TOMOGRAPHIC RECONSTRUCTION OF LABEL IMAGES USING GIBBS PRIORS

by

HSTAU Y. LIAO

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2005

UMI Number: 3187391

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This manuscript has been read and accepted for the Graduate Faculty in Computer Sicence in satisfaction of the dissertation requirements for the degree of Doctor of Philosophy.

Date	Dr. Gabor T. Herman, Chair of Examining Committee
Date	Dr. Theodore Brown, Executive Officer

Dr. Robert M. Haralick

Dr. T. Yung Kong

Dr. Yair Censor

THE CITY UNIVERSITY OF NEW YORK

Abstract

TOMOGRAPHIC RECONSTRUCTION OF LABEL IMAGES USING GIBBS PRIORS

by

HSTAU Y. LIAO

Advisor: Dr. Gabor T. Herman

Our aim is to produce a tessellation of space into small voxels and, based on only a few tomographic projections of an object, assign to each voxel a label that indicates one of the components of interest constituting the object. Examples of application are in the areas of electron microscopy, industrial non-destructive testing, cardiac imaging, etc.

Current approaches first reconstruct the density distribution from the projections and then segment (label) this distribution. We instead postulate a low level prior knowledge regarding the underlying distribution of label images and then directly estimate the label image based on the prior and the projections. In particular we show, in the binary (i.e., two labels) case, that the marginal posterior mode estimator outperfoms the widely-known maximum a posteriori probability estimator.

In terms of the label misclassification in the reconstructions, our direct labeling method was experimentally proved (in the binary case) to be superior to the current approaches, but performs less satisfactory under a detectability measure. We discuss possible improvements of our direct labeling methods.

This work is dedicated in loving memory to my grandfather Wen-tong.

Acknowledgments

There are indeed many, many individuals who have assisted me in many, many ways during my road to the doctorate, and to all I owe a debt of gratitude for all they have done for me.

The first person to whom this debt is owed is my advisor, Gabor T. Herman, for his support, invaluable suggestions and encouragement on issues that were beyond my research. I truly admire his creative way of thinking. Following him to CUNY from the University of Pennsylvania was one of the wisest decisions I ever made. I thank Alvaro De Pierro for introducing the two of us.

I would like to give an especially big thank-you to three individuals for their indispensable help, without which my research experiments could not have possibly existed: Roberto Marabini, Edgar Garduño, and Bruno M. Carvalho. I cannot forget, of course, Lajos Rodek, Florian Lengyel, Laszlo Rusko, Miroslaw Kalinowski, Deniz Sarioz, Stu W. Rowland, Eilat Vardi-Gonen, and Joel Dubowy.

I greatly appreciate the kindness, attentions and valuable time of my examining committee members: Yair Censor, for his comments and advice on optimization related issues; Robert M. Haralick, for his numerous and valid critiques, and T. Yung Kong. I also thank Jesse Barlow, Erkki Somersalos, and Stephan Altmueller for their suggestions.

I am indebted to −and happily not in debt because of− the institutions that supported me financially during my studies through various grants, fellowships and employment opportunities, specifically the National Institute of Health; the National Science Foundation; and The City University of New York.

Certainly research was not the only ingredient in the development of my dissertation. My life outside academia was sustained thanks to Justin Huhn whom I first met at UPenn

and who has been a big friend, Joe Driscoll and Atish Bagchi for their staunch help and advice on making big decisions, Charlie, Szu-ting, Andrea, Janet, Yi-ting, Jeremy, Joel, Ran, and Eilat.

I express gratitude to my parents, sisters, and brother who have always been there, and without whose encouragement, understanding, and unconditional love, this journey would not have even been begun.

Finally, I thank Argentina, in particular the Instituto Balseiro, the Instituto Politécnico Superior, and the Olimpíada Mathemática Argentina for providing me with a high quality education and igniting my desire to continue in it.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the most important imaging techniques, known as *computerized tomography* (CT) [39], has undoubtedly revolutionized diagnostic radiology in the last thirty years. In CT the density distribution within the human body is calculated from the measured attenuation of X-rays through the body. Since an enormous variety of densities may occur in the body, a lot of projections are necessary to ensure the accurate reconstruction of their distribution. Among the many other CT applications (such as angiography, astronomy, industrial non-destructive testing or reverse engineering, etc.) is electron tomography (ET) [27] of biological macromolecules, which aims at their structural determination, so that their biological functions can be inferred.

In many situations the ultimate aim is not the density distribution itself but rather a distribution of *labels* that correspond to one of the components (such as protein or RNA in macromolecules) constituting the object, and there are good reasons (such as damage by radiation) why only a few projections can be collected. Making use of the knowledge that the reconstruction should contain only a few labels to make up for the lack of availability of the number of projections typically required in CT is the essence of *discrete tomography*

[43]. Another powerful prior is the knowledge regarding the general shapes and sizes of characteristic structures, which can be expressed quantitatively in the form of a Gibbs distribution [94]. Such prior knowledge needs to be learned from a training set of typical correctly labeled images [59].

The focus of this dissertation is the incorporation of the Gibbs priors into discrete tomography, as motivated primarily by three-dimensional (3D) electron tomography. The reconstruction methods developed here are applicable not only to ET but also to many other areas, such as angiography, cardiac imaging, industrial non-destructive testing, etc, in all of which a labeling of the object is sought, based on a few (usually less than ten) projections. We think that when only a few noisy projections are available, a direct estimation of the unknown label image based on them should be significantly more effective and robust than current approaches of first reconstructing (using methods of CT) and then segmenting to obtain the label image. It turned out that in our experiments, our direct labeling approach did produce in general images with fewer misclassified voxels than a CT method known as ART with blobs (see, e.g., [69]) but performs less satisfactory in terms of a detectability measure. In this dissertation we emphasize on the optimization algorithms, as well as image modeling using Gibbs priors.

In Chapter 2 we explain briefly the physics of imaging macromolecules by electron tomography, current CT-type reconstruction techniques, and the motivation for using Gibbs priors within this context. In Chapter 3 we define a Gibbs distribution and present various models, each of which is associated with a particular statistics on the sizes and shapes of the characteristic structures. A Gibbs distribution is uniquely determined by a set of parameters, which can be determined based on a set of label images that are typical for a specific application area. Chapter 4 discusses most known parameter estimation methods. (Partial results of Chapters 3 and 4 are published in [59].) In Chapter 5 we pose the re-

construction task as an optimization problem and present several algorithms for estimating the unknown label image from its projections; results of experiments with 2D images are demonstrated in Chapter 6 (they also appeared partly in [57, 58, 60, 62, 63, 64]). Chapter 7 considers reconstruction of 3D volumes using "3D Gibbs priors," based on more realistic projection data (partial results of this chapter are published in [61]). All the experimental results are demonstrated for the binary (i.e., two labels) case. A methodology for objective comparison of algorithms is also treated. Finally, in Chapter 8 we give the conclusions.

A brief warning on the notational convention. A random variable or a random vector is usually denoted in bold font. We use $Prob(\,)$ to denote the probability of an event. For example, $Prob(\mathbf{x} = x)$ denotes the probability (or probability density, in the continuous case) that a random variable (or random vector) $\mathbf{x}$ takes value x. Similarly, by $Prob(\mathbf{x} = x|\mathbf{y} = y)$ we denote the conditional probability (or probability density) that $\mathbf{x}$ equals x, given that the random variable $\mathbf{y}$ takes value y. Very often, however, we abbreviate $Prob(\mathbf{x} = x)$ and $Prob(\mathbf{x} = x|\mathbf{y} = y)$ by, respectively, $P_1(x)$ and $P_2(x|y)$, for some probability functions P_1 and P_2.

Chapter 2

Motivation and specific objective

2.1 Electron microscopy of macromolecules

The understanding of the macromolecular mechanisms of the key cell functions has great impacts such as the discovery of new drugs. In particular, protein macromolecules can act as enzymes that accelerate reactions involving smaller molecules, transporters of molecules across membrane's, providers of cell rigidity, etc. [49]. Such capabilities are possible thanks to their three-dimensional (3D) structure (or *conformation*), and therefore its knowledge is essential in the inference of the biological functions of the proteins [53].

There are various technologies for learning the conformations of biological macromolecules, such as nuclear magnetic resonance spectroscopy, X-ray crystallography, and *electron microscopy* (EM); with the last one possessing important advantages over the other two. For example, it can produce images within a wide range of resolution (from the order of atomic resolution to about 0.1 μm), it gives the amplitude and phase of the Fourier transform of the projections, and it can work with non-crystalline specimens [15]. A transmission electron microscope (TEM) measures indirectly the projection of the Coulomb

potential distribution [26] of the specimen, which, following the terminologies in computerized tomography (CT), we term *density*.

To be more quantitatively precise, let z be the direction of the electron beam (see Figure 2.1), $\mathbf{r} = (r_x.r_y)$, $I(\mathbf{r})$ be the recorded projection image at the viewing screen, and $P(\mathbf{r})$ be

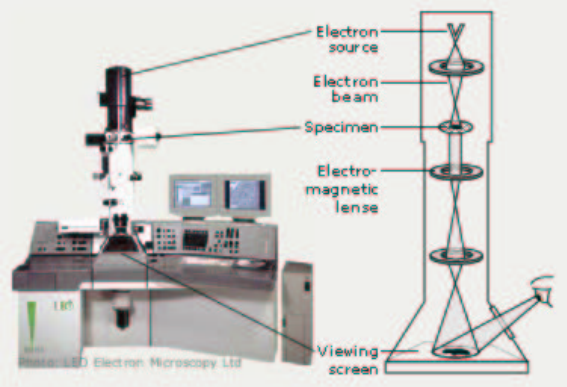

Figure 2.1: A transmission electron microscopy (by LEO Electron Microscopy Ltd) and its schematic operation. Picture courtesy of Nobelprize.org (www. nobelprize. org). Copyright ©2005 The Nobel Foundation.

the *projection* $P(\mathbf{r}) = \int_{-\infty}^{\infty} C(\mathbf{r},z)dz$ of the density (Coulomb potential) distribution $C(\mathbf{r},z)$ in the specimen. Then a very simplified model (ignoring noise and the *aperture function*, and assuming the *weak-phase approximation*) establishes that

$$\{\mathcal{F}[I(\mathbf{r})]\}(\mathbf{k}) = \{\mathcal{F}[P(\mathbf{r})]\}(\mathbf{k})W(\mathbf{k}), \tag{2.1}$$

where $\mathcal{F}$ is the two-dimensional *Fourier transform* [9], $\mathbf{k} = (k_x, k_y)$ is the spatial frequency, and $W(\mathbf{k})$ is known as the *contrast transfer function* (CTF) that takes into account the various imperfections (e.g., aberrations) of the lens. Several efforts have been made for correcting the CTF [87, 98].

The projection taking at different angles is made possible by tilting the specimen usually up to $\pm 70^\circ$ with respect to the direction of the electron beam. Such angular restriction is

due to the limitation on the thickness of the specimen, which is an important requirement in order for (2.1) to be reasonable.

A major difficulty in electron microscopy is the damage of the specimen caused by electron radiation [32]. One possible solution is *staining*, in which the specimen is covered by a metallic "cast" that is not destroyed by radiation. This technique, however, losses the internal structure below the stained surface, and the resulting resolution is too low for high resolution EM [90]. The other technique is *cryo EM*, in which the specimen being imaged is maintained at temperature low enough, so that the extent of the radiation damage can be reduced by as much as an order of magnitude [28, 33]. The challenge for achieving high resolution with EM of unstained specimens is therefore that doses that are high enough to get a good signal-to-noise (SNR) ratio lead to unacceptable specimen damage, while doses that are low enough to preserve the specimen generate images that may be too noisy for reconstruction.

Single particle techniques [77, 92] overcome the poor SNR by aligning and averaging a projection image with hundreds (or thousands) of "copies" of the same macromolecule randomly oriented. Unlike this type of technique, which is hampered by the impurities and heterogeneities of the macromolecules, *electron tomography* (ET) [3, 67] consists of the structural determination of "one of a kind" objects, such as whole cells, in which averaging methods are not applicable.

2.2 Current reconstruction techniques

In this section by "reconstruction" we mean the recovery of the density distribution (or *gray value*s) of the specimen from the projections, which are carried out by CT techniques. However, quite often the ultimate goal is not to recover such a distribution, but instead to

identify at each point of the reconstructed image the component (ice, protein, RNA, etc.) to which it corresponds. Such labeling (segmentation) of the reconstruction would require an additional step (see the end of this section). Later we discuss our approaches that will produce a label image *directly* from the projections. For now we concentrate on the current reconstruction techniques.

2.2.1 Gray value reconstruction

Under the assumptions for which (2.1) is approximately correct, it is in principle possible to reconstruct the density distribution (gray values) from the measured projections $I(\mathbf{r})$, after the correction for the CTF. The most known reconstruction techniques applied to EM are the *weighted back-projection* (WBP, which is related to the *convolution method* in [39, Chapter 8]), method like the *Fourier method* (see e.g., [39, Chapter 9]), and iterative methods, such as the *algebraic reconstruction techniques* (ART) [34] (see also Appendix F) and the *simultaneous iterative reconstruction techniques* [39, Chapter 12].

The WBP and the Fourier methods gained popularity early when iterative methods were considered to be too computational demanding. Since both the WBP and the Fourier methods use the Fourier transform, a well known efficient algorithm (the *Fast Fourier Transform*) that implements it is exploited. One way of evaluating a reconstruction method is to reconstruct a centrally located point. The resulting reconstruction is called the *impulse response* that one would hope that it is as close as possible to an *impulse* (Dirac's delta function or its multiple). In the case of WBP and when the number of projections is low, the point response contains high oscillations close to the origin [82]. As for the Fourier method, a problem with it is the necessity of interpolating in Fourier space.

Neither the WBP nor the Fourier method is capable of including the nature of the noise,

as do the iterative methods. The iterative methods can also incorporate nonlinear constraints in the algorithm. In particular, ART have been proved to outperform other techniques for various reconstruction tasks [40, 50, 68, 85].

In ART the image solution $C(x,y,z)$ is approximated (expanded) by a weighted sum of *basis functions*, each of which is a translated *basic basis function* $b(x,y,z)$:

$$C(x,y,z) = \sum_{i,j,k} c_{ijk} b\left(x - x_i, y - y_j, z - z_k\right). \tag{2.2}$$

(This is the reason why ART belong to the so-called *series expansion methods* [39, Chapter 11].) Usually a basic basis function has symmetries (with respect to a point) and is *localized*, meaning that the function support is bounded and is much smaller than the size of the image. A good example is a computer digitization (approximation) of an image by a weighted sum of basis functions with cubic support that are valued one inside the support and zero otherwise. The weights c_{ijk} (or *coefficients*) of the expansion are the values of the digitized image in each cubic voxel. The set of the center points of the basis functions defines the *grid* of the expansion. For cubic voxels, the corresponding grid is typically the *cubic grid*

$$S_\rho = \left\{ v = \rho\left(v_1, v_2, v_3\right) \mid v \in \mathbb{Z}^3 \right\}, \tag{2.3}$$

where $\rho > 0$ is the sampling unit. There is another type of grid in the literature of image reconstructions, which, combined with a particular basic basis function, yields efficient and successful electron microscopic reconstructions [66, 67]. This pair is the *body centered cubic* (BCC) grid (see Figure 2.2)

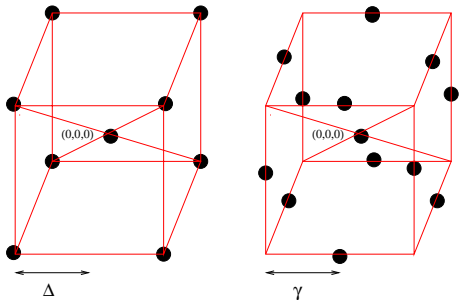

Figure 2.2: The body centered cubic (BCC) grid with sampling unit Δ and the face centered cubic (FCC) grid with sampling unit γ.

$$B_\Delta = \left\{ v = \Delta(v_1, v_2, v_3) \mid v \in \mathbb{Z}^3, v_1 \equiv v_2 \equiv v_3 (\text{mod} 2) \right\} \tag{2.4}$$

($\Delta > 0$ is the sampling unit) together with spherically symmetric basis functions known as *blobs*.

To be more precise, a blob is a *generalized Kaiser-Bessel window function* [54] in the radial direction. Such function has a maximum at the center of the support and decays monotonically and smoothly to zero at the boundary. Two parameters of a blob that concern us are the radius a of the support and the parameter α that controls how rapidly the blob function reaches to zero. A blob has the desired property that its spectrum amplitude vanishes very quickly outside the *effective bandwidth* [54].

In real applications, we can only collect a finite number of measured line integrals of the density distribution. Let w_l be one of the measured integrals along a line l, then in the absence of noise for an image function in the form of (2.2), it must be that

$$w_l = \sum_{i,j,k} c_{ijk} r_{ijk}(l), \tag{2.5}$$

where $r_{ijk}(l)$ denotes the integral of the basis function $b\left(x - x_i, y - y_j, z - z_k\right)$ along l, which is uniquely determined by the geometry of data collection. When noise is present,

as it is the case in reality, the equality in (2.5) becomes an approximation. Thus, given a vector w whose components are the measured line integrals, ART attempts to estimate the coefficients c_{ijk} by solving the system of linear equations

$$Rc = w, \tag{2.6}$$

where R, known as the *projection matrix*, contains all the $r_{ijk}(l)$ and c is the vector of coefficients. In the case where (2.6) is a consistent system, ART provides the minimum norm solution (see [39, Chapter 11] and also Appendix F).

Based on our finding about the various reconstruction algorithms that we just discussed, we use ART as *the* reconstruction algorithm against which our direct labeling approach will be compared. In the preliminary experiments on 2D images (Chapter 6) we employ the square grid (thus, basis functions are the usual square pixels); while for 3D images (Chapter 7), blobs on a BCC grid are exploited.

Before proceeding further, we define *reciprocal grids* [52] (see also Appendix G). Given a grid E, its reciprocal grid $\widetilde{E}$ is formed by the center points of the impulses that form the Fourier transform of impulses centered at the points of E. For example, the reciprocal grid of a cubic grid is another cubic grid $\widetilde{(S_\rho)} = \frac{1}{\rho^3} S_{1/\rho}$; whereas the reciprocal grid corresponding to a BCC grid is the *face centered cubic* (FCC) grid: for a given $\gamma > 0$ a FCC grid is defined by

$$F_\gamma = \left\{ v = \gamma(v_1, v_2, v_3) \mid v \in \mathbb{Z}^3, \sum_{p=1}^{3} v_p \equiv 0 (\mathrm{mod} 2) \right\}. \tag{2.7}$$

In Appendix G we show that $\widetilde{(B_\Delta)} = \frac{1}{(2\Delta)^3} F_{1/(2\Delta)}$. The associated voxels (i.e., the *Voronoi neighborhoods* [41] of the grid points) of S_ρ are cubes of side length ρ; whereas the asso-

ciated voxels of a FCC grid are rhombic-dodecahedra.

In order to apply ART with blobs, we need to decide on the values for the parameters Δ, a, and α. For the the sampling unit Δ of the BCC grid, we adopt the *equivalent grids* criterion [69], which establishes that two grids E and E' are "equivalent," from the viewpoint of image representation by blobs, if the following is the case. If we place identical and non-overlapping spheres on the grid points in each of the grids $\widetilde{E}$ and $\widetilde{E}'$, then the maximal radius of the spheres in each grid must be the same. Later we explain in Chapter 7 that our direct labeling approach will produce images that are defined on the FCC grid F_1. Therefore, for the purpose of comparison of approaches we will have to produce a segmentation of an ART-reconstructed image also on F_1. As a result, we need to find a BCC grid (on which blobs are placed in the ART algorithm) that is equivalent to F_1. In the same appendix we show that the required BCC grid is B_Δ with $\Delta = 1/\sqrt{2} = 0.70710678$.

In regards to the parameters a and α, we followed the criterion in [35]. Applied to our case, this criterion aims at optimally approximating a constant by a series expansion like (2.2), using blobs with a constant coefficient and on the BCC grid B_Δ. In Appendix H we give the details on how we determine that, for the value of Δ computed above, $a = 2.9174404$ and $\alpha = 17.18465792$.

2.2.2 Segmentation

In electron microscopy the concentration of the different components (e.g., protein, ice, etc.) can be determined with reasonably high accuracy. Therefore, it is a standard to include this information in the segmentation process by selecting a threshold which would yield the right estimate of relative concentrations. This is the segmentation technique we adopt for the current approaches when they are compared against our approaches.

2.3 Our approach

Our work is primarily motivated by cryo electron tomography, but the labeling methods we develop are targeted to all application areas in which only a few projections are available and the desired output is a label image. Our aim is to produce, based on the electron micrographs (projections), a tessellation of space into small voxels (in this case the rhombic dodecahedra of F_1), each labeled as containing ice or protein. Current approaches using methods of CT would first assign, based on the projections, to each voxel a gray value (which is the density of that voxel) and then would segment this gray value image to obtain the label image. However, typically the number of projections required in CT is much larger than that in discrete tomography, hence a reconstructed gray value image from only a few projections is likely to be very inaccurate, leading to an incorrect segmentation. A particular difficulty arises if thresholding [84] is used for segmentation (which is a common current practice) because at a resolution of 2.5 Å or better the density distributions corresponding to different labels greatly overlap (see Figure 2.3). Also, we are dealing with a very under-determined type of problem, in which the number of constraints (mea-

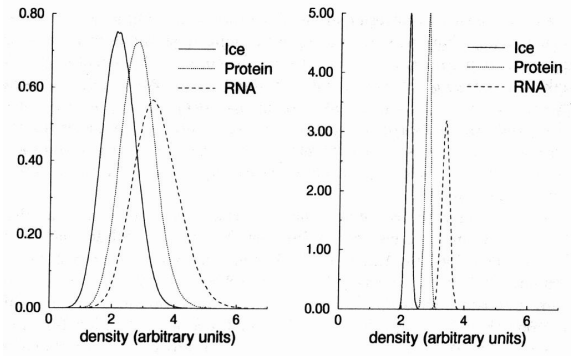

Figure 2.3: Histograms of the densities corresponding to volumes sampled using voxels of edge length equal to 2.5 Å (left) and 7.5 Å (right), obtained from volumes composed only of ice, only protein, or only RNA; from [11].

surements) is much smaller than the number of unknowns (labeling of the voxels).

To overcome such problems, we postulate a low level Gibbs prior on the underlying distribution of label images, and then directly estimate an optimum label image based on the prior and the measured projections. We attempt to express quantitatively the general shapes and sizes of characteristic structures using Gibbs priors. A Gibbs distribution assigns to every label image x a probability

$$\pi(x) = Z^{-1} \exp[-H(x)], \tag{2.8}$$

where Z is the normalizing factor, and $H(x)$ is referred to as the *energy* of x (see, e.g., [94]). We choose Gibbs priors as it has been experimentally demonstrated [12, 59] that for certain types of Gibbs distributions there are algorithms that are able to recover exceptionally "well" an unknown image that is a typical sample from the distribution, when provided with a few projections of the image and with the values of the parameters of the Gibbs distribution. A difficulty is that in a practical application the parameter values are not known to us. On the other hand, we usually have access to typical images of the application area. In th next chapter we address the problem of estimating the parameters from such sample images for the purpose of reconstruction.

To see the usefulness of image modeling by Gibbs priors in tomography of binary (i.e., two-label) images, consider the problem of reconstructing a binary image from its horizontal, vertical, and the NW diagonal projections, under the assumption that the image is a sample from a known Gibbs distribution. The inputs to the reconstruction problem are: the three projections of the image and the parameters that define the Gibbs distribution from which the image is assumed to be a sample. Figure 2.4 shows that it is possible to obtain a perfect reconstruction by combining the two types of information (the projections

and the prior).

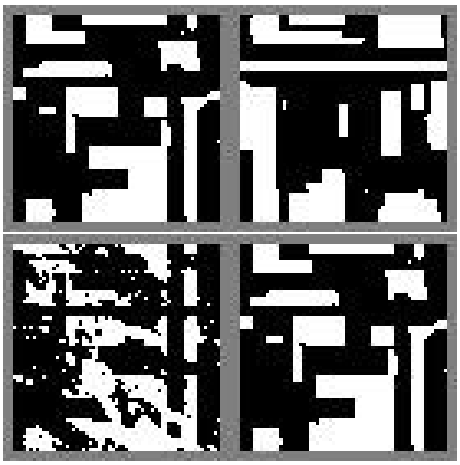

Figure 2.4: Binary tomography using a Gibbs prior. The top-left image is a random sample from a particular Gibbs distribution. The top-right image is another random sample image from the same distribution. The bottom-left image is a binary image with exactly the same projections (in three directions) as the one at the top-left. By combining both the prior information and the projection data into a reconstruction process, we obtained a perfectly reconstructed image shown at the bottom-right. The number of pixels whose label (color) differs from the label of the corresponding pixel in the top-left image is 1,763 for the top-right image, 822 for the bottom-left image, and 0 for the bottom-right image. All the images are of size 63×63. From [59].

Chapter 3

Image modeling using Gibbs priors

3.1 Gibbs distributions

For simplicity in the discussions, we concentrate for now on two-dimensional (2D) label images. Later, in Chapter 7, we study three-dimensional images. Let D be a fixed non-empty finite set that, for reasons that will become immediately obvious, we call the *domain*. In all the examples on 2D images in this dissertation, D is a square subset of the square lattice (i.e.,

$$D = \left\{ (v_1, v_2) \in \mathbb{Z}^2 \mid 0 \leq v_1, v_2 < V \right\}, \tag{3.1}$$

where $\mathbb{Z}$ denotes the set of all integers), but the definitions developed here are applicable to an arbitrary D. An element $d \in D$ is called a *point* in D. Any non-empty subset of D is called a *clique*. Given a clique q, a *configuration* (*over* q) is defined as a function g mapping from q into the set of labels $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_K\}$. In the special case where $q = D$, a configuration $g : D \to \Lambda$ is called a *label image* and will be denoted by x. (In the

2D case, since the Voronoi neighborhood [41] on the plane of a point in $\mathbb{Z}^2$ is a unit square, which is usually referred to as a *pixel*, sometimes we also call 2D label image a labeling of the set of pixels corresponding to D.) The set of all possible configurations over a clique q will be denoted as Λ^q; e.g., Λ^D is the set of all possible label images over D.

We define a *model* as a pair (Q, U) in which Q is a set of cliques and U is a function mapping the set $G = \bigcup_{q\in Q} \Lambda^q$ of all possible configurations over all cliques in Q into the real numbers. We refer to the value $U(g)$ as the *potential* of the configuration g. For a model $\mu = (Q, U)$, the μ-energy of an image x is defined as

$$H^\mu(x) = -\sum_{q\in Q} U(x|q), \tag{3.2}$$

where $x|q$ denotes the restriction of x to the clique q; i.e., $x|q$ is the configuration g over q such that, for all $d \in q$, $g(d) = x(d)$. Any model μ defines a *Gibbs distribution* π^μ over Λ^D as follows. The probability assigned to an image $x \in \Lambda^D$ is

$$\pi^\mu(x) = \frac{1}{Z} e^{-H^\mu(x)}, \tag{3.3}$$

where $Z = \sum_{x'\in\Lambda^D} e^{-H^\mu(x')}$. (In the discussion below, it will usually be understood what μ is at any particular stage. In such a case we will use H and π instead of H^μ and π^μ, respectively.)

Sometimes, for computational purposes, it is convenient to express images as vectors. From here on, we will allow x to denote either an image with domain D or a column vector of dimension $|D| = V^2$ (see Subsection 5.2.1); i.e., the elements (components) of the corresponding vector $x = \left(x_0, \cdots, x_{|D|-1}\right)^t$ are defined by $x_{v_2V+v_1} = x(v_1, v_2)$, $0 \le v_1, v_2 <$

V.

Other concepts that are needed in later discussions are the so-called closed neighborhood and neighborhood. Given a set of cliques Q of the domain D, the *closed neighborhood* κ_d of $d \in D$ with respect to Q is the subset of D defined by

$$\kappa_d = \left\{ d' \mid \text{for some } q \in Q,\ d \in q \text{ and } d' \in q \right\}. \tag{3.4}$$

and the *neighborhood* υ_d of d is defined as

$$\upsilon_d = \kappa_d - \{d\}. \tag{3.5}$$

To avoid repetitious discussion of trivial special cases, from now on we assume that a model (Q,U) is such that, for all $d \in D$, υ_d is nonempty (and so it is a clique, although not necessarily a clique of Q).

3.2 Parameters of a Gibbs distribution

3.2.1 Definition

Suppose that in some application of tomography of label images we believe that the efficacy of the process can be improved by modeling the distribution of the label images as they occur in that application by a Gibbs distribution. Let us assume that the size of the images (and, consequently, of D) is fixed. Also, assuming that the set Q of cliques of the model (Q, U) is fixed (for practical reasons one would try to keep the size of Q small), then the model (and the resulting Gibbs distribution) is uniquely determined by the values of U. Typically, there are available to us a number of images that are considered representative

samples of images in our application area. In the next chapter we discuss how to estimate of the *parameters* $U(g)$, where $g \in G$, based on these sample images. (This process may fail in the sense that even with "the best" estimate of these parameters, the model (Q, U) may not be adequate for modeling the sample images. In this case, we may wish to increase the size of Q.)

As we have just stated, a Gibbs distribution is uniquely determined by a model $\mu = (Q, U)$, where the domain of U is the set $G = \cup_{q \in Q} \Lambda^q$ of all possible configurations over Q. In a very general sense, we can consider a partition $G = \cup_{c=0}^{C} G_c$ (where $G_c \cap G_{c'} = \emptyset$ for $0 \leq c \neq c' \leq C$) of G such that for all g_1 and g_2 in G_c, $U(g_1) = U(g_2) = U_c$ $(0 \leq c \leq C)$. Consequently, the number of parameters to be estimated is reduced to $C+1$. However, it is very easy to show the following property: if a model μ' is obtained from a model μ by replacing U_c by $U_c - U_0$ for $0 \leq c \leq C$ then, for all images x, $\pi^{\mu'}(x) = \pi^{\mu}(x)$. Hence we may assume, without loss of generality, that $U_0 = 0$, and therefore the number of parameters to be estimated is only C.

The approach of the previous paragraph leads to an interesting simplification of (3.2) as follows. Given the domain D, let E_1 and E_2 be two sets of configurations; i.e., if $e \in E_1 \cup E_2$ then e is a configuration over a clique (not necessarily in Q). Let $N(E_1, E_2)$ denote the *number of times an element of* E_1 *appears in* E_2; i.e., the number of all possible pairs (e_1, e_2) such that $e_1 \in E_1$, $e_2 \in E_2$ and e_2 restricted to the domain of e_1 is equal to e_1 (which implies, in particular, that the domain of e_1 is a subset of the domain of e_2). We note that with this definition $N(E_1, E_2) = \sum_{e_1 \in E_1, e_2 \in E_2} N(\{e_1\}, \{e_2\})$, where $N(\{e_1\}, \{e_2\})$ is either one (if e_2 restricted to the domain of e_1 is equal to e_1) or zero (otherwise). Then it follows

that (3.2) can be re-written as

$$H(x) = -\sum_{c=1}^{C} N(G_c, \{x\})U_c. \tag{3.6}$$

3.2.2 Reducing the number of parameters

In practice it is desirable to deal with a few parameters rather than with many of them. We now discuss a general approach to producing reasonable partitions of G. Let T be a (necessarily finite) set of one-to-one mappings of an element of Q onto an element of Q (and so, for every $t \in T$, the domain and the range of t are elements of Q). A configuration g over a clique q is said to be *T-equivalent* to a configuration g' over a clique q' if there exists a (possibly empty) sequence $t_1, \cdots, t_S$ of mappings such that

1. for $1 \leq s \leq S$, $t_s \in T$ or $t_s^{-1} \in T$;
2. $q' = \{t_S \cdots t_1(d) \mid d \in q\}$; and
3. for all $d \in q$, $g'(t_S \cdots t_1(d)) = g(d)$.

Clearly, T-equivalence is an equivalence relation on G. The partitions that we use to reduce the number of parameters are based on the equivalence classes of such T-equivalences.

As an example, consider the domain D of (3.1) with the set of 3×3 cliques Q defined as follows. For $(v_1, v_2) \in D$, define the clique

$$q_{(v_1,v_2)} = \{(v_1 \oplus \delta_1, v_2 \oplus \delta_2) \mid \delta_1, \delta_2 \in \{-1, 0, 1\}\}, \tag{3.7}$$

where $\oplus$ denote addition in $\mathbb{Z}_V$; i.e., addition modulo V. Let

$$Q = \left\{q_{(v_1,v_2)} \mid (v_1, v_2) \in D\right\}. \tag{3.8}$$

As examples of one-to-one mappings, consider two mappings of $q_{(1,1)}$ onto itself: one is a rotation ρ defined by $\rho(v_1', v_2') = (v_2', 2 - v_1')$, for $(v_1', v_2') \in q_{(1,1)}$, and the other is a reflection ψ defined by $\psi(v_1', v_2') = (2 - v_1', v_2')$, for $(v_1', v_2') \in q_{(1,1)}$. For every $(v_1, v_2) \in D$ we also define two one-to-one mappings $\tau^h_{(v_1,v_2)}$ and $\tau^v_{(v_1,v_2)}$ of $q_{(v_1,v_2)}$ onto $q_{(v_1 \oplus 1, v_2)}$ and onto $q_{(v_1, v_2 \oplus 1)}$, respectively, by $\tau^h_{(v_1,v_2)}(v_1', v_2') = (v_1' \oplus 1, v_2')$, for all $(v_1', v_2') \in q_{(v_1,v_2)}$ and $\tau^v_{(v_1,v_2)}(v_1', v_2') = (v_1', v_2' \oplus 1)$, for all $(v_1', v_2') \in q_{(v_1,v_2)}$. The mappings $\tau^h_{(v_1,v_2)}$ and $\tau^v_{(v_1,v_2)}$ are, respectively, a horizontal translation and a vertical translation. If we now let T contain ρ, ψ, and $\tau^h_{(v_1,v_2)}$ and $\tau^v_{(v_1,v_2)}$ for all $(v_1, v_2) \in D$, then we find that any configuration defined on a 3×3 clique is T-equivalent to any other configuration which can be obtained from it by translations, rotations around the central point, and reflections in either horizontal or vertical central axis. One can reasonably argue that in many applications such T-equivalent configurations should be assigned the same potential. We will give a specific example in the next section.

3.3 Binary models

In the case when $|\Lambda| = 2$ we have *binary label images* or simply *binary images* defined on the domain D. Without loss of generality we set either $\Lambda = \{\text{black, white}\}$ or $\Lambda = \{0, 1\}$. In this dissertation, all the discussions will be concerned with binary images.

3.3.1 Our models

Following the idea introduced in Subsection 3.2.2, some of 3×3 configurations may be considered to be a particular *local feature* of one of the following types: a *black region*, a *white region*, a *convex corner*, a *concave corner*, or an *edge*. In Figure 3.1 we give

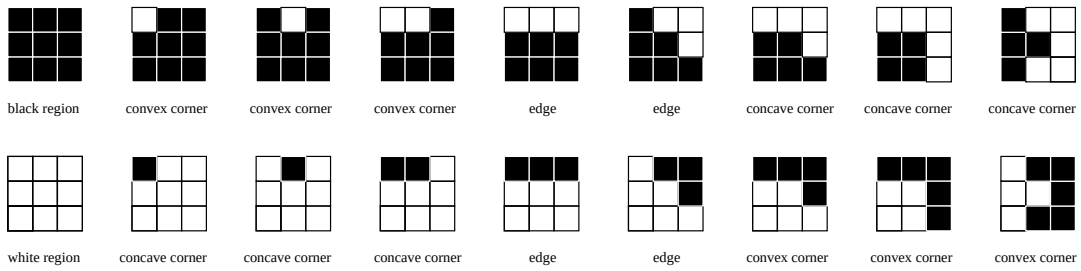

Figure 3.1: Configurations of a 3×3 clique that specify *local features* referred to as: a black region, a convex corner, a concave corner, an edge, and a white region.

examples of such special configurations. In fact, this set of examples is complete: e.g., a 3×3 configuration in an image is a convex corner if, and only if, it is *T*-equivalent to one of the configurations identified as a convex corner in Figure 3.1 (where, for completeness, we may assume that the configurations are over the clique $q_{(1,1)}$). The working definition of these five types of configurations is as follows. A configuration is one of these types only if there are r consecutive white pixels and $8-r$ consecutive black pixels among the eight external pixels of the clique. If the central pixel is black, then the configuration is a black region, a convex corner, an edge, or a concave corner respectively, if, and only if, $r = 0$, $1 \leq r \leq 2$, $r = 3$, or $4 \leq r \leq 5$, respectively (see the upper row of Figure 3.1). By switching the color (label) at each grid point of these local features, we will obtain the corresponding "opposite" local features (see the bottom row of the same figure). The opposite of a black region is a white region; the opposite of a convex corner is a concave corner (and vice-versa); the opposite of an edge is still an edge.

A Gibbs distribution based on the five local features of this example can be defined by a model in which $G = \cup_{c=0}^{5} G_c$, where G_0 is the set of all configurations that are not any of the above specified five local features, G_1 contains the black regions, G_2 the white regions, G_3 the edges, G_4 the convex corners, and G_5 the concave corners. In the rest of the thesis we will refer to models of this type as *our models*. The complete specification of one of

our models requires the five numbers U_1, U_2, U_3, U_4, and U_5 (recall that $U_0 = 0$). We adopt the convention of specifying a particular Gibbs distribution of this type by (U_1, U_2, U_3, U_4, U_5).

3.3.2 Ising models

We now briefly discuss how a much-studied family of Gibbs distributions can be incorporated into the same framework. For an *Ising model* (Q,U), D is determined by (3.1) and Q and the partition of $G = \cup_{q\in Q}\Lambda^q$ ($\Lambda = \{0,1\}$) are defined as follows. For $0 \le v_1, v_2 < V$, let $q^s_{(v_1,v_2)} = \{(v_1,v_2)\}$, $q^h_{(v_1,v_2)} = \{(v_1 \oplus \delta, v_2) \mid \delta \in \{0,1\}\}$ (pair of horizontal neighbors), and $q^v_{(v_1,v_2)} = \{(v_1, v_2 \oplus \delta) \mid \delta \in \{0,1\}\}$ (pair of vertical neighbors). The set of cliques Q is the union of $Q^s = \left\{q^s_{(v_1,v_2)} \mid 0 \le v_1, v_2 < V\right\}$ and $Q^p = \left\{q^h_{(v_1,v_2)} \mid 0 \le v_1, v_2 < V\right\} \cup \left\{q^v_{(v_1,v_2)} \mid 0 \le v_1, v_2 < V\right\}$. The partition $G = \cup_{c=0}^{2} G_c$ is such that a configuration g is in G_1 if, and only if, it is over some $q \in Q^s$ and it assigns the value 1 to the single element of q; a configuration g is in G_2 if, and only if, it is over some $q \in Q^p$ and it assigns the value 1 to both elements of q; and the remaining configurations (those that assign to at least one element in their domain the value 0) fall in G_0. Such an Ising model is completely specified by the pair (U_1, U_2). In Appendix A we show the equivalence of the definition of an Ising model as we have just given it to the more traditional one, such as the one provided by [94]. Here we note that the partition used in our definition could have been obtained by the T-equivalence classes under a T consisting of the already defined horizontal and vertical translations, together with a mapping of $q^h_{(0,0)}$ into $q^v_{(0,0)}$ by a rotation.

In this dissertation we do not report on the experiment with Ising models, since our own tests and evidence from the literature [14, 59] indicate that models with small cliques (such as Ising models) perform poorer than models with large cliques (such as our models).

3.4 Image Modeling

In this section we introduce a very general approach to drawing samples from a multi-dimensional probability distribution, such as a Gibbs distribution of our models. We show how we can increase the chances of appearance of a local feature by properly adjusting the parameters.

3.4.1 Markov chain Monte Carlo methods

In many cases, for example in order to estimate some characteristic (such as the expectation) of the (not necessarily binary) label images with respect to some probability distribution γ (e.g., a Gibbs distribution), it is necessary to obtain random sample images from γ. When γ is complex, drawing samples from it using standard analytical or numerical methods may be very difficult or even impossible in practice. In such situations, *Markov chain Monte Carlo* (MCMC) methods give a tractable way for sampling from, or estimating characteristics with respect to, such a distribution.

Given a probability distribution γ defined over the set of label images Λ^D and a mapping $n : \Lambda^D \to \mathbb{R}$, the expectation $\langle n(x) \rangle_\gamma$ is defined by

$$\langle n(x) \rangle_\gamma \doteq \sum_{x' \in \Lambda^D} n(x')\gamma(x'). \tag{3.9}$$

Unfortunately, it is seldom the case that this can be analytically computed. One known alternative is to generate a sequence $\left\{x^{(u)}\right\}_{u \geq 0}$ of independent random samples from the distribution γ and to estimate $\langle n(x) \rangle_\gamma$ by the empirical average (an approach justified by the

laws of large numbers; see e.g., [74]):

$$\overline{n}_S \doteq \frac{1}{S}\sum_{u=1}^{S} n(x^{(u)}), \tag{3.10}$$

where S is a positive integer; such method is known as *Monte Carlo.* (A Monte Carlo method is a method of solving problems by means of simulations that use random numbers.) It can be shown that $\overline{n}_S$ converges *almost surely* [89] (or *with probability 1*) to $\langle n(\mathbf{x})\rangle_\gamma$.

The fundamental idea of the MCMC technique is that when obtaining independent random samples from γ is prohibitively difficult, it may still be feasible to simulate a *Markov chain* that converges to γ [10]. A sequence of label images (or "states") $\left\{x^{(u)}\right\}_{u\geq 0}$ is said to be a Markov chain, if, for every non-negative integer u,

$$Prob\left(\mathbf{x}^{(u+1)} = x^{(u+1)} | \mathbf{x}^{(u)} = x^{(u)}, \cdots, \mathbf{x}^{(0)} = x^{(0)}\right) = Prob\left(\mathbf{x}^{(u+1)} = x^{(u+1)} | \mathbf{x}^{(u)} = x^{(u)}\right). \tag{3.11}$$

If, in addition, the right hand side of (3.11) is independent of u, the Markov chain is said to be *homogeneous.*

In a MCMC algorithm a homogeneous Markov chain of label images $\left\{x^{(u)}\right\}_{u\geq 0}$ is generated. At each step, given that the current state is $x^{(u)}$, the next state $x^{(u+1)}$ is generated according to a transition probability $Prob(\mathbf{x}^{(u+1)} = x^{(u+1)} \mid \mathbf{x}^{(u)} = x^{(u)})$, or simply $Tr(x^{(u+1)} \mid x^{(u)})$. The transition probability should be constructed so that the sequence converges to the target distribution γ. Starting from an arbitrary state $x^{(0)}$ and after a sufficiently long run of the algorithm, images obtained from the chain can be used as samples from γ, which enables Monte Carlo methods. The number of steps needed to achieve this is

commonly referred to as the *burn-in*. Sufficient conditions for a Markov chain to meet this property are *irreducibility* (i.e., for every pair of states, the probability of going from one state to any other state in a finite number of steps is positive) and *aperiodicity* (i.e., for an irreducible Markov chain, there is no partition of Λ^D into $r \geq 2$ classes $F_0, \cdots, F_{r-1}$, such that, for all k and for all $x \in F_k$,

$$\sum_{x' \in F_{k+1}} Tr\left(x \mid x'\right) = 1, \tag{3.12}$$

where, by convention, $F_r = F_0$) [10]. Convergence to the target distribution is in the *total variation* sense ([10, Chapter 4, Theorem 2.1]); i.e., let $\widetilde{\gamma}_u(x)$ be the distribution of the states corresponding to the u^{th} step (starting from an arbitrary state), then for every positive real number ε there is a u' such that for any $u > u'$

$$TV\left(\widetilde{\gamma}_u, \gamma\right) \doteq \frac{1}{2} \sum_{x' \in \Lambda^D} \left|\widetilde{\gamma}_u(x') - \gamma(x')\right| < \varepsilon. \tag{3.13}$$

Let $\widehat{n}_S = \frac{1}{S}\sum_{u=1}^{S} n(x^{(u)})$, where $\left\{x^{(u)}\right\}_{u \geq 0}$ is the Markov chain described above. It is important to mention (see [10, Chapter 3, Theorems 3.3 and 4.1] for proofs) that under the irreducibility condition and making use of the fact that Λ^D is finite, $\widehat{n}_S$ converges almost surely to the probabilistic average $\langle n(x) \rangle_\gamma$ defined in (3.9). Therefore, $\langle n(x) \rangle_\gamma$ can in principle be estimated by this empirical average. In our implementation, however, we take independent samples rather than consecutive samples, since this is feasible.

Given γ, various selections of the transition probability exist, which in turn give rise to different MCMC algorithms (e.g., the Metropolis algorithm and the Gibbs sampler that will be discussed shortly). In practice it is very common that the transition probabilities are

chosen so that the resulting Markov chain is *reversible*; i.e.,

$$\gamma(x)\,Tr\left(x' \mid x\right) = \gamma\left(x'\right)\,Tr\left(x \mid x'\right), \tag{3.14}$$

for all x and x' in Λ^D. (Such equations are also known as *detailed balance equation*s.) Many MCMC algorithms are hybrids or generalizations of the *Gibbs sampler* and the *Metropolis-Hastings algorithms*.

At each step of the Gibbs sampler [10], given the current image $x^{(u)}$, a *tentative* image $x^{(n)}$ is generated as being identical to $x^{(u)}$ except that at a randomly selected point d in D, the corresponding label has a value $\lambda \in \Lambda$ that is different from the value of $x^{(u)}$ at d; i.e., $x^{(n)}(d) = \lambda \neq x^{(u)}(d)$. (This, in particular, implies that $|\Lambda| \geq 2$.) The image $x^{(n)}$ is then accepted as $x^{(u+1)}$ with the *acceptance probability* $Acc(x^{(n)} \mid x^{(u)})$, which is the conditional probability

$$Prob(x^{(n)}(d) = \lambda \mid x^{(n)}(d') = x^{(u)}(d'), d' \in D - \{d\}). \tag{3.15}$$

If $x^{(n)}$ is not accepted, then $x^{(u+1)} = x^{(u)}$. (See also [91] for the general principle of the Gibbs sampler.) It is easy to see that in the binary case, $|\Lambda| = 2$, the conditional probability is equal to

$$\frac{\gamma(x^{(n)})}{\gamma(x^{(n)}) + \gamma(x^{(u)})}. \tag{3.16}$$

Another class of MCMC algorithms are the *Metropolis-Hastings algorithms* [23, 38]. In this family of algorithms, a tentative image $x^{(n)}$ is generated at each iterative step with probability $Cand(x^{(n)} \mid x^{(u)})$ (the *candidate generating density* or the *proposal density*).

When $x^{(u)} \neq x^{(n)}$, $x^{(n)}$ is accepted with the acceptance probability

$$Acc(x^{(n)} \mid x^{(u)}) = \min\left\{1, \frac{\gamma(x^{(n)})Cand(x^{(u)} \mid x^{(n)})}{\gamma(x^{(u)})Cand(x^{(n)} \mid x^{(u)})}\right\}; \tag{3.17}$$

otherwise $x^{(u+1)} = x^{(u)}$. Whenever $Cand(x^{(n)} \mid x^{(u)}) = Cand(x^{(u)} \mid x^{(n)})$ for every pair $x^{(u)}, x^{(n)}$ in Λ^D, the acceptance probability of the Metropolis-Hastings algorithm becomes

$$Acc(x^{(n)} \mid x^{(u)}) = \min\left\{1, \frac{\gamma(x^{(n)})}{\gamma(x^{(u)})}\right\}, \tag{3.18}$$

which is the original form by Metropolis et al. [70]. The particular case of the Metropolis-Hastings algorithms that implements (3.18) is referred to as the *Metropolis algorithm.* The Gibbs sampler and the Metropolis algorithm are two popular choices in various applications that use the MCMC method. We choose the latter, because in the binary case the "acceptance rate" (the percentage of times a move to a different image is made) is higher in the Metropolis algorithm than in the Gibbs sampler (compare the two formulas in the case when $\gamma(x^{(n)}) = \gamma(x^{(u)})$); see [38]. Also, another advantage (not used here) of the Metropolis algorithm is the possibility of selecting $x^{(n)}$ so that it differs from $x^{(u)}$ at more than one point. Note that both of the algorithms require knowledge of γ only up to a constant factor.

A very common question concerning the use of a MCMC method is, what is a reasonable burn-in? The answer can be given, for example, by giving a formula for the analytic upper bound of the total variation [79, 81]. Its computation, however, is intractable for all practical purposes. Therefore, we propose heuristic methods to estimate the burn-in, by measuring the number of steps after which the image "forgets" its initial state.

To obtain a typical sample from a Gibbs distribution, we simply set $\gamma = \pi$ and use the Metropolis algorithm. Later in Chapter 5, when we discuss reconstruction methods, we

apply a similar approach by sampling a distribution γ that takes into account a prior and the projection data.

3.4.2 Implementation

Generation of samples from a distribution γ that nowhere vanishes (i.e., $\gamma(x) \neq 0$, for $x \in \Lambda^D$), such as a Gibbs distribution, is computationally feasible using the Metropolis algorithm. This is because, as pointed out earlier, we do not ever have to compute a probability $\gamma(x)$, which would likely involve an impractically difficult calculation of a normalization factor (e.g., the factor Z of the Gibbs distribution in (3.3)) but only the ratio of two different $\gamma(x)$'s. Another fact that facilitates a low computational burden is that the change of the label at a point d only changes the configuration over cliques which are subsets of the closed neighborhood of d. For example, to generate samples from a Gibbs distribution whose set of cliques is the set Q defined in (3.8), the closed neighborhood $\kappa_{(2,2)}$ comprises $(2,2)$ itself and its 24 surrounding points, namely

$$
\begin{array}{ccccc}
(0,4) & (1,4) & (2,4) & (3,4) & (4,4) \\
(0,3) & (1,3) & (2,3) & (3,3) & (4,3) \\
(0,2) & (1,2) & (2,2) & (3,2) & (4,2) \\
(0,1) & (1,1) & (2,1) & (3,1) & (4,1) \\
(0,0) & (1,0) & (2,0) & (3,0) & (4,0)
\end{array}
$$

To proceed with the discussion but now restricted to the binary case, one definition is needed. Let g be a configuration over a clique q and let $d \in D - q$. We denote by g^d_λ ($\lambda \in \{0,1\}$) the configuration over $q \cup \{d\}$ for which $g^d_\lambda(d) = \lambda$ and $g^d_\lambda | q = g$. The *local*

interaction vector for d and g is defined to be $\mathbf{A}^d(g) = (A_1^d(g), \cdots, A_C^d(g))$, where $A_c^d(g) = N(G_c, \{g_1^d\}) - N(G_c, \{g_0^d\})$, for $1 \leq c \leq C$.

Using this notation, together with (3.3), (3.5), and (3.6), it is easy to derive that in a step of the Metropolis algorithm the ratio $\pi(x^{(n)})/\pi(x^{(u)})$ can be expressed as

$$\frac{\pi(x^{(n)})}{\pi(x^{(u)})} = \exp\left\{ \left[1 - 2x^{(u)}(d)\right] \sum_{c=1}^{C} A_c^d \left(x^{(u)} | \upsilon_d\right) U_c \right\}. \tag{3.19}$$

For our models (respectively, the Ising models), the value of $\sum_{c=1}^{C} A_c^d \left(x^{(u)} | \upsilon_d\right) U_c$ is uniquely determined by the configuration on the 24 (respectively, four) points in the neighborhood of d (in other words, once we know this vector, we no longer need to know what d is; this is due to the T-equivalence of configurations under translations). Hence, prior to running the Metropolis algorithm, these 2^{24} (respectively, 2^4) possible values can be precalculated and stored in a table. During the running of the algorithm, the needed value is obtained from the table by a simple look-up based on the labels of the points surrounding d (see [93], where this idea was first introduced).

To insure that the algorithm has been run long enough (burn-in) to provide a typical sample of the distribution, we initialized the Metropolis algorithm with two different images: a blank black image and another that is completely white. Then both cases were run until they "stabilized" with images of similar energy and similar number of white pixels. The time for the stabilization process is measured in *cycles*: in each cycle, $|D|$ steps of the Metropolis algorithms are applied. For example, for the binary images of size 63×63 shown below, we observe that the burn-in is approximately $2 \cdot 10^4$ cycles.

3.4.3 Modeling using our models

Here we show sample images from Gibbs distributions defined by some of our models. Specifically, we will consider various choices for the five potentials corresponding to the five types of configurations: U_1 for black regions, U_2 for white regions, U_3 for edges, U_4 for convex corners, and U_5 for concave corners.

Four random samples of size 63×63 from different Gibbs distribution are shown in Figure 3.2. We can see for example that a higher U_3, which controls the "edginess", gives rise to an "edgy" image (bottom-left); while a typical image from a distribution with higher U_4 (for convex corners) has numerous small objects (top-right). All the samples in the figure were produced by running the algorithm for $3 \cdot 10^4$ cycles.

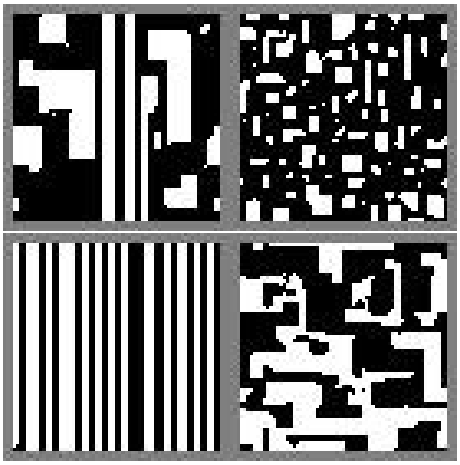

Figure 3.2: Sample images (63×63) from our models using parameters (1.2, 1.2, 1.2, 0.52, 0.2) (top-left), (1.2, 1.2, 1.2, 0.9, 0.2) (top-right), (1.2, 1.2, 1.4, 0.52, 0.2) (bottom-left), and (1.2, 1.2, 1.2, 0.52, 0.6) (bottom-right). Note that the difference between the top-left and any other image is caused by a change in only one parameter. With respect to the top-left image the top-right image has a higher U_4, which results in more numerous but smaller image objects; the bottom-left image is much more "edgy" because of its higher U_3; and in the bottom-right image we can see more concave corners due to the higher U_5.

Chapter 4

Estimation of the parameters of Gibbs priors

4.1 Introduction

Estimating the parameters of a Gibbs distribution is a relatively recent research topic. Depending on the problem formulation, several levels of complexity may exist in estimating the parameters. If the training samples are noiseless realizations of a single Gibbs distribution, then the task is simply to estimate the parameters of that distribution [5, 8, 19]. When the sample images are contaminated by noise parameterized with unknown values, these need to be estimated too [96]. The complexity increases when each of the images is put together from regions that are themselves samples from different Gibbs distributions (e.g., when multiple textures are present in the images) that needed to be segmented [95]. For the purpose of tomographic reconstruction, sample images that are available to us will be noiseless and, for simplicity, will be assumed to come from a single distribution. Experimental evidence [12, 59] shows that surprisingly good reconstructions are achievable under

this assumption.

In Figure 2.4 we have illustrated that if we have the projections of a binary image and the Gibbs distribution from which the image is a sample, then we may obtain a very good reconstruction. Hence, given a typical sample collection of images in a certain application area (we refer to this collection as the *training set* and denote it by X_{tr}), it is worthwhile to estimate a Gibbs distribution that may give rise to such a sample collection. Since a Gibbs distribution is defined by a model (Q,U), we need to discover both the set of cliques Q and the corresponding potentials defined by U. Here we assume that the set of cliques Q is chosen beforehand, as well as the partition $\{G_c \mid 0 \leq c \leq C\}$ of G, and therefore the only remaining task is to estimate the U_c for $1 \leq c \leq C$ (recall that $U_0 = 0$). We discuss a number of previously proposed methods for doing this. We restrict the discussion to the binary case $|\Lambda| = 2$; however, most of the methods can be easily extended to the general case.

4.2 Parameter estimation methods

In the literature the estimation methods that implicitly assume noiseless training samples from a single Gibbs distribution fall into two main classes: those that estimate the expected number of the local features (see Subsection 3.3.1) by counting them (the *heuristic method* [12] and the *Markov chain Monte Carlo maximum likelihood* (MCMCML) *method* [20]) and those that estimate some conditional probabilities (or function of these) given that a certain configuration over a neighborhood (see Section 3.1) occurs in the training images. Methods of the latter class do the estimation by counting configurations over neighborhoods; they are the *histogram method* [19], the *modified histogram method* [36], *Borges' approach* [8], the *coding method* [4], and the *maximum pseudo-likelihood method* [5, 6]. We will discuss all the above methods, but we will not show experimental results of the

modified histogram method, of the coding method, or of the MCMCML method. For the first two, there have been reports [8, 56] on their weaknesses from the theoretical and practical points of view; while for the MCMCML method, we found that the proposed algorithm is inappropriate for our models.

4.3 Counting configurations over cliques

4.3.1 The heuristic approach

This method [12], applicable only to binary images (see Section 3.3), defines, for $1 \leq c \leq C$,

$$U_c = \rho \left[\ln \left(N(G_c, X_{tr}) / |G_c| + 1 \right) - \ln \left(N(G_0, X_{tr}) / |G_0| + 1 \right) \right]. \tag{4.1}$$

(recall that $N(G_c, X_{tr})$ is the number of times a configuration in G_c appears in the training set; see Subsection 3.2.1). The constant ρ is determined by an additional criterion; here we attempt to select ρ so that the expected number of pixels labeled white (or 1) in a sample image from the resulting distribution equals the average number of white pixels in the images of the training set. For reconstruction, however, this constant need not to be considered separately, because it can be absorbed into the temperature of the annealing schedule.

In the context of Figure 3.1, a higher potential using this definition implies that the local feature of the corresponding type occurs with a higher frequency in the sample collection.

4.3.2 Markov chain Monte Carlo maximum likelihood method

Proposed in [20], the MCMCML method finds a parameter vector $\widehat{\mathbf{U}} = (\widehat{U}_1, \cdots, \widehat{U}_C)^t$ such that the likelihood of observing the sample images in X_{tr} is maximum. This condition is equivalent (see Appendix B) to finding a $\widehat{\mathbf{U}}$ such that the expected value of $N(G_c, \{x\})$ with respect to $\pi_{\widehat{\mathbf{U}}}$ equals to $N(G_c, X_{tr}) / |X_{tr}|$. We denote this expected value by $\langle N_c(\{x\}) \rangle_{\widehat{\mathbf{U}}}$. Starting from a parameter vector, the method consists of a gradient search algorithm that iteratively adjusts the parameters until the ML condition is met within a certain tolerance. During the process some characteristics, such as $\langle N_c(\{x\}) \rangle_{\mathbf{U}}$, are estimated using a Markov chain Monte Carlo (MCMC) method.

We think that the general idea behind this parameter estimation method is interesting but impractical, at least for our models. In Appendix B we give a detail explanation of the method and discuss why it may not be useful in practice. The main reason is that when the likelihood function (such as that based on our models) is not well-behaved and with relatively high "oscillations" (i.e., a "small" change in the parameter can cause a large change in the likelihood), it is not adequate to apply a gradient search algorithm (such as the conjugate gradient method), due to its second-order type of approximation.

4.4 Counting configurations over neighborhoods

Except for the pseudo-likelihood method and the coding method (to be discussed shortly), all the estimation methods that count configurations over neighborhoods solve a system of linear equations, in which the unknowns are the parameters and the matrix system is uniquely determined by all the possible (neighborhood) configurations. The right hand side of the system is estimated from the observed counts, which varies from method to

method. Although the discussion is restricted to the binary case, the generalization to the case of more than two labels is straight-forward. We now explain how the system is formed.

A well-known property of Gibbs distributions [94] is the following. For every d in D and every label image (not necessarily binary) x in Λ^D:

$$\begin{aligned} Prob(\mathbf{x}(d) = x(d) \mid \mathbf{x}(d') = x(d'), d' \in \upsilon_d) = \\ Prob(\mathbf{x}(d) = x(d) \mid \mathbf{x}(d') = x(d'), d' \in D - \{d\}), \end{aligned} \tag{4.2}$$

where, according to our convention, $\mathbf{x}(d)$ is the associated random vector. A random vector satisfying (4.2), is said to be a *Markov random field* [94, 37]. The strategy for estimating the parameters of a Gibbs prior is to form a system of linear equations by relating (3.19) to (4.2), where the unknowns are the parameters and the ratio $\frac{\pi(x^{(n)})}{\pi(x^{(u)})}$ can be estimated by counting the configurations over neighborhoods.

If we restrict attention to the binary case, given two images $x^{(u)}$ and $x^{(n)}$ that differ only at one point d (without loss of generality, we assume $x^{(n)}(d) = 1$ and $x^{(u)}(d) = 0$), we can express the conditional probability

$$z = Prob(\mathbf{x}^{(n)}(d) = x^{(n)}(d) \mid \mathbf{x}^{(n)}(d') = x^{(n)}(d'), d' \in \upsilon_d) \tag{4.3}$$

as, using (4.2) and Bayes' formula,

$$z = \frac{Prob(\mathbf{x}^{(n)}(d') = x^{(n)}(d'), d' \in D)}{Prob(\mathbf{x}^{(n)}(d') = x^{(n)}(d'), d' \in D - \{d\})} = \frac{\pi(x^{(n)})}{\pi(x^{(u)}) + \pi(x^{(n)})}; \tag{4.4}$$

z is different from zero and one, because $\pi(x)$ nowhere vanishes by definition. Note that (4.4) is a particular case of (3.16), in which γ has been replaced by π. After some algebra,

(4.4) is equivalent to

$$\frac{\pi(x^{(n)})}{\pi(x^{(u)})} = \frac{z}{1-z}. \tag{4.5}$$

Taking the natural logarithm on both sides, we have, using (3.19) with $x^{(u)}(d) = 0$, the equation

$$\sum_{c=1}^{C} A_c^d(x^{(u)}|\upsilon_d)U_c = \log\frac{z}{1-z}. \tag{4.6}$$

(We note that, in the case where $x^{(u)}(d) = 1$ and $x^{(n)}(d) = 0$, the right hand side of (4.6) should be replaced by $\log\frac{1-z}{z}$.) By considering all the possible configurations over a neighborhood, we then have formed a system of linear equations, one for each configuration, on the parameters U_c $(c = 1,..,C)$. The right hand side, however, is function of the conditional probability z, which is unknown but can be estimated from the training set; e.g., by counting the number of times a certain configuration occurs and, out of those many, the number of times the value at the center is 1. Since in practice the number of all possible configurations is likely to be much larger than the number of counts in the training set, which implies large variance in the estimates of the conditional probabilities, it is desirable to partition the set of all possible configurations into some classes, each one of them yielding the same z, for a given set of parameters.

Let

$$\Omega = \{(d,g) \mid d \in D \text{ and } g \text{ is a configuration over } \upsilon_d\}; \tag{4.7}$$

recall that υ_d is the neighborhood of d (Section 3.1). Let us partition Ω by the condition

that two items belong to the same class of the partition if, and only if, they share the same local interaction vector $A^d(g)$ (Subsection 3.4.2). Let $\Omega = \cup_{b=1}^{B} \Omega_b$ be this partition and let $A^b = (A_1^b, \cdots, a_C^b)$ be the unique local interaction vector for the elements of Ω_b. We also define, for $\lambda \in \{0,1\}$ and $1 \leq b \leq B$, $\Omega_b^\lambda = \{g_\lambda^d \,|\, (d,g) \in \Omega_b\}$. As a result of this partitioning, we have a system of B linear equations, such that for the b-th equation ($b = 1, \cdots, B$)

$$\sum_{c=1}^{C} A_c^b U_c = \widehat{\left(\log \frac{z}{1-z}\right)}_b, \tag{4.8}$$

where $\widehat{\left(\log \frac{z}{1-z}\right)}_b$ denotes an estimate of $\left(\log \frac{z}{1-z}\right)_b$. For our models and for the Ising model, B is respectively equal to 1997 and 5.

4.4.1 The histogram method

The histogram method [19] aims at satisfying the system of equations (4.8) in the least-squares sense. In this method the conditional probability z for the b-th class is estimated by the classical estimate $N(\Omega_b^1, X_{tr}) / \left[N(\Omega_b^1, X_{tr}) + N(\Omega_b^0, X_{tr})\right]$. This implies that the right hand side of becomes (4.8)

$$\widehat{\left(\log \frac{z}{1-z}\right)}_b = \log \frac{N(\Omega_b^1, X_{tr})}{N(\Omega_b^0, X_{tr})}. \tag{4.9}$$

The advantage of this estimate is that it can be given in closed form and is very easy to calculate. Nonetheless, this estimator is well-defined only if $N(\Omega_b^\lambda, X_{tr}) \neq 0$ for $\lambda \in \{0,1\}$, which is troublesome since the condition is likely to be violated if we are given a small number of sample images. Eliminating too many equations from the system to avoid the cases where $N(\Omega_b^0, X_{tr}) = 0$ or $N(\Omega_b^1, X_{tr}) = 0$ can lead to rank-deficiency and precludes a unique solution to the least-squares problem. Another disadvantage is that when the counts

$N(\Omega_b^\lambda, X_{tr})$ ($\lambda \in \{0,1\}$) are low, the classical estimate may be too biased. In [8] another method is proposed, which does not require the elimination of the equations and provides better parameter estimation in most of the cases that we consider.

4.4.2 The modified histogram method

Another method also aimed at improving the histogram method is the so called *modified histogram method.* Nonetheless, since it has been reported in [8] that the corresponding estimator is inconsistent, we do not evaluate this method; but instead, for completeness, we discuss it in Appendix D.

4.4.3 Borges' method

To simplify notations, we drop the subindex b and let N_1, N_0, and N denote respectively $N(\Omega_b^1, X_{tr})$, $N(\Omega_b^0, X_{tr})$, and $N(\Omega_b^1, X_{tr}) + N(\Omega_b^0, X_{tr})$. As opposed to the histogram method that estimates first the ratio $\frac{z}{1-z}$ by $\frac{N_1}{N_0}$ and then takes the natural logarithm, Borges' method [8] estimates directly $\log \frac{z}{1-z}$ based on the observations N_1 and N_0. The estimator, denoted by $\Xi(N_1, N_0)$, is defined to be the minimum mean squared error estimator. Thus

$$\widehat{\left(\log \frac{z}{1-z}\right)}_b = \Xi(N_1, N_0) \tag{4.10}$$

is an expectation with respect to the random variable N_1

$$\sum_{N_1=0}^{N} C_N^{N_1} z^{N_1} (1-z)^{N_0} \left(\Xi(N_1, N_0) - \log \frac{z}{1-z}\right)^2, \tag{4.11}$$

where the factor $C_N^{N_1} z^{N_1}(1-z)^{N_0}$ comes from the fact that N_1 follows a binomial law with parameters N and z.

The variable z in (4.11) is unknown, therefore Borges adopts a Bayesian approach in which z is assumed to have a prior uniform distribution on the interval $[0,1]$, thereby integrating z out in (4.11) from 0 to 1 and setting $\Xi(N_1,N_0)$ (for $N_1 = 0,\cdots,N$ and $N_0 = N - N_1$ and noting that the terms in the expectation are non-negative) to be the minimizer of

$$\int_0^1 z^{N_1}(1-z)^{N_0}\left(\Xi(N_1,N_0) - \log\frac{z}{1-z}\right)^2 dz. \tag{4.12}$$

Solving (4.12) gives the analytical expression for the estimate

$$\Xi(N_1,N_0) = \begin{cases} 0, & \text{if } N_1 = N_0, \\ \frac{1}{N_0+1} + \cdots + \frac{1}{N_1}, & \text{if } N_1 > N_0, \\ -\frac{1}{N_1+1} - \cdots - \frac{1}{N_0}, & \text{if } N_1 < N_0. \end{cases} \tag{4.13}$$

Borges in fact solves the system (4.8) by the weighted least square method. The weights ζ are the square roots of the mean squared error of the estimate $\Xi(N_1,N_0)$, which can be estimated by substituting the value of $\Xi(N_1,N_0)$ given by (4.13) in (4.12), yielding

$$\zeta = \left[\frac{\pi^2}{3} - \frac{1}{N+1}\sum_{N_1=0}^{N}\Xi^2(N_1,N-N_1)\right]^{-\frac{1}{2}}. \tag{4.14}$$

For $N_0 + N_1 > 400$, [8] reports that a good approximation to ζ is $\sqrt{3+0.05N}$.

4.4.4 The coding method

The strategy of this method [4] is the following. Partition the domain D into disjoint subsets D^h (i.e., $D^h \cap D^{h'} = \emptyset$ and $\bigcup_h D^h = D$), each called a *code*, in such a way that no two points in D^h belong to a same clique. The total number of possible codings is thus related to the maximum size of cliques in Q. (For example, for our models, where the cliques are all 3×3, there would be at least nine codings; while for the Ising models, this number is at least two.) Then the parameters of a Gibbs distribution are chosen so to maximize the (conditional) likelihood (assuming that the training samples are statistically independent)

$$\prod_{x \in X_{tr}} Prob\left(\mathbf{x}(d) = x(d)), d \in D^h \mid \mathbf{x}(d') = x(d'), d' \in D - D^h\right). \tag{4.15}$$

Using (4.2), it is easy to show that the term inside the product can in turn be decomposed in the product form

$$\prod_{d \in D^h} Prob\left(\mathbf{x}(d) = x(d) \mid \mathbf{x}(d') = x(d'), d' \in \upsilon_d\right) \tag{4.16}$$

The parameters can then be estimated (more details are given in the next subsection) using some standard gradient ascent methods. Due to the convexity of the conditional likelihood (see [36]), searching for the global maximum efficiently is possible. However, because each coding gives a set of estimated parameters, it is not clear how to combine different sets optimally [56].

4.4.5 The pseudo-likelihood method

The author of [4] later proposed in [5, 6] to maximize, instead, the *pseudo-likelihood* that is essentially of the same form as the likelihood (4.16) but is "extended" to the whole

image: replace D^h by D in (4.16). Although the formula is no longer a true likelihood, good parameter estimation results can be obtained using a relatively small training set.

This method aims at finding parameters U_c $(c = 1, ..., C)$ that maximize the log-pseudo-likelihood

$$\mathcal{L}_{\mathbf{U}}(X_{tr}) = \sum_{d \in D, x \in X_{tr}} \log \left[Prob\left(\mathbf{x}(d) = x(d) \mid \mathbf{x}(d') = x(d'), d' \in \upsilon_d \right) \right]. \tag{4.17}$$

To see the dependency on the parameters, the probability in (4.17) can be written, using (4.6), as

$$Prob\left(\mathbf{x}(d) = x(d) \mid \mathbf{x}(d') = x(d'), d' \in \upsilon_d \right) = \begin{cases} \dfrac{1}{1 + \exp\left[-\mathbf{U}^t \mathbf{A}^d(x|\upsilon_d) \right]}, & \text{if } x(d) = 1, \\ \dfrac{1}{1 + \exp\left[\mathbf{U}^t \mathbf{A}^d(x|\upsilon_d) \right]}, & \text{if } x(d) = 0. \end{cases} \tag{4.18}$$

That is, the conditional probability depends solely on the local interaction vector and the parameters, which is an already known result from (4.6) but now stated in words. If we bring in the partition over Ω into B classes (see the beginning of this section) and take into account the classification in (4.18), the sum in (4.17) can be rearranged and rewritten as (denote $\mathbf{U}^t \mathbf{A}^b = \sum_{c=1}^{C} A_c^b U_c$)

$$\mathcal{L}_{\mathbf{U}}(X_{tr}) = \sum_{b=1}^{B} \left[N(\Omega_b^1, X_{tr}) \ln \left(\frac{1}{1 + \exp\left[-\mathbf{U}^t \mathbf{A}^b \right]} \right) + N(\Omega_b^0, X_{tr}) \ln \left(\frac{1}{1 + \exp\left[\mathbf{U}^t \mathbf{A}^b \right]} \right) \right], \tag{4.19}$$

whose gradient is

$$\nabla \mathcal{L}_{\mathbf{U}}(X_{tr}) = \sum_{b=1}^{B} \left[\frac{1}{1+\exp\left[\mathbf{U}^t \mathbf{A}^b\right]} \left(N(\Omega_b^1, X_{tr}) - N(\Omega_b^0, X_{tr}) \exp\left[\mathbf{U}^t \mathbf{A}^b\right] \right) \mathbf{A}^b \right]. \quad (4.20)$$

The function in (4.19) is convex in $\mathbf{U}$ [36], therefore a local search algorithm suffices to find a solution that maximizes (4.19). In particular, we use a steepest ascent algorithm for the search.

A sufficient condition for the uniqueness of the solution [36] is that there are C terms with index $b_c = b_1, \cdots, b_C$ in (4.19), such that $N(\Omega_{b_c}^1, X_{tr}) \cdot N(\Omega_{b_c}^1, X_{tr}) > 0$ and the vectors $\mathbf{A}^{b_1}, \cdots, \mathbf{A}^{b_C}$ are linearly independent.

4.5 Evaluation

In this section we evaluate the various approaches to estimating the parameters of a Gibbs prior. To do this we select Gibbs distributions using our models, each one of the four illustrated in Figure 3.2, and we generate some random samples and use them as the training set. We use the indicator ε to measure the success of the estimation method, which is defined to be the squared norm:

$$\varepsilon = \sum_{c=1}^{C} (U_c - \widehat{U}_c)^2, \quad (4.21)$$

where U_c and $\widehat{U}_c$ (for $1 \leq c \leq C$) are the actual and estimated values, respectively, of the potentials determining the Gibbs distribution. We find that similar results are obtained if we use a different indicator; e.g., $\varepsilon = \max_c \left| U_c - \widehat{U}_c \right|$.

Most of the estimation methods known in the literature have been applied to Gibbs prior models that are quite simple. For example, [8, 19, 36] all dealt with Gibbs distributions having neighborhood of size 3×3 and the number of parameters is either one or two. Here we are interested in the performance of the different estimation methods for our models with 5×5-neighborhoods and five parameters.

We have mentioned seven methods for estimating the parameters from noiseless training sample images. However, for reasons that are already given, we do not evaluate the coding method, the modified histogram method, nor the MCMCML method.

The dependence of ε on the number of samples for the four distributions is shown in Figure 4.1 and 4.2. The error bars are the sample standard deviations obtained by repeating ten times each experiment for a fixed number of samples. Clearly, both Borges' and the pseudo likelihood (PL) method are able to recover the parameters in all the four cases; and so can the histogram method, except that the latter requires, in general, more samples to reach to, within a certain tolerance, the original parameters. For images such as those from the distribution (1.2, 1.2, 1.4, 0.52, 0.2) the PL method gives a poor estimate when the number of training images is low. This is attributed to the lack of uniqueness in the solutions (see the end of Subsection 4.4.5). Also, for the same distribution, the error bar corresponding to one sample is smaller than that of ten samples in the histogram method, because we did not take into account those estimates that resulted from a rank deficient system. Finally, the heuristic method fails to converge to a similarly low value of ε, although its error bars are practically non existent. Figure 4.3 shows typical sample images from distributions estimated by the heuristic method; they do not resemble those from the corresponding original distributions.

We do not report on the heuristic method for the distribution (1.2, 1.2, 1.2, 0.52, 0.6) because its expected number of white pixels is 2,110, while the same item for the distribu-

tions defined by (4.1) applied to the training set is never greater than 2,000 for any ρ. Thus, our proposed way of selecting the ρ in (4.1) is not guaranteed to produce a result. However, as pointed out already, this is not essential for reconstruction (to be discussed in the next chapter), since the ρ can be absorbed into the temperature of the annealing schedule in simulated annealing.

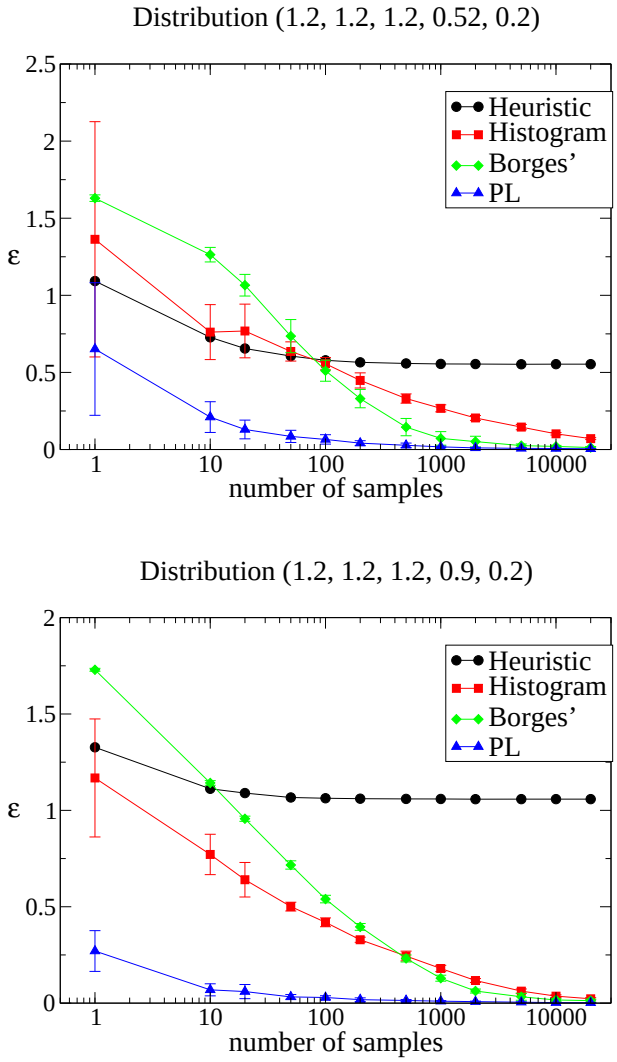

Figure 4.1: Parameter estimation error defined by the indicator ε of (4.21) as a function of the number of training images for two of the four distributions in Figure 3.2.

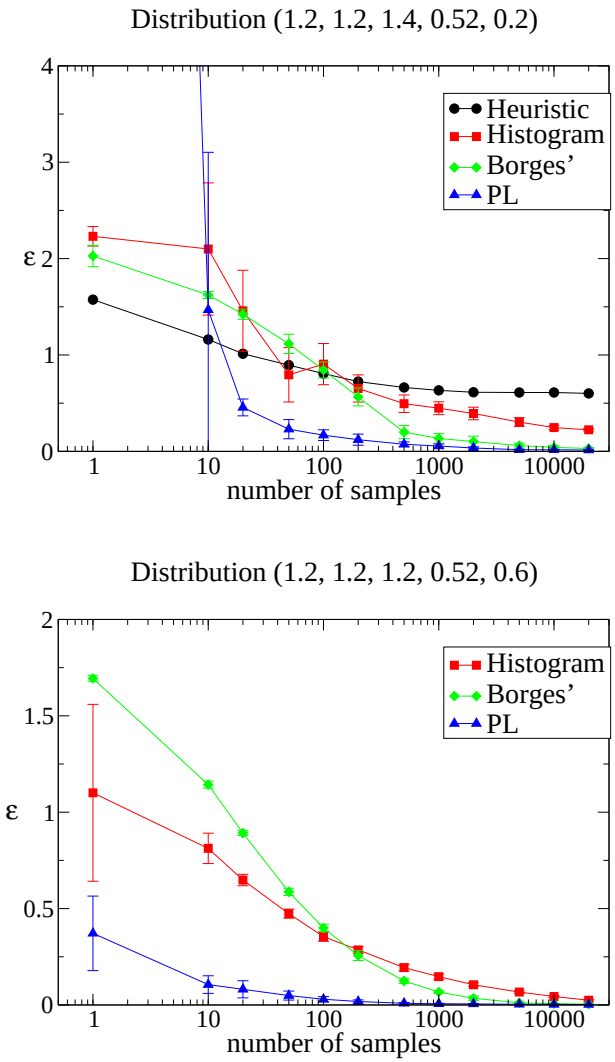

Figure 4.2: Parameter estimation error defined by the indicator ε of (4.21) as a function of the number of training images for two of the four distributions in Figure 3.2.

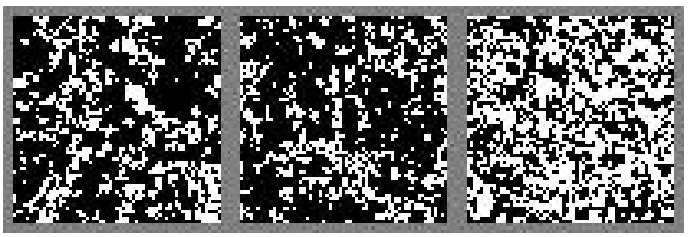

Figure 4.3: Typical samples (after $3 \cdot 10^4$ cycles) from the estimated distribution using the heuristic method, corresponding to the distribution (1.2, 1.2, 1.2, 0.52, 0.2) (left), (1.2, 1.2, 1.2, 0.9, 0.2) (center) and (1.2, 1.2, 1.4, 0.52, 0.2) (right).

Chapter 5

Reconstruction of label images

5.1 Introduction

Our general approach to solving a tomographic reconstruction problem is to postulate a low level Gibbs prior on the underlying distribution of label images, combining it with the information coming from the measurement data based on a Bayesian point of view, and then optimizing (minimizing) a *Bayes risk* that is defined as the expectation of some *cost function* with respect to the *posterior probability*. The posterior probability is proportional to the product of the prior probability and the *likelihood* that is defined to be the conditional probability of the measurement data given the label image. The optimizer of the (Bayes) risk (also called *Bayes estimator*) is adopted as the solution of the reconstruction problem.

In Chapter 3 we have defined Gibbs distributions and have discussed how we can generate samples from it once we know its parameters. We have shown in Chapter 4 how to estimate the parameters of a Gibbs distribution from label images that are assumed to be representative samples in a particular application area. In this chapter we discuss some commonly used cost functions, the implications on their respective Bayes estimators, and

the ways to find such estimators efficiently.

5.2 The posterior probability

5.2.1 Label images

As pointed out in Section 3.1, a label image is treated as either a labeling of points in the domain D (or of their Voronoi neighborhoods) or a $|D|-$ dimensional column vector $x = (x_0, \cdots, x_{|D|-1})^t$, where $x_i \in \Lambda$ (the set of labels), for $i = 0, \cdots, |D|-1$. We assume that there is a prior distribution that assigns to every label image x a probability $\pi(x)$ of the form (3.3), which is a Gibbs distribution.

5.2.2 Gray value images

Let $\mathbf{Y}$ be a set of *gray value images* each one of which is an I-dimensional vector $y = (y_0, \cdots, y_{I-1})^t$**,** where $y_i \in Y$ (the set of *gray values*, $Y \subseteq \mathbb{R}$), for $1 \leq i \leq I$. We assume that there is a conditional distribution that, given a label image x, assigns a probability $\phi(y|x)$ to every gray value image y. (Since $\mathbf{Y}$ is not necessarily finite, it would be more precise to say that $\phi(y|x)$ is the probability density function defining the conditional distribution of the gray value image x given the label image x. For the sake of brevity, we will continue to refer to a notation such as $\phi(y|x)$ as a "probability" rather than a "probability density function.") Unless otherwise stated, we work with the special case in which $I = |D|$ and, for every label x_i, there is a distribution which assigns (independently) a probability $\varphi(y_i|x_i)$ to every gray value y_i, for $0 \leq i \leq I-1$. Consequently,

$$\phi(y|x) = \prod_{i=0}^{I-1} \varphi(y_i|x_i). \tag{5.1}$$

For example, in electron microscopy applications, a gray value y_i is the density of the Voronoi neighborhood of the lattice point i, for $0 \leq i \leq I-1$; and the probabilities $\varphi(y_i|x_i)$ intend to capture the histograms in Figure 2.3; i.e., the uncertainty of the density around the average density, denoted by μ_{x_i}, corresponding to the label x_i.

5.2.3 Measurements

Let **W** be a set of *measurement vectors* each one of which is a J-dimensional column vector $w = (w_0, \cdots, w_{J-1})^t$, where $w_j \in W \subseteq \mathbb{R}$, for $0 \leq j \leq J-1$. We assume that there is a conditional distribution that, given a gray value image y, assigns a probability $\chi(w|y)$ to every measurement vector w. Usually w is related to a linear transformation z of the gray value image

$$z = Ry, \tag{5.2}$$

where R is of size $J \times I$ with fixed coefficients and none of the rows of R is a null vector. In this dissertation two cases are studied: one in which R is the identity matrix (thus $J = I$) and the other one in which R is the projection matrix [39] (see also Subsection 2.2.1).

The first case is based on the current approaches of first reconstructing the gray-value image followed by a segmentation. If the gray-value reconstruction were perfect; i.e., the measurement is the gray value image itself ($w = Ry$), then the resulted segmented image would be the best one can expect from current techniques. This idealistic situation could provide us with an "upper bound" of the quality of the label reconstruction by a current approach.

The second case reflects real tomography applications, in which the measurement data is a set of projections, each one of which is usually modeled as the measured (hence, noisy)

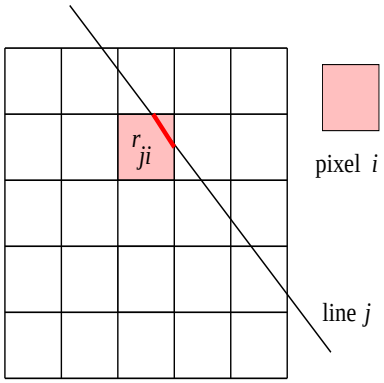

Figure 5.1: An illustration in 2D of the entry r_{ji}, as the length of intersection between the line j and the pixel i, of the projection matrix.

integrals of the density of the object being reconstructed along lines that are either parallel to each other or share a point. In our framework this can be modeled by setting the entry r_{ji} of R to be the length of intersection between the line j and the Voronoi neighborhood of the point i; see Figure 5.1 for an example with a two-dimensional image of size $|D| = 25$.

For the experiments reported here, we work with the special case in which $\chi(w|y)$ has a product form; i.e., given a gray value image y, there is (for $0 \leq j \leq J-1$) a conditional probability $\upsilon_j(w_j|y)$ of the j-th measurement being w, and

$$\chi(w|y) = \prod_{j=0}^{J-1} \upsilon_j(w_j|y). \tag{5.3}$$

5.2.4 The posterior probability

The *likelihood* $\eta(w|x)$ of measuring w given a label image x, assuming (as reasonable) that w does not depend on x given y, is the integral (or sum if Y is discrete)

$$\eta(w|x) = \int_{\mathbf{Y}} \chi(w|y)\phi(y|x)dy. \tag{5.4}$$

By invoking Bayes' rule, the posterior probability $\theta(x|w)$ of x given the measurement w is proportional to the product of the prior probability $\pi(x)$ and the likelihood $\eta(w|x)$

$$\theta(x|w) = \frac{\pi(x)\eta(w|x)}{Prob(w)} \propto \pi(x)\eta(w|x). \tag{5.5}$$

5.3 Optimization criteria: the MAP and the MPM estimators

Optimization is a widely used technique in computer vision (e.g., image restoration [31], image reconstruction [43, 59, 93], image segmentation [55], stereo [2], motion [72], optical flow [45], texture [17], etc.) possibly due to various uncertainties in vision processes, such as noise in sensed images, and ambiguities in visual interpretation. Because the exact or perfect solution rarely exists, inexact but optimal (in some sense) solutions are usually sought instead.

Since the model for the prior as well as the model for likelihood of the data are available to us by assumption, it seems natural to seek a solution from the Bayesian point of view. Within this framework, the most popular is the *maximum a posteriori probability* (MAP) approach [56]. Other known criteria are the *marginal posterior mode* (MPM) and the *minimum mean square* (MMS) [94] approaches. Unlike the former two, which are discussed shortly, the MMS solution by definition does not correspond to a label image, even though all three estimators aim at minimizing the Bayes risk for some cost functions.

Bayesian approaches are not the only ones. Among the non-Bayesian optimization-based techniques are, e.g., the *maximum likelihood* (ML) approach (in which the label image that maximizes the likelihood function is considered to be the solution), the *maxi-*

mum entropy principle [46], the *minimum description length* principle [78] (this principle is related to ML and MAP under some considerations [56]), etc. Although there have been scientific and philosophical controversies about the appropriateness in inference and decision making (see, e.g., [16]) of the Bayesian or non-Bayesian methods, the Bayes criteria are still popular in computer vision, as represented by the MAP approach in image modeling based on Gibbs distributions [56].

To help the reader in recalling the various abbreviations or acronyms that appear in the rest of this dissertation, they are summarized in Table 5.1.

In Bayesian estimation theory, estimators are studied in terms of *cost functions* [56]. For a given measurement w, let $\widehat{x}(w)$ denote an estimator that hopefully is "close" to the unknown x. The cost of estimating a true x by $\widehat{x}$ is measured by a symmetric "distance" or cost function $\widetilde{C}(x,\widehat{x}) \geq 0$ with the convention that $\widetilde{C}(x,x) = 0$. Given a cost function $\widetilde{C}(x,\widehat{x})$, a posterior probability $\theta(x|w)$, and a measurement vector w, the (Bayes) risk is defined as

abbreviation	meaning	described in (Sub)section	on page
MAP	maximum a posteriori probability	5.3	54
MPM	marginal posterior mode	5.3	55
ML	maximum likelihood	5.3	52
MCMC	Markov chain Monte Carlo	3.4.1	23
MML	mean-by-the-mode likelihood	5.4.1	57
MM-MAP	mean-by-the-mode MAP	5.4.1	57
MM-MPM	mean-by-the-mode MPM	5.4.1	57
PL	pseudo-likelihood	5.4.2	58
P-MAP	pseudo-MAP	5.4.2	58
P-MPM	pseudo-MPM	5.4.2	58
CA-MAP	coordinate ascent MAP	5.8.1	71
SG-MAP	semi-global MAP	5.8.2	72

Table 5.1: Abbreviations or acronyms that are appear in the rest of this dissertation.

$$\widetilde{R}(\widetilde{C},\widehat{x},w) = \sum_{x\in\Lambda^D} \widetilde{C}\,(x,\widehat{x}(w))\,\theta(x|w). \tag{5.6}$$

An estimator that minimizes this risk is known as a *Bayes estimator*:

$$\widehat{x}(w) = \arg\min_{x^*} \widetilde{R}(\widetilde{C},x^*,w). \tag{5.7}$$

The quality of a reconstruction method depends on both the prior model and the estimator (or the cost function). Two popular cost functions are the *all-or-none cost function* and the *Hamming cost function.*

The all-or-none cost function is defined by

$$\widetilde{C}_0\,(x,\widehat{x}) = \begin{cases} 0 & \text{if } \widehat{x} = x, \\ 1, & \text{otherwise.} \end{cases} \tag{5.8}$$

In this case

$$\widetilde{R}\left(\widetilde{C}_0,\widehat{x},w\right) = \sum_{x\in\Lambda^D} \theta(x|w) - \theta(\widehat{x}(w)|w) = 1 - \theta(\widehat{x}(w)|w). \tag{5.9}$$

Clearly, the corresponding estimator is the maximizer of the posterior probability; i.e.,

$$\widehat{x}(w) = \arg\max_{x\in\Lambda^D} [\theta(x|w)], \tag{5.10}$$

which is generally referred to as the *maximum a posteriori probability* (MAP) estimator.

For the Hamming cost function

$$\widetilde{C}_1\,(x,\widehat{x}) = |\{i\,:\,x_i \neq \widehat{x}_i\}|, \tag{5.11}$$

the risk is

$$\widetilde{R}\left(\widetilde{C}_1,\widehat{x},w\right)=\sum_{x\in\Lambda^D}\sum_{i=0}^{|D|-1}\varepsilon\left(x_i,\widehat{x}_i\right)\theta(x|w), \tag{5.12}$$

where

$$\varepsilon\left(x_i,\widehat{x}_i\right)=\begin{cases}0 & \text{if } \widehat{x}_i=x_i,\\ 1, & \text{otherwise.}\end{cases} \tag{5.13}$$

Rearranging the summations in (5.12), we get that

$$\widetilde{R}\left(\widetilde{C}_1,\widehat{x},w\right)=\sum_{i=0}^{|D|-1}\left[\sum_{x\in\Lambda^D}\theta(x|w)-\sum_{x\in\Lambda^D,x_i=\widehat{x}_i}\theta(x|w)\right]=\sum_{i=0}^{|D|-1}\left[1-\theta(\widehat{x}_i|w)\right], \tag{5.14}$$

where $\theta(\widehat{x}_i|w)$ denotes the marginal of $\theta(\widehat{x}|w)$ with respect to the component i. It is easy to see that the corresponding Bayes estimator is the label image whose i-th component is the label that maximizes the marginal posterior; i.e.,

$$\widehat{x}_i=\arg\max_{x_i^*\in\Lambda}\theta(x_i^*|w), \tag{5.15}$$

for $i=0,\cdots,|D|-1$. Such estimator is known in the literature as the *marginal posterior mode* (MPM) estimator.

Regarding the choice of either the MAP or the MPM criterion, there have been controversies regarding their suitability for model-based vision problems (see, e.g., [25, 94]). From the risk minimization point of view, the all-or-none cost function of the MAP criterion is a rather rough measure, since an image that differs from the true image everywhere

has the same distance as those that fail only at one point; whereas the Hamming cost function of the MPM estimator does not make difference between "scattered" and "aggregated" misclassification: a modest number of misclassification are rather harmless if they are scattered, as they would be interpreted as isolated misclassification, which is not the case if they aggregate into some artifact. From the probabilistic point of view, the MAP estimate is by definition the image with the highest posterior probability; whereas the same probability for the MPM estimate may be very low.

The authors in [1, 83] developed new estimators that are sort of a compromise between the MAP and the MPM estimators. The idea is to construct a cost function that assigns positive values to subregions of the image, whose points have been *simultaneously* misclassified. When the subregions are the individual points in the image, we have the MPM estimator; whereas a subregion being the entire image itself corresponds to the MAP estimator. However, it has also been reported that the achievement of specific "effects" in the estimator cannot rely solely on the design of the cost function, because of the dominant influence of the prior model [83]. Here we test only the MAP and the MPM criteria combined with various Gibbs priors from the point of view of their appropriateness for reconstruction purpose.

5.4 Approximations to the posterior probability

There are two challenges concerning the estimations. First, computing the MAP or the MPM estimators implies respectively the finding of the maximum of the posterior probability or its marginal. Unfortunately, closed forms for the optima do not exist and neither do efficient deterministic algorithms for finding them. In this situation MCMC methods (see Subsection 3.4.1) might be the only feasible recourse. In particular for the MAP esti-

mator, because of the non-linearity and the non-convexity of the posterior probability, the optimum cannot be obtained via local search techniques, unless a suboptimal estimator is sought. (We found that a local method, such as the one known as *iterated conditional mode* [7], yields results that are not nearly as good as those provided by our approaches.) Furthermore, the posterior is defined on labels, rather than on continuous numerical variables; therefore, some kind of combinatorial optimization technique needs to be employed.

The second challenge has to do with the fact that the complicated nature of the posterior probability (especially because the likelihood term involves a multidimensional integration) does not make possible an efficient sampling (in the MCMC method); or at least we are not aware of such. Therefore, we investigated the existence of alternative approaches that can be efficiently implemented, and that at the same time deliver good reconstructions. We propose two approximations to the likelihood function; the *mean-by-the-mode likelihood* (MML) and the *pseudo likelihood* (PL).

5.4.1 The mean-by-the-mode likelihood (MML) approximation

In this case the likelihood in (5.4), viewed as an expectation with respect to the probability $\chi(w|y)$, is approximated by the *mean-by-the-mode likelihood* (MML) that is defined as the maximum at the mode

$$\eta_m(w|x) \doteq \max_y [\chi(w|y)\phi(y|x)]. \tag{5.16}$$

The introduction of the MML in (5.10) and (5.15) by replacing $\eta(w|x)$ by $\eta_m(w|x)$ in (5.5), gives rise to, respectively, the *mean-by-the-mode MAP* (MM-MAP) *estimator*

$$x^{mMAP} = \arg\max_x \theta_m(x|w), \tag{5.17}$$

and the *mean-by-the-mode MPM* (MM-MPM) *estimator*

$$x_i^{mMPM} = \arg\max_{x_i \in \Lambda} \theta_m(x_i|w). \tag{5.18}$$

for $i = 0, \cdots, I-1$. (Note that $\eta_m(w|x)$ may not define a valid probability distribution on **W**, because its integral over **W** is in general different from one; therefore it is a probability function up to a constant of proportionality that is being ignored as part of the approximation.)

5.4.2 The pseudo likelihood (PL) approximation

We approximate the likelihood in (5.4) by the *pseudo likelihood* (PL*)* that is defined as

$$\eta_{PL}(w|x) \doteq \prod_{j=0}^{J-1} \varsigma_j(w_j|x), \tag{5.19}$$

where $\varsigma_j(w_j|x)$, for $j = 0, \cdots, J-1$, is the likelihood (probability) of observing w_j given x; i.e., it is the marginal

$$\varsigma_j(w_j|x) = \int_{W_j} \eta(w'|x) dw', \tag{5.20}$$

where

$$W_j = \left\{ w' : w'_j = w_j \right\} \tag{5.21}$$

(the integral in (5.20) is replaced by the sum if **W** is discrete). The *pseudo posterior* (PP) $\theta_{PL}(x|w)$ is obtained by replacing $\eta(w|x)$ by $\eta_{PL}(w|x)$ in (5.5). The introduction of the PL

in (5.10) and (5.15) by replacing $\eta(w|x)$ by $\eta_{PL}(w|x)$ in (5.5) gives rise to, respectively, the *pseudo-MAP* (P-MAP) *estimator*

$$x^{PMAP} = \arg\max_{x} \theta_{PL}(x|w), \tag{5.22}$$

and the *pseudo-MPM* (P-MPM) *estimator*

$$x_i^{PMPM} = \arg\max_{x_i \in \Lambda} \theta_{PL}(x_i|w), \tag{5.23}$$

for $i = 0, \cdots, I-1$.

The PL approximation is justified when, e.g., the measurements are very noisy, which is likely to be the case in electron tomography. The PL approximation assumes that the likelihood can be factorized into a product of marginals, each of which corresponds to a single component of the measurement vector. In other words, given the label image, the measurements are assumed to be statistically independent. This is valid in many tomographic reconstruction problem settings (see [42, 86, 88], to mention a few), but unfortunately it is not so in our framework, because of the dependencies coupled by the gray values. Intuitively, however, it is reasonable to argue that these dependencies are rather "weak" when the measurements are very noisy, which is likely to be the case in practice. Hence, the PL is in general a good approximation to the true likelihood. Below we give a formal justification for this approximation by giving a relationship between PL and the true likelihood under the normality assumption.

5.5 Normality assumption on the likelihood

The two approximations to the posterior were necessary because a direct sampling of the posterior is intractable in practice. This is due the fact that the likelihood is defined as a multi-dimensional integration (5.4), which has to be evaluated repeatedly and efficiently during the optimization. When $\chi(w|y)$ and $\phi(y|x)$ are both normally distributed, there exists a closed form for the likelihood. However, a direct manipulation of the closed form is impractical due to the necessity of inverting a large (covariance) matrix. We do not use this form in the optimization, but instead we provide its relationship with the two approximations just introduced. We consider normal distributions because they are reasonably realistic (except for the negativity of the gray values and the measurements, whose effect can be kept minimal by appropriate parameter selection). The discussions here are restricted to the case when all the covariance matrices are non-singular. Some cases with singularities are treated in Section 5.6.

Let $\phi(y|x)$ be a normal distribution with mean vector μ_x and covariance matrix Σ_x; i.e.,

$$\phi(y|x) = Normal\,(\mu_x, \Sigma_x) \tag{5.24}$$

and $Y = \mathbb{R}$. Here we work with the special case in which $\Sigma_x = \mathrm{diag}_{0\le i\le I-1}(\sigma_{xi}^2)$ is positive definite. Assume that the measurement is a (Gaussian) noisy linear transformation, as in (5.2), of the gray value image; i.e.,

$$\chi(w|y) = Normal\,(Ry, \Sigma_w)\,, \tag{5.25}$$

where Σ_w is the covariance matrix of the noise; here we consider the case in which $\Sigma_w = \mathrm{diag}_{0\le j\le J-1}(\sigma_{wj}^2)$ is also positive definite.

At this point it is convenient to view the gray value image y and the measurement vector w as instances, respectively, of the associated random vectors $\mathbf{y}$ and $\mathbf{w}$. Accordingly, (5.24) implies that

$$\mathbf{y} \sim Normal\,(\mu_x, \Sigma_x) \tag{5.26}$$

and (5.25) implies that

$$\mathbf{w} = R\mathbf{y} + \mathbf{n}, \tag{5.27}$$

where $\mathbf{n}$ is the random vector associated with the measurement noise; i.e., $\mathbf{n} \sim Normal\,(0, \Sigma_w)$.

A result in probability theory [47] establishes that $R\mathbf{y}$ is normally distributed random vector with mean $R\mu_x$ and covariance $R\Sigma_x R^t$. Knowing this and assuming (as reasonable) that $R\mathbf{y}$ and $\mathbf{n}$ are uncorrelated, we have the probability law for the likelihood

$$\eta(w|x) = Normal\left(R\mu_x, R\Sigma_x R^t + \Sigma_w\right). \tag{5.28}$$

More explicitly, if we set $\mu_w = R\mu_x$ and $\Sigma_{wx} = R\Sigma_x R^t + \Sigma_w$, then

$$\eta(w|x) = \frac{1}{(2\pi)^{J/2}\,|\Sigma_{wx}|^{1/2}} \exp[-\frac{1}{2}(w-\mu_w)^t \Sigma_{wx}^{-1}(w-\mu_w)], \tag{5.29}$$

where Σ_{wx} is also positive definite, because so are Σ_x and Σ_w by assumption.

5.5.1 The mean-by-the-mode likelihood (MML) under the normality assumption

We establish a relationship between the MML and the true likelihood. Using (5.24) and (5.25), we can write (recall that Σ_x and Σ_w are non-singular by assumption)

$$\chi(w|y)\phi(y|x) = \frac{1}{C_w C_x}\exp\left[-\frac{1}{2}(w-Ry)^t\Sigma_w^{-1}(w-Ry) - \frac{1}{2}(y-\mu_x)^t\Sigma_x^{-1}(y-\mu_x)\right], \quad (5.30)$$

where $C_w = (2\pi)^{J/2}|\Sigma_w|^{1/2}$ and $C_x = (2\pi)^{I/2}|\Sigma_x|^{1/2}$ (both independent of y) are the normalizing constants for $\chi(w|y)$ and $\phi(y|x)$, respectively. Next, if we denote

$$\widetilde{q}(y) = -\frac{1}{2}(w-Ry)^t\Sigma_w^{-1}(w-Ry) - \frac{1}{2}(y-\mu_x)^t\Sigma_x^{-1}(y-\mu_x), \quad (5.31)$$

then $\widetilde{q}(y)$ is a quadratic in y. It is convenient to write $\widetilde{q}(y)$ in the form

$$\widetilde{q}(y) = -\frac{1}{2}(y-\mu)^t\Sigma^{-1}(y-\mu) + \delta, \quad (5.32)$$

for some matrix Σ and vectors μ and δ that are independent of y. After some algebra, we get that

$$\begin{aligned} \Sigma^{-1} &= R^t\Sigma_w^{-1}R + \Sigma_x^{-1} \\ \mu &= -\tfrac{1}{2}\Sigma a \\ \delta &= \Theta - \tfrac{1}{4}a^t\Sigma a \end{aligned} \quad (5.33)$$

where $\Theta = \mu_x^t\Sigma_x^{-1}\mu_x + w^t\Sigma_w^{-1}w$, $a^t = -2\left(w^t\Sigma_w^{-1}R + \mu_x^t\Sigma_x^{-1}\right)$, and Σ is positive definite (because so are Σ_x and Σ_w). This suggests that $\chi(w|y)\phi(y|x)$ has the "shape" of a normal distribution with mean μ and covariance matrix Σ. Noting (5.30), (5.32) and the fact that

the mean coincides with the mode in a normal distribution, (5.16) can be written as

$$\max_y [\chi(w|y)\phi(y|x)] = \frac{1}{C_w C_x} \exp[\widetilde{q}(\mu)] = \frac{e^{\delta}}{C_w C_x}. \tag{5.34}$$

Meanwhile, for the likelihood in (5.4), we use (5.30), (5.32), and the normalization condition:

$$\int_{\mathbf{Y}} \frac{1}{(2\pi)^{I/2} |\Sigma|^{1/2}} \exp[-\frac{1}{2}(y-\mu)^t \Sigma^{-1}(y-\mu)] dy = 1. \tag{5.35}$$

to express it as

$$\int_{\mathbf{Y}} \chi(w|y)\phi(y|x) dy = \int_{\mathbf{Y}} \frac{1}{C_w C_x} \exp[\widetilde{q}(y)]\, dy = \frac{(2\pi)^{I/2} |\Sigma|^{1/2} e^{\delta}}{C_w C_x}. \tag{5.36}$$

By comparing (5.34) with (5.36) and recalling (5.4) and (5.16), we conclude that

$$\eta_m(w|x) = (2\pi)^{-I/2} |\Sigma|^{-1/2} \eta(w|x). \tag{5.37}$$

From the relations in (5.33), we can see that if the noise in the measurements is very high, then $\Sigma^{-1} \approx \Sigma_x^{-1}$; whereas if the noise is relatively low, then $\Sigma^{-1} \approx R^t \Sigma_w^{-1} R$, which does not depends on the labels and therefore can be ignored during the optimization with respect to x.

5.5.2 The pseudo likelihood (PL) under the normality assumption

Clearly, in this case the marginal distributions $\varsigma_j(w_j|x)$ in (5.20) becomes, noting (5.28),

$$\varsigma_j(w_j|x) = Normal\left(\mu_{wj}, \sigma^2_{PLj}\right), \tag{5.38}$$

where, for $0 \leq j \leq J-1$, μ_{wj} is the j-th component of $\mu_w = R\mu_x$ and $\sigma^2_{PLj} = r_j^t \Sigma_x r_j + \sigma^2_{wj} > 0$, with r_j being the transpose of the j-th row of R. Hence, the PL becomes

$$\eta_{PL}(w|x) = Normal\left(\mu_w, \Sigma_{PL}\right), \tag{5.39}$$

where we have set $\Sigma_{PL} = \text{diag}_{0\leq j\leq J-1}(\sigma^2_{PLj})$. A closer look tells us that Σ_{PL} is in fact the diagonal part of Σ_{wx} in (5.29); thus we can set $\Sigma_{wx} = \Sigma_{PL} - \Re$, where the entries of $\Re$ are the off-diagonal elements of Σ_{wx} but with the negative sign. If $\eta_{PL}(w|x)$ is a good approximation to $\eta(w|x)$, then Σ_{PL} is also good approximation to Σ_{wx}. Assuming that $\left|\Sigma^{-1}_{PL}\Re\right| < 1$, one can think of Σ^{-1}_{PL} as the 0^{th} order approximation to $\Sigma^{-1}_{wx} = (\Sigma_{PL} - \Re)^{-1}$ in the series expansion

$$\Sigma^{-1}_{wx} = \Sigma^{-1}_{PL}\left(I_J - \Sigma^{-1}_{PL}\Re\right)^{-1} = \Sigma^{-1}_{PL}\left(I_J + \Sigma^{-1}_{PL}\Re + \ldots\right). \tag{5.40}$$

The condition $\left|\Sigma^{-1}_{PL}\Re\right| < 1$ is generally true when, e.g., the noise level is high, which is the case in electron tomography; hence, the PL is a good approximation to the true likelihood.

5.6 Trivial assignments of the probabilities

Prior to the discussion of the optimization algorithms in the next section, we study here some trivial assignments of the probabilities $\chi(w|y)$ and $\phi(y|x)$ of Section 5.2, as these may provide us with the best performance that one can expect from a particular optimization algorithm.

5.6.1 The gray value image is uniquely determined

The first trivial case is when given a label image x the gray value image is uniquely determined by that x and is equal to μ_x; i.e., $\phi(y|x) = \delta(y - \mu_x)$ (δ is the Dirac's delta function). This is the most common setting in discrete tomography; see, e.g., [14, 22, 75, 93]. In this case, noting (5.4) and (5.3), the likelihood is

$$\eta(w|x) = \int_{\mathbf{Y}} \chi(w|y)\phi(y|x)dy = \chi(w|\mu_x) = \prod_{j=0}^{J-1} \upsilon_j(w_j|\mu_x). \tag{5.41}$$

Since the likelihood is a product of probabilities of the measurements, it coincides with the PL; and therefore the P-MAP and the P-MPM estimators coincide, respectively, with the MAP and the MPM estimators. The MML is undefined for this trivial assignment.

5.6.2 Measurement is the gray value image

The second trivial assignment (already anticipated in Subsection 5.2.3) in which we are interested is when the measurement is the gray value image itself and is noiseless; i.e., $\chi(w|y) = \delta(w - y)$, $I = J$, and $\mathbf{W} = \mathbf{Y}$. Consequently, using (5.1) and (5.4),

$$\eta(w|x) = \int_{\mathbf{Y}} \chi(w|y)\phi(y|x)dy = \phi(w|x) = \prod_{i=0}^{I-1} \varphi(w_i|x_i). \tag{5.42}$$

Using an argument similar to that given in the previous subsection, here we also have that the PL is the true likelihood, but the MML is undefined.

This special case is relevant to the current approaches of first reconstructing the gray value image and then segmenting it to obtain a label image. Since under the assumption of this subsection the reconstruction of the gray value image is exactly the gray value image itself, a segmentation of it is the best that we can expect from current approaches when using a few projections. An optimum way of segmenting a gray value image is the *minimum-error-rate classifier* [21]

$$x^{TRAD} = \arg\max_{x} \prod_{i=0}^{I-1} \varphi(w_i|x_i) Prob(x_i), \tag{5.43}$$

where $Prob(x_i)$ is the probability of occurrence of the label x_i at point i, which can be estimated from a training set of typical label images.

In real applications, since a reconstructed gray value image is usually "far" from the exact one, a different segmentation procedure is employed. One of the standard techniques has been already addressed in Section 2.2.

5.7 Algorithms for finding the optimum label image

5.7.1 Introduction

In Section 5.4 we have mentioned that we need to resort to MCMC-type of algorithms if we are interested in the global optimum. In particular, since the search space is discrete and of high dimension, methods like simulated annealing [13, 51] may be the only feasible recourse for finding the MAP estimators. We discard other combinatorial search techniques,

such as *evolutionary algorithms* [24], due to the impractically large memory storage. On the other hand, for the MPM estimator one can estimate the modes simply by sampling the posterior probability. In fact, both simulated annealing and the sampling of a probability distribution can be carried out by MCMC algorithms (see Subsection 3.4.1).

In this section we address our general approaches to finding the MAP and the MPM estimators for the MML and the PL approximations (i.e., formulas (5.17), (5.18), (5.22), and (5.23)), as alternatives to the existing local search techniques.

Since usually the design of an optimization algorithm depends strongly on the mathematical formulation, in the next two sections we explain the details of our approaches and the estimators that result from them. In particular we present in Section 5.8 three estimators for the MML approximation, which we term the *CA-MAP*, the *SG-MAP*, and the *MM-MPM* estimators, and in Section 5.9 we discuss two estimators for the PL approximation: the *P-MAP* and the *P-MPM* estimators.

5.7.2 Local optimization methods

Sometimes methods that only guarantee a local optimum are preferred over global methods, simply because of the ease of their implementations. A local method is usually iterative, and its solution can be strongly dependent on the starting image. Unless either a good choice of the starting image is assured or a local optimum is proved to be as "useful" as (if not equivalent to) the global optimum, it has been suggested in [56] that local methods are inferior to global methods. This conclusion was based on the local methods called *iterated conditional mode* (ICM) [7], *expectation maximization* [18, 56], and *relaxation labeling* [80]; all tested for image restoration using Gibbs priors. We tested (see later in Section 6.5) the performance of the ICM method in the context of our reconstruction problem and

found that the quality (to be defined later) of the reconstructions is significantly lower than that produced by our proposed methods.

5.7.3 A global method: simulated annealing

In addition to drawing samples from a given distribution, sampling algorithms such as the Metropolis algorithm (see Subsection 3.4.1) are also employed in the so-called *simulated annealing* [13, 51]. Often used as a combinatorial optimization technique, simulated annealing simulates the physical annealing process, in which a substance is melted and then slowly cooled down to reach a low energy configuration. To achieve this, simulated annealing successively applies a sampling algorithm to the function $[\gamma(x)]^{\beta}$, where $\gamma(x)$ is the (positive) function being maximized (with respect to x) and $\beta = 1/T$ is a positive real number. The parameter T is usually referred to as the *temperature*. Initially, T is set to a high value, so that the graph of $[\gamma(x)]^{1/T}$ is nearly flat. At each subsequent step of the sampling algorithm, T is decreased to a lower value according to an *annealing schedule*. A theorem of [31] gives an annealing schedule that guarantees that the process converges (almost surely) to a global maximum. The conditions of the theorem are sufficient but not necessary. In practice, however, the annealing schedule in [31] is too slow and faster schedules are used instead. We follow the procedure proposed in [93], where, for efficiency and computational cost saving, T is kept fixed during a fixed number of *cycles* (see Subsection 3.4.2) before its value is lowered and the sample image with the highest $[\gamma(x)]^{1/T}$, for the current T, is selected as the starting image for the next temperature.

5.7.4 Our general approaches to finding the MAP and the MPM estimators for the PL and the MML approximations

Given the measurement vector, our goal is to find the MAP estimate and the MPM estimate, both based on the posterior probability. Our first challenge is the non-convexity and the non-linearity of the posterior that impede us from using local methods to find a global optimum. Although local methods are very efficient and easy to implement, they can only guarantee local solutions that can be highly sensitive to the starting image.

This leads us to consider only global methods, such as the simulated annealing (for the MAP estimate) and sampling methods (for the MPM estimate). Since a simulated annealing process consists of a sequence of sampling algorithms (in particular, MCMC based), the optimization task becomes a sampling task. Due to the fact that a MCMC algorithm (e.g., the Metropolis algorithm; see Subsection 3.4.1) is inherently slow, it is critical that the ratio $\gamma(x^{(n)})/\gamma(x^{(u)})$ can be evaluated efficiently in each step of the Metropolis algorithm. The distribution γ in this case is proportional to $\pi(x)\eta'(w|x)$, where $\eta'(w|x) = \eta_m(w|x)$ in the MML approximation and $\eta'(w|x) = \eta_{PL}(w|x)$ in the PL approximation. To be precise, the target distribution (see Subsection 3.4.1) is either $\gamma(x)$, if we wish to obtain the MPM estimate, or the distribution proportional to $[\gamma(x)]^{1/T}$ (see previous subsection), if we need the MAP estimate. Nevertheless, since the temperature parameter T can be absorbed into the computation of (pre-generated pseudo-) random numbers [93], we only need to be concerned with the distribution γ itself for both estimates. We have mentioned in Subsection 3.4.2 that in order to speed up the Metropolis algorithm, all the possible values of the log-ratio $\log\left[\pi\left(x^{(n)}\right)/\pi\left(x^{(u)}\right)\right]$ should be pre-calculated and stored in a look-up table [93]. Following this idea, we also pre-calculate all the possible values (or their approximations)

of the log-ratio

$$\log\left[\frac{\eta'\left(w|x^{(n)}\right)}{\eta'\left(w|x^{(u)}\right)}\right], \tag{5.44}$$

where, again, $\eta'(w|x) = \eta_m(w|x)$ in the MML approximation and $\eta'(w|x) = \eta_{PL}(w|x)$ in the PL approximation. In the rest of this chapter we give the details of the log-ratio under the various circumstances.

5.8 Algorithms for the MML approximation

In this case we are interested in sampling the target distribution $\theta_m(x|w) \propto \pi(x)\eta_m(w|x)$ (see Subsection 5.4.1). For the MM-MAP estimate, we must find, according to (5.17) and (5.16),

$$x^{mMAP} = \arg\max_x \theta_m(x|w) = \arg\max_x \left\{\pi(x)\max_y\left[\chi(w|y)\phi(y|x)\right]\right\}. \tag{5.45}$$

We propose two optimization algorithms: one is the *coordinate ascent* (CA) *approach* [63] and the other is the *semi-global* (SG) *approach* [64], and we call the corresponding estimators as the CA-MAP estimator and the SG-MAP estimator.

In Subsection 5.8.3 we explain how we obtain the MM-MPM estimator (that was defined in Subsection 5.4.1).

5.8.1 MAP estimator by the coordinate ascent (CA) approach: the CA-MAP estimator

Let $O_1(x,y)$ and $\widetilde{y}(x)$ be functions that are defined by

$$O_1(x,y) = \chi(w|y)\phi(y|x) \tag{5.46}$$

and

$$\widetilde{y}(x) = \arg\max_y [O_1(x,y)]\,. \tag{5.47}$$

Then−using (5.16)−both $\widetilde{y}(x)$ and

$$O_1\left(x,\widetilde{y}(x)\right) = \max_y [O_1(x,y)] = \eta_m\left(w|x\right) \tag{5.48}$$

are functions of x, and (5.45) becomes

$$x^{mMAP} = \arg\max_x \{\pi(x) O_1\left(x,\widetilde{y}(x)\right)\}\,. \tag{5.49}$$

In the CA approach [63] we start from a gray value image (see Appendix E) and alternately maximize

$$O_2(x,y) = \pi(x) O_1(x,y) \tag{5.50}$$

with respect to (the coordinate) x and with respect to (the coordinate) y. The two maximizations that we term respectively the *x-step* and the *y-step* are carried out using global methods. The first one is performed by simulated annealing via the Metropolis algorithm

and the second one, under the normality assumption, is a quadratic optimization problem (see Appendix F.1). Since the algorithm is greedy with respect to each coordinate, this approach is local, and therefore only a local optimum is guaranteed.

Following the general idea of Subsection 5.7.4, for computational efficiency of an x-step aiming at maximizing $O_2(x,y)$, we pre-calculate the log-ratio−using (5.50) and (5.46)−

$$\log\left[\frac{O_1(x^{(n)},\widetilde{y}_0)}{O_1(x^{(u)},\widetilde{y}_0)}\right]=\log\left[\frac{\phi\left(\widetilde{y}_0|x^{(n)}\right)}{\phi\left(\widetilde{y}_0|x^{(u)}\right)}\right], \tag{5.51}$$

where where $\widetilde{y}_0$ is the initial gray value image or the one computed in the y-step immediately before the current x-step. Note that according to (5.44), the log-ratio that we really need is
$\log\left[\eta_m\left(w|x^{(n)}\right)/\eta_m\left(w|x^{(u)}\right)\right]$, which is in this case−using (5.46) and (5.48)−

$$\log\left[\frac{\eta_m\left(w|x^{(n)}\right)}{\eta_m\left(w|x^{(u)}\right)}\right]=\log\left[\frac{O_1(x^{(n)},\widetilde{y}(x^{(n)}))}{O_1(x^{(u)},\widetilde{y}(x^{(u)}))}\right]=\log\left[\frac{\phi\left(\widetilde{y}(x^{(n)})|x^{(n)}\right)}{\phi\left(\widetilde{y}(x^{(u)})|x^{(u)}\right)}\right]. \tag{5.52}$$

5.8.2 MAP estimator by the semi-global (SG) approach: the SG-MAP estimator

This approach (algorithm B of [64]) is a variant of the CA approach. We estimate x^{mMAP} of (5.49) by starting from a gray value image and then run one single x-step, *during* which several y-steps are carried out. That is, suppose that y_0 is the starting gray value image, then instead of applying an *entire* x-step (which consists of simulated annealing via the Metropolis algorithm) based on y_0 as in the CA approach, we run Metropolis algorithm partially for only a (fixed) number of cycles. Suppose that, as a result, we get label image

x'. We then apply a y-step based on this x', which produces $\widetilde{y}(x')$, and continue running the Metropolis algorithm based on this $\widetilde{y}(x')$ for the same number of cycles; and so forth.

This process of constantly updating the gray value image while running simulated annealing is justified by noting that if we could evaluate $\widetilde{y}(x)$ for *every* sampled x (which is prohibitive, due to the typically enormous number of samples x involved in a Metropolis algorithm), then we would get the exact x^{mMAP}. Thus we term this approach the *semi-global approach.* In the experiments we update of the gray value image every 5,000 cycles−no significant improvements were observed by using lower (up to 500) cycles. To increase algorithmic efficiency, we use some additional tunning parameters that are discussed in [64].

The log-ratio that we pre-calculate in this case is

$$\log\left[\frac{O_1(x^{(n)},\widetilde{y}(x'))}{O_1(x^{(u)},\widetilde{y}(x'))}\right] = \log\left[\frac{\phi\left(\widetilde{y}(x')|x^{(n)}\right)}{\phi\left(\widetilde{y}(x')|x^{(u)}\right)}\right], \tag{5.53}$$

where $\widetilde{y}(x')$ is either the starting gray value image or the one produced by the most recent y-step. The log-ratio (5.53) is another approximation to (5.52).

5.8.3 MM-MPM estimator

As stated in Subsection 5.7.4, in order to obtain the MM-MPM estimator of (5.18), we need to consider the same target distribution as the one used in the MAP estimator above, namely $\pi(x)O_1\left(x,\widetilde{y}(x)\right)$ in (5.49), since both estimators can be found by sampling the same distribution. In this case we compute the log-ratio of (5.44) exactly as in (5.53).

5.9 Algorithms for the PL approximation

Following the discussions in Subsection 5.7.4, here we concentrate on the target distribution $\theta_{PL}(x|w) \propto \pi(x)\eta_{PL}(w|x)$ for both the MAP and the MPM estimate, which−recall (5.5), (5.22), and (5.23)−give rise to the P-MAP estimator and the P-MPM estimator. In this case, using (5.39), the log-ratio in (5.44) under the normality assumption is

$$\log\left[\frac{\eta_{PL}\left(w|x^{(n)}\right)}{\eta_{PL}\left(w|x^{(u)}\right)}\right] = \sum_{j:r_{ij}\neq 0}\left[\log\frac{\sigma_{PLj}\left(x^{(u)}\right)}{\sigma_{PLj}\left(x^{(n)}\right)} + \frac{\left(r_j^t\mu_{x^{(u)}} - w_j\right)^2}{2\sigma^2_{PLj}\left(x^{(u)}\right)} - \frac{\left(r_j^t\mu_{x^{(n)}} - w_j\right)^2}{2\sigma^2_{PLj}\left(x^{(n)}\right)}\right], \tag{5.54}$$

where $x^{(n)}$ and $x^{(u)}$ differ only at the i-th point (recall Subsection 3.4.1) and, from Subsubsection 5.5.2,

$$\sigma^2_{PLj} = r_j^t\Sigma_x r_j + \sigma^2_{wj}. \tag{5.55}$$

Since we would also like to test the special assignment of Subsection 5.6.1, in which the PL is the true likelihood, we compute the log-ratio (5.54), by taking into account (5.25) and (5.41) and setting

$$\sigma^2_{PLj} = \sigma^2_{wj}. \tag{5.56}$$

In order for the the log-ratio in (5.54) to be pre-calculated exactly and efficiently stored, the components of r_j (for $0 \leq j \leq J-1$) should have only a few (one, in our experiments) possible non-zero values (see Subsections 6.1 and 7.4.2). Specifically, suppose that there

are exactly T_j non-zero components of r_j, all having a same value. Then, since we are dealing with binary images, there can be only $T_j + 1$ possible values for the expressions $r_j^t \mu_{x^{(u)}}$, $r_j^t \mu_{x^{(n)}}$ and $r_j^t \Sigma_x r_j$ in (5.54) and (5.55). Each value depending solely on the number of grid points (from 0 to T_j) that are labeled 1 (or 0) along the line corresponding to the j-th measurement. Therefore, given w_j and σ_{wj}^2 ,we need only to precalculate and store $T_j + 1$values for the j-th measurement.

5.10 Summary

We summarize this chapter by highlighting our five estimators. We have proposed two approximations to the likelihood function, each one of which used in a MPM-type or a MAP-type of estimator (see Table 5.2).

estimator	MML approximation	PL approximation
MAP	CA-MAP/SG-MAP	P-MAP
MPM	MM-MPM	P-MPM

Table 5.2: Our proposed five estimators.

Chapter 6

Experiments on 2D images

In this chapter we apply our approaches (namely the five estimators CA-MAP, SG-MAP, MM-MPM, P-MAP, and P-MPM from the previous chapter; see Table 5.2) to reconstructing two-dimensional (2D) binary images. Given some phantoms that are label images, we generate their corresponding gray value images (one gray value image for each phantom). and recover them from simulated projections taken from the gray value images. An indicator that tells us how good these reconstructions are is the average percentage of misclassified pixels. We first describe our experimental settings in Section 6.1.

In Section 6.2 we compare the MAP estimator against the MPM estimator, under the trivial assignment of Subsection 5.6.1, in which a gray value image is uniquely determined given the label image. We know that in this case the P-MAP and the P-MPM estimators coincide, respectively, with the MAP and the MPM estimators.

In Section 6.3 we study the performance of the MML approximation (see Section 5.8) by comparing the CA-MAP, SG-MAP, and the MM-MPM estimators.

We then evaluate in Section 6.4 the P-MAP and the P-MPM estimators under the PL approximation (see Section 5.9).

Finally, in Section 6.5, we test our five estimators and compare them with the ICM method [7] (see Subsection 5.7.2) and a current method (namely ART with pixels; see Section 2.2) of first reconstructing the gray value image and then segmenting it to obtain the label image. We also investigate the approach of Subsection 5.6.2 in which we assume a perfect reconstruction of the gray value image and then segment it optimally.

6.1 Choice of the experimental variables

We take 2D binary (two-label) images of size $I = 63 \times 63$ with the set of labels $X = \{0,1\}$. In Sections 6.2, 6.3, and 6.4 we reconstruct images that are typical samples from the Gibbs distribution of our model (1.2, 1.2, 1.2, 0.52, 0.2) $-$Subsection 3.4.3. The parameters are such that the prior assigns higher probabilities to images that have relatively large uniform regions (which, in this case, are labeled 1) over a background (labeled 0); such type of images is quite common in many discrete tomography problems; see e.g., [14, 22, 75]. Later, in Section 6.5, we reconstruct label images that represent the cross section of a macromolecule in four different conformations.

For each one of the label images, we randomly picked a gray value image (with $I = 63 \times 63$ and $Y = \mathbb{R}$). The product structure of $\phi(y|x)$ implies that for a label image, a gray value image can be sampled by sampling at each pixel according to the probability $\varphi(y_i|x_i)$, for $1 \leq i \leq I$.

Normality is considered in all the algorithms. The $\varphi(y_i|x_i)$, for $1 \leq i \leq I$, was assumed to be normally distributed with mean μ_{x_i} and variance $\sigma^2_{x_i}$, where $\mu_{x_i} = \sigma^2_{x_i}$, $\mu_0 = 4$, and $\mu_1 = 9$. However, for the data simulation, to avoid negative density (gray value), the value of y at a pixel was set to zero whenever it was negative (with probability less than 0.028 if $x_i = 0$ and less than 0.01 if $x_i = 1$) according to $\varphi(y_i|x_i)$. There is no particular reason

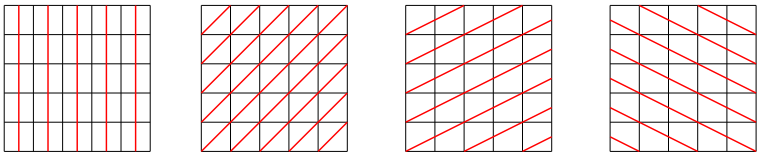

Figure 6.1: Illustration on a 5×5 image of the chosen projection lines, so that the length of intersection of a line with a pixel is the same in the direction with tangent equal to (respectively from left to right) infinity, 1, 0.5, and -0.5. 90 degree rotations of these directions give the directions with tangents equal to, respectively, 0, -1, -2, and 2.

for choosing these values of mean and variance, except for the fact that they reflect the idea of overlapping gray values in higher resolution electron microscopy, as shown by the left histogram in Figure 2.3. The projections forming the measurement vector w were simulated as follows. There were either four or eight projections using parallel lines in each projection, such that the tangent of the angle, denoted by α, between these lines and the "positive horizontal direction" is 0, infinity, -1, and 1 in the case of four projections and it was 0, infinity, -1, 1, -0.5, 0.5, -2, and 2 in the case of eight projections. We chose parallel lines so that the non-zero lengths of intersections of a line with the pixels are all the same in one direction; see Figure 6.1. The distance between two consecutive parallel lines was $l \cdot \max(\cos\alpha, \sin\alpha)$, where l is the length of a side of a pixel. This choice of the location of the lines is not a necessary condition for validity of the our approaches, but it simplifies their implementation (especially in the PL approximation; see Section 5.9). For each experiment we formed a J-dimensional column vector $z = (z_1, \ldots, z_J)^t$ from the sums of the pixel values in the gray value image along the J lines in all the projections.

We chose $\upsilon_j(w_j|y)$, for $1 \leq j \leq J$, to be normally distributed with mean z_j, and we initially intended to set the variance to be $\sigma^2_{wj} = N \cdot z_j$. We call N the *noise level*, whose possible values were chosen to be $0.25, 1.0,$ or 4.0. Since it is possible that $z_j = 0$, which would not produce a positive definite (diagonal) covariance matrix Σ_w in (5.25), we replace

σ_{wj}^2 by $N \cdot \widetilde{w}_j = N \cdot \max\left(\mu_0, w_j\right)$, where w_j is sampled from $\upsilon_j(w_j|y)$. The preceding is how we modeled $\upsilon_j(w_j|\mathbf{y})$ in the algorithms. For the simulated noisy projections, to avoid negative w_j we set $w_j = \widetilde{w}_j$, for $j = 0, \cdots J-1$. The justification for this replacement is the following. The original variance $N \cdot z_j$ is proportional to the mean z_j, for $0 \leq j \leq J-1$, which implies that (i) higher measurement suffer from higher noise, but (ii) the ratio between the latter and the former is lower for higher measurement. This is in agreement with real applications, at least as a first order approximation. Since a real noise process is not exactly normally distributed anyway, we expect that the replacement of z_j by $\widetilde{w}_j$ in σ_{wj}^2 will not significantly violate the properties (i) and (ii) for small N. Next, since by definition the variance must be non-negative, we use μ_0, which is the minimum value attainable by $\left(R\mu_x\right)_j$, as the lower bound for $\widetilde{w}_j$.

For the three MAP estimators (CA-MAP, SG-MAP, and P-MAP) we ran $5 \cdot 10^4$ cycles of the Metropolis algorithm for each temperature T, where $1/T$ ranged from 0.5 to 1.4 in increments of 0.05. We did not observe significant improvements by increasing the range (from 0.5 up to 2) of $1/T$. For the two MPM estimators (the MM-MPM and the P-MPM) we took a total of 1,000 samples for computing the mode, and each sample was obtained by running $2 \cdot 10^4$ cycles (which is sufficient for the burn-in). A mode based on 1,000 samples was practically the same as that based on 5,000 samples.

For a current approach of Section 2.2, we used ART with pixels, which was set to be the initial gray value image in the CA approach (see Appendix E); and also the method of Subsection 5.6.2, which assumes a perfect reconstruction of the gray value image and then thresholds it optimally (which is likely to yield a result that is the best we can expect from the current approach).

Table 6.1: Percentage of misclassification of the exact MAP estimator and the exact MPM estimator (N is the noise level).

Method	$N = 0.25$	$N = 1.0$	$N = 4.0$
MAP with 4 projections	1.5±0.9	5.0±3.1	12.8±5.1
MPM with 4 projections	1.6±0.9	4.5±2.1	11.4±4.0
MAP with 8 projections	0.7±0.2	2.3±1.0	7.9±3.2
MPM with 8 projections	0.7±0.3	2.0±0.7	6.6±2.0

6.2 MAP vs MPM

In this section we compare the performance of the MAP estimator versus that of the MPM estimator. Under the trivial assignment of Subsection 5.6.1, in which a gray value image is uniquely determined given the label image, the P-MAP and the P-MPM estimators are exactly the MAP and the MPM estimators, respectively. Specifically, we implement (5.54), in conjunction with (5.56).

In Table 6.1 we report on the quality of both the MAP estimator and the MPM estimator. The significance of the differences in reconstruction quality between the two estimators is measured using the pairwise *t*-test [71]. In the case of $N = 1.0$ with four projections, we found that the MPM is better than the MAP with a *P*-value of about 0.025. The difference is even more significant in the rest of the cases ($N = 1.0$ with eight projections and $N = 4.0$) in which the *P*-value is less than 0.0025. However, in the case of $N = 0.5$, we found no such significance. Actual reconstructions of one phantom are depicted in Figure 6.2.

6.3 MML approximation

In this case we compute and compare the CA-MAP, SG-MAP, and the MM-MPM estimates. (see Sections 5.4.1 and 5.8). The details of the implementations were explained in

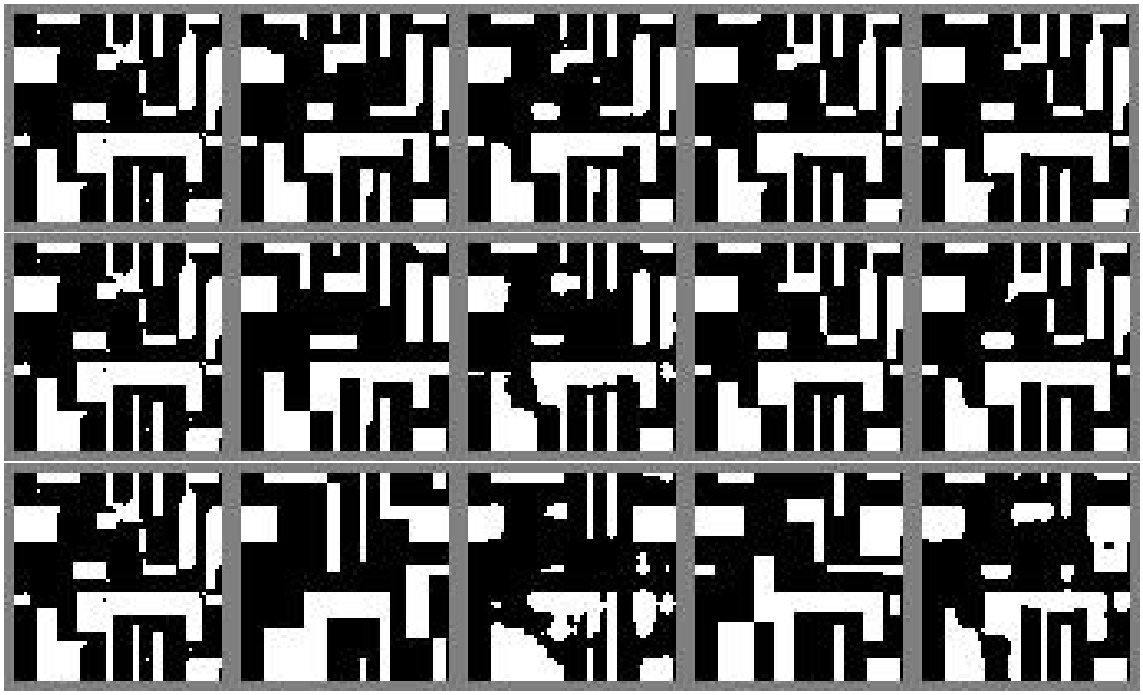

Figure 6.2: MAP estimator vs. MPM estimator. From left to right in the top row are a phantom, its MAP and MPM estimates from four projections, and the MAP and MPM estimators from eight projections; all the reconstructions correspond to the noise level $N = 0.5$ and the number of misclassification are 164, 137, 42, and 46. The central row, with the same arrangement, corresponds to $N = 1.0$ with 416, 313, 129, and 109 misclassification. The bottom row, also with the same arrangement, corresponds to $N = 4.0$ with 930, 777, 666, and 470 misclassifications.

Section 5.8 and further in Appendix E for the CA approach.

6.3.1 CA-MAP and SG-MAP estimators

Figure 6.3 shows the actual reconstructions of one phantom. Table 6.2 reports on the quality of the two estimates for the three noise levels. We can see that the SG approach outperforms the CA approach when eight projections are used. Otherwise, it seems that the frequent update (y-step) of the gray value image during the optimization (x-step) of the label image (see Subsection 5.8.3) does not help, and the SG technique provides less satisfactory results

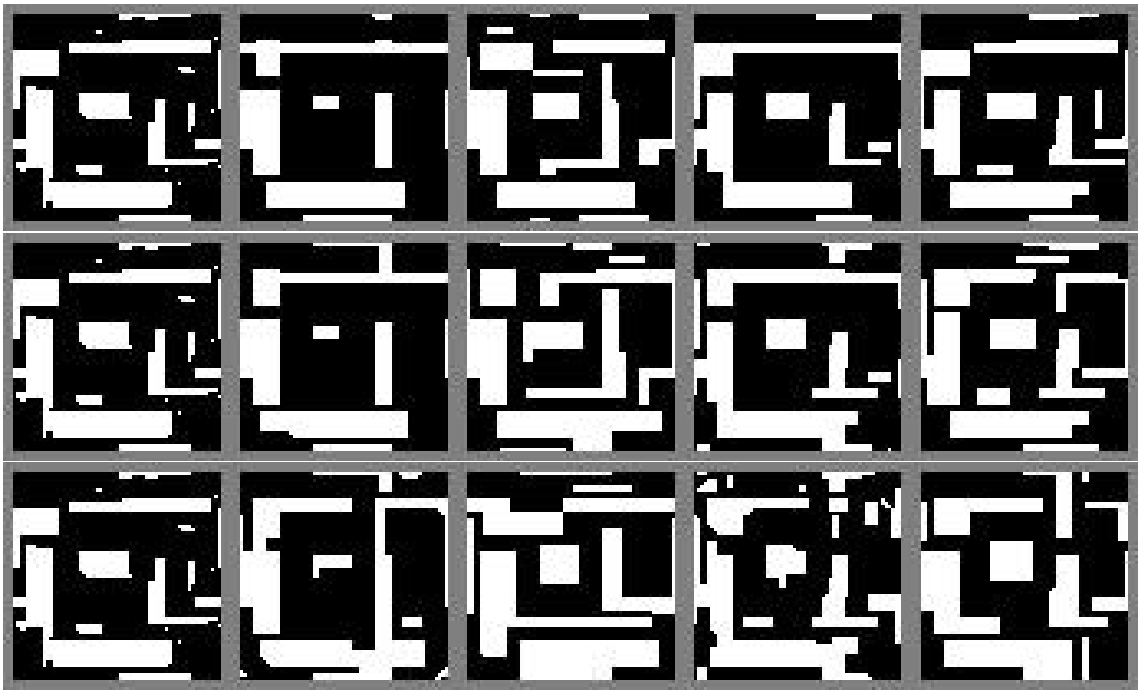

Figure 6.3: CA-MAP and SG-MAP estimators. From left to right in the top row are a phantom, its reconstructions using the Coordinate Ascent (CA) approach and the Semi Global (SG) approach from four projections, and then (in the same order; CA approach followed by SG approach) from eight projections.; all the reconstructions correspond to the noise level $N = 0.25$ and the number of misclassification are 538, 530, 295, and 183. The central and the bottom row, with the same arrangement as the top row, corresponds to, respectively, $N = 1.0$ and $N = 4.0$. The number of misclassification are 581, 720, 469, and 469 for the central row, and 718, 1030, 722, and 729 for the bottom row.

Table 6.2: Percentage of misclassification of the CA-MAP and the SG-MAP estimators (N is the noise level).

Method	$N = 0.25$	$N = 1.0$	$N = 4.0$
CA-MAP with 4 projections	11.5±4.0	12.1±4.0	15.2±4.4
SG-MAP with 4 projections	12.0±4.5	15.7±4.7	22.4±4.6
CA-MAP with 8 projections	6.5±2.4	8.5±2.7	16.0±3.1
SG-MAP with 8 projections	4.1±1.5	7.6±2.2	15.3±3.5

than the CA technique. In the CA approach the superiority (though with a P-value of about 0.015) of using four projections over that of eight projections in the noisiest case may be due to the fact that the performance of the CA approach, which is inherently local, depends strongly on the starting gray value image, which may not have been a good candidate for this case.

6.3.2 MM-MPM estimator

Table 6.3 reports on the quality of the the MM-MPM estimate (see Subsection 5.8.3) for the three noise levels, and Figure 6.4 shows the actual reconstructions of one phantom. Comparing Table 6.3 with Table 6.2, we find that the MM-MPM estimator is significantly (P-value less than 10^{-3}) better than the SG-MAP estimator in the case of four projections with $N = 1.0$ and $N = 4.0$ and eight projections with $N = 0.25$. Also, the performances of the MM-MPM and the CA-MAP estimators are similar, except for the case of eight projections with $N = 0.25$.

Table 6.3: Percentage of misclassification of the MM-MPM estimator (N is the noise level).

Method	$N = 0.25$	$N = 1.0$	$N = 4.0$
MM-MPM with 4 projections	10.7±4.1	12.1±3.8	15.7±3.7
MM-MPM with 8 projections	1.8±0.5	7.9±2.6	15.3±3.3

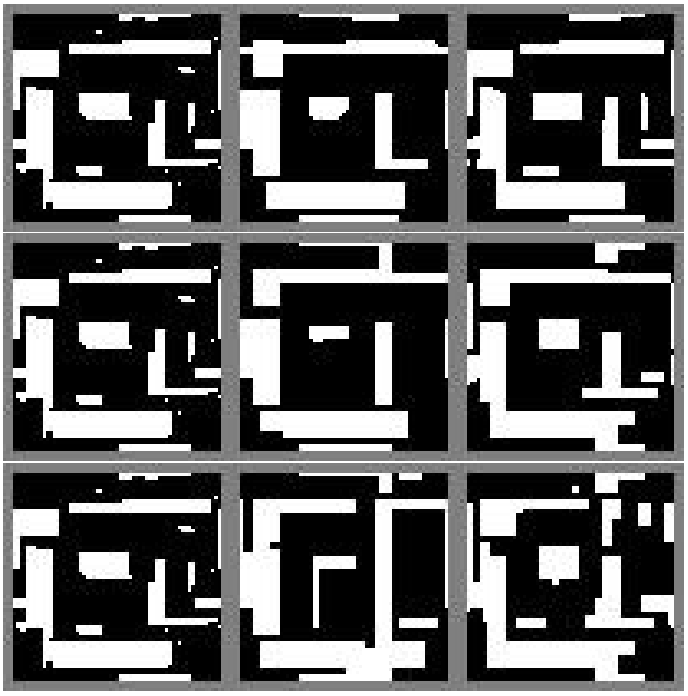

Figure 6.4: MM-MPM estimator. From left to right in the top row are a phantom, its MM-MPM estimate from four projections, and its MM-MPM estimate from eight projections.; the reconstructions correspond to the noise level $N = 0.25$ and the number of misclassification are 448 and 104. The central and the bottom row, with the same arrangement as the top row, corresponds to, respectively, $N = 1.0$ and $N = 4.0$. The number of misclassification are 581 and 470 for the central row, and 758 and 688 for the bottom row.

Table 6.4: Percentage of misclassification in the PL approximation (N is the noise level).

Method	$N = 0.25$	$N = 1.0$	$N = 4.0$
P-MAP with 4 projections	7.0±4.0	9.4±5.1	14.9±6.5
P-MPM with 4 projections	5.4±2.2	7.2±3.3	11.9±4.3
P-MAP with 8 projections	3.4±1.4	4.8±1.8	9.8±3.5
P-MPM with 8 projections	2.8±1.0	4.0±1.4	7.7±2.7

6.4 PL approximation

The details of the implementation of this approximation were discussed in Subsection 5.9, mainly by the formula (5.54) combined with (5.55). We compare the P-MAP and the P-MPM estimators.

Table 6.4 reports on the quality of the two estimators for the three noise levels. Here we confirm the superiority of the P-MPM estimator to the P-MAP estimator with P-values that are less than 10^{-4} for all the noise levels. Figure 6.5 shows the actual reconstructions. Compared to the MML approximation above, we note a significantly better performance of the PL approximation in almost all the six cases (four and eight projections, each combined with the three noise levels). The exception is eight projections with the lowest noise, in which the MM-MPM estimate appears to be the best. This can be explained by recalling that one of the assumptions of the PL (MML) approximation is that the noise should be relatively high (low), in the normality case (recall Sections 5.5.1 and 5.5.2).

6.5 A more realistic application

In this section we reconstruct binary images that represent a cross-section of a macromolecule in four conformations, by using the ICM method [7] (see Subsection 5.7.2), our five estimators CA-MAP, SG-MAP, MM-MPM, P-MAP, and P-MPM , as well as current

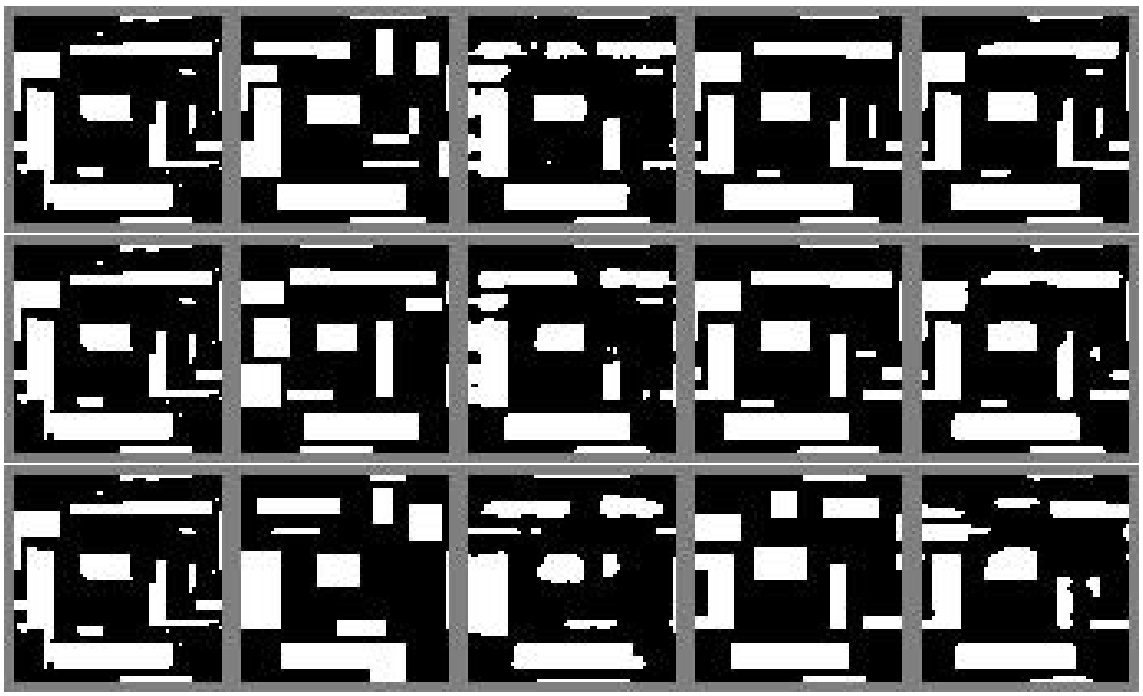

Figure 6.5: PL approximation. From left to right in the top row are a phantom, its P-MAP and P-MPM estimates from four projections, and the P-MAP and P-MPM estimators from eight projections; all the reconstructions correspond to the noise level $N = 0.25$ and the number of misclassification are 600, 391, 182, and 186. The central and the bottom row, with the same arrangement as the top row, corresponds to, respectively, $N = 1.0$ and $N = 4.0$. The number of misclassification are 628, 453, 251, and 214 for the central row and 893, 719, 482, and 480 for the bottom row.

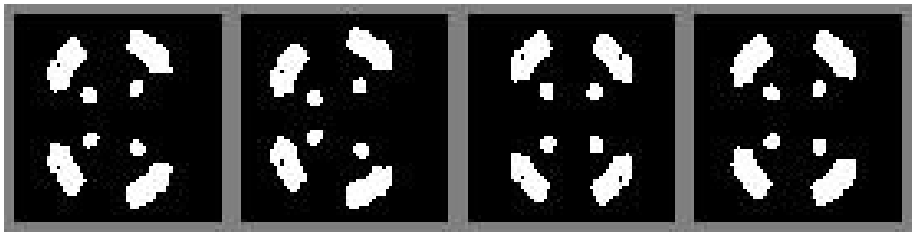

Figure 6.6: A cross section of a macromolecule in four conformations.

techniques of first reconstructing the gray value image and then segmenting it; i.e., the method of Section 2.2 and of Subsection 5.6.2. The latter assumes a perfect reconstruction of the gray value image and then segments it optimally, which is likely to yield a result that is the best we can expect from a current approach. All the reconstructions are based on eight projections with noise level $N = 1.0$.

Our approaches require the knowledge of the Gibbs distribution from which the unknown image is a typical sample (see the discussions in Subsection 3.2.1). To that end, for each of the four images, we estimate the Gibbs prior parameters based on our models (see Subsection 3.3.1), using the other three images. The estimation is carried out by the pseudo-likelihood method of Subsection 4.4.5.

The four conformations are depicted in Figure 6.6, Table 6.5 reports on the quality of

Table 6.5: Percentage of misclassification (N is the noise level).

Method	8 projections, $N = 1.0$
optimal thresholded gray value image	6.8±0.2
optimal thresholded gray value reconstruction	7.7±0.2
ICM	9.5±0.1
CA-MAP	4.5±0.1
SG-MAP	4.2±0.5
MM-MPM	5.9±0.3
P-MAP	4.1±0.3
P-MPM	3.5±0.2

the eight estimators, and Figure 6.7 shows the actual reconstructions of one phantom. We note that as far as the number of misclassification is concerned, current techniques perform worse than our methods. In particular, our P-MPM estimator, even though it is unable to fully recover an important feature in the original image, has the fewest misclassification. Consequently, a different indicator of the reconstruction quality, which takes into account this capability of recovering features should also be considered, and this will be done in the next chapter. We also note that in general MAP-type estimates are better than the MPM-type estimators at recovering small features, but the reconstructed structures are more blocky.

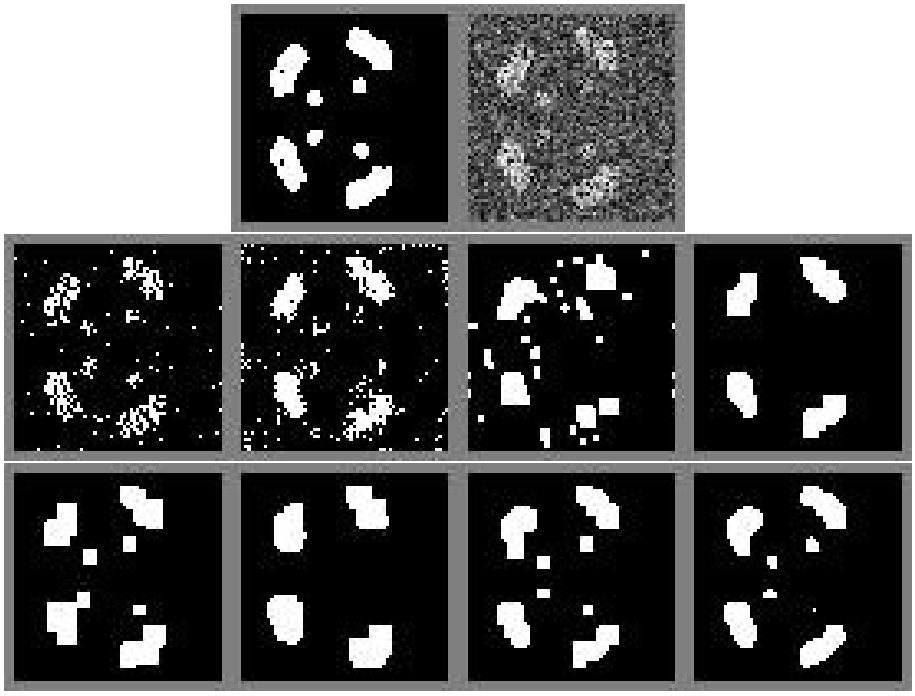

Figure 6.7: Reconstructions of an image representing the cross section of a macromolecule. From left to right, in the top row are one phantom and its gray value image. In the central row are an optimal thresholding of its gray value image (corresponding to an "ideal" reconstruction by current approaches), the reconstruction by the an ART-wth-pixel current approach, by the ICM method, and the CA-MAP estimate. The number of misclassification are respectively 272 and 319, 387, 181. In the bottom row are the reconstructions using the estimators SG-MAP, MM-MPM, P-MAP, and P-MPM. The number of misclassification are respectively 175, 247, 147, and 130.

Chapter 7

3D Image Reconstruction

We can in principle reconstruct a (three-dimensional) 3D image slice by slice using the techniques for (two-dimensional) 2D images illustrated in the previous chapters. In each slice we would use a Gibbs prior like the one described so far, which is defined on a 2D square grid. However, this strategy is not likely to be the best one, as the smoothness can only be imposed on each individual slice but not between the slices. In this chapter we develop fully 3D reconstruction techniques by establishing Gibbs distributions on 3D label images.

7.1 3D Gibbs distribution

7.1.1 The face-centered cubic grid

In Chapter 3 we discussed 2D Gibbs distributions (GD) defined on images whose domain is a subset of the square grid, which is the "standard" 2D grid. However, we need not to be restricted to this type of grid, since GDs can be defined on any grid. A natural extension of

the square grid to 3D is the usual cubic grid (2.3), whose associated voxels are cubes.

An important disadvantage of cubic voxels is that faces that meet at an edge are perpendicular to each other, which makes the graphic display of a surface appear blocky [41]. This effect is less pronounced in the face-centered cubic (FCC) grid (2.7), whose voxels are rhombic-dodecahedra. Also, a FCC grid point has fewer (12) face-or-edge neighbors (i.e., neighboring grid points with their corresponding voxels sharing a face or an edge) than a cubic grid point that has 18 face-or-edge neighbors, which implies less computational burden in the sampling of Gibbs distributions. Furthermore, these neighbors of a FCC grid point are more "evenly" distributed than those of a cubic grid point.

For 3D label images we define the domain D as subset of the FCC grid F_1

$$D = \left\{ v = (v_1, v_2, v_3) \mid v \in F_1 \text{ and } 0 \leq v_p < V_p \text{ for } p = 1,2,3 \right\}, \tag{7.1}$$

where the V_p are positive integers. (In this chapter D denotes the domain defined by (7.1), rather than by (3.1).)

As before (see Section 3.1), a label image x is a configuration over D, which for convenience also denotes a vector $\left(x_0, \cdots, x_{|D|-1}\right)^t$, such that $x_{\lfloor (v_3 V_1 V_2 + v_2 V_1 + v_1)/2 \rfloor} = x(v_1, v_2, v_3)$, for $(v_1, v_2, v_3) \in D$ (where for a real number α, $\lfloor \alpha \rfloor$ denotes the largest integer smaller than or equal to α).

7.1.2 Local features

We consider the set of cliques (see Sections 3.1 and 3.2)

$$Q = \left\{ q_{(v_1, v_2, v_3)} \mid (v_1, v_2, v_3) \in D \right\}, \tag{7.2}$$

where

$$q_{(v_1,v_2,v_3)} = \left\{ (v_1 \oplus_1 \delta_1, v_2 \oplus_2 \delta_2, v_3 \oplus_3 \delta_3) \mid \sum_{p=1}^{3} \delta_p \equiv 0 (\mathrm{mod} 2) \right\}, \tag{7.3}$$

$\delta_p \in \{-1,0,1\}$, and $\oplus_p$ denote the addition in $\mathbb{Z}_{V_p}$, for $p = 1,2,3$; i.e., addition modulo V_p.

To reduce the number of parameters of a GD, we use a T-equivalence relation on the set G of all possible configurations on Q containing the translation $\tau^1_{(v_1,v_2,v_3)}$ that maps $q_{(v_1,v_2,v_3)}$ onto $q_{(v_1 \oplus_1 1, v_2, v_3)}$. The T-equivalence relation also contains the translations $\tau^2_{(v_1,v_2,v_3)}$ and $\tau^3_{(v_1,v_2,v_3)}$ in the other two directions, which are defined in a similar way. (We do not need rotations or reflections because our description of the local features that follows, together with the three translations, is complete.) Consider, e.g., all the configurations on the clique $q_{(2,2,2)}$; see Figure 7.1. We are particularly interested in the following eight types of configurations (local features) on $q_{(2,2,2)}$.

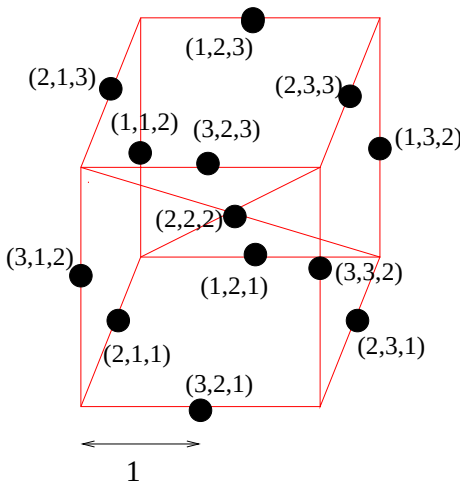

Figure 7.1: Clique $q_{(2,2,2)}$ in our 3D Gibbs Distribution model, composed of the grid point $(2,2,2)$ and its twelve neighboring grid points in the face-centered cubic grid.

- *black region*: All the points in $q_{(2,2,2)}$ are labeled 0.

- *white region*: All the points in $q_{(2,2,2)}$ are labeled 1.

(See Figure 7.2.)

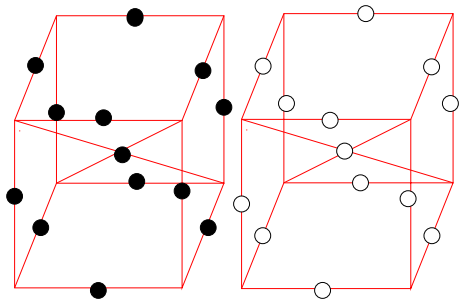

Figure 7.2: Local features named black region (left) and white region (right).

A configuration is a *wall* if for both labels there are at least three points with that label, and there are three points in $q_{(2,2,2)}$ with the following properties: all the points on the plane (called Π) that is determined by the three points have the same label, all the points in one open half-space determined by Π are labeled 1, and all the points in the other open half-space are labeled 0. Among walls, we distinguish the following two types.

- *Cartesian wall*: The normal to Π has direction parallel to one of $(1,0,0)$, $(0,1,0)$, or $(0,0,1)$.

- *regular wall*: All walls that are not a Cartesian wall.

(See Figure 7.3.)

We also introduce *convex corners* and *concave corner*. A concave corner is an "opposite feature" of a convex corner; i.e., the former can be obtained from the latter by switching the label (from 1 to 0 and from 0 to 1) at each point, and vice-versa. For example, the white region and the black region are opposite features; whereas the opposite feature of a wall is still a wall. There are two types of convex corners and concave corners.

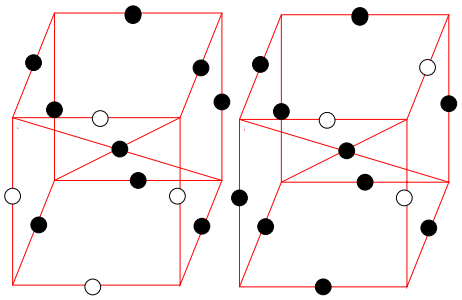

Figure 7.3: Examples of a Cartesian wall (left) and of a regular wall (right).

- *small convex corner*: All the points in $q_{(2,2,2)}$ are labeled 0, except for one or two points (different from the center $(2,2,2)$) that are labeled 1. In the latter case the two points must have two coordinates in common.

- *small concave corner*: is the opposite feature of a small convex corner.

- *large convex corner*: Not a wall, and all the points in $q_{(2,2,2)}$ are labeled 0, except for the center $(2,2,2)$ and at least three (but at most five) other points that are labeled 1. Consider all the half-lines, each one of which with origin at $(2,2,2)$ and passes through a point labeled 1. Then all the (grid) points inside the convex hull formed by the half-lines (which is a cone with vertex $(2,2,2)$) must be labeled 1.

- *large concave corner*: is the opposite feature of a large convex corner.

(See Figure 7.4.)

We will refer to GDs defined by these eight local features as *our 3D models*. The complete specification of one of our GD models requires the eight parameters (potentials) U_c ($1 \leq c \leq 8$) in the format $(U_1,\ldots,U_8)$, where U_1 corresponds to the black regions, U_2 the white regions, U_3 the Cartesian walls, U_4 the regular walls, U_5 the small convex corners, U_6 the small concave corners, U_7 the large convex corners, and U_8 the large concave corners.

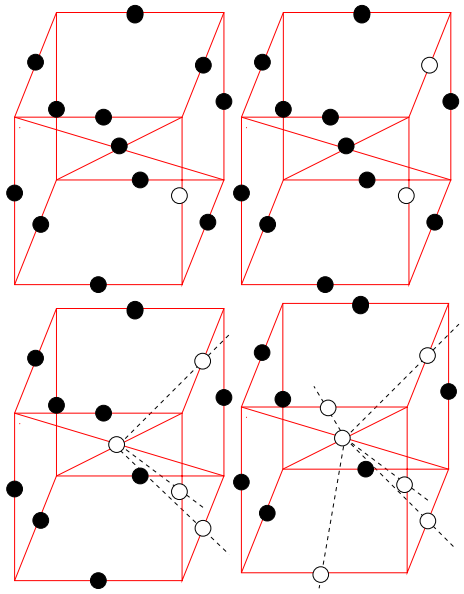

Figure 7.4: Examples of a small convex corner (top) and of a large convex corner (bottom).

7.2 Image modeling

Here we show sample images from Gibbs distributions defined by some of our 3D models. Specifically, we consider various choices for the eight potentials (parameters) corresponding to the eight types of configurations. A particular combination of the parameters determines the general shapes and sizes of the characteristic structures.

In the same manner as for 2D images, we employ the Metropolis algorithm (see Subsection 3.4.1) for the sampling and for the reconstruction. To improve efficiency, we make use of look-up tables for speeding up the sampling process, very similar to what is done in [93].

Figure 7.5 shows the cross-sections of typical 3D sample images from ten different GDs, whose parameters are reported in Table 7.1. The domain D is defined by $V_1 = V_2 = 64$ and $V_3 = 42$ in (7.1). For convenience, *domain* also refers to the set of rhombic-dodecahedra that are the associated voxels of all the FCC grid points in D, including those in the bound-

aries (a consequence of our "toroidal" cliques in (7.3)). A cross-section is obtained by cutting the image by a plane that passes through grid points and is parallel to one of the Cartesian directions. It is possible to select the parameters in such a way that the typical images of the distribution have relatively large uniform regions over a background, as occur in images of biological macromolecules. Surface renderings of four of the ten samples are depicted in Figure 7.6.

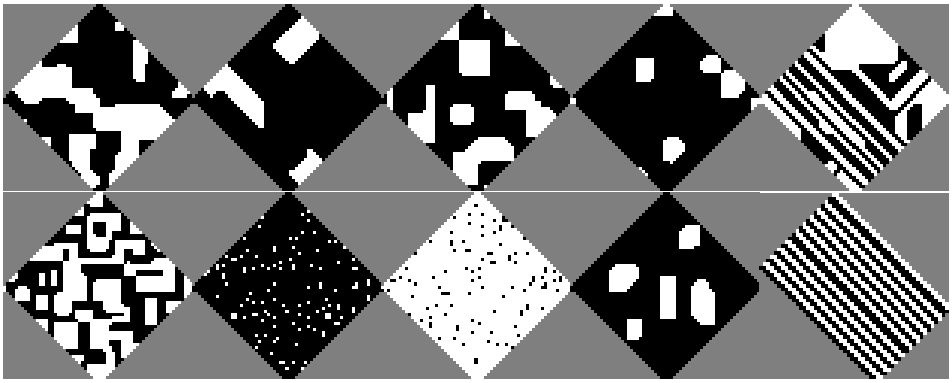

Figure 7.5: Cross-sections (the 30^{th}slice) of typical images corresponding to ten of our 3D GD models with the parameters reported in Table 7.1. The first nine cross-sections are normal to the direction 3, and the last one is normal to the direction 1.

sample	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8
1	1.2	1.2	1.2	1.2	0.52	0.5	0.52	0.5
2	-	-	-	1.0	-	-	-	-
3	-	-	1.1	-	-	-	0.5	0.48
4	-	-	-	-	0.6	-	0.6	-
5	-	-	1.4	-	-	-	-	-
6	-	-	-	1.4	-	-	-	-
7	-	-	-	-	1.2	-	-	-
8	-	-	-	-	-	1.2	-	-
9	-	-	-	-	-	-	1.2	-
10	-	-	2.0	-	-	-	-	-

Table 7.1: Parameters of the Gibbs distributions based on our GD models. Sample images are depicted in Figure 7.5 (from left to right, top to bottom). The "-" symbol indicates that the parameter value equals to the one in the same column for sample 1.

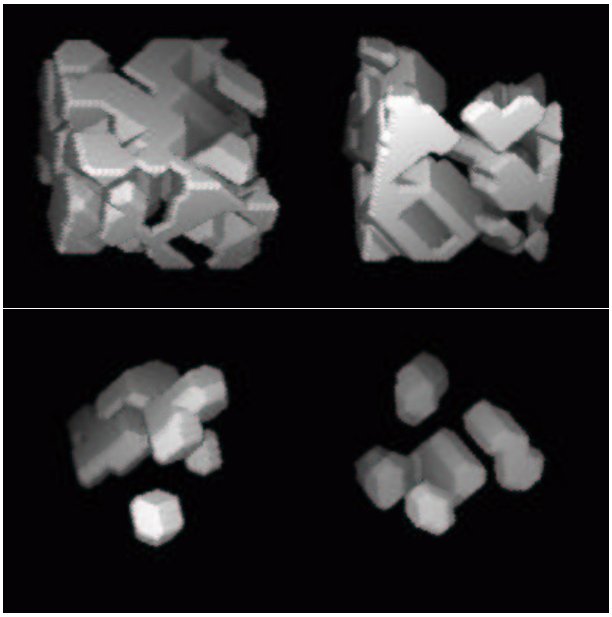

Figure 7.6: Surface rendering of the samples 1, 3, 4, and 9 (from left to right and top to bottom).

7.3 Image reconstruction methods

7.3.1 A current approach

To compare our approaches of reconstructing directly the label image against conventional approaches of first reconstructing the gray value image followed by a segmentation, we need to choose a “good” gray-value reconstruction algorithm that would then facilitate the segmentation process.

As stated in Section 2.2, based on the fact that ART has proved to outperform most known algorithms for various tasks [40, 50, 68], we choose ART combined with blobs on a BCC grid, whose parameters a, α, and Δ were determined in the same section. The relaxation parameter and the number of iterations were optimized based on some training

images and according to a *task* (to be defined in the next section).

As for the segmentation of the reconstructed gray value image, we adopt the technique discussed in Section 2.2.

7.3.2 Our approaches

Results from the experiments on 2D images suggest that the PL approximation gives better reconstruction quality than the MML approximation. Hence, here we only test the former, namely we compute the P-MAP estimate of (5.22) and the P-MPM estimate of (5.23), by means of a simulated annealing (see Subsection 5.7.3) and the Metropolis algorithm (see Subsection 3.4.1). Here we found that we can run fewer cycles for the samplings than in the 2D case. In particular, we ran 3,000 cycles to obtain one sample (in total 500 samples were taken for one P-MPM estimate). We observed that typical energy does not significantly change even if we increased the number of cycles to $5 \cdot 10^5$. We ran 10^4 cycles for each temperature of the simulated annealing for one P-MAP estimate. The annealing schedule was such that $1/T$ ranged from 0.1 to 1.25 with intervals of 0.05. The number of cycles and the annealing schedule were not optimized.

7.4 Experimental details

Unlike what we have done in the experiments on 2D images, here the (simulated) projection data are defined based on mathematically described images, consisting of spheres of various sizes. We estimated the label images from the projection data, using the conventional approach and our approaches, as explained in Section 7.3.

In the experiments with 2D images we have used the average percentage of misclas-

sification as an indicator of the reconstruction quality. In fact, this percentage δ (or more correctly, $100-\delta$) can be regarded to as a *figure of merit* (FOM) in the context of statistical-hypothesis-testing based methodology [29, 44] for the evaluation of the relative efficacy of two reconstruction methods. A FOM measures how helpful a reconstructed image is for solving a specific problem (or achieving a task) in the application area.

In the experiments in this chapter we are interested in the appearance of the reconstructions and the detectability of small structures. To that end, for each reconstruction we use two measures: one is δ and the other is 100 times the area under a *receiver operating characteristic* (ROC) curve [73, 97] (details on the latter follow below).

7.4.1 Phantoms

For the phantoms, we considered 3D two-label images with domain size such that $V_1 = V_2 = 64$ and $V_3 = 42$ (so that $|D| = 86,016$), and the set of labels $X =$ {ice, protein}. We used black (0) to represent ice and white (1) to represent protein. There were ten phantoms representing biological macromolecules. Each phantom (ten in total) is a discretization of nine white spheres arranged symmetrically in a pyramidal structure on a black background. Near the surface of the top (single) and bottom (four) spheres there are four possible locations for a little sphere that can be either present or absent with probability 0.5 (so the total number of distinct phantoms is $2^{5\times 4}$, but we randomly selected ten of them for the experiments). Figure 7.7 shows the case when all the little spheres are present, and Table I.2 of Appendix I reports on the exact locations and the sizes of the spheres. The discretization was done by labeling 1 a grid point (in D) if it is inside or on a sphere and 0 otherwise. The purpose of introducing the little spheres is to evaluate the detectability of small structures.

We assumed that the phantoms were samples from one of our GD models; therefore we

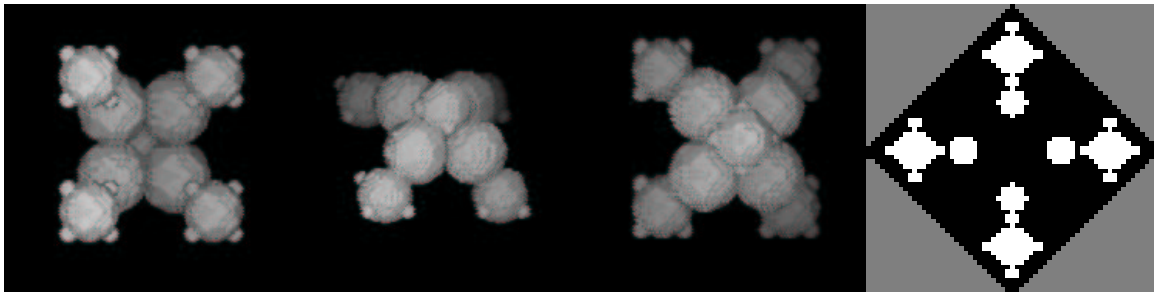

Figure 7.7: Surface renderings of a phantom and one of its cross-sections.

needed to estimate its eight parameters. To that end, we randomly selected 100 (different from the previous ten) training "phantoms" and used the pseudo-likelihood method [6] (see Subsection 4.4.5); obtaining (1.34, 1.51, 0.57, 0.65, 0.80, 0.08, 0.34), with accuracy up to two decimals. The number of training images turned out to be more than sufficient, since the estimates do not differ significantly from those based on 1,000 or even on 10 training images.

7.4.2 Projection data

Very similar to the models we used in the experiments on 2D images, here we also assumed normality in the algorithms. In particular, $\varphi(y_i|x_i)$ $(1 \leq i \leq I)$ was assumed to be normally distributed with mean μ_{x_i} and variance $\sigma^2_{x_i}$, where $\mu_{x_i} = \sigma^2_{x_i}$, $\mu_0 = 4$, and $\mu_1 = 9$. The model for $\chi(w|y)$ should become clear as we explain how we simulated the projections forming the measurement vector $w = (w_0, \cdots, w_{J-1})^t$. In each projection we took parallel lines that pass through all the grid points in the domain D, and the lengths of intersection with a rhombic-dodecahedron voxel have all the same value. Nine projections with such a property are in the direction $(0,1,1)$, $(0,-1,1)$, $(1,0,1)$,$(-1,0,1)$, $(1,1,0)$, $(-1,1,0)$, $(1,0,0)$,$(0,1,0)$, and $(0,0,1)$; see Figure 7.8. It is easy to see that the lengths of intersection for the first six projections have all value $\sqrt{2}$, and they have value 2 for the last three projections Neither

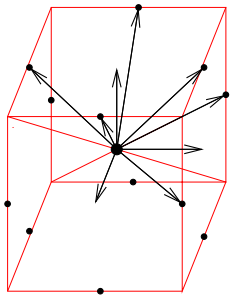

Figure 7.8: Nine direction projections; any line along one of the directions and passes through a grid point will intersect the voxels with equal length.

the condition of passing through all the grid points, nor that the intersection length must be equal in one projection is necessary for the validity of the our approaches, but they together simplify the implementation. For the jth line ($j = 0, \cdots, J-1$) we computed the line integral z_j of the gray values inside the mathematically described images as follows. Let l_0 and l_1 be respectively the lengths of intersection of the line with the spheres and with the background. (We note that, as a consequence of our "toroidal" cliques in (7.3), $l_0 + l_1$ is the number of intersected grid points multiplied by a factor 2 or $\sqrt{2}$, depending on the direction of the projection.) Then z_j was assumed to come from a normal distribution with mean and variance equal to $\mu_0 l_0 + \mu_1 l_1$, where $\mu_0 = 4$ and $\mu_1 = 9$. To avoid negativities, we set z_j to 0 whenever it was sampled negative (with probability less than 0.028 in only a few measurements). There is no particular reason for choosing these values of mean and variance, except for the fact that they clearly reflect the idea of overlapping gray values in higher resolution electron microscopy, as shown by the left histogram in Figure 2.3. Our $\upsilon_j(w_j|y)$ in the algorithms was considered to be normally distributed with mean z_j, and we initially intended to set the variance to be $\sigma^2_{wj} = N \cdot z_j$ (recall that N is the noise level, whose possible values are chosen to be $0.01, 1.0,$ or 4.0). To avoid non-positive variance, we replaced σ^2_{wj} by $N \cdot \widetilde{w}_j = N \cdot \max\left(\mu_0, w_j\right)$, where w_j is sampled from $\upsilon_j(w_j|y)$. Finally,

the simulated measurement was taken to be $\widetilde{w}_j$, as we have done with 2D images (see Section 6.1 for justifications).

The sequence of projections given in the previous paragraph is the one used in the ART algorithms, and the sequence of lines in one projection were such that no blob intersects two consecutive lines. Specifically, they were picked row by row: every eleventh within one row and every eleventh row (both with wrapping around)−the number 11 is co-prime with 64 and 42.

7.4.3 Detectability by the area under a ROC curve

In quantifying the detectability of small structures, we evaluate the reconstructed labels at the voxels (each of which called a *target*) forming the 20 potential locations for a little sphere. Each location comprises of either 16 (for the top little spheres) or 19 (for the bottom little spheres) voxels. In a decent reconstruction it is expected that a target has most of the times the same label as that in the phantom. One can thus create a function that indicates the "signalness" based on target labels in each potential location. A high (or low) signalness corresponds to the presence (or absence) of a little sphere. It is reasonable to set such a function to be the sum of the reconstructed target labels (so it is exactly the number of targets labeled 1). If one selects a particular threshold and judges a location to contain a little sphere if, and only if, the function value is above the threshold, then one will get the number of *true positives* (locations correctly judged to contain a little sphere) and the number of *false positives* (locations judged to contain a little sphere when indeed there is not in the phantom).

A plot of the fraction (over the total number of locations) of true positives versus the fraction of false negatives at various threshold values is known as the ROC curve [97]. It

tells us how well a reconstruction algorithm detects small structures. Due to the discrete nature, there are only a finite number of possible fractions, which means that we will only get some sample points of the curve, which is then formed by linearly interpolating these points. Typically, for comparison of two ROC curves, the area under the curve is used.

7.5 Experimental Results

In Table 7.2 we report on the quality of the reconstructions using nine projections by the current ART-based approach, the P-MPM estimator, and the P-MAP estimator. Figure 7.9 shows the actual reconstructions of one phantom in one cross-section. The significance of the differences in reconstruction quality between the two estimators is measured using the pairwise *t*-test. As far as δ (percentage of misclassification) is concerned, in the case of $N = 0.01$, the P-MPM estimate is significantly (*P*-value less than 0.001) the best, followed by the P-MAP estimate, and then the ART estimate. For $N = 1.0$ the P-MPM estimate is still significantly the best, while the comparison between the other two estimates is not significant. For $N = 4.0$, the ART estimate is significantly the best, followed by the P-MPM estimate, and then the P-MAP estimate.

According to the detectability measure (by the area under a ROC curve), for $N = 0.01$, the differences among the three estimates are not significant, and neither are the differences between the P-MPM and the P-MAP estimates at the three noise levels. For $N = 1.0$ and $N = 4.0$, ART estimate is significantly the best.

Above we report on the results using nine projections. We also evaluated the three estimators based on six projections (the first six from the list of nine projections in Subsection 7.4.2) and found that (see Table 7.3 and Figure 7.10), in terms of percentage of misclassification δ, ART estimate is no longer significantly better than the P-MPM estimate for

Table 7.2: Quality of reconstruction using nine projections, according to δ (percentage of misclassification averaged over the ten testing phantoms; left table) and the detectability measure (100×area under a ROC curve; right table). N is the noise level.

Method	$N = 0.01$	$N = 1.0$	$N = 4.0$	$N = 0.01$	$N = 1.0$	$N = 4.0$
ART	1.36±0.04	1.61±0.05	2.15±0.08	98±4	95±5	95±4
P-MPM	0.97±0.04	1.35±0.07	2.33±0.06	98±3	85±7	73±9
P-MAP	1.19±0.05	1.64±0.08	2.62±0.09	97±4	84±8	68±6

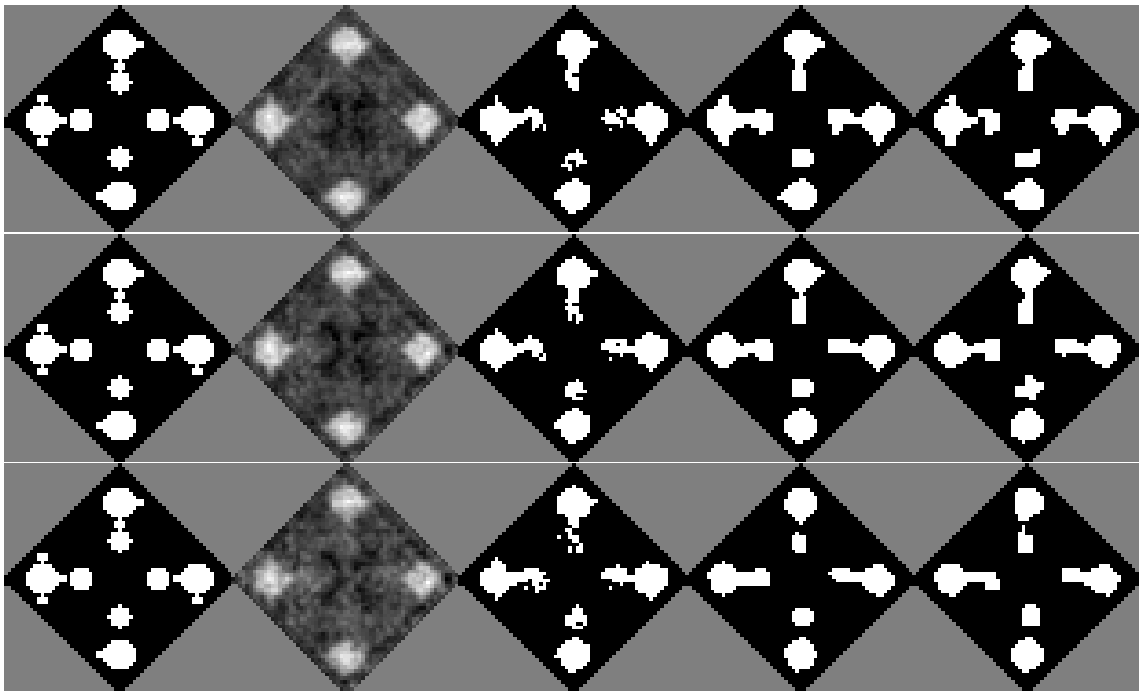

Figure 7.9: Reconstructions of a 3D phantom (one cross-section). From left to right in the top row are a phantom, its reconstruction using ART, an optimal segmentation of the ART reconstruction, the P-MPM and P-MAP estimators; all the reconstructions correspond to the noise level $N = 0.01$ and the number misclassification (in the 3D object) are 1,231, 888, and 1,083. The central row, with the same arrangement, corresponds to $N = 1.0$ with 1,435, 1,225, and 2,339 misclassification. The bottom row, also with the same arrangement, corresponds to $N = 4.0$ with 1,922, 2,069, and 2,339 misclassifications. The size of a phantom is 86,016.

Table 7.3: Quality of reconstruction using six projections, according to δ (percentage of misclassification averaged over the ten testing phantoms; left table) and the detectability measure ($100\times$area under a ROC curve; right table). N is the noise level.

Method	$N = 0.01$	$N = 1.0$	$N = 4.0$	$N = 0.01$	$N = 1.0$	$N = 4.0$
ART	1.36 ± 0.04	1.61 ± 0.05	2.15 ± 0.08	98 ± 4	95 ± 5	93 ± 7
P-MPM	1.29 ± 0.07	1.94 ± 0.07	3.25 ± 0.05	90 ± 8	85 ± 4	67 ± 9
P-MAP	1.54 ± 0.07	2.24 ± 0.07	3.54 ± 0.06	90 ± 7	81 ± 8	60 ± 8

$N = 4.0$. Furthermore, for each of the other two noise levels we note that the difference of the δs of the two estimates has increased, as compared to the case with nine projections. These observations are an indication of the higher effectiveness of our direct labeling P-MPM approach relatively to the ART approach, as the number of projections decreases. However, as far as detectability is concerned, ART is significantly the best for all the three noise levels.

Compared to the experiments on 2D images, the not-so-favorable results by the new approach on 3D images, under the detectability measure may be explained by the fact that the cliques of our 3D (Gibbs prior) models are relatively smaller (when looking at a cross-section) than those of our (2D) models. As pointed out in [14, 59], the reconstruction quality using small neighborhoods in the Gibbs prior can be poor compared to that when using larger ones.

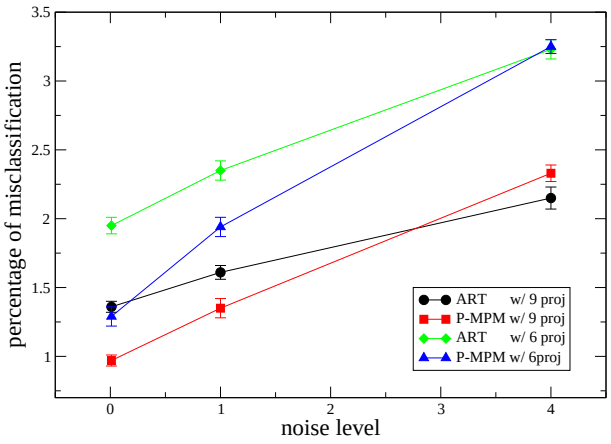

Figure 7.10: Averaged percentage of misclassification δ versus noise level for the ART and the P-MPM estimators. At each noise level, the difference of the δs of the two estimators is larger when a lower number of projections is used.

Chapter 8

Conclusions

8.1 Summary and Contributions

Reconstruction of label images from only a few projections has great usefulness and potentials in the areas of angiography, non-destructive material testing, cardiac imaging, etc. Our work is primarily motivated by the electron microscopy of biological macromolecules, in which the density (gray value) in a voxel corresponding to one type of matter is not uniquely determined by that matter.

We have proposed five techniques that *directly* produce, based on a few noisy projections, a labeling of the space tessellated into pixels (in 2D) or rhombic-dodecahedra (in 3D) voxels. Because this type of problem is very "under-determined" (low number of projections versus lots of voxels), we conjectured that, as opposed to our methods, current approaches of first reconstructing the density distribution followed by a segmentation would not be reliable.

Our approaches are based on a (Bayesian) statistical model, in which the prior probability is in the form of a Gibbs distribution. We have adopted and evaluated several methods

for estimating the parameters of the Gibbs prior based on images that are typical in the application area, Such images are given higher probability according to the prior.

We have created 2D and 3D Gibbs prior models that have, respectively, five and eight parameters. By varying the parameters, we were able to generate a variety of distributions, each of which with typical images that have characteristic structures with particular shapes and sizes. Such diversity should useful in many application areas.

We have implemented and studied two Bayesian estimators for the purpose of label reconstruction: the MAP estimator and the MPM estimator. Direct optimization of the objective function by either the MAP or the MPM criterion is infeasible. Therefore, we have made two approximations to the likelihood function: the pseudo likelihood (which is justified when data noise is high) and the mean-by-the-mode likelihood (which makes sense when noise is low). We have developed five reconstruction techniques based on the two estimators combined with the two approximations.

Our experiments on 2D images indicate that our proposed techniques outperforms an ART-with-pixel-based current approach and even an "ideal" current approach (in which the gray value image is perfectly reconstructed), in terms of the percentage of misclassification. Among our techniques, a MPM-type estimator combined with the pseudo likelihood approximation (i.e., the P-MPM estimator) proved to be the best in general, under the measure.

In our experiments on 3D images, we introduced another measure for the detectability of small structures, which is defined by the area under a ROC curve. The result indicate that although the P-MPM estimator is superior to an ART-with-blob based current approach in terms of the percentage of misclassification (except at the highest noise level, for which the two approaches have similar performance, based on six projections), it is not so under the detectability measure.

The main differences between the two sets of experiments are: (i) ART with pixel basis was used in 2D and ART with blob basis in 3D, and (ii) the nature of the Gibbs prior. Our 3D reconstructions may be improved by "enhancing" the Gibbs prior model. For example, the cliques of our 3D (Gibbs prior) models are relatively smaller (when looking at a cross-section) than those of our (2D) models, and reconstruction quality using small cliques can be poor compared to that when using larger ones. In regards to the enhancement of Gibbs prior model and improvement of the reconstruction, we can address in addition some interesting future works that are summarized below.

8.2 Future works

In estimating the Gibbs distribution from which typical images of an application area are assumed to be samples, we independently created a model and then estimated the parameters of the Gibbs prior from the images. An interesting open problem is to determine also the model from the images. This means choosing an appropriate set of cliques and a partition on the set of their configurations, in such a way that efficient reconstruction algorithms can be implemented (e.g., the strategy of using look-up tables requires that size of the cliques cannot be too large).

A disadvantage of the MAP estimator is over-smoothing. A weakness of the MPM criterion is the lack of distinction between "scattered" and "aggregated" misclassification: a modest number of misclassification are rather harmless if they are scattered, as they would be interpreted as isolated misclassification, which is not the case if they aggregate into some artifact. Therefore, it is important to develop and study new estimators that do not suffer from such undesirable properties.

A Gibbs distribution can be viewed as an undirected graphical model, in which the

voxels are the nodes and there is an edge between two voxels if, and only if, they belong to a same subregion in the image. Because the current proposed algorithms are Monte Carlo based, even with the use of look-up tables to speed up the reconstruction process, they are still slower than some deterministic algorithms that are derived from a directed graphical model (e.g., the forward-backward propagation algorithm for training a neural network). Even though directionality seems counter-intuitive in tomographic applications, it might be still worthwhile to explore its usefulness from the algorithmic efficiency point of view.

Appendix A

Ising models

For an Ising model as defined in [94], the energy (expressed using the notations and conventions of this thesis) is defined by

$$\begin{aligned} H(x) &= -U_1^{Ising} \Sigma_{\{(v_1,v_2)\} \in Q^s} [2x(v_1,v_2)-1] \\ &- U_2^{Ising} \Sigma_{\{(v_1,v_2),(v_1',v_2')\} \in Q^p} [2x(v_1,v_2)-1][2x(v_1',v_2')-1], \end{aligned} \tag{A.1}$$

where $U_1^{Ising} = \frac{J}{kT}$ and $U_2^{Ising} = \frac{mB}{kT}$. (For the discussions that follow, the precise meanings of the physical constants J, k, T, m, and B are irrelevant.)

Noting that

$$\sum_{\{(v_1,v_2),(v_1',v_2')\} \in Q^p} x(v_1,v_2) = \sum_{\{(v_1,v_2),(v_1',v_2')\} \in Q^p} x(v_1',v_2') \tag{A.2}$$

$$= 2 \sum_{\{(v_1,v_2)\} \in Q^s} x(v_1,v_2) \tag{A.3}$$

and observing (a trivial consequence of (3.3)) that if the difference between two energies $H(x)$ and $H'(x)$ is a constant, then the corresponding Gibbs distributions are the same, we

get that the Gibbs distribution whose energy is $H(x)$ is identical to the Gibbs distribution whose energy is

$$\begin{aligned} H'(x) &= -(2U_1^{Ising} - 8U_2^{Ising}) \sum_{\{(v_1,v_2)\}} \in Q^s x(v_1,v_2) \\ &- 4U_2^{Ising} \sum_{\{(v_1,v_2),(v_1',v_2')\}} \in Q^p x(v_1,v_2)x(v_1',v_2'). \end{aligned} \tag{A.4}$$

Using the notion introduced in Section 3.2, we see that the term $\sum_{\{(v_1,v_2)\}} \in Q^s x(v_1,v_2)$ in (A.4) is $N(G_1,\{x\})$ and the term $\sum_{\{(v_1,v_2),(v_1',v_2')\}} \in Q^p x(v_1,v_2)x(v_1',v_2')$ in (A.4) is $N(G_2,\{x\})$. Hence, by (3.6), our definition of an Ising model will give rise to the same Gibbs distribution as defined by (A.1), provided only that $U_1 = 2U_1^{Ising} - 8U_2^{Ising}$ and $U_2 = 4U_2^{Ising}$. Clearly, the converse is also the case: any Gibbs distribution defined using our definition of an Ising model can also be obtained by the appropriate selection of U_1^{Ising} and U_2^{Ising} in (A.1).

Appendix B

MCMCML method

Recall from Subsection 4.3.2 that the MCMCML method aims at finding a parameter vector $\widehat{\mathbf{U}} = (\widehat{U}_1, ..., \widehat{U}_C)^t$ such that $\langle N_c(\{x\})\rangle_{\widehat{\mathbf{U}}}$ equals to $N(G_c, X_{tr}) / |X_{tr}|$, by gradually adjusting the parameters until this condition is met, using a gradient search algorithm (which, according to [20], is the conjugate gradient method) and Markov chain Monte Carlo (MCMC) methods.

B.1 The conjugate gradients method

This method [65] allows us to find the local minimum of an objective function $\vartheta(\mathbf{U})$. It is iterative and greedy (and therefore only local optimum is guaranteed). Starting from a $\mathbf{U}^{(0)}$, in each iterative step, it minimizes locally at the current $\mathbf{U}^*$ a quadratic function $\vartheta_2(\mathbf{U})$ that is the Taylor expansion of $\vartheta(\mathbf{U})$ up to the second order

$$\begin{aligned} \vartheta_2(\mathbf{U}^* + \mathbf{U}) &= \vartheta(\mathbf{U}^*) + \sum_{c=1}^{C} \frac{\partial \vartheta}{\partial U_c}(\mathbf{U}^*) U_c + \frac{1}{2} \sum_{1 \le c,c' \le C} \frac{\partial^2 \vartheta}{\partial U_c \partial U_{c'}}(\mathbf{U}^*) U_c U_{c'} \\ &= \vartheta(\mathbf{U}^*) + \mathbf{U}^t \nabla \vartheta(\mathbf{U}^*) + \frac{1}{2} \mathbf{U}^t \mathcal{H} \mathbf{U}, \end{aligned} \tag{B.1}$$

where, for a vector $\mathbf{U}$, $\mathbf{U}^t$ denotes its transpose, $\nabla\vartheta(\mathbf{U}^*)$ is the gradient of ϑ at $\mathbf{U}^*$ and the matrix $\mathcal{H}$ (whose entries are the second order partial derivatives) is the *Hessian* matrix of ϑ at $\mathbf{U}^*$. A sequence of vectors $\mathbf{U}^{(1)},...,\mathbf{U}^{(C)}$, such that $(\mathbf{U}^c)^t\,\mathcal{H}\mathbf{U}^{c'}=0$, for $1\leq c\neq c'\leq C$, is said to be a sequence of *conjugate directions* with respect to $\mathcal{H}$. Instead of the local gradient for going downhill, the algorithm uses the conjugates directions. The algorithm (see [65] and [76]) proceeds as follows. Let $p^{(0)}=-\nabla\vartheta\left(\mathbf{U}^{(0)}\right)$. For $m\geq 0$, assuming that $p^{(m)}$ is known, the algorithm produces

$$o^{(m)} = -\nabla\vartheta\left(\mathbf{U}^{(m)}\right), \tag{B.2}$$

$$\mathbf{U}^{(m+1)} = \mathbf{U}^{(m)}+\frac{\left(o^{(m)}\right)^t p^{(m)}}{\left(p^{(m)}\right)^t\mathcal{H}p^{(m)}}p^{(m)}, \tag{B.3}$$

$$p^{(m+1)} = o^{(m+1)}+\frac{\left(o^{(m+1)}\right)^t o^{(m+1)}}{\left(o^{(m)}\right)^t o^{(m)}}p^{(m)}. \tag{B.4}$$

The algorithm converges in C steps if $\vartheta(\mathbf{U})\equiv\vartheta_2(\mathbf{U})$. It is easy to verify [76] that the following important property holds. Assuming that $\vartheta(\mathbf{U})\equiv\vartheta_2(\mathbf{U})$, then $\mathbf{U}^{(m+1)}$ in (B.3) is the minimizer of ϑ along the line in the direction $p^{(m)}$, which passes through $\mathbf{U}^{(m)}$. This means that the algorithm can run without ever having to compute the Hessian matrix $\mathcal{H}$, as long as we define $\mathbf{U}^{(m+1)}$ as having this property. A desirable property of the conjugate gradient method is that if the vicinity of the minimum has the shape of a long and narrow valley, the minimum is reached in much fewer steps than would be the case using another gradient search algorithm known as the method of *steepest descent* [65], in which the (line) search for the minimum is simply along the direction of the negative gradient.

Our objective function is the negative of the log-likelihood. Below we give its formula along with the formula of its gradient, both needed in the conjugate gradient algorithm.

B.2 Maximum likelihood

For a given parameter vector $\mathbf{U} = (U_1, \ldots, U_C)^t$ the likelihood of an image x is (using (3.3) and (3.6))

$$\pi_{\mathbf{U}}(x) = \frac{1}{Z_{\mathbf{U}}} \exp\left[\sum_{c=1}^{C} N_c(\{x\}) U_c\right] = \frac{1}{Z_{\mathbf{U}}} \exp\left[\mathbf{N}^t(x)\mathbf{U}\right], \tag{B.5}$$

where the subindex $\mathbf{U}$ in $\pi_{\mathbf{U}}(x)$ is to emphasize the dependency on $\mathbf{U}$, $\mathbf{N}^t(x) = (N_1(\{x\}), \ldots, N_C(\{x\}))$, and $Z_{\mathbf{U}}$ is the partition function

$$Z_{\mathbf{U}} = \sum_{x \in \Lambda^D} \exp\left[\mathbf{N}^t(x)\mathbf{U}\right] \tag{B.6}$$

The maximum likelihood (ML) estimator $\widehat{\mathbf{U}}$ is obtained by maximizing the log-likelihood (divided by the size of the training set X_{tr})

$$L_{\mathbf{U}}(X_{tr}) = \frac{1}{|X_{tr}|} \sum_{x \in X_{tr}} \log \pi_{\mathbf{U}}(x) = \overline{\mathbf{N}}^t(X_{tr})\mathbf{U} - \log Z_{\mathbf{U}}, \tag{B.7}$$

where $\overline{\mathbf{N}}^t(X_{tr}) = \frac{1}{|X_{tr}|} \sum_{x \in X_{tr}} \mathbf{N}^t(x)$, with respect to $\mathbf{U}$. Taking the partial derivatives of $L_{\mathbf{U}}(X_{tr})$ with respect to each component of $\mathbf{U}$, we have that, for $c = 1, \ldots, C$,

$$\frac{\partial [L_{\mathbf{U}}(X_{tr})]}{\partial U_c}(\mathbf{U}) = \overline{N}_c(X_{tr}) - \sum_{x \in \Lambda^D} N_c(\{x\}) \frac{\exp[\mathbf{N}^t(x)\mathbf{U}]}{Z_{\mathbf{U}}}, \tag{B.8}$$

where the ratio on the right-hand side is precisely $\pi_{\mathbf{U}}(x)$, which means that the second term is the expected value of $N_c(\{x\})$ with respect to $\pi_{\mathbf{U}}$; i. e.,

$$\frac{\partial [L_{\mathbf{U}}(X_{tr})]}{\partial U_c}(\mathbf{U}) = \overline{N}_c(X_{tr}) - \langle N_c(\{x\}) \rangle_{\mathbf{U}}. \tag{B.9}$$

At $\widehat{\mathbf{U}}$, the log-likelihood achieves an extreme, so its gradient must be zero; i.e.,

$$\langle N_c(\{x\})\rangle_{\widehat{\mathbf{U}}} = \overline{N}_c(X_{tr}). \tag{B.10}$$

In other words, the ML estimator is such that the expected value of $N_c(\{x\})$ coincides with that of the training set.

B.3 Importance sampling

Although any characteristic with respect to $\pi_{\mathbf{U}}$ (for a given $\mathbf{U}$) can be estimated using a MCMC method, sampling $\pi_{\mathbf{U}}$ for every needed value of $\mathbf{U}$ becomes infeasible in terms of CPU time. Nonetheless, based on a standard technique for variance reduction in Monte Carlo methods that is known as *importance sampling*, one can define, for a given $\widetilde{\mathbf{U}}$, a neighborhood of $\widetilde{\mathbf{U}}$ inside which samples from $\pi_{\widetilde{\mathbf{U}}}$ can be reused without having to re-sample. To that end, a characteristic with respect to $\pi_{\mathbf{U}}$ must be expressed in terms of some characteristics (could be more than one) with respect to $\pi_{\widetilde{\mathbf{U}}}$, whose samples are already available.

Before giving the formula for the objective function, we derive an alternative expression for the gradient of the log-likelihood that permits us to implement importance sampling. In (B.14) we show that knowing the gradient of the log-likelihood as a function of $\mathbf{U}$ is essentially the same as knowing $\langle N_c(\{x\})\rangle_{\mathbf{U}}$. (The other term $\overline{N}_c(X_{tr})$ is computed once from the training set and is fixed.) This expected value is, by definition,

$$\langle N_c(\{x\})\rangle_{\mathbf{U}} = \sum_{x\in\Lambda^D} N_c(\{x\})\,\frac{\exp\left[\mathbf{N}^t(x)\mathbf{U}\right]}{\sum_{x'\in\Lambda^D}\exp\left[\mathbf{N}^t(x')\mathbf{U}\right]}, \tag{B.11}$$

where the ratio is $\pi_{\mathbf{U}}(x)$ and its denominator (independent of x) is the expanded normalizing factor $Z_{\mathbf{U}}$. By replacing the potential $U_{c'}$ by $U_{c'} - \widetilde{U}_{c'} + \widetilde{U}_{c'}$ in both the numerator and denominator and rearranging terms, we get that

$$\langle N_c(\{x\})\rangle_{\mathbf{U}} = \frac{\sum_{x\in\Lambda^D} N_c(\{x\}) \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right] \exp\left[\mathbf{N}^t(x)\widetilde{\mathbf{U}}\right]}{\sum_{x'\in\Lambda^D} \exp\left[\mathbf{N}^t(x')\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right] \exp\left[\mathbf{N}^t(x')\widetilde{\mathbf{U}}\right]}. \tag{B.12}$$

Now, by dividing both the numerator and the denominator by $Z_{\widetilde{\mathbf{U}}}$ and noting (B.5), we conclude that the numerator and the denominator are, respectively, the expectations $\left\langle N_c(\{x\}) \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}$ and $\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}$; i.e.,

$$\langle N_c(\{x\})\rangle_{\mathbf{U}} = \frac{\left\langle N_c(\{x\}) \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}}{\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}}. \tag{B.13}$$

Hence, the partial derivatives of the log-likelihood are (using the result in (B.13) for $\langle N_c(\{x\})\rangle_{\mathbf{U}}$ in (B.14)):

$$\frac{\partial\left[L_{\mathbf{U}}(X_{tr})\right]}{\partial U_c}(\mathbf{U}) = \overline{N}_c(X_{tr}) - \langle N_c(\{x\})\rangle_{\mathbf{U}}. \tag{B.14}$$

$$\frac{\partial\left[L_{\mathbf{U}}(X_{tr})\right]}{\partial U_c}(\mathbf{U}) = \overline{N}_c(X_{tr}) - \frac{\left\langle N_c(\{x\}) \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}}{\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}}, \tag{B.15}$$

for $c = 1,\ldots,C$. At this point we have two expressions for evaluating the gradient of the log-likelihood, namely (B.14) and (B.15). We show later that in implementations with MCMC methods the latter should be avoided.

We now derive the objective function to be minimized. Maximizing the log-likelihood

in (B.7) is the same as minimizing, for a fixed $\widetilde{\mathbf{U}}$, the negative of the log-likelihood. Based on (B.11) and (B.12), one can easily conclude that

$$\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}} = \frac{Z_{\mathbf{U}}}{Z_{\widetilde{\mathbf{U}}}}. \tag{B.16}$$

Therefore, the negative of the log-likelihood equals to

$$\begin{aligned} -L_{\mathbf{U}}(X_{tr}) &= -\overline{\mathbf{N}}^t(X_{tr})\mathbf{U} + \log\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}} + \log Z_{\widetilde{\mathbf{U}}} \\ &= -L_{\widetilde{\mathbf{U}}}(X_{tr}) - \overline{\mathbf{N}}^t(X_{tr})\left(\mathbf{U}-\widetilde{\mathbf{U}}\right) + \log\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}, \end{aligned} \tag{B.17}$$

which implies that we need only to consider the objective function

$$\vartheta(\mathbf{U}) = -\overline{\mathbf{N}}^t(X_{tr})\left(\mathbf{U}-\widetilde{\mathbf{U}}\right) + \log\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}, \tag{B.18}$$

since $-L_{\widetilde{\mathbf{U}}}(X_{tr})$ is simply a reference value that does not intervene in the optimization.

B.4 Implementation

By sampling with respect to $\pi_{\widetilde{\mathbf{U}}}$, the partial derivatives of the log-likelihood in (B.15) can be empirically determined by first estimating expectations, such as the denominator in the same equation, as

$$\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}} \approx \frac{1}{S}\sum_{u=1}^{S} \exp\left[\mathbf{N}^t(x^{(u)})\left(\mathbf{U}-\widetilde{\mathbf{U}}\right)\right], \tag{B.19}$$

where x_u $(u = 1,..,S)$ are samples from $\pi_{\widetilde{\mathbf{U}}}$. It is pointed out in [20] that in order to avoid large variances in the Monte Carlo estimates, it is necessary that $\mathbf{U}$ does not differ too much

from $\widetilde{\mathbf{U}}$, by some measure. The following criteria, also suggested in [20], try to justify this statement:

1. the total variation between the distributions $\pi_{\widetilde{\mathbf{U}}}$ and $\pi_{\mathbf{U}}$ is below some threshold and
2. for each image x sampled from $\pi_{\widetilde{\mathbf{U}}}$, let $\xi(x)$ be a weight function defined by

$$\xi(x) = \mathbf{N}^t(x)\left(\mathbf{U} - \widetilde{\mathbf{U}}\right), \tag{B.20}$$

then the criterion establishes that the difference $\max_x[\xi(x)] - \min_x[\xi(x)]$ is bounded by some small positive number ϖ, where the maximum and the minimum is taken among the (independent) sample images $\left\{x^{(u)}\right\}_{1\le u\le S}$.

According to the authors in [20], the number ϖ should be chosen so that the estimate of $\left\langle \exp\left[\mathbf{N}^t(x)\left(\mathbf{U} - \widetilde{\mathbf{U}}\right)\right]\right\rangle_{\widetilde{\mathbf{U}}}$ in (B.19), which is

$$\frac{1}{S}\sum_{u=1}^{S}\exp\left[\xi(x^{(u)})\right] = \frac{\exp[\max_x[\xi(x)]]}{S}\sum_{u=1}^{S}\exp\left\{\xi(x^{(u)}) - \max_x[\xi(x)]\right\}, \tag{B.21}$$

is not likely to contain too many nearly-zero terms in the summation. This would avoid "wasting" too many samples and hence give a better estimate of the expectation. Similarly, one can approximate numerator in (B.15) by

$$\frac{\exp[\max_x[\xi(x)]]}{S}\sum_{u=1}^{S}N_c(\{x_u\})\exp\left\{\xi(x^{(u)}) - \max_x[\xi(x)]\right\}. \tag{B.22}$$

It is not stated in [20] which criterion is used, but we think that none of the them by itself is useful enough, because nothing is said about the number of samples S. We introduce a new criterion based on error propagation analysis, but we also use in conjunction the criterion (ii) solely to avoid numerical overflows or underflows.

In the rest of the appendix we explain what we believe to be a better founded version of the MCMCML method, by giving the details of the evaluation of all the needed quantities, namely the gradient and the objective function. We will make precise our criterion as we develop the discussions, but first we define a *search interval*. At the end, we present an algorithm based on our version of the MCMCML method.

Given a direction p and a parameter $\mathbf{U}$, a parameter $\widetilde{\mathbf{U}}$ is said to be within the search interval (with respect to $\mathbf{U}$ and along p) if the second and third criteria are satisfied for that $\widetilde{\mathbf{U}}$.

In the MCMCML method two quantities need to be evaluated repeatedly. They are the gradient of the log-likelihood in (B.14) (or (B.15)) and the objective function in (B.17). In our chosen conjugate gradient algorithm (which does not compute the Hessian), for each evaluation of the gradient (which is followed by the computation of the corresponding conjugate direction), several evaluations of the objective function may be necessary until a minimum along the conjugate direction is found.

We now explain why we choose (B.14) over (B.15) to estimate the gradient, even though the estimation of (B.14) implies re-sampling at the current parameter. In our experiments, we observed that, if we take more than one sample (i.e., $S > 1$), the variance of the random variable $N_c(\{x\})$ in (B.15) is significantly "amplified" by the exponential operation in the same equation. This implies that the empirical averages will be dominated by outliers. Therefore, in order to get a reasonable estimate of the gradient, either the search interval needs to be very small or a (impractically) large number of samples (from $\pi_{\widetilde{\mathbf{U}}}$) are required. Either choice leads to an extremely slow computation. If $S = 1$, the exponential terms in the numerator and the denominator cancel out, meaning that the gradient (at $\mathbf{U}$), yields (possible very) different estimates using different $\widetilde{\mathbf{U}}$'s, which is an undesirable property. On the other hand, if we sample at the current $\mathbf{U}$ (i.e., if $\widetilde{\mathbf{U}} = \mathbf{U}$) then (B.15) reduces to (B.14),

which does not contain the exponential factors and is a more robust estimation formula.

We now turn to the estimation of errors that are propagated when applying the formulas (B.18) and (B.14). Given a label image x coming from some Gibbs distribution $\pi_{\mathbf{U}}$, the quantities $N_c(\{x\},\mathbf{U})$, for $c = 1,...,C$, (the extra argument $\mathbf{U}$ is to emphasize the dependency) are random variables and so are functions defined on them. In particular, we note that in both formulas the first term, unlike the second term, is dependent on the data (over which we do not have control). In order to obtain a reasonable estimate, consequently, we should make the error (standard deviation) of the second term as small as possible, certainly not exceeding that of the first one.

For the formula (B.18), given the (unknown) parameter vector $\mathbf{U}^*$ that generated the training samples, two other parameters vectors $\mathbf{U}$ and $\widetilde{\mathbf{U}}$, and S images sampled from $\pi_{\widetilde{\mathbf{U}}}$, we define some auxiliary functions

$$\omega_0\left(\mathbf{U},\mathbf{U}^*\right) = \sum_{c=1}^{C} \overline{N}_c(X_{tr},\mathbf{U}^*)\left(U_c - \widetilde{U}_c\right), \tag{B.23}$$

$$\omega_1\left(\mathbf{x},\widetilde{\mathbf{U}},\mathbf{U}\right) = \sum_{c=1}^{C} N_c(\{\mathbf{x}\},\widetilde{\mathbf{U}})\left(U_c - \widetilde{U}_c\right), \tag{B.24}$$

$$\omega_2\left(\mathbf{x},\widetilde{\mathbf{U}},\mathbf{U}\right) = \exp\left[\omega_1(\mathbf{x},\widetilde{\mathbf{U}},\mathbf{U})\right], \tag{B.25}$$

$$\omega_3\left(\widetilde{\mathbf{U}},\mathbf{U}\right) = \frac{1}{S}\sum_{u=1}^{S} \omega_2\left(\mathbf{x}^{(u)},\mathbf{U},\mathbf{U}^*\right), \tag{B.26}$$

$$\omega_4\left(\widetilde{\mathbf{U}},\mathbf{U}\right) = \log\left[\omega_3\left(\widetilde{\mathbf{U}},\mathbf{U}\right)\right]. \tag{B.27}$$

Only the function $\omega_0(\mathbf{U}, \mathbf{U}^*)$ depends on the training set. Then (B.18) can be approximated as

$$\vartheta(\mathbf{U}) = \vartheta(\mathbf{U}, \widetilde{\mathbf{U}}, \mathbf{U}^*) = -\omega_0(\mathbf{U}, \mathbf{U}^*) + \omega_4\left(\widetilde{\mathbf{U}}, \mathbf{U}\right), \tag{B.28}$$

Given a random variable ω, let σ_ω be its standard variation. To find the variance

$$\sigma^2_\vartheta = \sigma^2_{\omega_0} + \sigma^2_{\omega_4}, \tag{B.29}$$

we apply the error propagation rules (see Appendix C) and determine that, at first order,

$$\sigma^2_{\omega_3} = \frac{1}{S^2}\sum_{u=1}^{S}\sigma^2_{\omega_2}(\mathbf{x}^{(u)}) = \frac{\sigma^2_{\omega_1}}{S^2}\sum_{u=1}^{S}\overline{\omega}^2_2(\mathbf{x}^{(u)}); \tag{B.30}$$

therefore

$$\sigma^2_{\omega_4} = \frac{1}{\overline{\omega}^2_3}\sigma^2_{\omega_3} = \frac{\sum_{u=1}^{S}\overline{\omega}^2_2(\mathbf{x}^{(u)})}{\left[\sum_{u=1}^{S}\overline{\omega}_2(\mathbf{x}^{(u)})\right]^2} = \frac{1}{S}\sigma^2_{\omega_1}, \tag{B.31}$$

where we have assumed that all the $\omega_1(\mathbf{x}^{(u)})$, for $u = 1,\ldots,S$, have the same variance. We have then from (B.29) and (B.31),

$$\sigma^2_\vartheta = \sigma^2_{\omega_0} + \frac{1}{S}\sigma^2_{\omega_1} \tag{B.32}$$

The error due to ω_0, on the other hand, is not under our control and is determined by the variances of $\overline{N}_c(X_{tr}, \mathbf{U}^*)$, for $c = 1,\ldots,C$, which in turn depends on the size of the training

set X_{tr} and $\mathbf{U}^*$:

$$\sigma^2_{\omega_0} = \sum_{c=1}^{C} \sigma^2_{\overline{N}_c(X_{tr},\mathbf{U}^*)} \left| U_c - \widetilde{U}_c \right|^2. \tag{B.33}$$

Meanwhile, $\sigma^2_{\omega_1}$ can be controlled and written as

$$\sigma^2_{\omega_1} = \sum_{c=1}^{C} \sigma^2_{N_c(\{\mathbf{x}\},\widetilde{\mathbf{U}})} \left| U_c - \widetilde{U}_c \right|^2. \tag{B.34}$$

Equation (B.32) implies that $\sigma^2_{\vartheta} \geq \sigma^2_{\omega_0}$. Making $\sigma^2_{\omega_1} = 0$ is practically infeasible (unless $\mathbf{U} = \widetilde{\mathbf{U}}$, which implies sampling at every evaluated $\mathbf{U}$). On the other hand, one can establish conditions under which

$$\sigma^2_{\omega_0} \sim \frac{1}{S}\sigma^2_{\omega_1}, \tag{B.35}$$

where the symbol "$\sim$" means "is of the same order as." As the MCMCML algorithm converges, supposedly $\mathbf{U} \approx \widetilde{\mathbf{U}} \approx \mathbf{U}^*$, therefore (B.35) is satisfied if we set $S = |X_{tr}|$ and noting that

$$\sigma^2_{\overline{N}_c(X_{tr},\mathbf{U}^*)} \approx \frac{1}{|X_{tr}|}\sigma^2_{N_c(\{\mathbf{x}_{tr}\},\mathbf{U}^*)} \approx \frac{1}{|X_{tr}|}\sigma^2_{N_c(\{\mathbf{x}\},\widetilde{\mathbf{U}})}, \tag{B.36}$$

for $c = 1,\ldots,C$, where $\mathbf{x}_{tr}$, viewed as a random vector, is an element of X_{tr}.

As for the error in estimating the gradient based on (B.14) that also has two terms (the right term is "controllable" but left term is not), a similar analysis tells us again that the number of sample images S should not be smaller than the number of training images $|X_{tr}|$, if we want to keep the controllable error smaller than the non-controllable error.

We are now in position to define the third criterion. It establishes that $S \geq |X_{tr}|$ and that $\left|U_c - \widetilde{U}_c\right|$, for $c = 1,\ldots,C$, must be small enough so that (B.35) is satisfied. In practice we found that in fact if this criterion is satisfied, then so is the second criterion.

In summary, our specific MCMCML parameter estimation algorithm is the following.

1. Compute $\overline{N}_c(X_{tr})$ for $c = 1,\ldots,C$.

2. Initialize the estimate $\mathbf{U}^{(0)}$ with arbitrary entries (not too far from the truth, if possible) and set $l = 0$ and $\widetilde{\mathbf{U}} = \mathbf{U}^{(0)}$.

3. Set $\widetilde{\mathbf{U}} = \mathbf{U}^{(l)}$ and generate $10\,|X_{tr}|$ samples from $\pi_{\widetilde{\mathbf{U}}}$.

4. Estimate the gradient of the log-likelihood at $\mathbf{U}^{(l)}$ using (B.14) and the corresponding conjugate direction, using (B.4).

5. For the current conjugate direction, define a search interval on which the estimate is considered robust based both on the second and the third criteria. We take $\varpi = 25$.

6. Compute $\mathbf{U}^{(l+1)}$ by minimizing the objective function within each search interval.

7. If the Euclidean norm $\left\| \vartheta\left(\mathbf{U}^{(l+1)}\right) - \vartheta\left(\mathbf{U}^{(l)}\right) \right\|$ is larger than a small positive value (in this case 10^{-6}), set l to $l+1$ and go to step 3; otherwise, stop.

Appendix C

Error propagation analysis

Given two random variables ω_1 and ω_2, let ω_3 be a function of ω_1 and ω_2

$$\omega_3 = f(\omega_1, \omega_2). \tag{C.1}$$

(As a result, ω_3 is also a random variable.) For a random variable ω, let $\overline{\omega}$ and σ_ω denote, if exist, its mean and standard variation. If we approximate ω_3 by its first order Taylor expansion with respect to ω_1 and ω_2 and around $f(\overline{\omega}_1, \overline{\omega}_2)$, we get

$$\omega_3 = f(\omega_1 + \overline{\omega}_1, \omega_2 + \overline{\omega}_2) \approx f(\overline{\omega}_1, \overline{\omega}_2) + \omega_1 \frac{\partial f}{\partial \omega_1}(\overline{\omega}_1, \overline{\omega}_2) + \omega_2 \frac{\partial f}{\partial \omega_2}(\overline{\omega}_1, \overline{\omega}_2) \tag{C.2}$$

and therefore

$$\overline{\omega}_3 \approx f(\overline{\omega}_1, \overline{\omega}_2) + \overline{\omega}_1 \frac{\partial f}{\partial \omega_1}(\overline{\omega}_1, \overline{\omega}_2) + \overline{\omega}_2 \frac{\partial f}{\partial \omega_2}(\overline{\omega}_1, \overline{\omega}_2). \tag{C.3}$$

Subtracting (C.3) from (C.2), we get that

$$\omega_3 - \overline{\omega}_3 \approx (\omega_1 - \overline{\omega}_1)\,\frac{\partial f}{\partial \omega_1}(\overline{\omega}_1, \overline{\omega}_2) + (\omega_2 - \overline{\omega}_2)\,\frac{\partial f}{\partial \omega_2}(\overline{\omega}_1, \overline{\omega}_2), \tag{C.4}$$

from which it is easy to see that if ω_1 and ω_2 are statistically independent, then

$$\sigma^2_{\omega_3} \approx \sigma^2_{\omega_1}\left[\frac{\partial f}{\partial \omega_1}(\overline{\omega}_1, \overline{\omega}_2)\right]^2 + \sigma^2_{\omega_2}\left[\frac{\partial f}{\partial \omega_1}(\overline{\omega}_1, \overline{\omega}_2)\right]^2. \tag{C.5}$$

Assuming ω_3 is linear in ω_1 and ω_2, all the "approximate" signs become "equal" signs. In particular, for $k_2, k_3 \in \mathbb{R}$, if $\omega_3 = k_1\omega_1 + k_2\omega_2$ then $\sigma^2_{\omega_3} = k_1^2\sigma^2_{\omega_1} + k_2^2\sigma^2_{\omega_2}$. In case of the product or division, i.e., $\omega_3 = \omega_1 \cdot \omega_2$ or $\omega_3 = \dfrac{\omega_1}{\omega_2}$ then $\left(\dfrac{\sigma_{\omega_3}}{\overline{\omega}_3}\right)^2 \approx \left(\dfrac{\sigma_{\omega_1}}{\overline{\omega}_1}\right)^2 + \left(\dfrac{\sigma_{\omega_2}}{\overline{\omega}_2}\right)^2$ (assuming $\omega_2 \cdot \omega_3 \neq 0$). As for exponential and logarithmic operations, if $\omega_3 = \exp(\omega_1)$, then $\sigma_{\omega_3} \approx \overline{\omega}_3 \cdot \sigma_{\omega_1}$, and if $\omega_3 = \log(\omega_1)$, then $\sigma_{\omega_3} \approx \dfrac{\sigma_{\omega_1}}{\overline{\omega}_1}$.

Appendix D

The modified histogram method

This method was proposed in [36] and briefly discussed in Subsection 4.4.2. For the discussion that follows, we restrict attention to a single local interaction vector. To simplify notation, we drop the subindex b and let N_1, N_0, and N denote respectively $N(\Omega_b^1, X_{tr})$, $N(\Omega_b^0, X_{tr})$, and $N(\Omega_b^1, X_{tr}) + N(\Omega_b^0, X_{tr})$. Recall that in the histogram method the estimate on the right hand side of the system (4.8), which we now denote it by $\mathbf{H}(N_1, N_0)$, is defined as

$$\mathbf{H}(N_1, N_0) = \log\left(\frac{N_1}{N_0}\right) \tag{D.1}$$

provided that $N_1 \cdot N_0 \neq 0$ (otherwise, the corresponding equation is eliminated). In the modified histogram method a new estimate $\mathbf{H}_m(N_1, N_0)$ is used. Depending on the value of N, the authors in [36] propose to do the following: for two given non-negative integer numbers r_1 and r_2, if $N < r_1$ then drop the equation; if $r_1 \leq N \leq r_2$ then use the estimate $\mathbf{H}(N_1, N_0)$ provided by the histogram method; and finally if $N > r_2$ then use $\mathbf{H}_m(N_1, N_0)$,

which is defined as the least-squares solution of the system

$$\sum_{n=0}^{N} C_N^n z_j^n (1-z_j)^{N-n} \mathbf{H}_m(n, N-n) = \ln\left(\frac{z_j}{1-z_j}\right), \tag{D.2}$$

for some selected $z_1, \ldots, z_J$ in the open interval $(0,1)$. It is suggested in [36] that z_j come from the set Z, where

$$Z = \{z \mid z = 0.5 \pm (0.05 + 0.01k), k = 0, \ldots, K_1\} \tag{D.3}$$

in the case $r_1 \leq N < r_2$, and

$$Z = \{z \mid z = 0.5 \pm (0.05 + 0.01k), k = 0, \ldots, K_2\} \tag{D.4}$$

when $N = r_2$. For the experiments in [36], the size of a closed neighborhood is 3×3; therefore, the authors state that for a small training set an important portion of the discarded data in the histogram method usually would correspond to $N = 5, \ldots, 11$. Furthermore, the bias in the estimate is not very significant for $N > 11$; therefore, r_1 and r_2 can be set to be 5 and 11, respectively. As for K_1 and K_2, the authors suggest $K_1 = 15$ and $K_2 = 25$, arguing that these values work fine in their examples, but they also state that no specific strategy has been followed for such a choice.

Some disadvantages of this method that are inherited from the histogram method are (i) for $r_1 \leq N \leq r_2$, the estimator $\mathbf{H}(N_1, N_0)$ does not provide estimate when $N_0 \cdot N_1 = 0$, which may occur often even for the model in [36] and (ii) consequently dropping the corresponding equation for which this occurs may lead to rank-deficiency system. Another drawback reported in [8] of the modified histogram method is that the estimate $\mathbf{H}_m(N_1, N_0)$ sometimes may be negative and that the estimator is in fact inconsistent.

Appendix E

More on the coordinate ascent approach

By substituting (5.30) and (5.46) into (5.50), we get that

$$O_2(x,y) = \left(\frac{1}{ZC_wC_x}\right) \exp\left[-\frac{1}{2}(w-Ry)^t\Sigma_w^{-1}(w-Ry) - \frac{1}{2}(y-\mu_x)^t\Sigma_x^{-1}(y-\mu_x)\right], \quad \text{(E.1)}$$

where $C_w = (2\pi)^{J/2}\,|\Sigma_w|^{1/2}$ and $C_x = (2\pi)^{I/2}\,|\Sigma_x|^{1/2}$, with $|\Sigma_w|^{1/2} = \prod_{j=1}^{J}\sigma_{wj}$ and $|\Sigma_x|^{1/2} = \prod_{i=1}^{I}\sigma_{xi}$. We are interested only in the case where C_w does not depend explicitly on either x and y; since neither does the factor Z, both C_w and Z can be ignored during the optimization process in the the Coordinate Ascent (CA) algorithm.

The CA method starts with an initial $y^{(0)}$. It is taken to be the gray value image produced by a reconstruction based on the (noisy) projections, using the *Algebraic Reconstruction Technique* (ART) described in [39, Chapter 11] (see also Appendix F) with 256 cycles through the data, initial image vector with all its components set to zero and relaxation parameter equal to 0.5. The data were accessed so that all the lines in one projection were processed before going to the next projection, and the sequence of projections (with no repetition in one cycle) were chosen so that the angle between any pair of consecutive

projections directions was as large as possible. The CA method then alternately optimizes with respect to x (the *x-step*) and then with respect to y (the *y-step*), as described in the pseudo-code of Table E.1. Note that there is no dependency on x of the function $\chi(w|y)$ in (5.46) _and thus also in (5.50)_.

Since C_x does not depend on y, maximizing (E.1) with respect to y is equivalent to minimizing the quadratic in the exponent in (E.1), which is the $\widetilde{q}(y)$ in (5.31). Because Σ_x and Σ_w are positive definite, the minimizer of $\widetilde{q}(y)$ is unique (see, e.g., Section 12.1 of [39]). This assures that our termination test of $x^l = x^{l-1}$, for $l = 1, 2, \ldots$, implies that $y^{(l)} = y^{(l-1)}$. (Our way of indexing for x is different from that for y, so that it is not confused with a Metropolis step inside the *x-step*.)

Among the various methods for the minimization of (5.31) (e.g., conjugate gradient methods, the steepest descent method, etc.) we resort to ART, which is an easy-to-implement row-action iterative algorithm with relatively fast convergence properties. The main idea is to construct from the original system $Ry = z$ of equations (that is most likely to be inconsistent) a consistent system of linear equations, in such a way that there is a one-to-one onto continuous mapping between the solutions of this system and I-dimensional vectors y. Furthermore, the squared (Euclidean) norm of a solution of the new system is equal to the value of $\widetilde{q}(y)$ for the corresponding y. An ART algorithm that yields iterates converging to the (unique) minimum norm solution of the new system can then be adapted to converge to the minimizer of the quadratic. In [39] an algorithm is described that optimizes $\widetilde{q}(y)$ for matrices Σ_x and Σ_w that are proportional to the identity matrices of appropriate sizes. Generalized to our case, the algorithm for finding y that minimizes $\widetilde{q}(y)$ is the following (see Appendix F.1 for the derivations).

During the l-th y-step (see the pseudo-code in Table E.1)

Initialization
$y^{(0)}, l \leftarrow 1$
Repeat
x-step
$x^l = \arg\max_x \left[\pi(x)\phi(y^{(l-1)}|x)\right]$
y-step
$y^{(l)} = \arg\max_y \left[\chi(w|y)\phi(y|x^l)\right]$
$l \leftarrow l+1$
Until
termination test == true

Table E.1: Pseudo-code for the Coordinate Ascent approach.

$$u^{(0)} \quad \text{is} \quad \text{the } J\text{-dimensional column vector of zeros,} \tag{E.2}$$

$$y^{(l,0)} = \mu_{x^{(n)}}, \tag{E.3}$$

$$u^{(m+1)} = u^{(m)} + c^{(m)}\sigma_{wj_m}, \tag{E.4}$$

$$y^{(l,m+1)} = y^{(l,m)} + c^{(m)}\Sigma_x r_{j_m} \tag{E.5}$$

with

$$c^{(m)} = \frac{w_{j_m} - r_{j_m}^t y^{(l,m)} - \sigma_{wj_m}^t u^{(m)}}{\sigma_{wj_m}^t \sigma_{wj_m} + r_{j_m}^t \Sigma_x r_{j_m}}, \tag{E.6}$$

where j_m denotes $m \pmod J + 1$, σ_{wj_m} and r_{j_m} are the respective transposes of the j_m-th row of $\Sigma_w^{1/2}$ and of R ($r^{(m)}$, $y^{(l,m)}$, and $\mu_{x^{(n)}}$ are all column vectors, for $m = 0, 1, \ldots$). Specifically, for the experiments, we chose 256 cycles through the data (i.e., the output of the n-th y-step is $y^{(l)} = y^{(l,256J)}$), and the data were accessed in the same way as was done for obtaining the $y^{(0)}$.

Appendix F

Algebraic Reconstruction Techniques

First we present an algebraic reconstruction technique [39, Chapter 11] for solving a consistent system of linear equations

$$Ag = b \tag{F.1}$$

with S unknowns and L equations. For $1 \leq l \leq L$, let A_l be the transpose of the l-th row of A and b_l be the l-th component of **b**. The following algorithm is a relaxation method for solving the system (F.1) of equalities (Section 11.2 of [39]).

$$g^{(0)} \quad \text{is} \quad \text{the } S\text{-dimensional column vector of zeros,} \tag{F.2}$$

$$g^{(m+1)} \quad = \quad g^{(m)} + c^{(m)} A_{k_m}, \tag{F.3}$$

with

$$c^{(m)} = \frac{\left(b_{k_m} - A_{k_m}^t g^{(m)}\right)}{A_{k_m}^t A_{k_m}}, \tag{F.4}$$

where $k_m = m \,(\mathrm{mod}\, K) + 1$. The sequence of the $g^{(m)}$ converges to the (unique) minimum (Euclidean) norm solution of the system (F.1).

F.1 An application: finding the minimizer of a quadratic function

To find the minimizer y^* of the quadratic $\widetilde{q}(y)$ in (5.31), consider the system

$$\left[\Sigma_w^{1/2} \;\; R\Sigma_x^{1/2}\right] \begin{bmatrix} u \\ v \end{bmatrix} = \left(w - R\mu_{x^{(n)}}\right), \tag{F.5}$$

in which the system matrix $\left[\Sigma_w^{1/2} \;\; R\Sigma_x^{1/2}\right]$ is of size $J \times (J+I)$, the unknown $\begin{bmatrix} u \\ v \end{bmatrix}$ is a column vector of size $J+I$, and the right hand side $\left(w - R\mu_{x^{(n)}}\right)$ is a column vector of size J. The following result is easy to prove.

Lemma: For any J-dimensional vector u and I-dimensional vector v, $\begin{bmatrix} u \\ v \end{bmatrix}$ is a solution of (F.5) if, and only if, there exists a I-dimensional vector y such that

$$u = u(y) = \Sigma_w^{-1/2}(w - Ry) \tag{F.6}$$

and

$$v = v(y) = \Sigma_x^{-1/2}\left(y - \mu_{x^{(n)}}\right). \tag{F.7}$$

This lemma implies in particular that (F.5) is a consistent system of linear equations. The squared norm of any of its solutions can be written, in view of (F.6) and (F.7), as

$$(w-Ry)^t\Sigma_w^{-1}(w-Ry)+(y-\mu_{x^{(n)}})^t\Sigma_x^{-1}(y-\mu_{x^{(n)}}), \tag{F.8}$$

which is precisely the quadratic $\widetilde{q}(y)$ of (5.31). It follows that if $u^{(m)}$ and $v^{(m)}$ are sequences of K-dimensional, respectively J-dimensional, vectors such that $\begin{bmatrix} u^{(m)} \\ v^{(m)} \end{bmatrix}$ converges to the minimum norm solution of (F.5) and if, for all m, we define

$$y^{(n,m)}=\Sigma_x^{1/2}v^{(m)}+\mu_{x^{(n)}}, \tag{F.9}$$

then $y^{(n,m)}$ converges to the minimizer of $\widetilde{q}(y)$ in (5.31).

In order to apply the general algorithm described in (F.2)-(F.4) to obtain a sequence $\begin{bmatrix} u^{(m)} \\ v^{(m)} \end{bmatrix}$ that converges to the minimum norm solution of (F.5), we observe that for the special case

$$A_{j_m}^t=\begin{bmatrix}\sigma_{wj_m}^t & r_{j_m}^t\Sigma_x^{1/2}\end{bmatrix} \tag{F.10}$$

and

$$b_{j_m}=w_{j_m}-r_{j_m}^t\mu_{x^{(n)}}^{(m)}, \tag{F.11}$$

where σ_{wj_m} and r_{j_m} are as defined after (E.6). This combined with (F.9) yields the algorithm described in (E.2)-(E.6).

As noted just above, the sequence $y^{(n,m)}$ converges to a minimizer of $q(y)$ as m goes to infinity.

Appendix G

BCC grid and FCC grid as reciprocal grids

We use the results from [30] regarding a BCC grid and an FCC grid as reciprocal grids, For a grid E, let

$$shah(E) \equiv \sum_{\overline{z} \in E} \delta_{\overline{z}}, \tag{G.1}$$

where $\delta_{\overline{z}}$ be the Dirac's delta function centered at $\overline{z}$. Then we have that

$$\widehat{shah\,(B_\Delta)} = \frac{1}{(2\Delta)^3} shah\left(F_{1/2\Delta}\right). \tag{G.2}$$

and

$$\widehat{shah\,(F_\Delta)} = \frac{1}{(2\Delta)^3} shah\left(B_{1/2\Delta}\right), \tag{G.3}$$

where, for a multivariate function or *generalized function* [48] $f(x)$, the Fourier transform $\widehat{f}(\xi)$ (if exists) is defined as

$$\widehat{f}(\xi) \equiv \int_{-\infty}^{\infty} e^{-2\pi i x\xi} f(x)dx. \tag{G.4}$$

Suppose now that we are given the FCC grid F_1 and we need to determine a BCC grid B_Δ that is equivalent to F_1. From (G.3), clearly $\widehat{shah\,(F_1)} = \frac{1}{8} shah\left(B_{1/2}\right)$, which implies that the maximal radius in $B_{1/2}$ will be $\frac{1}{2}$. On the other hand, $\widehat{shah\,(B_\Delta)} = \frac{1}{(2\Delta)^3} shah\left(F_{1/2\Delta}\right)$, and therefore the maximal radius in $F_{1/2\Delta}$ is going to be $\frac{\sqrt{2}}{4\Delta}$, which has to be equal to $\frac{1}{2}$. This gives $\Delta = \frac{1}{\sqrt{2}}$.

Appendix H

Determining optimal blob parameters

We find the parameters a and α of a blob, so that a series expansion using blobs with a same coefficient on the BCC grid B_Δ "best" approximates a constant [35]. Such expansion can be thought of as the *convolution* [9] of a blob $b(x,y,z)$ with impulses centered at points in B_Δ. The approximation is done in the Fourier space, where the expansion is transformed into a multiplication of the FT of a blob [54] (denoted by $\widehat{b}$) with impulses centered at the points in the reciprocal grid $\frac{1}{(2\Delta)^3}F_{1/(2\Delta)}$. Since the FT of a constant is an *impulse* (Dirac's delta function or its multiple) at the origin, the goal is to make this multiplication resemble an impulse as much as possible. This can be achieved with high accuracy by tuning the parameters a and α in such a way that $\widehat{b}$ crosses zero exactly at the location of the (reciprocal) grid points. Since there are only two unknowns (a and α) and $\widehat{b}$ is spherically symmetric, we can impose such zero crossing conditions at only two different distances from the origin. Clearly, since $\widehat{b}$ vanishes very rapidly, for higher effectiveness of the approximation, it makes sense to consider grid points that are the nearest and the second nearest to the origin. It is easy to see that in the reciprocal grid these two distances

are $\frac{1}{\sqrt{2}\Delta}$ and $\frac{1}{\Delta}$. Meanwhile, $\widehat{b}$ crosses zero a distance d from the origin if and only if [54]

$$(2\pi a d)^2 - \alpha^2 = c^2, \tag{H.1}$$

where c is a zero of Bessel function (of first kind). Therefore, if we consider two such equations with, e.g., the first and the fifth zeros (respectively, 6.38016189 and 19.4094152)of Bessel function and d equal to $\frac{1}{\sqrt{2}\Delta}$ and $\frac{1}{\Delta}$, respectively, then we get (for $\Delta = \frac{1}{\sqrt{2}}$) a=2.9174404 and α=17.18465792.

Appendix I

Parameters of the spheres in the 3D phantoms

In Table I.2 we report on the exact locations and the sizes of the spheres of the phantom shown in Figure 7.7.

large spheres	coordinate 1	coordinate 2	coordinate 3	radius
top	b	b	b	6.3
middle	d	d	d	10.9
middle	d	e	d	10.9
middle	e	d	d	10.9
middle	e	e	d	10.9
bottom	a	a	a	6.3
bottom	a	c	a	6.3
bottom	c	a	a	6.3
bottom	c	c	a	6.3

Table I.1: Locations of the centers of the large spheres in the phantom of Figure 7.7 with values: $a = 10.5$, $b = 31.5$, $c = 52.5$, $d = 21$, and $e = 42$.

little spheres	coordinate 1	coordinate 2	coordinate 3
top	$b \pm f$	b	b
top	b	$b \pm f$	b
bottom	$a \pm g$	$a \pm g$	a
bottom	$a \pm g$	$c \pm g$	a
bottom	$c \pm g$	$a \pm g$	a
bottom	$c \pm g$	$c \pm g$	a

Table I.2: Locations of the centers of the little spheres in the phantom of Figure 7.7 with values: $a = 10.5$, $b = 31.5$, $c = 52.5$, $f = 7.9$, and $g = 5.59$. All the radius are 2.1, and the two "$\pm$" signs in each row mean that all the four possible combinations are considered.

Bibliography

[1] A.J. Baddeley. An error metric for binary images. In W. Forstner and S. Ruwiedel, editors, *Robust Computer Vision: Quality of Vision Algorithms*, pages 59–78. Wichmann, Karlsruhe, 1992.

[2] S. Barnard. Stochastic stereo matching over scale. *Int. J. Comput. Vision*, 3:17–32, 1989.

[3] W. Baumeister, R. Grimm, and J. Waltz. Electron tomography of molecules and cells. *Trends Cell Biol.*, 9:81–85, 1999.

[4] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statist. Soc. Ser. B*, 2:192–236, 1974.

[5] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.

[6] J. Besag. Efficiency of pseudo-likelihood estimation for simple Gaussian fields. *Biometrika*, 64:616–618, 1977.

[7] J. Besag. On the statistical analysis of dirty pictures. *J. Roy. Statist. Soc. Ser. B*, 48:259–302, 1986.

[8] C.F. Borges. On the estimation of Markov random field parameters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:216–224, 1999.

[9] R.N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-HIll Book Company, New York, third edition, 1999.

[10] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer-Verlag, New York, 1999.

[11] J.M. Carazo, C.O. Sorzano, E. Rietzel, R. Schröder, and R. Marabini. Discrete tomography in electron microscopy. In G.T. Herman and A. Kuba, editors, *Discrete Tomography: Foundations, Algorithms and Applications*, pages 405–416. Birkhäuser, Boston, 1999.

[12] B.M. Carvalho, G.T. Herman, S. Matej, C. Salzberg, and E. Vardi. Binary tomography for triplane cardiography. In A. Kuba, M. Sámal, and A. Todd-Pokropek, editors, *Information Processing in Medical Imaging*, pages 29–41. Springer-Verlag, Berlin, 1999.

[13] V. Cerney. Thermodynamical approach to the traveling salesmen problem: an efficient simulation algorithm. *J. Optimiz. Theory Appl.*, 45:41–51, 1985.

[14] M.T. Chan, G.T. Herman, and E. Levitan. Probabilistic modeling of discrete images. In G.T. Herman and A. Kuba, editors, *Discrete Tomography: Foundations, Algorithms and Applications*, pages 213–235. Birkhäuser, Boston, 1999.

[15] W. Chiu. What does electron cryomicroscopy provide that x-ray crystallography and nmr spectroscopy cannot? *Annu. Rev. Biophys. Bio.*, 22:233–255, 1993.

[16] J.J. Clark and A.L. Yuille. *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers, Norwell, MA, 1990.

[17] G. Cross and A.K. Jain. Markov random field texture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5:25–39, 1983.

[18] A.P. Dempster, N.M. Laird, and D.B. Bubin. Maximum likelihood from incomplete data via EM algorithm. *J. Royal Stat. Soc., Series B*, 39:1–38, 1977.

[19] H. Derin and H. Elliot. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:39–55, 1987.

[20] X. Descombes, R.D. Morris, J. Zerubia, and M. Berthod. Estimation of Markov random field prior parameters using Markov chain Monte Carlo maximum likehood. *IEEE Trans. Image Proc.*, 8:954–963, 1999.

[21] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2001.

[22] J.A. Fessler. Segmented attenuation correction for pet. In *Proc. IEEE Nuc. Sci. Symp. and Med. Imag. Conf.*, pages 1182–1184, Orlando, FL, 1992. IEEE.

[23] G.S. Fishman. *Monte Carlo*. Springer, New York, 1996.

[24] D.B. Fogel. An introduction to simulated evolutionary optimization. *IEEE Trans. on Neural Networks*, 5:3–14, 1994.

[25] C. Fox and G.K. Nicholls. Exact MAP states and expectations from perfect sampling: Greig, Porteous and Seheult revisited. In A. Mohammad-Djafari, editor, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, pages 252–263. AIP conference proceedings, Melville, NY, 2001.

[26] J. Frank. *Electron Tomography: Three-Dimensional Imaging with the Transmission Electron Microscope*. Plenum Press, New York, 1992.

[27] J. Frank. *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*. Academic Press, San Diego, CA, 1996.

[28] Y. Fujiyoshi. The structural study of membrane proteins by electron crystallography. *Adv. Biophys.*, 35:25–80, 1998.

[29] S.S. Furuie, G.T. Herman, T.K. Narayan, P. Kinahan, J.S. Karp, R.M. Lewitt, and S. Matej. A methodology for testing for statistically significant differences between fully 3–D PET reconstruction algorithms. *Phys. Med. Biol.*, 39:341–354, 1994.

[30] E. Garduño. *Visualization and Extraction of Structural Components from Reconstructed Volumes*. PhD thesis, University of Pennsylvania, Philadelphia, USA, 2002.

[31] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741, 1984.

[32] R.M. Glaeser. Limitations to significant information in biological electron microscopy as a result of radiation damage. *J. Ultrastruct. Res.*, 36:466–82, 1971.

[33] R.M. Glaeser, V.E. Cosslett, and U. Valdre. Low temperature electron microscopy: radiation damage in crystalline biological materials. *J. Microsc.*, 12:133–38, 1971.

[34] R. Gordon, R. Bender, and G.T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theoret. Biol.*, 29:471–482, 1970.

[35] J.J. Green. Approximation with the radial basis functions of Lewitt. In J. Leversley, I. Anderson, and J.C. Mason, editors, *Algorithms for Approximation IV*, pages 212–219. University of Huddersfield, Huddersfield, UK, 2002.

[36] M.I. Gürelli and L. Onural. On a parameter estimation method for Gibbs-Markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:424–430, 1994.

[37] X. Guyon. *Random fields on a network. Modeling, statistics, and applications.* Spring Verlag, New York, NY, probability and its applications edition, 1995.

[38] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[39] G.T. Herman. *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*. Academic Press, New York, 1980.

[40] G.T. Herman. Algebraic reconstruction techniques in medical imaging. In C.T. Leondes, editor, *Medical Imaging, Systems Techniques and Applications - Computational Techniques*, pages 1–42. Gordon and Breach Science Publishers, Amsterdam, 1997.

[41] G.T. Herman. *Geometry of Digital Spaces*. Birkhäuser, Boston, MA, 1998.

[42] G.T. Herman, A.R. De Pierro, and N. Gai. On methods for maximum a posteriori image reconstruction with a normal prior. *J. Visual Comm. Image Represent.*, 3:316–324, 1992.

[43] G.T. Herman and A. Kuba, editors. *Discrete Tomography: Foundations, Algorithms and Applications*. Birkhäuser, Boston, 1999.

[44] G.T. Herman and L.B. Meyer. Algebraic reconstruction techniques can be made computationally efficient. *IEEE Trans. Med. Imag.*, 12:600–609, 1993.

[45] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artif. Intell.*, 17:185–203, 1981.

[46] E. Jaynes. On the rationale of maximum-entropy methods. *Proc. IEEE*, 70:939–952, 1982.

[47] I.M. Jacobs J.M. Wozencraft. *Principles of Communication Engineering.* Waveland Press, Prospect Hights, IL, 1990.

[48] R.P. Kanwal. *Generalized Functions: Theory and Applications*. Birkhäuser, Boston, 2004.

[49] G. Karp, editor. *Cell and Molecular Biology: Concepts and Experiments*. John Wiley and Sons, Inc., New York, 1996.

[50] P.E. Kinahan, S. Matej, J.S. Karp, G.T. Herman, and R.M. Lewitt. A comparison of transform and iterative reconstruction techniques for a volume-imaging PET scanner with a large acceptance angle. *IEEE Trans. Nucl. Sci.*, 42:2281–2287, 1995.

[51] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[52] C. Kittel. *Introduction to Solid State Physics*. John Wiley & Sons, New York, 7th edition, 1996.

[53] K. Kyte, editor. *Structure in Protein Chemistry*. Garland Publishers, New York, 1995.

[54] R.M. Lewitt. Multidimensional digital image representations using generalized Kaiser-Bessel window functions. *J. Opt. Soc. Amer. A*, 7:1834–1846, 1990.

[55] S.Z. Li. Invariant surface segmentation through energy minimization with discontinuities. *Int. J. Comput. Vision*, 5:161–194, 1990.

[56] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, Tokyo, 2001.

[57] H.Y. Liao and G.T. Herman. Reconstruction of label images from a few projections as motivated by electron microscopy. In *Proc. IEEE* 28^{th} *Annual Northeast Bioeng. Conf.*, pages 205–206, Philadelphia, PA, 2002. IEEE.

[58] H.Y. Liao and G.T. Herman. Tomographic reconstruction of label images from a few projections. *Electronic Notes in Discrete Mathematics*, 12, 2003.

[59] H.Y. Liao and G.T. Herman. Automated estimation of the parameters of Gibbs priors to be used in binary tomography. *Discrete Appl. Math.*, 139:149–170, 2004.

[60] H.Y. Liao and G.T. Herman. A method for reconstructing label images from a few projections, as motivated by electron microscopy. In *Proc. IEEE International Symposium on Biomedical Imaging*, pages 551–554, Arlington, VA, 2004. IEEE.

[61] H.Y. Liao and G.T. Herman. Discrete tomography with a very few views, using Gibbs priors and a marginal posterior mode. *Electronic Notes in Discrete Mathematics*, 20, 2005.

[62] H.Y. Liao and G.T. Herman. Reconstruction by direct labeling in discrete tomography, using Gibbs priors and a marginal posterior mode approach. In *Proc. IEEE* 31^{st} *Annual Northeast Bioeng. Conf.*, pages 134–135, Hoboken, NJ, 2005. IEEE.

[63] H.Y. Liao and G.T. Herman. A coordinate ascent approach to tomographic reconstruction of label images from a few projections. *Discrete Appl. Math.*, to appear.

[64] H.Y. Liao and G.T. Herman. A method for reconstructing label images from a few projections, as motivated by electron microscopy. *Ann. Oper. Res.*, to appear.

[65] D.G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, New York, 1969.

[66] R. Marabini, G.T. Herman, and J.M. Carazo. 3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs). *Ultramicrosc.*, 72:53–65, 1998.

[67] R. Marabini, E. Rietzel, R. Schröder, G.T. Herman, and J.M. Carazo. Three-dimensional reconstruction from reduced sets of very noisy images acquired following a single-axis tilt schema: Application of a new three-dimensional reconstruction

algorithm and objective comparison with weighted backprojection. *J. Struct. Biol.*, 120:363–371, 1997.

[68] S. Matej, G.T. Herman, T.K. Narayan, S.S. Furuie, R.M. Lewitt, and P. Kinahan. Evaluation of task-oriented performance of several fully 3–D PET reconstruction algorithms. *Phys. Med. Biol.*, 39:355–367, 1994.

[69] S. Matej and R.M. Lewitt. Efficient 3D grids for image reconstruction using spherically-symmetric volume elements. *IEEE Trans. Nucl. Sci.*, 42:1361–1370, 1995.

[70] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[71] D.C. Montgomery and G.C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, New York, third edition, 2002.

[72] D. Murray and B. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:220–228, 1987.

[73] T.K. Narayan. *Evaluation of Image Reconstruction Algorithms by Optimized Numerical Observers*. PhD thesis, University of Pennsylvania, Philadelphia, USA, 1998.

[74] A. Papoulis. *Probability and Statistics*. Prentice-Hall, Inc., New Jersey, USA, 1990.

[75] C. Pellot, A. Herment, M. Sigelle, P. Horain, H. Maitre, and P. Peronneau. A 3D reconstruction of vascular structures from two X-ray angiograms using an adapted simulated annealing algorithm. *IEEE Trans. Med. Imag.*, 13:48–60, 1994.

[76] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, New York, 1993.

[77] M. Rademacher, T. Wagenknecht, A. Verschoor, and J. Frank. Three-dimensional reconstruction from a single-exposure, random conical tilt series applied to the 50 S ribosomal subunit of escherichia coli. *J. Microsc.*, 146:113–136, 1987.

[78] J. Rissanen. A universal prior for integers and estimation by minimal description length. *Ann. Stat.*, 11:416–431, 1983.

[79] G.O. Roberts and R.L. Tweedie. Bounds on regeneration times and convergence rates for Markov chains. *Stoch. Proc. Appl.*, 80:211–229, 1999.

[80] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation oerations. *IEEE Trans. Syst. Man Cyber.*, 6:420–433, 1976.

[81] J.S. Rosenthal. Minorization conditions and convergence rates for Markov Chain Monte Carlo. *J. Amer. Statist. Assoc.*, 90:558–566, 1995.

[82] S.W. Rowland. Computer implementation of image reconstruction formulas. In G.T. Herman, editor, *Image Reconstruction from Projections: Implementation and Applications*, pages 9–79. Springer-Verlag, Berlin, 1979.

[83] H. Rue. New loss function in Bayesian imaging. *J. Am. Stat. Assoc.*, 90:900–908, 1995.

[84] P.K. Sahoo, S. Soltani, A.K.C. Wong, and Y.C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41:233–260, 1996.

[85] S.H.W. Scheres, R. Marabini, S. Lanzavecchia, F. Cantele, Rutten T., S.D. Fuller, J.M. Carazo, R.M. Burnett, and C. San Martin. Classification of single projection reconstructions for cryo-electron microscopy data of icosahedral viruses. *J. Struct. Biol.*, 151:79–91, 2005.

[86] L.A. Shepp and Y. Vardi. Maximum likelihood reconstruction in positron emission tomography. *IEEE Trans. Med. Imag.*, 1:113–122, 1982.

[87] C.O.S. Sorzano, R. Marabini, G.T. Herman, Y. Censor, and J.M. Carazo. Transfer function restoration in 3D electron microscopy via iterative data refinement. *Phys. Med. Biol.*, 49:509–522, 2004.

[88] S. Sotthivirat and J.A. Fessler. Image recovery using partitioned-separable paraboloidal surrogate coordinate ascent algorithms. *IEEE Trans. Image Process.*, 11:306–317, 2002.

[89] H. Stark and J.W. Woods, editors. *Probability and Random Processes with Applications to Signal Processing*. Prentice Hall, Upper Saddle River, NJ, third edition, 2001.

[90] S. Subramaniam and J. L.S. Mine. Three-dimensional electron microscopy at molecular resolution. *Annu. Rev. Biophys. Biomol. Struct.*, 33:141–55, 2004.

[91] L. Tierney. Markov chains for exploring posterior distributions. *Ann. Stat.*, 22:1701–1762, 1994.

[92] M. Van Heel. Angular reconstitution: a posteriori assignment of projection directions for 3D reconstructions. *Ultramicrosc.*, 21:111–24, 1987.

[93] E. Vardi, G.T. Herman, and T.Y. Kong. Speeding up stochastic reconstructions of binary images from limited projection directions. *Linear Algebra Appl.*, 339:75–89, 2001.

[94] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer, Berlin, second edition, 2003.

[95] C.S. Won and H. Derin. Unsupervised sgmentation of noisy and textured images using Markov random fields. *CVGIP: Graph. Models Image Proc.*, 54:308–328, 1992.

[96] J. Zhang. The mean field theory in EM procedures for Markov random fields. *IEEE Trans. Image Process.*, 40:2570–2583, 1992.

[97] X. Zhou, N.A. Obuchowski, and D.K. McClish. *Statistical Methods in Diagnostic Medicine*. John Wiley & Sons, New York, 2002.

[98] J.P. Zubelli, R. Marabini, C.O.S. Sorzano, and G.T. Herman. Three-dimensional reconstruction by Chahine's method from electron microscopic projections corrupted by instrumental aberrations. *Inv. Prob.*, 19:933–949, 2003.

Index