency in quadratic 0-1 optimization," *Mathematical Programming*, vol. 28, pp. 121–155, 1984.

[83] F. R. Hansen and H. Elliott, "Image segmentation using simple Markov field models," *Computer Vision, Graphics and Image Processing*, vol. 20, pp. 101–132, October 1982.

[84] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems on Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[85] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their application," *Biometrika*, vol. 57, pp. 97–109, 1970.

[86] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 695–702, Washington, DC, USA, July 2004.

[87] F. Heitz and P. Bouthemy, "Multimodal estimation of discontinuous optical flow using Markov random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1217–1232, December 1993.

[88] F. Heitz, E. Memin, P. Perez, and P. Bouthemy, "A parallel multiscale relaxation algorithm for image sequence analysis," in *Proceedings of International Colloquiumon Parallel Image Processing*, Paris, France, June 1991.

[89] F. Heitz, P. Perez, and P. Bouthemy, "Multiscale minimization of global energy functions in some visual recovery problems," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 59, no. 1, pp. 125–134, 1994.

[90] F. Heitz, P. Perez, E. Memin, and P. Bouthemy, "Parallel visual motion analysis using multiscale Markov random fields," in *Proceedings of Workshop on Motion*, Princeton, October 1991.

[91] H. P. Hiriyannaiah, G. L. Bilbro, and W. E. Snyder, "Restoration of piecewise-constant images by mean-field annealing," *Journal of the Optical Society of America A*, vol. 6, pp. 1901–1912, December 1989.

[92] R. Huang, V. Pavlovic, and D. N. Metaxas, "A graphical model framework for coupling MRFs and deformable models," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 739–746, Washington, DC, USA, June 2004.

[93] R. Hummel and S. Zucker, "On the foundations of relaxation labeling processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 267–287, May 1983.

[94] H. Ishikawa, "Exact optimization for markov random fields with convex priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1333–1336, October 2003.

[95] H. Ishikawa, "Transformation of general binary mrf minimization to the first order case," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1234–1249, June 2011.

[96] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[97] F. C. Jeng and J. M. Woods, "Compound Gauss — Markov random fields for image estimation," *IEEE Transactions on Signal Processing*, vol. 39, pp. 683–697, March 1991.

[98] B. Jeon and D. A. Landgrebe, "Classification with spatio-temporal interpixel class dependency contexts," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 30, pp. 663–672, July 1992.

[99] J. M. Jolion and A. Rosenfeld, *A Pyramid Framework for Early Vision, Engineering and Computer Science*. Dordrecht, Netherlands: Kluwer Academic Publisher, 1994.

[100] T. R. Jones, I. H. Kang, D. B. Wheeler, R. A. Lindquist, A. Papallo, D. M. Sabatini, P. Golland, and A. E. Carpenter, "Cellprofiler analyst: Data exploration and analysis software for complex image-based screens," *BMC Bioinformatics*, vol. 9, p. 482, November 2008.

[101] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[102] Z. Kato, "Segmentation of color images via reversible jump MCMC sampling," *Image and Vision Computing*, vol. 26, pp. 361–371, March 2008.

[103] Z. Kato, M. Berthod, and J. Zerubia, "A hierarchical Markov random field model for image classification," in *Proceedings of International Workshop on Image and Multidimensional Digital Signal Processing*, Cannes, France, September 1993.

[104] Z. Kato, M. Berthod, and J. Zerubia, "Multiscale Markov random field models for parallel image classification," in *Proceedings of International Conference on Computer Vision*, pp. 253–257, Berlin, Germany, May 1993.

[105] Z. Kato, M. Berthod, and J. Zerubia, "Parallel image classification using multiscale Markov random fields," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 137–140, Minneapolis, USA, April 1993.

[106] Z. Kato, M. Berthod, and J. Zerubia, "A hierarchical Markov random field model and multi-temperature annealing for parallel image classification," *Computer Graphics and Image Processing: Graphical Models and Image Processing*, vol. 58, pp. 18–37, January 1996.

[107] Z. Kato, M. Berthod, and J. Zerubia, "Parallel image classification using multiscale Markov random fields," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 137–140, Minneapolis, April 1993.

[108] Z. Kato, M. Berthod, J. Zerubia, and W. Pieczynski, "Unsupervised adaptive image segmentation," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 2399–2402, Detroit, Michigan, USA, May 1995.

[109] Z. Kato and T. C. Pong, "A Markov random field image segmentation model using combined color and texture features," in *Proceedings of International Conference on Computer Analysis of Images and Patterns*, (W. Skarbek, ed.), pp. 547–554, Warsaw, Poland, September 2001.

[110] Z. Kato and T. C. Pong, "Video object segmentation using a multicue Markovian model," in *Joint Hungarian-Austrian Conference on Image Processing and Pattern Recognition*, (D. Chetverikov, L. Czuni, and M. Vincze, eds.), pp. 111–118, Veszprem, Hungary: KEPAF, OAGM/AAPR, Austrian Computer Society, May 2005.

[111] Z. Kato and T. C. Pong, "A Markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, pp. 1103–1114, October 2006.

[112] Z. Kato and T. C. Pong, "A multi-layer MRF model for video object segmentation," in *Proceedings of Asian Conference on Computer Vision*, (P. J. Narayanan, S. K. Nayar, and H.-Y. Shum, eds.), pp. 953–962, Hyderabad, India: Springer, January 2006.

[113] Z. Kato, T. C. Pong, and G. Q. Song, "Multicue MRF image segmentation: Combining texture and color," in *Proceedings of the International Conference on Pattern Recognition*, pp. 660–663, Quebec, Canada, August 2002.

[114] Z. Kato, T. C. Pong, and G. Q. Song, "Unsupervised segmentation of color textured images using a multi-layer MRF model," in *Proceedings of International Conference on Image Processing*, pp. 961–964, Barcelona, Spain, September 2003.

[115] Z. Kato, J. Zerubia, and M. Berthod, "Image classification using Markov random fields with two new relaxation methods: Deterministic pseudo annealing and modified Metropolis dynamics," Research Report 1606, INRIA, Sophia Antipolis, France, February 1992.

[116] Z. Kato, J. Zerubia, and M. Berthod, "Satellite image classification using a modified Metropolis dynamics," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 573–576, San Francisco, California, USA, March 1992.

[117] Z. Kato, J. Zerubia, and M. Berthod, "Bayesian image classification using Markov random fields," in *Maximum Entropy and Bayesian Methods*, (A. Mohammad-Djafari and G. Demoment, eds.), pp. 375–382, Dordrecht Netherlands: Kluwer Academic Publisher, 1993.

[118] Z. Kato, J. Zerubia, and M. Berthod, "Unsupervised parallel image classification using a hierarchical Markovian model," Research Report 2528, INRIA, Sophia Antipolis, France, April 1995.

[119] Z. Kato, J. Zerubia, and M. Berthod, "Unsupervised parallel image classification using a hierarchical Markovian model," in *Proceedings of International Conference on Computer Vision*, pp. 169–174, Cambridge, MA, USA, June 1995.

[120] Z. Kato, J. Zerubia, and M. Berthod, "Unsupervised parallel image classification using Markovian models," *Pattern Recognition*, vol. 32, pp. 591–604, April 1999.

[121] D. Kersten, P. Mamassian, and A. Yuille, "Object perception as Bayesian inference," *Annual Review of Psychology*, vol. 55, pp. 271–304, 2004.

[122] C. Kervrann and F. Heitz, "A statistical model-based approach to unsupervised texture segmentation," in *Proceedings of Scandinavian Conferences on Image Analysis*, pp. 284–288, Tromso, Norway, May 1993.

[123] C. Kervrann and F. Heitz, "Statistical deformable model-based segmentation of image motion," *IEEE Transactions on Image Processing*, vol. 8, pp. 583–588, 1999.

[124] S. Khan and M. Shah, "Object based segmentation of video using color, motion and spatial information," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 746–751, Kauai, Hawaii, December 2001.

[125] R. Kindermann and J. L. Snell, *Markov Random Fields and their Applications*. Providence, RI: American Mathematical Society, 1980.

[126] S. Kirkpatrick, C. Gellatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.

[127] P. Kohli, M. P. Kumar, and P. H. Torr, "$\mathcal{P}^3$ & beyond: Move making algorithms for solving higher order functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1645–1656, September 2009.

[128] P. Kohli, L. Ladicky, and P. Torr, "Robust higher order potentials for enforcing label consistency," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[129] A. Kokaram, *Motion Picture Restoration*. London: Springer, 1998.

[130] V. Kolmogorov, "QPBO algorithm," *software*, 2007.

[131] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts-a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1274–1279, 2007.

[132] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 147–159, February 2004.

[133] V. Krylov, G. Moser, S. Serpico, and J. Zerubia, "Enhanced dictionary-based SAR amplitude distribution estimation and its validation with very high-resolution data," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 8, no. 1, pp. 148–152, 2011.

[134] V. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, "Supervised high resolution dual polarization SAR image classification by finite mixtures and copulas," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 3, pp. 554–566, 2011.

[135] S. Kumar and M. Hebert, "Discriminative fields for modeling spatial dependencies in natural images," in *Proceedings of Neural Information Processing Systems*, 2003.

[136] S. Kumar and M. Hebert, "A hierarchical field framework for unified context-based classification," in *Proceedings of International Conference on Computer Vision*, pp. 1284–1291, 2005.

[137] P. V. Laarhoven and E. Aarts, *Simulated Annealing: Theory and Applications*. Dordrecht: Kluwer Academic Publisher, 1987.

[138] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, "Graph cut based inference with co-occurrence statistics," in *Proceedings of European Conference on Computer Vision*, (K. Daniilidis, P. Maragos, and N. Paragios, eds.), pp. 239–253, Crete, Greece, September 2010.

[139] J.-M. Laferte, P. Perez, and F. Heitz, "Discrete Markov modeling and inference on the quad-tree," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 390–404, 2000.

[140] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning*, pp. 282–289, 2001.

[141] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 799–813, August 1989.

[142] S. Lakshmanan and H. Derin, "Gaussian Markov Random fields at multiple resolution," in *Markov Random Fields*, pp. 131–157, San Diego: Academic Press, 1993.

[143] M. Leskó, Z. Kato, A. Nagy, I. Gombos, Z. Török, L. V. Jr, and L. Vígh, "Live cell segmentation in fluorescence microscopy via graph cut," in *Proceedings of the International Conference on Pattern Recognition*, pp. 1485–1488, Istambul, Turkey, August 2010.

[144] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. New York NY: Springer, 3rd Edition, 2009.

[145] E. Littmann and H. Ritter, "Adaptive color segmentation — a comparison of neural and statistical methods," *IEEE Transactions on Neural Networks*, vol. 8, pp. 175–185, January 1997.

[146] S. Liu-Yu, "Reconnaissance de formes par vision par ordinateur: application à l'identification de foraminifères planctoniques," PhD thesis, University of Nice, Sophia Antipolis, France, June 1992.

[147] F. Marques, J. Cunillera, and A. Gasull, "Hierarchical segmentation using compound Gauss-Markov random fields," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, San Francisco, California, USA, March 1992.

[148] J. L. Marroquin, "Probabilistic solution of inverse problems," PhD thesis, MIT-Artificial Intelligence Lab., USA, 1985.

[149] P. Masson and W. Pieczynski, "SEM Algorithm and unsupervised statistical segmentation of satellite images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, pp. 618–633, May 1993.

[150] E. Memin, "Algorithmes et architectures parallèles pour les approches markoviennes en analyse d'image," PhD thesis, University of Rennes I, France, 1993.

[151] E. Memin, F. Heitz, and F. Charot, "Efficient parallel non-linear multigrid relaxation algorithms for low-level vision applications," *Journal of Parallel Distributed Computing*, vol. 29, pp. 96–103, August 1995.

[152] D. Metaxas, *Physics-based Deformable Models: Applications to Computer Vision, Graphics and Medical Imaging*. Kluwer Academic Publisher, 1997.

[153] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[154] W. Michiels, E. H. L. Aarts, and J. Korst, *Theoretical Aspects of Local Search*. New York, NY: Springer, 2007.

[155] M. Miller and L. Younes, "Group actions, homeomorphisms, and matching: A general framework," *International Journal of Computer Vision*, vol. 41, pp. 61–84, February 2001.

[156] M. I. Miller, U. Grenander, O. J. A., and D. L. Snyder, "Automatic target recognition organized via jump-diffusion algorithms," *IEEE Transactions on Image Processing*, vol. 6, pp. 157–174, January 1997.

[157] R. Morris, X. Descombes, and J. Zerubia, "Fully Bayesian image segmentation — an engineering perspective," in *Proceedings of International Conference on Image Processing*, Santa Barbara, USA, October 1997.

[158] G. Moser, V. Krylov, S. Serpico, and J. Zerubia, "High resolution SAR-image classification by Markov random fields and finite mixtures," in *Proceedings of SPIE IS&T/SPIE Electronic Imaging*, pp. 1–8, San Jose, USA, January 2010.

[159] G. Moser, S. B. Serpico, and J. Zerubia, "Dictionary-based Stochastic Expectation Maximization for SAR amplitude probability density function estimation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 1, pp. 188–199, 2006.

[160] J. Moussouris, "Gibbs and Markov random system with constrainsts," *Journal of Statistical Physics*, vol. 10, pp. 11–33, January 1974.

[161] D. Mumford, "The Bayesian rationale for energy functionals," in *Geometry-Driven Diffusion in Computer Vision*, (B. Romeny, ed.), pp. 141–153, Boston, MA: Kluwer Academic Publisher, 1994.

[162] D. Mumford, "Pattern theory: A unifying perspective," in *Perception as Bayesian Inference*, (D. Knill and W. Richards, eds.), pp. 25–62, Cambridge University Press, 1996.

[163] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[164] E. Nagy, Z. Balogi, I. Gombos, M. Akerfelt, A. Bjorkbom, G. Balogh, Z. Torok, A. Maslyanko, A. Fiszer-Kierzkowska, K. Lisowska, P. Slotte, L. Sistonen, I. Horvath, and L. Vigh, "Hyperfluidization-coupled membrane microdomain reorganization is linked to activation of the heat shock response in a murine melanoma cell line," in *Proceedings of National Academy Science USA*, pp. 7945–7950, 2007.

[165] R. B. Nelsen, *An Introduction to Copulas*. New York, NY: Springer, 2nd Edition, 2006.

[166] J. C. Noordam, G. W. Otten, A. J. M. Timmermans, and B. v. Zwol, "High-speed potato grading and quality inspection based on a color vision system," in *Proceedings of SPIE Machine Vision Applications in Industrial Inspection*, (K. W. T. Jr., ed.), pp. 206–220, 2000.

[167] J.-M. Odobez and P. Bouthemy, "MRF-based motion segmentation exploiting a 2D motion model robust estimation," in *Proceedings of International Conference on Image Processing*, pp. 628–631, Washington, DC, USA, October 1995.

[168] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar images*. New Jersey, NJ: SciTech Publishing, 2004.

[169] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *International Journal of Computer Vision*, vol. 46, pp. 223–247, 2002.

[170] G. Parisi, *Statistical Field Theory*. Westview Press, 1998.

[171] P. Perez, "Champs markoviens et analyse multirésolution de l'image: Application à l'analyse du mouvement," PhD thesis, University of Rennes I, France, 1993.

[172] P. Perez and F. Heitz, "Multiscale Markov random fields and constrained relaxation in low level image analysis," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pp. 61–64, San Francisco, California, USA, March 1992.

[173] P. Pérez and F. Heitz, "Restriction of Markov random fields on graphs. Application to multiresolution image analysis," Research Report 2170, INRIA, March 1994.

[174] H. Permuter, J. Francos, and I. Jermyn, "A study of Gaussian mixture models of colour and texture features for image classification and segmentation," *Pattern Recognition*, vol. 39, pp. 695–706, April 2006.

[175] W. Pieczynski, "Statistical image segmentation," in *Proceedings of Machine Graphics and Vision*, pp. 261–268, Naleczow, Poland, May 1992.

[176] S. Rajasekaran, "On the convergence time of simulated annealing," Research Report MS-CIS-90-89, University of Pennsylvania, Department of Computer and Information Science, USA, November 1990.

[177] S. Raman, B. Parvin, C. Maxwell, and M. H. Barcellos-Ho, "Geometric approach to segmentation and protein localization in cell cultured assays," in *Advances in Visual Computing*, pp. 427–436, November 2005.

[178] A. Rangarajan and R. Chellappa, "Markov random field models in image processing," in *Handbook of Brain Theory and Neural Networks*, (A. M.A, ed.), pp. 564–567, Cambridge, MA: MIT Press, 1995.

[179] A. Rangarajan, B. Manjunath, and R. Chellappa, "Markov random fields and neural networks with applications in early vision problems," in *Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections*, (I. Sethi and A. Jain, eds.), Amsterdam: Elsevier Science Publishers, 1991.

[180] B. Reddy and B. Chatterji, "An FFT-based technique for translation, rotation and scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, 1996.

[181] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, pp. 195–239, April 1984.

[182] E. Rignot and R. Chellappa, "Maximum a posteriori classification of multifrequency, multilook, synthetic aperture radar intensity data," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 10, no. 4, pp. 573–582, 1993.

[183] M. Rochery, I. H. Jermyn, and J. Zerubia, "Higher order active contours and their application to the detection of line networks in satellite imagery," in *Proceedings of Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, New York, NY: ICCV, Nice, France, October 2003.

[184] Y. Rosanov, *Markov Random Fields*. New York, NY: Springer, 1982.

[185] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary MRFs via extended roof duality," in *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, USA, June 2007.

[186] M. Rousson and N. Paragios, "Shape priors for level set representations," in *Proceedings of European Conference on Computer Vision*, pp. 78–92, Copenhagen, Denmark, 2002.

[187] C. Russell, D. Metaxas, C. Restif, and P. Torr, "Using the P^n Potts model with learning methods to segment live cell images," in *International Conference on Computer Vision*, pp. 1–8, Rio de Janeiro, Brazil, October 2007.

[188] C. Samson, L. Blanc-Feraud, G. Aubert, and J. Zerubia, "A level set model for image classification," *International Journal of Computer Vision*, vol. 40, no. 3, pp. 187–197, 2000.

[189] S. J. Sangwine and R. E. N. Horne, eds., *The Colour Image Processing Handbook*. London: Chapman & Hall, 1998.

[190] M. Schneider, P. Fieguth, W. Karl, and A. Willsky, "Multiscale statistical methods for the segmentation of images," *IEEE Transactions on Image Processing*, vol. 9, pp. 442–455, March 2000.

[191] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge: MIT Press, 2001.

[192] J. Sethian, *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge Monograph on Applied and Computational Mathematics*. Cambridge University Press, 1999.

[193] T. Shima, S. Sugimoto, and M. Okutomi, "Comparison of image alignment on hexagonal and square lattices," in *Proceedings of International Conference on Image Processing*, pp. 141–144, Hong Kong, China, September 2010.

[194] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, pp. 2–23, 2009.

[195] M. Sigelle, C. Bardinet, and R. Ronfard, "Relaxation of classification images by a Markov field technique — application to the geographical classification of Bretagne region," in *Proceedings of European Association of Remote Sensing*, Eger, Hungary, September 1992.

[196] M. Sigelle and R. Ronfard, "Modèles de Potts et relaxation d'images de labels par champs de markov," *Traitement du Signal*, vol. 9, pp. 449–458, March 1993.

[197] T. Simchony, R. Chellappa, and Z. Lichtestein, "Image estimation using 2-D noncausal Gauss Markov random fields," in *Image Restoration*, (A. Katsaggelos, ed.), pp. 109–141, Springer, 1991.

[198] R. H. Swendsen and J.-S. Wang, "Nonuniversal critical dynamics in Monte Carlo simulations," *Physical Review Letters*, vol. 58, pp. 86–88, 1987.

[199] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1068–1080, June 2008.

[200] H. L. Tan, S. B. Gelfand, and E. J. Delp, "A cost minimization approach to edge detection using simulated annealing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 3–18, January 1991.

[201] A. Trouvé, "Massive parallelization of simulated annealing: A mathematical study," in *Simulated Annealing: Parallelization Techniques*, (R. Azencott, ed.), pp. 145–164, John Wiley & Sons, 1992.

[202] Z. Tu, X. Chen, A. Yuille, and S.-C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of Computer Vision*, vol. 63, pp. 113–140, July 2005.

[203] Z. Tu and S.-C. Zhu, "Image segmentation by data-driven Markov chain Monte Carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 657–673, May 2002.

[204] F. Tupin, H. Maitre, J.-F. Mangin, J.-M. Nicolas, and E. Pechersky, "Detection of linear features in SAR images: Application to road network extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, pp. 434–453, March 1998.

[205] N. Vandenbroucke, L. Macaire, and J. Postaire, "Color image segmentation by supervised pixel classification in a color texture feature space. Application to soccer image segmentation," in *Proceedings of the International Conference on Pattern Recognition*, pp. 621–624, Barcelona, Spain, 2000.

[206] J. Verbeek and B. Triggs, "Scene segmentation with CRFs learned from partially labeled images," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 1553–1560, January 2008.

[207] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, June 2008.

[208] Ľubor Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr, "What, where and how many? combining object detectors and CRFs," in *Proceedings of European Conference on Computer Vision*, (K. Daniilidis, P. Maragos, and N. Paragios, eds.), pp. 424–437, Crete, Greece: Springer, September 2010.

[209] A. Voisin, V. Krylov, G. Moser, S. Serpico, and J. Zerubia, "Classification of very high resolution sar images of urban areas," Technical Report, INRIA Sophia Antipolis Mediterranee, 2011.

[210] A. Voisin, V. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, "Multichannel hierarchical image classification using multivariate copulas," in *IS&T/SPIE Electronic Imaging 2012, Proceedings of SPIE, volume 8296, 82960K*, pp. 22–26, San Francisco, USA, January 2012.

[211] A. Voisin, G. Moser, V. Krylov, S. B. Serpico, and J. Zerubia, "Classification of very high resolution SAR images of urban areas by dictionary-based mixture models, copulas and Markov random fields using textural features," in *Proceedings of SPIE*, p. 78300O, 2010.

[212] H. M. Wallach, "Conditional random fields: An introduction," Technical Report MS-CIS-04-21, University of Pennsylvania, USA, February 2004.

[213] Y. Wang and S.-C. Zhu, "Analysis and synthesis of textured motion: Particles and waves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1348–1363, October 2004.

[214] G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer, 2003.

[215] J. H. Woods, "Two-dimensional discrete Markovian fields," *IEEE Transactions on Information Theory*, vol. 18, pp. 232–240, March 1972.

[216] F. Y. Wu, "The Potts model," *Reviews of Modern Physics*, vol. 54, pp. 235–268, January 1982.

[217] C. Xu and J. L. Prince, "Snakes, shapes, gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, pp. 359–369, March 1998.

[218] S. Yu and M. Berthod, "A game strategy approach for image labeling," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 32–37, 1995.

[219] J. Zerubia and R. Chellappa, "Mean field approximation using compound Gauss-Markov random field for edge detection and image restoration," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, USA, 1990.

[220] J. Zerubia and R. Chellappa, "Mean field annealing using Compound Gauss-Markov Random fields for edge detection and image estimation," *IEEE Transactions on Neural Networks*, vol. 8, pp. 703–709, July 1993.

[221] J. Zerubia and C. Graffigne, *Analyse d'images: filtrage et segmentation*, ch. Segmentation contour: Approches statistiques. Masson, 1995.

[222] J. Zerubia, Z. Kato, and M. Berthod, "Multi-temperature annealing: A new approach for the energy-minimization of hierarchical Markov random field models," in *Proceedings of the International Conference on Pattern Recognition*, pp. 520–522, Jerusalem, Israel, October 1994.

[223] J. Zerubia and F. Ployette, "Detection de contours et restauration d'image par des algorithmes deterministes de relaxation. Mise en oeuvre sur la machine a connexions CM2," Research Report 1291, INRIA, September 1991.

[224] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.