Some Estimation and Restoration Techniques for Statistical Image Analysis

Graeme Ernest Barclay Archer

A dissertation submitted to the

University of Glasgow

for the degree of

Doctor of Philosophy

Department of Statistics

December 1994.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Abstract

Some Estimation and Restoration Techniques for Statistical Image Analysis

Graeme Ernest Barclay Archer
Department of Statistics
University of Glasgow
Submitted for the degree of Doctor of Philosophy
December 1994

This thesis is concerned with statistical image analysis: the estimation of parameters within image models, and how to produce restorations of degraded scenes which are the most probable *given* the parameter estimates and the data. We develop algorithms for estimation within hierarchical and empirical Bayesian models, and compare results with non–Bayesian methods. The empirical behaviour of parameter estimates under different algorithms are studied in a simulation exercise and compared with their theoretical behaviour. We sample realisations from Markov random fields using the Metropolis algorithm, and propose a resampling technique to assess convergence. An alternative to the EM algorithm (EMA), the Image Space Reconstruction algorithm (ISRA), is extended and compared with the EMA. A technique for increasing the rate of ISRA–convergence is investigated. Finally, an adaption of a method to prevent over–smoothing of image discontinuities is fully automated. The effect of user–supplied parameter values on the image restoration quality is investigated via a simulation study; the effects are found to be negligible.

Acknowledgements

This work would not have been possible without the patient support and guidance of my supervisor, Professor D. Michael Titterington. It is a pleasure to record my thanks to him for all of his assistance, which started long before the work on my Ph.D. : not only was he one of my undergraduate lecturers, he was also responsible for providing me with my first paid employment! To Dr. James W. Kay, who was my first supervisor and who first interested me in Image Analysis, I owe another large debt of gratitude.

I am very grateful to the Science and Engineering Research Council, whose financial support made my postgraduate study possible.

My family, who doggedly read every statistical word I write, deserve a special vote of thanks. It must seem a very long time to my parents, Ernie and Eveline, since they enrolled me in school, perhaps little knowing for how long I would remain within full time education (twenty years at the last count!). I thank them, and my grandparents Rhoda and Barclay Robertson, for all their love and support over the years.

During my years at Glasgow University several special friendships began. In particular, I would like to thank and send love to Andrew Noble, Eileen Wright, Catriona Hayes, Rodney Wolff, and in particular Charles Neeson.

The thesis is dedicated to Charles, and my family.

Glasgow, December 1994.

Contents

Chapter 1

Introduction

1.1 Image Analysis

Our work is concerned with statistical image analysis – that is, the restoration of noisy, blurred images, using sound statistical techniques.

> *'and what is the use of a book,' thought Alice, 'without pictures or conversation?'*[1]

In accordance with these sentiments we present and discuss many pictures in the course of our dissertation, to highlight the effect of the techniques we propose.

The term "image analysis" describes the restoration and interpretation of remotely sensed–data, for example: ultra–sonic scans of patients' internal organs, or satellite data concerning land usage. Long regarded as a branch of digital image processing by engineers, who applied various deterministic filtering techniques to the noisy, blurred data (see, for example, [40]), there was an explosion of statistical interest in the subject in the mid–to–late 1980s, due largely to the seminal papers of Geman and Geman ([32]) and Besag ([6]). We summarise their main results below, but their real value was to place the various problems of image

[1] Lewis Carroll, "Alice's Adventures in Wonderland".

analysis (from simple processing to classification, from satellite data to emission tomography) firmly within the paradigm of Bayesian statistical inference, thus allowing the application of sound estimation and inferential techniques.

A further dramatic result has been the re–emergence of Bayesianism in many other areas of statistics, due to the technology developed for image analysis (mainly the practice of using Markov Chain Monte Carlo methods to sample from previously intractable posterior distributions).

To motivate the thesis, we here present (without any statistical detail) a real example of image processing. It is with the processing of images, rather than the classification of their contents, that we shall be concerned. The left hand side of Figure 1.1 is an ultra–sonic scan of a human heart, received from Stobhill Hospital in Glasgow. The data is very distorted, due to blurring and noise in the recording process. Using one of the techniques of the thesis (that of Chapter 5) we can form an estimate of what the true picture of the heart should look like; the result can be seen on the right hand side of the figure, showing a clear improvement over the data in terms of information displayed.

We now present an overview of the rest of this chapter: in the next section, we introduce the notation we will need; following this we provide a short history of the subject, in order to explain some concepts to which we shall subsequently often refer, as well as giving our work its historical setting. Also in this section we detail our attraction to the Bayesian paradigm for statistical inference. Finally, we provide a summary of the rest of the thesis, mentioning the main results from each of the chapters.

1.2 Notation

We follow closely the notation of Besag in [6]. Let S be a two–dimensional array of pixels, labelled with integers $\{1, 2, \ldots, n\}$. We assume that the true, unknown image, x, is a realisation on S of a random vector $X = \{X_1, X_2, \ldots, X_n\}$. The data, y, is a realisation of a random vector $\{Y_t : t \in T\}$, caused by a stochastic degradation D of the true image:

$$D : X \longrightarrow Y.$$

Throughout this thesis, $S \equiv T$, but this is not necessary in general. In Emission Tomography, for example (see [80]), the data space is (usually) a two–dimensional pixellated region, while the image space is the shape of the body–part under investigation.

We use $p(\cdot)$ to denote a generic probability distribution, or density function, according to whether the random variable (r.v.) under consideration is discretely or continuously valued.

We now make two assumptions:

ONE. The random variables $\{Y_i\}$ are conditionally independent and have the same conditional density function, dependent on X. Thus the joint density function of y given x, i.e. the likelihood function of the data, is

$$p(y \mid x) = \prod_{i=1}^{n} p(y_i \mid x).$$

In fact, in all the applications we discuss, the dependency between y_i and the true image extends only to a subset of S, B_i, say. The size and shape of B_i is determined by the particular application: see Section 2.2 for details. □.

TWO. The true image x is a realisation of a locally dependent Markov random field (m.r.f.). Let Ω be the sample space for X. The m.r.f. assumption requires that the following two conditions are met.

1. $p(X = x) > 0$, for all $x \in \Omega$,

2. $p(X_i = x_i \mid X_j = x_j, j \neq i) = p(X_i = x_i \mid X_j = x_j, j \in \delta_i)$.

We say δ_i is the "m.r.f. neighbourhood" of pixel i; what makes the use of m.r.f.'s so appealing is that by defining δ_i in a local manner we induce a unique joint distribution for $p(x)$ (see [5]). We have only to ensure that the neighbourhood structure is defined in such a way that $i \in \delta_j \Leftrightarrow j \in \delta_i$. By "local manner" we mean that the neighbourhood of i should consist of those pixels which are "geographically" close. We call a first–order neighbourhood the set of pixels to the immediate north,south, east and west of i. The second–order neighbourhood consists of these four pixels plus the four diagonal pixels adjacent to i. □.

Both the data–model $p(y \mid x)$ and the image–model $p(x)$ will contain *parameters*, the estimation of which will be our concern in Chapters 2 and 3. We defer discussion of them until that time. For the moment, note that the two models can be combined with Bayes' theorem (see, for example, [16]), to form the posterior distribution of x given y,

$$p(x \mid y) \propto p(y \mid x)p(x). \tag{1.1}$$

If we wished the most probable estimate of x, the *maximum a posteriori* (m.a.p.) estimate, we would maximise (1.1) with respect to (w.r.t.) x.

1.3 Seminal work

We here discuss two papers which in our opinion have greatly shaped the direction of research in this area. Of course most new papers are a synthesis of work that has gone before; but occasionally one emerges which seems to exert a particulary powerful influence on subsequent research effort. That of Geman and Geman ([32]) could perhaps be such a paper. The authors made the comparison between images and statistical mechanics, and used methods from that area to solve some of the problems in image restoration. In particular, they introduced the techniques of simulated annealing (see [63], where the concept of annealing is applied to problems of optimisation) and Gibbs sampling.

Later (in Section 2.3.1) we detail the severe computational problems involved with maximising $p(x \mid y)$; the Gemans proposed maximising $\{p(x \mid y)\}^{1/t}$, where t is a control parameter corresponding to the temperature of a physical system : an annealing schedule is the reduction of temperature (slowly) over time, producing realisations that settle upon the mode of the posterior density.

The authors formulated the Gibbs sampler (a variation of the Metropolis algorithm [70]), for sampling from the posterior density at a particular temperature. Basically, each site of the graph is visited infinitely often (in practice, each site is visited a large number of times). A new value for the site is chosen from the local conditional probability distribution. For example, if $\Omega = \{1, 2, \ldots, c\}$ and at a particular iteration we have selected site l, then we set $x_l = f$, where $f \in \Omega$, with probability $p(x_l = f \mid y, x_{S\backslash l})$. (The set $S\backslash l$ is the set of sites omitting site l.) The sequence of realisations thus formed, say $x^{(1)}, x^{(2)}, \ldots$, forms a Markov Chain

with equilibrium distribution $p(x \mid y)$. In Section 2.3.2 we discuss the difficulties of assessing when equilibrium has been reached; Gelman ([31]) discusses the relationship between Gibbs sampling and the non–iterative techniques of rejection and importance sampling. Convergence difficulties notwithstanding, the use of the Gibbs sampler (and variants) to sample from complicated multi–dimensional integrals has freed much of Bayesian statistics from the charge frequently levelled against it: that philosophically it is agreeable and elegant, but that any feasible posterior distribution function is almost certainly analytically intractable. With the Gibbs sampler, one no longer has to approximate the desired posterior with a manageable distribution.

The work of Julian Besag, in [5] and [6], was highly successful in popularising the use of m.r.f.s to describe the unknown image. In [6] he proposed an iterative method for estimating the mode of $p(x \mid y)$, less computationally demanding than simulated annealing. The method was called Iterated Conditional Modes, since it involves visiting each site in turn, and at site l it chooses $\hat{x}_l$ to maximise $p(x_l \mid y, \hat{x}_{S\setminus l})$. Using Bayes' Theorem, Assumptions ONE and TWO, and allowing that $p(y_i \mid x) = p(y_i \mid x_i)$ (indicating that no blurring occurs between the image–space and the data–space), we see that $p(x_l \mid y, \hat{x}_{S\setminus l}) \propto p(y_l \mid x_l) \times p(x_l \mid \hat{x}_{\delta_l})$, and so the method is particularly simple to implement.

In the same paper, Besag proposed the pseudo–likelihood estimator of the parameters in the prior distribution (we make use of this method in Chapter 2), which again exploits the local dependency structure of m.r.f.s to choose the parameter value which maximises $\prod_l p(x_l \mid x_{\delta_l})$ w.r.t. the parameter of interest.

An overview of the random field models used for image processing, and the

techniques used to sample from them, is provided by Dubes and Jain in [23].

1.3.1 Why *Bayesian* Image Analysis?

We are attracted to the Bayesian paradigm for all applications of statistical estimation and inference, and therefore also for image analysis, because we believe that any attempt to render scientific investigation as somehow "objective" is doomed to failure. The assumptions of the experimenter play a vital role in the direction of his or her research and Bayesianism forces these assumptions to be modelled explicitly, rather than allowing them to be swept under a frequentist carpet. Thus, rather than finding the essential subjectivity of the Bayesian approach a *weakness*, we regard it as the system's major strength. As de Finetti says ([27]):

> *in deductive logic, if one utilises only part of the hypothesis, the set of conclusions will be smaller but still correct; whereas in inductive logic, if one neglects a part of the information (unless it happens to be irrelevant) the conclusion drawn is incorrect.*

For a defence of the subjectivist interpretation of probabilities, and the demonstration that a coherent individual's belief about an experiment can be (should be) described by a probability distribution, see [27], already cited. In [101], the authors provide a measure of the disagreement between the *a priori* beliefs of an individual, and the results of an experiment. Finally, for a splendid rebuttal of frequentist statistics and a lucid explanation of the role of Bayes Theorem in scientific reasoning, see the book by Howson and Urbach, [55].

1.4 Summary of results

In Chapter **2**, we attempt to estimate parameters and restore the image "simulataneously" (albeit iteratively), in an hierarchical Bayesian framework. We present four estimation–restoration algorithms, with varying degrees of complexity in the underlying assumptions, and find fairly good image restoration but less successful parameter estimation. We simulate what we hope are true realisations from m.r.f.'s, in order to judge how well parameter estimation is proceeding, and detail some of the difficulties inherent in this area. We present a resampling technique to help judge when these simulations are successful. Finally, we carry out a simulation exercise to better examine the effects of the two most successful algorithms, and conclude that the simpler of the two is the more efficient.

In Chapter **3** we turn our attention to the empirical Bayesian paradigm: we estimate the parameters by maximum likelihood, and "plug–in" these estimates to obtain what we hope is the m.a.p. estimate of the true image. We provide an iterative procedure to reach the m.a.p. restoration from the empirical Bayes one, and compare our results with other standard "plug–in" estimators from the literature, as well as two optimal ones.

In numerical work, and in contrast to Chapter **2**, we see improvement in the parameter estimation, and less success with image restoration. An examination of the joint and some profile likelihood surfaces suggests a reason for why this may be.

The work of the bulk of Chapter 3 appears in [2].

Chapter **4** views the image restoration problem as belonging to the class

of *incomplete data problems.* Such motivation leads to consideration of the Expectation–Maximisation algorithm (EMA), particularly as expounded by Vardi and Lee ([98]). We discuss another algorithm, similar in mechanism, but with simpler motivation, called the iterative Image Space Reconstruction algorithm (ISRA; see [15]), and compare the two in a wide variety of situations. For detailed comparison we turn to image deblurring, finding excellent visual results with both, with slight evidence to suggest quicker convergence for the EMA, but better results for the ISRA. We then attempt to increase the rate of convergence of the ISRA but here the results are less impressive, and indeed we have been unable to prove convergence of the adapted algorithm.

The work of this chapter appears in [3].

Finally, Chapter **5** is a contribution to the problem of discontinuity detection. Since in many areas of application it is the detection of discontinuities, or edges, which is of prime interest, it is important to have image restoration algorithms which do not oversmooth such features. We take an algorithm by Abdallah and Kay ([9]), which relied on user–supplied parameter values, and fully automate it, with some success. We compare three different edge detection methods, and discuss why the technique is not more successful. To end, we explain how we feel the technique should be advanced.

Figure 1.1: An example of image processing - on the left, a distorted picture of a human heart; on the right, a restoration.

Chapter 2

An hierarchical Bayesian approach to simultaneous parameter estimation and image restoration

2.1 Introduction

In our problem of "blind" image restoration, we are faced with the difficulty of not only restoring an unseen image, x, but also of estimating the unknown parameters within the image models. In this chapter, we wish to remain as far as possible within the full Bayesian paradigm, to which we have admitted our attraction. The paradigm is straightforward: express all of our uncertainty in terms of probabilities, then manipulate into existence, using the calculus of probabilities, the appropriate posterior probability distribution. Let us now attempt to do this.

First of all, we observe a set of records, y, which we presume to depend on an unobserved true image, x, which is being corrupted by a point spread function (p.s.f.) V (assumed known) and unknown noise process which has variance ϕ.

Thus we will require to specify a form for

$$p(y \mid x, \phi). \tag{2.1}$$

Note that we are assuming the p.s.f. known – of course to be fully Bayesian we should assign V a probability distribution, and one can of course do this ([46],[105]). However, in many real–life image processing applications the elements of V can be accurately estimated via some off–line experiment and so we feel not too guilty about assigning V the privilege of constancy. There is a "blurring" matrix H corresponding to every V – discussion of these structures is deferred until our experimental sections.

We must further express our uncertainty about x in terms of a probability distribution – we suppose that this depends on one unknown parameter, β, and thus we will need to specify

$$p(x \mid \beta). \tag{2.2}$$

Thus we have introduced 2 parameters ϕ, β which require estimation – strictly speaking, in the Bayesian setting, they are random variables and must be assigned distributions:

$$p(\phi \mid \Phi) \tag{2.3}$$

and

$$p(\beta \mid \Theta). \tag{2.4}$$

Later we shall specify the parameters Φ, Θ; here let us say that we will assume them known constants and such that β and ϕ can be assumed *a priori* independent. We have now specified all the objects about which we are uncertain and can proceed to make inference concerning them by using Bayes' Theorem to combine

(2.1,2.2, 2.3,2.4):

$$p(x, \phi, \beta \mid y) \propto p(y \mid x, \phi) \times p(x \mid \beta) \times p(\phi) \times p(\beta) \tag{2.5}$$

Hierarchical modelling

This form of Bayesian modelling, the most "correct", was given a forceful exposition in the 1970s by Lindley and Smith (see [65],[85]) who found Bayesian estimates in the Normal linear model with lower mean square errors (m.s.e.'s) than the standard least square estimates (l.s.e.'s). A model with n hierarchies is called an n–stage model; we restrict attention to the case of a completely specified model after two stages, i.e. $n = 2$. Parameters ϕ and β we call *hyper–parameters*; Lindley and Smith, who attribute the "hyper" terminology to I.J.Good, assumed ϕ known, or estimated it in the "standard" fashion: that is, as some residual sum–of–squares divided by appropriate degrees of freedom. Although we prefer to assign a prior distribution, it will be seen that with prior ignorance this does indeed lead to the usual estimate of variance.

More recently, hierarchical models have been used very successfully in the construction of medical expert systems (for example, see [37],[38],[88]). From this field has come the practice of drawing a *directed acyclic graph* (d.a.g.) to represent the conditional probability distributions. Such a graph can greatly simplify understanding of complicated problems; a d.a.g. of our model, assuming β, ϕ each depend on 2 parameters, appears in Figure 2.1. We follow the convention outlined in [103] and use round nodes to represent unobserved random variables (our true image and the model parameters); square nodes signify observed r.v.s

(i.e. the data); double square nodes represent fixed quantities in prior distributions; arrows denote dependencies between the probability distributions. This is known as a d.a.g. because the arrows denote the directed Markov assumption: for example, *given* x, then we see y is independent of β, α, ν.

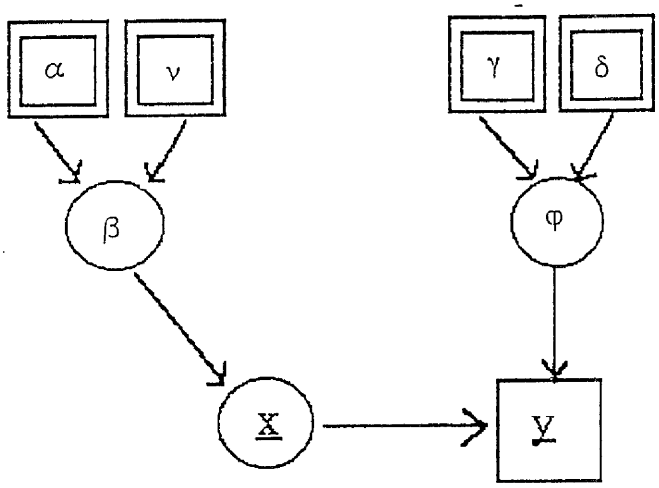


Figure 2.1: A directed acyclic graph representation of the Markovian structure of the probability distributions under consideration

More discussion of this approach to Bayesian inference can be found in [67],[54]; an important application to image analysis is in [20].

2.2 Algorithm *normal*

Introduction

Here we make the problem nearly as simple as possible, by assuming that our true unknown image has no spatial dependency structure – in other words that every pixel is stochastically independent of every other. Let the image

be composed of square pixels, arranged in n_r rows and n_c columns. If, for all $n = n_r \times n_c$ pixels, each x_i is a realisation of a $N(0, (2\beta)^{-1}))$ random variable, then straightforwardly:

$$p(x \mid \beta) \propto \beta^{n/2} \exp(-\beta \sum_{i=1}^{n} x_i^2). \tag{2.6}$$

For β, we will assume prior ignorance and assume a tractable vague distribution:

$$p(\beta) \propto \beta^{1/2}. \tag{2.7}$$

In retrospect, this is a rather odd choice of β prior, and perhaps a more appropriate one would have been $p(\beta) \propto \beta^{-1/2}$. Our choice of prior is very improper.

The set of records we will assume are generated by a Gaussian degradation, i.e.

$$Y_i \mid X, \phi \sim N(\sum_{j \in B_i} h_{ij} x_j, \phi), \tag{2.8}$$

In fact (Assumption ONE of Chapter 1) we further assume *conditional independence* of the data given the image, so that

$$p(y \mid x, \phi) = \prod_{i=1}^{n} p(y_i \mid x, \phi).$$

Discussion of H, the point spread matrix

We take the point–spread matrix H to be a row-stochastic block Toeplitz matrix in which each block is also Toeplitz. If the light from a pixel in the true image–space spreads into a $B = (2b + 1) \times (2b + 1)$ area in the data–space, then the size of the block–bandwidth of H is B.Every pixel i in the true image–space will have such a block of pixels associated with it in the data–space, and so we

say that "B_i is the blurring neighbourhood of pixel i". Further, we say that the point spread function (p.s.f.) of H has bandwidth b. The p.s.f. itself is constructed from a symmetric blurring vector v, a vector of positive components and of dimension b such that

$$\sum_{i=1}^{b} v_i = 1.$$

Then, if we denote the p.s.f. by V, we have that

$$(V)_{i,j} = v_i \times v_j \text{ for } i, j = 1, ..., b.$$

The H corresponding to V will be large, sparse and banded.

In a similar fashion as for β, we assign the following prior density to ϕ:

$$p(\phi) \propto \phi^{-1/2}. \tag{2.9}$$

Combining equations (2.6,2.7,2.8, 2.9) using Bayes' theorem, we see that the log of the posterior density function (2.5) is

$$\begin{aligned} \log(p(x, \beta, \phi \mid y)) \quad \propto \quad & -\frac{(n+1)}{2} \log \phi + \frac{(n+1)}{2} \log \beta \\ & -\beta \sum_{i=1}^{n} x_i^2 - \frac{1}{2\phi} \sum_{i=1}^{n} (y_i - \sum_{j \in B_i} h_{ij} x_j)^2. \end{aligned}$$

To find the *maximum a posteriori* estimates of x, β, ϕ, i.e. the most probable estimates of x, β, ϕ *given* the data y, we have to find the maximum of this function with respect to the 3 quantities of interest.

If we write $\mathrm{L} = \log(p(x, \beta, \phi \mid y))$, we can see that

$$\frac{\partial \mathrm{L}}{\partial \beta} = \frac{(n+1)}{2\beta} - \sum_{i=1}^{n} x_i^2 = 0 \tag{2.10}$$

when

$$\beta = \frac{(n+1)}{2 \sum_{i=1}^{n} x_i^2}, \tag{2.11}$$

and

$$\frac{\partial \mathrm{L}}{\partial \phi} = -\frac{(n+1)}{2\phi} + \frac{1}{2\phi^2}\sum_{i=1}^{n}(y_i - \sum_{j\in B_i} h_{ij}x_j)^2 = 0 \tag{2.12}$$

when

$$\phi = \frac{\sum_{i=1}^{n}(y_i - \sum_{j\in B_i} h_{ij}x_j)^2}{n+1}, \tag{2.13}$$

and

$$\frac{\partial \mathrm{L}}{\partial x_m} = -2\beta x_m + \frac{1}{\phi}\sum_{i=1}^{n} h_{im}(y_i - \sum_{j\in B_i} h_{ij}x_j) = 0 \tag{2.14}$$

when

$$x_m = \frac{1}{2\phi\beta}\sum_{i=1}^{n} h_{im}(y_i - \sum_{j\in B_i} h_{ij}x_j), \tag{2.15}$$

for $m = 1, 2, \ldots, n$.

These normal equations suggest the following algorithm for simultaneous parameter estimation and image restoration, which we call "normal", after the prior distribution chosen for x:

Algorithm **normal**

1. Choose $\hat{x}^{old}$.

2. Evaluate

$$\hat{\beta} = \frac{n+1}{\sum_{i=1}^{n}\hat{x}_i^{old^2}},$$

$$\hat{\phi} = \frac{\sum_{i=1}^{n}(y_i - \sum_{j\in B_i} h_{ij}\hat{x}_j^{old})^2}{n+1},$$

$$\hat{x}_m^{new} = (1/2\hat{\phi}\hat{\beta})\sum_{i=1}^{n} h_{im}(y_i - \sum_{j\in B_i} h_{ij}\hat{x}_j^{old}),$$

   for $m = 1, 2, ..., n$.

3. Check for convergence of $\hat{x}^{new}$: if

   YES $\longrightarrow$ **STOP**

   NO $\longrightarrow$ set $\hat{x}^{new} := \hat{x}^{old}$ and go to step 2. □

2.2.1 Experiments with Algorithm *normal*

Four test images were employed in the investigation of the effectiveness of this algorithm. Each of these images was convolved with two point spread functions (created as detailed above), and further degraded with the addition of independent Gaussian noise, mean zero, and s.d. σ. We used two values of σ, making a total of 4 test images and 4 blur/noise combinations:

Images:

I1: each pixel is a realisation from distribution (2.6), with $\beta = 0.05$.

I2: as I1, save that $\beta = 2.0$.

I5: artificial image "im.con", which contains many sharp discontinuities.

I6: artificial image "im.surfs", with *no* sharp discontinuities.

(Images **I3** and **I4** are introduced later.)

Degradations:

B1: 3×3 p.s.f.: $v = (0.3, 0.4, 0.3)^T$.

B2: 7×7 p.s.f.: $v = (0.04, 0.12, 0.18, 0.32, 0.18, 0.12, 0.04)^T$.

N1: $\sigma = 2.0$ (so $\phi_{\text{true}} = 4.0$).

N2: $\sigma = 5.0$ (so $\phi_{\text{true}} = 25.0$).

In the sequel, "I1B1N1" refers to the set of data formed by the convolution of image I1 with p.s.f. B1 and the addition of noise with level N1, and so on. In the following tables, "no.iters" refers to the number of iterations required for the

algorithm to reach convergence; convergence being assumed when

$$mse(\hat{x}^{iter} - \hat{x}^{iter-1}) = n^{-1} \sum_{i=1}^{n} (\hat{x}_i^{iter} - \hat{x}_i^{iter-1})^2 \leq \epsilon. \tag{2.16}$$

We chose a value of ϵ to be 0.5. Some examples of these test images/data can be seen in Figures 2.2 and 2.3.

Results:

I1 : $\hat{x}^{(0)} = x$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I1B1N1	0.05	4.0	349.47	5.15	10.06	4
I1B1N2	0.05	25.0	889.32	26.27	10.06	3
I1B2N1	0.05	4.0	193067.50	4.43	10.06	3
I1B2N2	0.05	25.0	55.08	25.34	10.06	2

I1 : $\hat{x}^{(0)} = y$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I1B1N1	0.05	4.0	207.96	5.13	10.06	3
I1B1N2	0.05	25.0	139.96	26.13	10.06	3
I1B2N1	0.05	4.0	138.73	4.39	10.06	2
I1B2N2	0.05	25.0	31.86	25.38	10.06	2

I2 : $\hat{x}^{(0)} = x$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I2B1N1	2.0	4.0	191.87	3.98	0.33	2
I2B1N2	2.0	25.0	1199.21	25.044	0.33	2
I2B2N1	2.0	4.0	8301.28	4.02	0.33	2
I2B2N2	2.0	25.0	51882.99	25.08	0.33	2

I2 : $\hat{x}^{(0)} = y$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I2B1N1	2.0	4.0	1191.46	4.02	0.33	3
I2B1N2	2.0	25.0	204.86	24.94	0.33	3
I2B2N1	2.0	4.0	212.23	4.00	0.33	2
I2B2N2	2.0	25.0	34.54	24.94	0.33	2

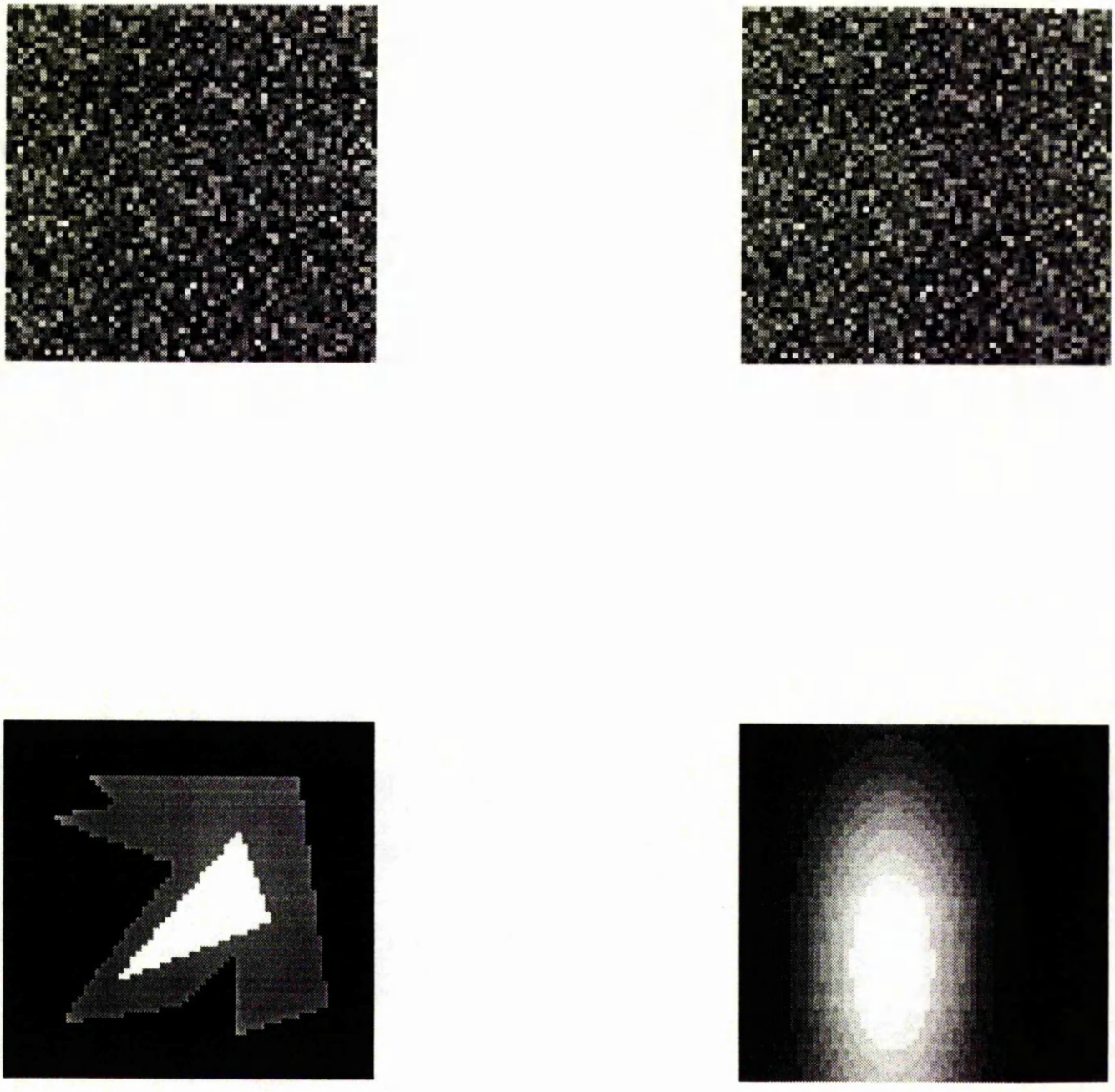


Figure 2.2: Some of the true test images. From top left to bottom right: (i) I1, (ii) I2, (iii) I5 and (iv) I6.

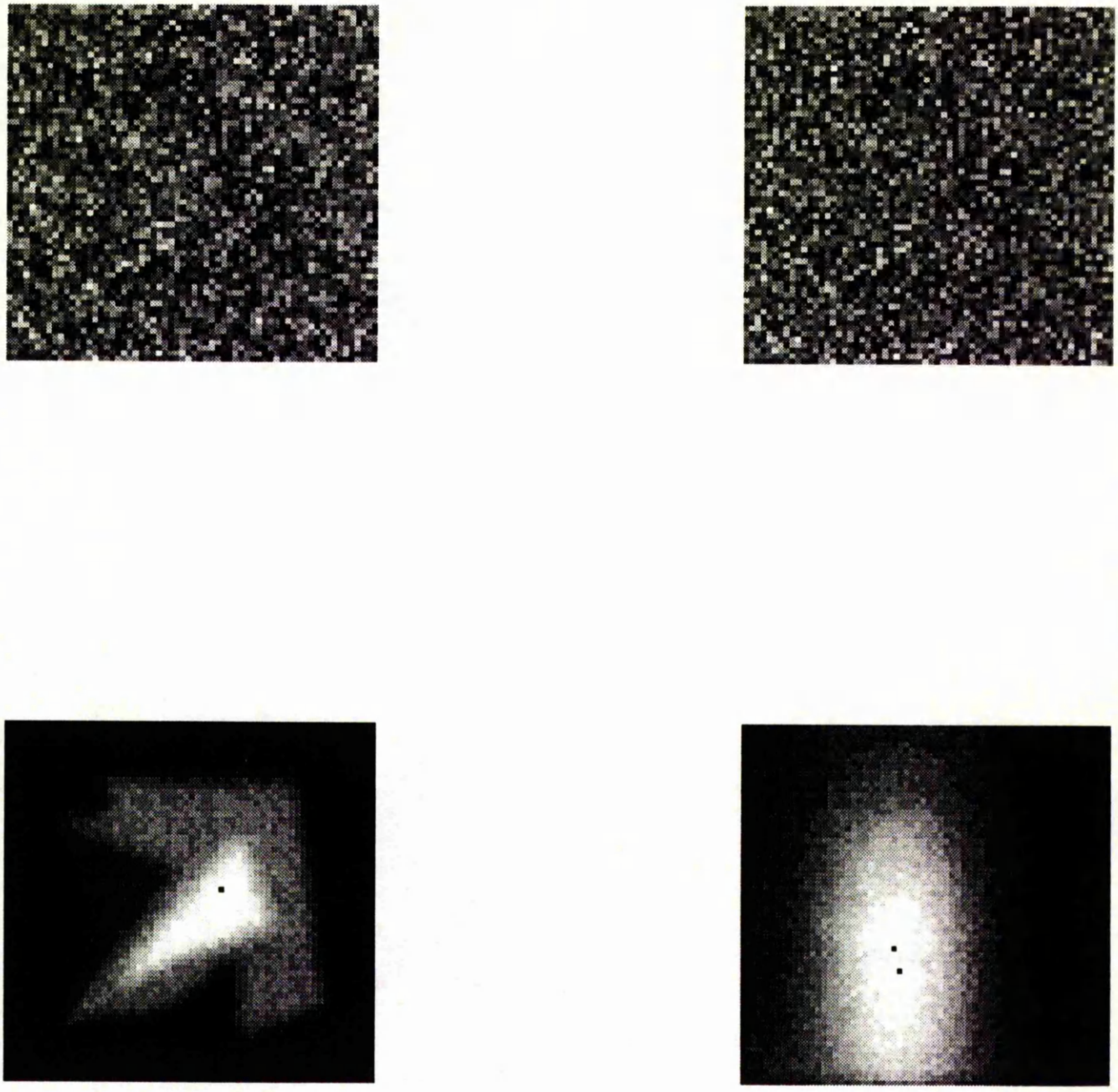


Figure 2.3: Some of the data created as detailed in Section 2.2.1. From top left to bottom right we have (i) I1B1N1, (ii) I2B1N2, (iii) I5B2N1, (iv) I6B2N2.

I5 : $\hat{x}^{(0)} = x$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I5B1N1	–	4.0	276606.90	15171.88	15226.54	39
I5B1N2	–	25.0	860373.24	15194.12	15226.54	31
I5B2N1	–	4.0	9319.21	15126.22	15226.54	6
I5B2N2	–	25.0	942.97	15144.78	15226.54	5

I5 : $\hat{x}^{(0)} = y$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I5B1N1	–	4.0	20793.54	15171.00	15226.54	39
I5B1N2	–	25.0	17.78	15153.06	15226.49	30
I5B2N1	–	4.0	92.194	15109.93	15226.54	6
I5B2N2	–	25.0	25.13	15115.81	16226.53	5

I6 : $\hat{x}^{(0)} = x$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I6B1N1	–	4.0	1444494.52	19019.69	19086.34	40
I6B1N2	–	25.0	32.98	19014.84	19086.31	31
I6B2N1	–	4.0	2717.51	18923.46	19086.34	6
I6B2N2	–	25.0	161.75	18940.41	19086.34	5

I6 : $\hat{x}^{(0)} = y$

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I6B1N1	–	4.0	775319.13	19019.63	19086.34	41
I6B1N2	–	25.0	157.05	19033.20	19086.34	30
I6B2N1	–	4.0	47428042.31	18927.15	19086.34	7
I6B2N2	–	25.0	642657.33	18955.41	19086.34	6

Discussion:

For I1,I2, images generated directly from the simple Gaussian model (2.8), the estimation of noise variance ϕ is very good, yet disastrously bad for the artificial images I5,I6. In all cases, estimation of the m.r.f. parameter β is too high – the parameter exhibits a tendency for vast inflation. We outline some reasons why this may be expected in the discussion of the next algorithm.

In plots of $\hat{\beta}$ against iteration number (Figure 2.4), some interesting patterns can be discerned, particularly for the artificial images I5 and I6. There seems strong evidence of a "blur" effect: that is, the examples employing the larger p.s.f., while not in general producing lower estimates of β than is the case for those using the smaller p.s.f., do reach convergence more quickly. Put another way, these estimates of β "blow–up" more quickly. This behaviour seems to be replicated in image I1, but not in I2. For images I5 and I6, the estimate of β is for a long time acceptably low: unfortunately m.s.e. convergence is not achieved until the estimates enlarge. Examination of the plots of $mse(\hat{x}^{iter}, \hat{x}^{iter-1})$ vs iteration number (Figure 2.5) re-inforce this pattern. Again there is a blur effect for the artificial images, in that consecutive iteration m.s.e. drops away more rapidly for those examples using the larger of the two point spread functions. The plots of variance estimate against iteration number (Figure2.6) highlight the difference between the performance of the algorithm for the simulated and the artificial images. For the Gaussian simulations, I1 and I2, estimates of ϕ converge to close to the true value, regardless of whether or not the initial estimate of the true image was the data or the truth itself. For all 4 images, there are again marked blur effects.

In all cases there is a remarkable "image—m.s.e." effect: regardless of blur, noise or starting estimate of x, each image has a value of mean square error (between the final image estimate and the truth) to which it is unshakeably drawn. This is because the very large estimates of β lead the final restorations to be equivalent to zero, regardless of the true x. This behaviour – replicated often in future algorithms – might also suggest that the most important factor in the

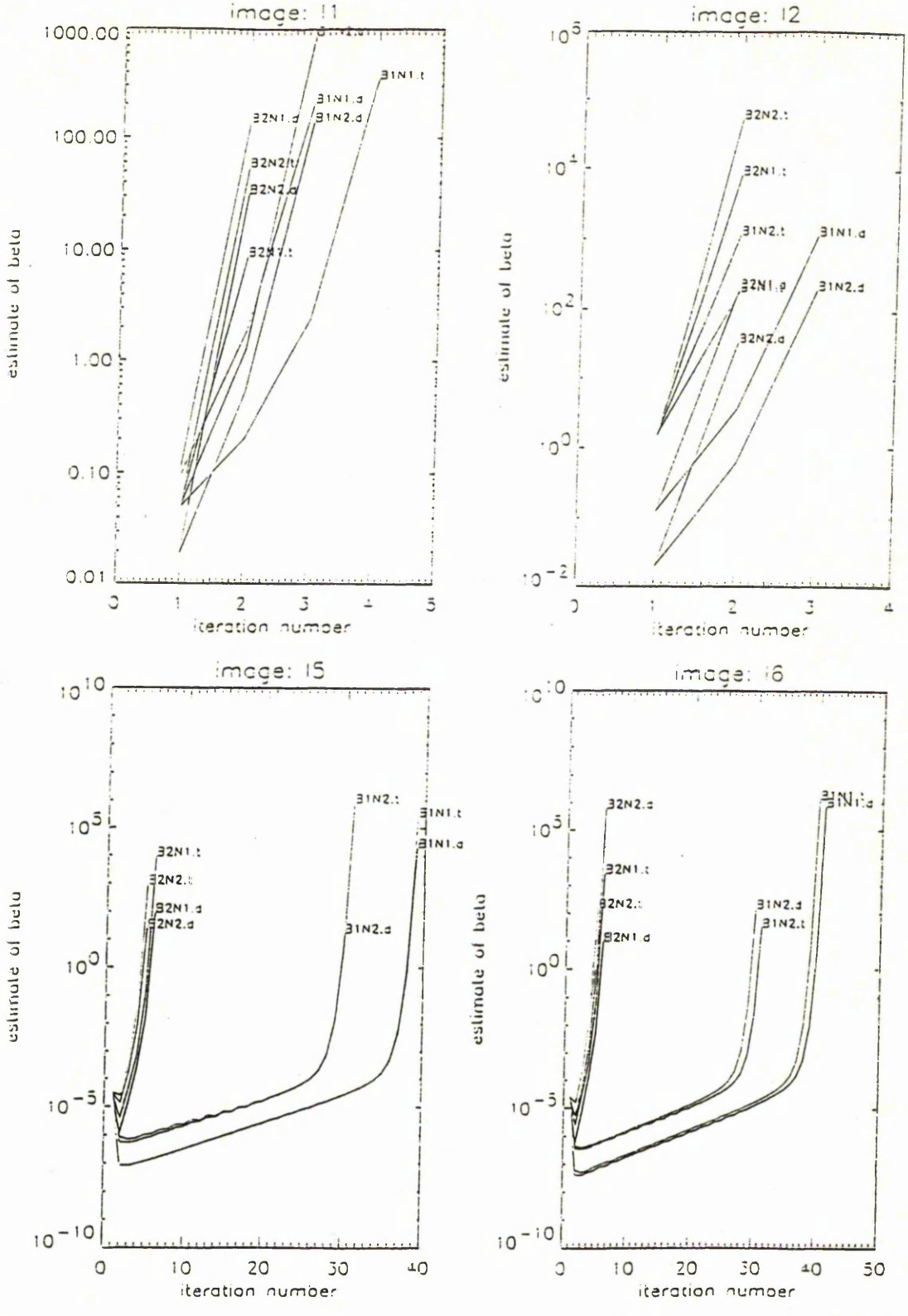


Figure 2.4: Plots of the estimate of beta against iteration number.

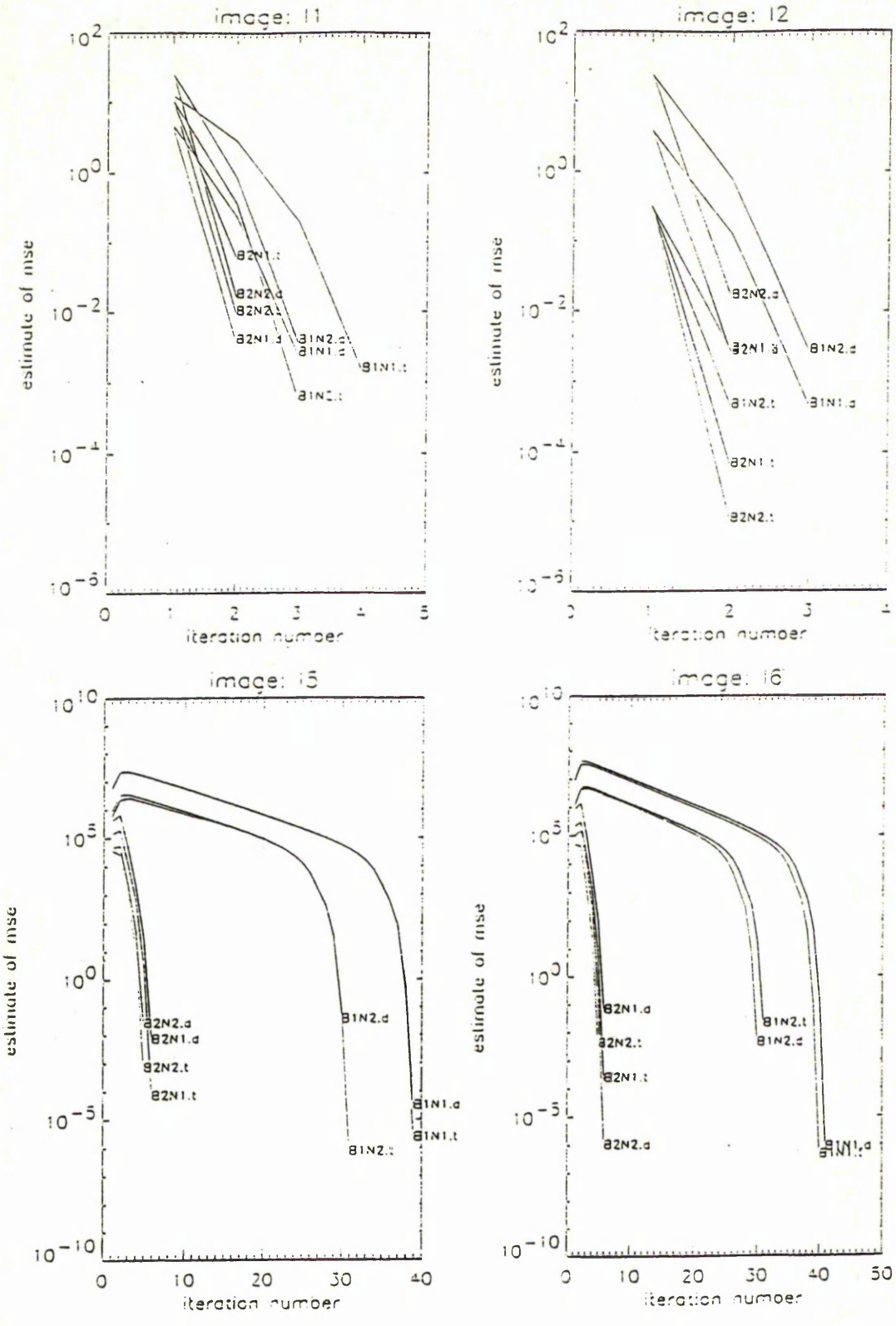


Figure 2.5: Plots of m.s.e. between successive iterations.

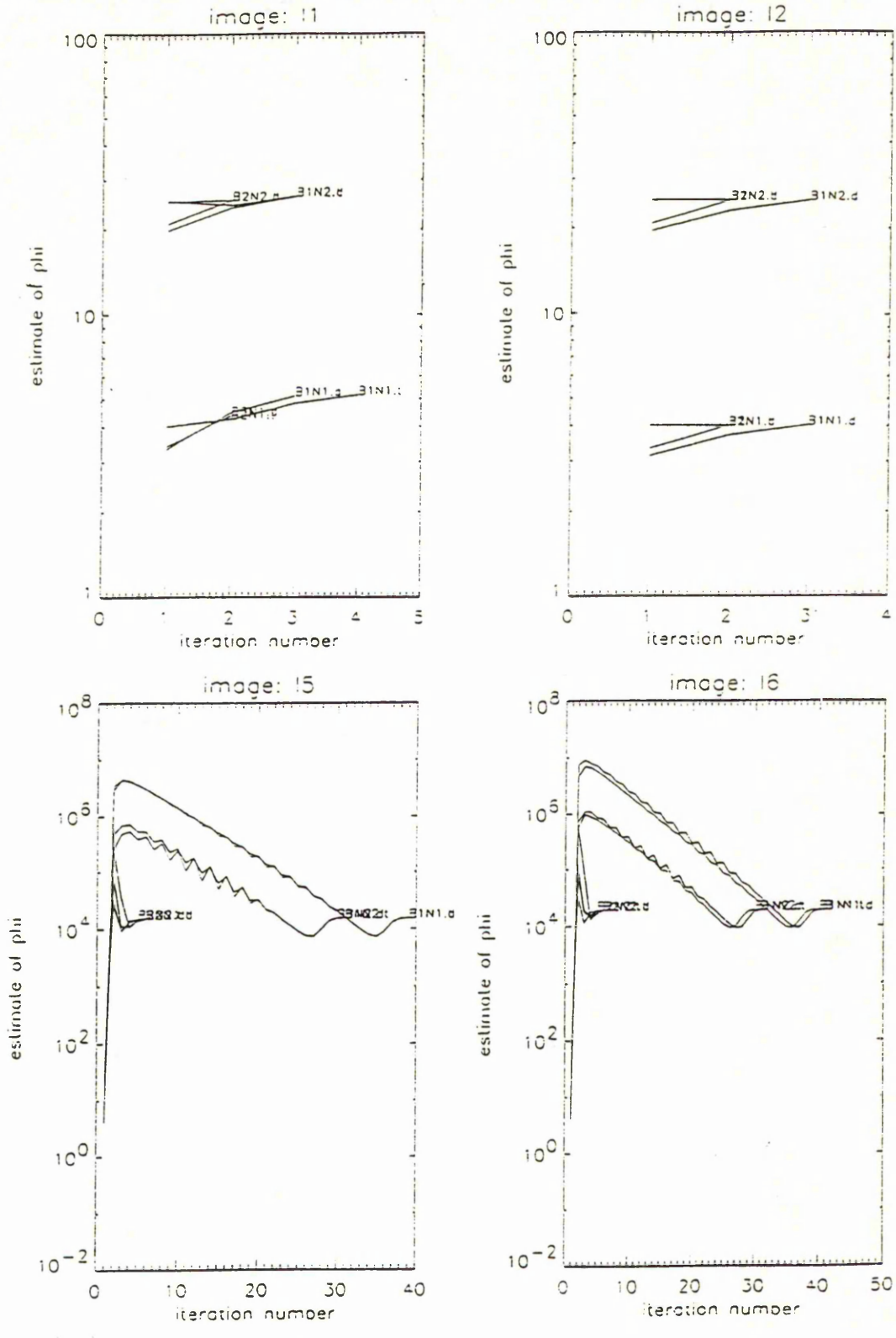


Figure 2.6: Plots of the variance estimator against iteration number.

image restoration problem is, unsurprisingly, the image itself.

Parameter estimation may be so poor due to the over–simplified assumptions we have made about x, and also because of the cyclical nature of the algorithm. At each iteration, estimates of β and ϕ are constructed, treating the current estimate of x as though it were the truth. These estimates are then used to update that of x, and so on. Small errors can therefore be promulgated through the system of formulae leading, for example, to the rapid increase in the value of the β estimate.

In Figures 2.7 and 2.8 we show some of the resulting reconstructions. We note that, although parameter estimation is generally unsuccessful, we have obtained some not unpleasing visual restorations of the image.

2.3 Algorithm *vague*

More realistic prior distributions for the unknown image

The assumption of stochastic independence between the pixel values in a real image is unappealing, if not frankly unbelievable, and it is now relaxed. Henceforward, we shall model x as though it were the realisation of a locally dependent Markov random field (m.r.f.) (Assumption TWO from Chapter 1). These structures are amenable to image processing because they allow the simple modelling of local continuity, i.e. if a pixel takes a certain value then it is more likely that pixels "close by" will be of like value, rather than radically different. Of course this simple m.r.f. specification takes no account of edges, which matter

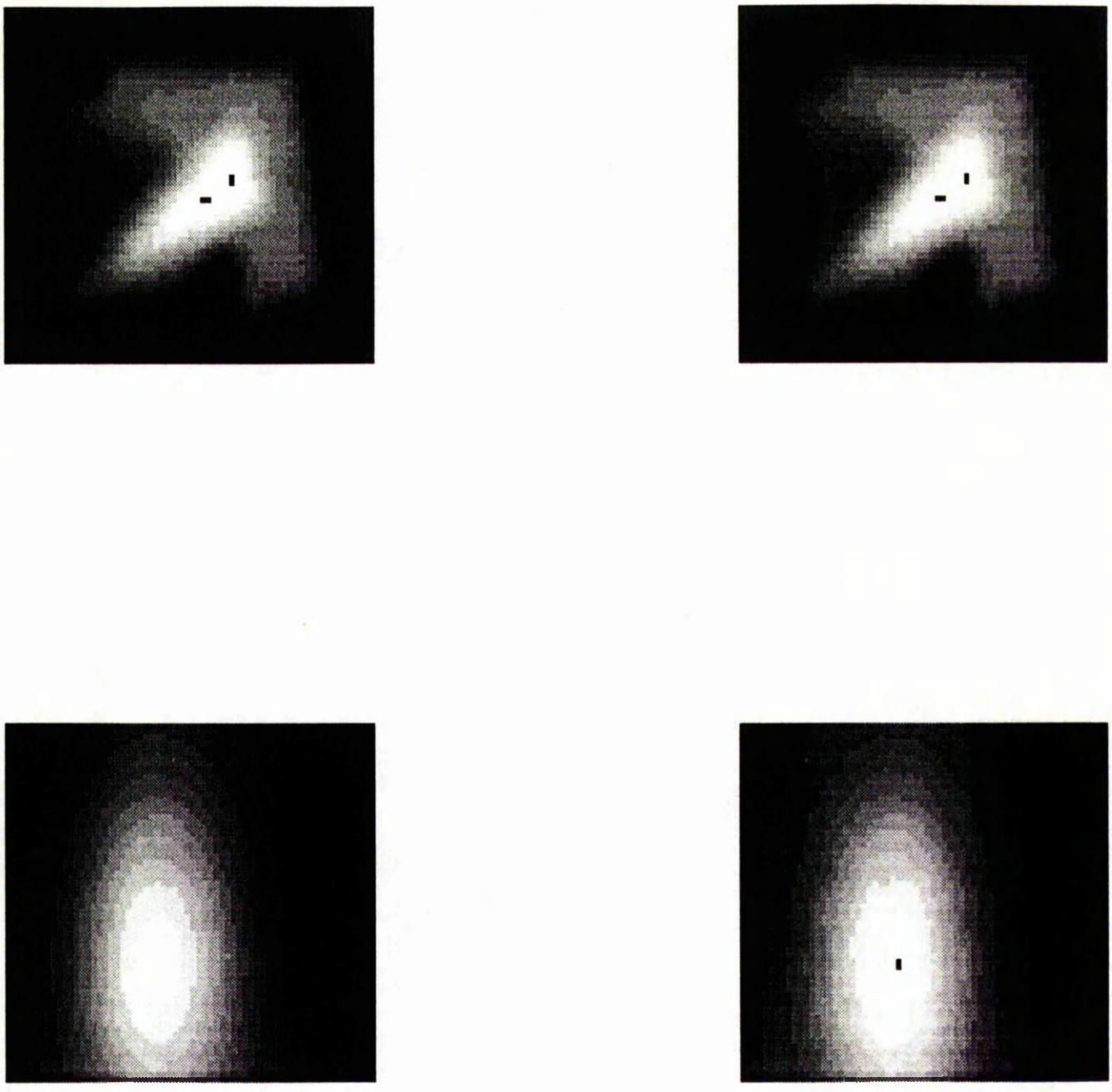


Figure 2.7: Some of the restorations from Algorithm **normal**. From top left to bottom right we have (i) I5B2N2, starting point = truth, (ii) I5B2B2, starting point = data, (iii) I6B1N1, starting point = truth, and (iv) I6B1N1, starting point = data

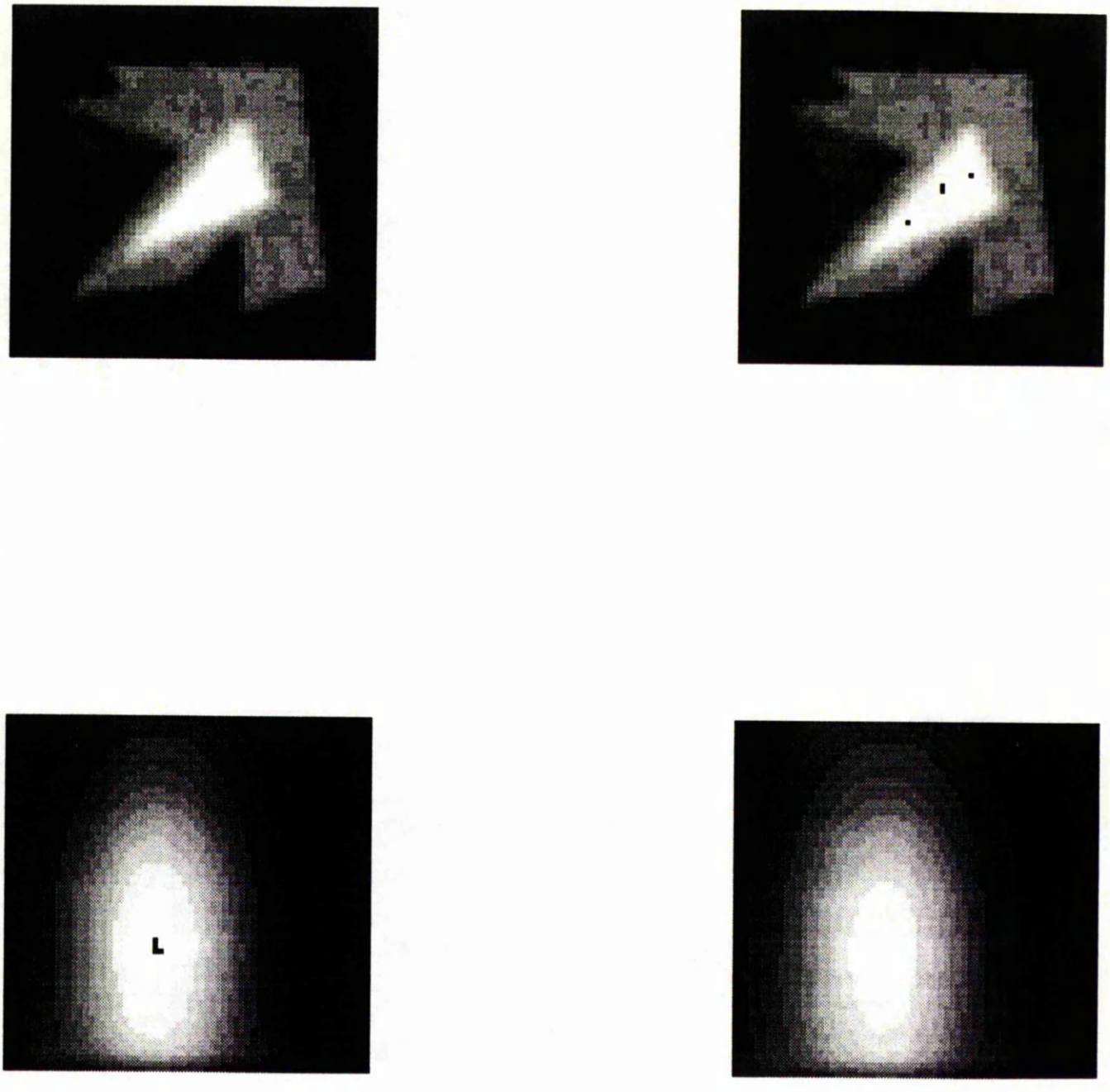


Figure 2.8: Some more restorations from Algorithm **normal**. From top left to bottom right we have (i) I5B1N1, starting point = truth, (ii) I5B1B1, starting point = data, (iii) I6B2N2, starting point = truth, and (iv) I6B2N2, starting point = data

we address in a later chapter; for many images (such as I6) it is hopefully not too extravagant to believe the m.r.f. is an appropriate representation of reality.

Thanks to the Hammersley–Clifford theorem (see, for example, [14], or [5]), we can model the local neighbourhood structure, as discussed in Chapter 1, of an m.r.f. via a Gibbs' distribution:

$$p(x \mid \beta) = Z(\beta)^{-1} \exp\{-\beta x^T A x\} \tag{2.17}$$

where $Z(\beta)$ is the normalising constant required to ensure the distribution is proper, i.e. that it sums to 1. Z makes the Gibbs distributions intractable: consider a discrete–valued image of n pixels, each pixel taking a value from $\{1, 2, \ldots, c\}$. Then the sample space of x is $\{1, 2, \ldots, c\}^n$, and since calculation of Z involves summing (or integrating, in the case of continuously–valued images) over every possible value of x, the computational cost is clearly prohibitive. (Consider, for example, the very simple case of a binary, 4×4 image: there are 2^{16} possible realisations of such an image and the probability of each must be evaluated and summed to calculate Z.) However, if the rank of A is n we can say, for a continuous x:

$$\begin{aligned} Z(\beta) &= \int_{\text{all } x} \exp\{-\beta x^T A x\} \mathrm{d}x \\ &= c\beta^{-n/2}, \end{aligned}$$

where c is a constant, independent of β. Thus $Z(\beta) \propto \beta^{-n/2}$.

Here, we call A the *smoothing* or *regularisation* matrix, and it is used to dictate the *order* of the m.r.f.; as with H, A will also be of block–Toeplitz form, with each block Toeplitz. In general, when the order of the m.r.f is p, then $A = Q_p^T Q_p$,

where Q_p is of order $(n_r - p)^2 \times n_r^2$ and has the form $D_p \otimes D_p$, where $\otimes$ denotes the Kronecker product. D_p is the $(n_r - p) \times n_r$ matrix with (i, k) element:

$$\{D_p\}_{i,k} = (-1)^k \begin{pmatrix} p \\ k \end{pmatrix},$$

for $i = 1, 2, ..., n_r - p$; $k = j - i$, $j = i, \ldots, i + p$ (see [60] for details). We assume Q_p is of full rank $(n_r - p)^2$ and thus

$$r = \text{rank}(A) = (n_r - p)^2.$$

For example, if we have specified that x is a realisation from a second–order m.r.f. (an 8–nearest–neighbours prior), then

$$x^T A x = \sum_{i \sim j} (x_i - x_j)^2$$

where $i \sim j$ means that pixel i and pixel j are neighbours in the m.r.f. definition.

Thus, we now specify the prior distribution for x as

$$p(x \mid \beta) \propto \beta^{r/2} \exp\{-\beta x^T A x\}. \tag{2.18}$$

Note that for $r < n$, this makes x a realisation of a singular normal distribution.

We also here specify general, natural forms for the priors of β, ϕ, i.e. we specify that they follow an inverse Gamma distribution:

$$p(\phi) \propto \phi^{l-1} \exp\{-1/m\phi\} \tag{2.19}$$

and

$$p(\beta) \propto \beta^{d-1} \exp\{-\beta/k\} \tag{2.20}$$

where $\phi, \beta, l, m, d, k > 0$. By "natural" we mean "mathematically tractable"; it is said that these are *conjugate* priors (see, for example, Section 6.3 of [16]) because

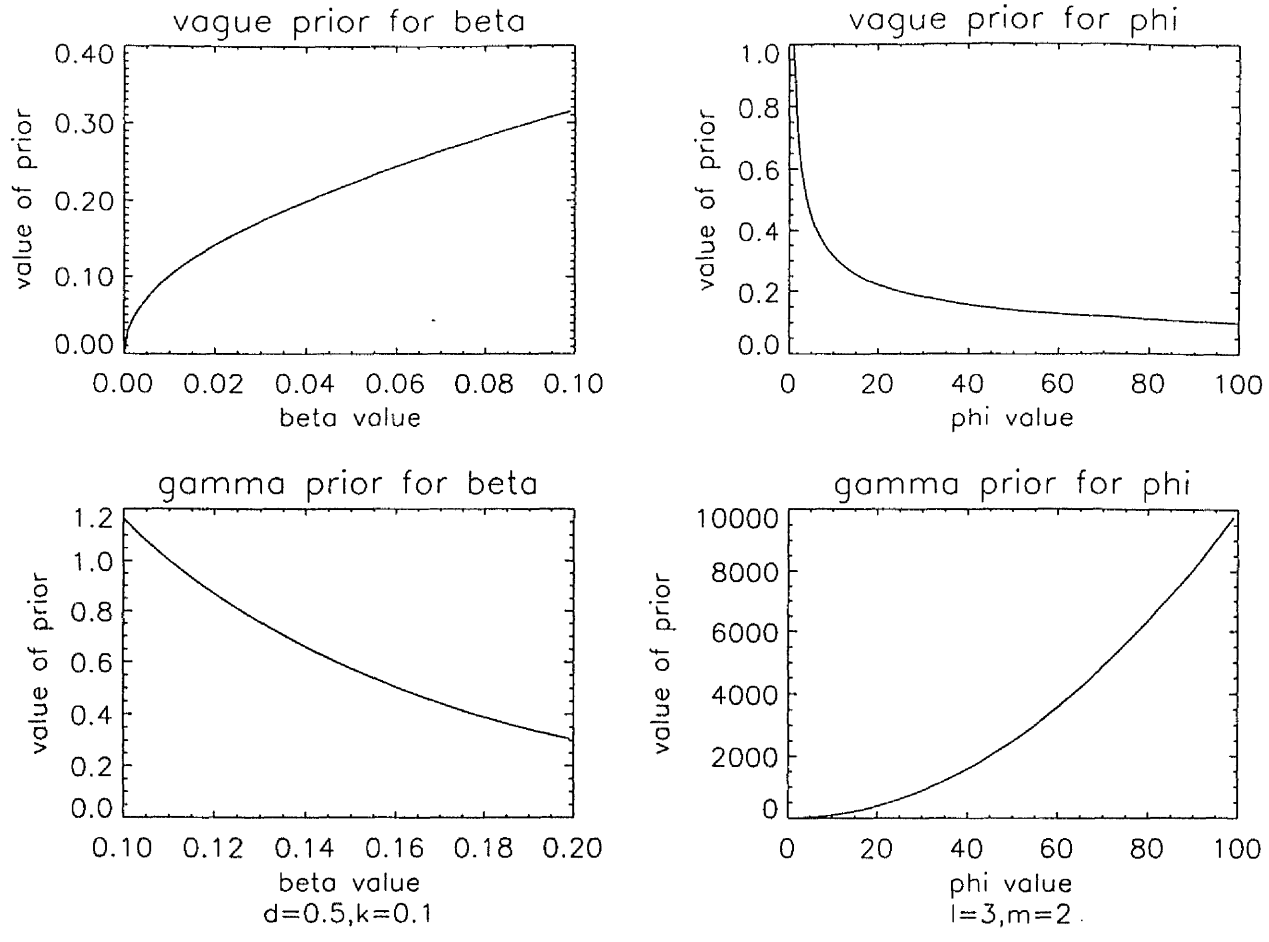


Figure 2.9: Different representations of the Gamma priors used to represent our ignorance concerning ϕ, β.

they will combine neatly with the Gaussian likelihood term when computing posterior densities. Although with the advent of modern Bayesian techniques such as the Gibbs Sampler (see [32]) this conjugacy between prior and likelihood is no longer a *sine qua non* of practical Bayesian inference, it is the case that the use of the Gamma density is rather uncontroversial; we shall specify in advance the values of k, d, l, m to be used. Figure 2.9 displays some such choices graphically.

In this algorithm, we fix the parameters at $d = 3/2, l = 1/2, k = m = \infty$ so that β, ϕ follow distributions (2.7,2.9) as before. Later, we shall relax some of these conditions.

We presume that y is caused by a Gaussian degradation, as in Algorithm

normal.

Combining our uncertainties with Bayes' Theorem again, we see that the log posterior density is now

$$\begin{aligned}\log\{p(x,\beta,\phi \mid y)\} \quad \propto \quad & -\frac{n+1}{2}\log(\phi)+\frac{r+1}{2}\log(\beta) \\ & -\frac{1}{2\phi}\|y-Hx\|^2-\beta x^T Ax. \qquad (2.21)\end{aligned}$$

The stationarity equations for β and ϕ take the form:

$$\frac{\partial}{\partial\beta}=\frac{(r+1)}{2\beta}-x^TAx \qquad (2.22)$$

and

$$\frac{\partial}{\partial\phi}=-\frac{n+1}{2\phi}+\frac{1}{2\phi^2}\|y-Hx\|^2. \qquad (2.23)$$

We choose x to maximise $p(x,\phi,\beta \mid y)$, which is equivalent to choosing x to minimise $\beta x^TAx+\frac{1}{2\phi}(y-Hx)^T(y-Hx)$. Differentiating this w.r.t. x we obtain:

$$\frac{\partial}{\partial x}=2\beta Ax+\frac{1}{\phi}H^THx-\frac{1}{\phi}H^Ty,$$

which equals zero when

$$(\frac{1}{\phi}H^TH+2\beta A)x=\frac{1}{\phi}H^Ty,$$

i.e. when

$$\begin{aligned}x &= (\frac{1}{\phi}H^TH+2\beta A)^{-1}\frac{1}{\phi}H^Ty \\ &= (H^TH+\lambda A)^{-1}H^Ty, \qquad (2.24)\end{aligned}$$

$$(2.25)$$

where $\lambda=2\phi\beta$.

These stationarity equations suggest the following fixed–point algorithm:

Algorithm **vague**

1. Choose $\hat{x}^{old}$.

2. Evaluate

$$\hat{\beta}_{\text{vague}} = \frac{r+1}{2\sum_{i\sim j}(\hat{x}_i^{old} - \hat{x}_j^{old})^2}, \tag{2.26}$$

$$\hat{\phi} = \frac{\|y - H\hat{x}^{old}\|^2}{n+1},$$

$$\hat{x}^{new} = N(\lambda)y$$

where $N(\lambda) = (H^T H + \lambda A)^{-1} H^T$ and $\lambda = (2\hat{\phi}\hat{\beta})$.

3. Check for convergence of $(\hat{\beta}, \hat{\phi}, \hat{x}^{new})$: if

YES $\longrightarrow$ **STOP**

NO $\longrightarrow$ set $\hat{x}^{old} := \hat{x}^{new}$ and go to step 2. □

At each iteration of the algorithm, we are using the estimates of β, ϕ to *regularise* our image estimate. See Section 3.6 in Chapter 3 for discussion of the relationships between our algorithms and the methods of regularisation.

Exploitation of the structure of H, A.

The advantage of the block-Toeplitz structure adopted for H and A now becomes clear: Toeplitz matrices can be well–approximated by circulant matrices, and the eigen–structure of circulant matrices is well understood (see [43, 58]). Further, we need only store the first row of each $n \times n$ matrix in order to capture all the information within that matrix.

Thus our computationally forbidding Algorithm **vague** can in effect be carried out by 3 straightforward discrete fast Fourier transforms (f.f.t.s) (see [40] and [91]): if $\{h_i\}, \{a_i\}$ are the sets of eigenvalues of H, A respectively, obtained by carrying out discrete f.f.t.s on their first rows, and w_i is the i'th component of the discrete f.f.t. of the data y, then we can write our estimate of the image x as

$$\hat{x}(\hat{\lambda}) = \sum_{i=1}^{n} \{ \frac{h_i (w_i * y)}{\mid h_i \mid^2 + \hat{\lambda} a_i} \} \times w_i.$$

2.3.1 Generating true images from m.r.f. prior distributions

When we use a restoration technique on a "real–life" image, the m.r.f. prior is a representation of what we intuitively feel should be correct in an image – i.e. we expect to see and would like to preserve local continuity. Since a picture of a heart, say, is very definitely *not* a realisation of an m.r.f., it follows that there does not exist a "true" value of β to be estimated; rather we aim for a "good" value of β in the sense of an aesthetically pleasing and informative restoration.

However, if the true image is indeed a realisation from

$$p(x \mid \beta) = Z(\beta)^{-1} \exp\{-\beta \sum_{i \sim j} (x_i - x_j)^2\} \tag{2.27}$$

then we would hope that the estimate of β is good not just in the sense that it leads to a visually pleasing image restoration but also that it should be "close" to the value used to generate the original image.

It is impossible to obtain such realisations directly due to the intractability of the normalising constant $Z(\beta) = \sum_{\underline{x}} \exp -\beta \sum_{i \sim j} (x_i - x_j)^2$ in the Gibbs repre-

sentation. We can obviate this difficulty and simulate such an image by using the ingenious Metropolis–Hastings Algorithm (see [70],[53]) which can be explained within the context of image analysis as follows (see, for example, [4] or [78]):

The Metropolis–Hastings Algorithm

Aim: to simulate a realisation from (2.27).

1. Choose $\hat{x}^0$ arbitrarily.

2. Choose $\hat{x}_i$ from $\hat{x}$ at random.

   Pick m such that $m \sim N(0, \tau^2)$.

   (We deal with the specification of τ shortly.)

   Make $\tilde{x}_i = \hat{x}_i + m$.

3. Calculate $\lambda = p(\tilde{x} \mid \beta)/p(\hat{x} \mid \beta)$

   where $\tilde{x} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_{i-1}, \tilde{x}_i, \hat{x}_{i+1}, ..., \hat{x}_n)$.

   It is the calculation of this ratio which renders unnecessary the calculation of the hideous $Z(\beta)$.

4. Accept $\tilde{x}$ with probability $p = \min(1, \lambda)$.

5. Repeat from step 2 above until convergence.

The sequence of images $\hat{x}$ so constructed form a Markov Chain with equilibrium distribution (2.27) ([87]). □

Convergence considerations

Methods such as this implementation of the Metropolis–Hastings algorithm, and the Gibbs sampler ([32]), combine such simplicity, elegance and usefulness that they have led to what may be only slightly over–stated as the renaissance of Bayesian statistics. (For a very small sample, see references: [7],[8], [37]). Indeed even non–Bayesian statisticians have made use of the flourishing ideas ([35]). However, these Monte Carlo Markov Chain (MCMC) methods suffer from a major drawback: how can we be certain that any particular sequence of realisations has reached the desired equilibrium distribution ([100])? Clearly this is of prime importance to our work, as we wish to test the efficacy of proposed estimators.

Green and Han ([47]) have shown that the speed of convergence is dependent upon the transition matrix of the Markov Chain (MC); in practice this eigen–analysis would usually be ominously complicated. Frigessi et al ([28]) show, again through an eigen–analysis, that for Ising models, a simplified binary version of the prior distribution that we are considering, the Metropolis–Hastings algorithm will converge more quickly than the Gibbs sampler.

Other work ([41]) involves the examination of auxiliary and antithetic variables, but most authors advocate the plotting of a simple summary statistic after each iteration with convergence being assumed when the plot of the statistics versus time has stabilised ([86],[13]). For example, Smith and Roberts ([87]), acknowledging that empirical evidence can never prove with certainty that convergence has been achieved (no more than such evidence could ever prove or disprove any statistical hypothesis), suggest some output analysis along these lines, offering the caveat that observing a scalar statistic involves the risk of ignoring

the multidimensional behaviour of the MC.

Kirkland ([62]) used time series techniques to check the stability of the pseudolikelihood estimator of the parameter in the Ising model , and suggests values for the number of realisations of the MC to be discarded before the assumption of convergence would be "safe". Gray ([41]) emphasises, again for the Ising model, that the initial configuration of the image is important.

The curiosity here is that Bayesians have fallen upon sampling theory to rescue them from the intractability of their complicated distributions: however, at least these methods are being used to sample from *correct* distributions.

The impossibility of using only a sample to make inference about a population was recognised by David Hume in 1777 when he articulated what became known as the "Scandal of Philosophy" (see [57], also [55]).

Here, I shall content myself with plotting the pseudo–likelihood estimator of β ([6]) at every iteration and assuming that equilibrium has been reached if this plot appears to converge to the correct value. In the next section we detail a method which may be of use in determining if the target value of β has in fact been reached.

There remains the question of choosing a value for τ; any positive value is of course valid, but the value actually selected will greatly affect the speed of convergence ([4]).

In our simulations, carried out to capture a realisation from (2.27), assumed to be of the second order, we used a value of $\tau = 2.0$, and attempted to simulate (1) from an m.r.f. with $\beta = 0.05$, and (2) from one where $\beta = 0.20$. The algorithm was run for $100,000$ and $500,000$ iterations for these two β values respectively:

in Figure 2.10 we see the results. For $\beta = 0.05$ the estimator we are tracking does indeed appear to have converged to close to the correct value; however for the larger value of β the algorithm appears to be "stuck" at a value of about 0.16: it is of course impossible to tell if the fault lies with the estimator, or with the sampler – the sampler we use is a very simple one. We will proceed with the two resulting images as though they were in fact realisations from m.r.f.s, although (illogical though it be) we have greater faith that this is in fact the case with the image corresponding to $\beta = 0.05$. Henceforth, this image is termed "I3", and the other m.r.f. realisation is "I4".

One point to mention is that in the definition of $p(x)$ we have used, β is basically a scale parameter and can therefore be altered by scaling x. Consideration of this would lead to τ being chosen appropriately for the two m.r.f. simulations.

2.3.2 Using the bootstrap technique to investigate β.

Above, we have detailed the problems involved in deciding whether or not a particular sample generated by the Metropolis– Hastings algorithm is in fact from the required distribution. Here we develop a technique based on the bootstrap procedure of Efron (see [24]) which we hope will aid us in that task. Although we can never know with certainty if the sample comes from the target distribution, we will use the bootstrap method to produce a range of plausible values for the parameter in the distribution. Again, we should emphasise that simply because the target value of β belongs to such a "confidence" interval, we could not make the inference that the image is a target realisation. However, if the true value of β is *not* within the interval, we would be fairly confident that the Metropolis–

Hastings procedure had not reached the appropriate equilibrium distribution. We offer the procedure as another method of weighing evidence, in other words, and make no strong inferential claims.

We first outline the bootstrap method, and then detail how we adapt it for our data structure.

THE BOOTSTRAP

Given some data x, which arise from an unknown distribution F, how can we make inference about a statistic calculated on x, called $s(x)$? Suppose the data is generated so that

$$x = (x_1, \ldots, x_n)^T \sim \text{ i.i.d. } F,$$

where "i.i.d." stands for independent and identically distributed. Bootstrapping proceeds by non–parametrically estimating the true d.f. F, using the sample x. Many resamples are taken from the original x, each providing a re–estimate of the statistic $s(x)$. If you could (re)sample an infinite amount of data, the non–parametric estimate of the distribution function (d.f.) of the statistic would tend to the true d.f. So we hope that with (large, finite) B resamples, the sampled distribution of the statistic *will* be close enough to its true distribution, allowing us to draw inference about the true value of the parameter.

To estimate F non–parametrically, put probability mass $(1/n)$ on each data point:

$$\hat{F} : \text{ probability } 1/n \text{ on } x_i, i = 1, \ldots, n.$$

A bootstrap sample from $\hat{F}$ is a sample drawn at random and with replacement

from the original sample, i.e.

$$\hat{F} \longrightarrow (x_1^*, \ldots, x_n^*)^T = x^*.$$

For every bootstrapped sample we draw, we recalculate the statistic of interest, $s^{*b} = s(x^{*b}), b = 1, \ldots, B$, and we use the resampled statistics to assess the accuracy of the original s (and by implication this provides information on the parameter in which we are interested). Information about s is drawn through this resample–distribution. For example, the so–called *percentile* confidence interval, with nominal confidence 0.95, is produced by selecting the 0.025 and 0.975 percentiles of $\{s^{*b}\}$ as endpoints. Note that this doesn't provide a confidence interval for the parameter which $s(x)$ is estimating – although we use the produced interval as though this were the case. Indeed, the inventor of the bootstrap, in [25], uses a percentile interval of the distribution of the sample correlation coefficient to make inference about the population correlation value. This abuse has caused these intervals to be derided (see, for example, Tukey's contribution to the discussion in [25]) and labelled *seductive* confidence intervals.

It is easy to see that in the case of long tailed resample distributions, or where such distributions are skewed, the percentile interval will be misleading. Many resamplers are therefore drawn to using the bootstrap technique to estimate the more robust quantities of the mean and standard deviation of the resample distribution. The bootstrap estimates of these quantites are, respectively,

$$\bar{s^*} = B^{-1} \sum_b s^{*b},$$

and

$$sd_{BS}(s) = \sqrt{\frac{\sum_b (s^{*b} - \bar{s^*})^2}{B-1}}.$$

An interval estimate for s based on standard normal asymptotic theory is then given by

$$\bar{s^*} \ (+/-) \ t(B-1;\alpha) sd_{BS}(s) \times \sqrt{B/(B-1)},$$

where $t(\nu;\alpha)$ refers to the 100αth percentile of the student–t distribution, evaluated on ν degrees of freedom.

BOOTSTRAPPING IMAGES

Having defined the bootstrap, how could it help us in our examination of the output from Metropolis–Hastings? We are using the pseudo–likelihood estimate of the population parameter β to indicate when to stop sampling. If we applied the bootstrap to the sample we select as "truth", and used the above techniques to produce a resample distribution for the pseudo–likelihood statistic, this could help us decide if the true parameter value has indeed been reached.

Our "data" then will be x, the true image. The pseudo–likelihood estimator (see Equation (2.39)) takes the part of $s(x)$. However x is very definitely *not* composed of i.i.d. elements, without which condition there is no guarantee of convergence of the estimated d.f. to true F. To cope with this difficulty, we adapt an argument of Hanna, in [52], who uses bootstrapping in a comparative exercise of some Air Quality models. He advocates the blocking of environmental data into homogenuous units, and resampling from within each unit, then recombining the resampled sub–units into one resampled vector. Our idea is to take advantage of the fact that the m.r.f. definition supposes the dependency structure in x is *local* : the value taken by pixel i, given the values of the pixels in its neighbourhood, is independent of the distribution of the values in the rest of the scene, i.e. $S \backslash \delta_i$. For

example, suppose x is 64×64, then we could divide it into eight non–overlapping 8×8 subsets (we choose 8 as an example only; and the subsets need not be non–overlapping). Within each bootstrap cycle, take the first pixel from *each* of the sub–blocks, resample these, and then take the set of second pixels, resample, and continue, until the entire image has been resampled. Hopefully, each *set* of pixels from which we resample are "far enough apart" to be independent of one another, as well as identically distributed. For emphasis: we are *not* resampling within each of the blocks, one block at a time. Rather, we take one pixel from each block at each bootstrap iteration, and resample these.

More formally, let S, the set of indices of x, be partitioned into ω subsets,each of which is of size W. Write $S = \{S_1, \ldots, S_\omega\}$. The block of x belonging to S_j we denote by $x_{[j]}$, each $x_{[j]}$ consisting of W pixels. The blocked bootstrapping procedure is as follows:

For $b = 1, \ldots, B$ **do begin:**

resample W points from $x_{[1]} \longrightarrow x^{*b}_{[1]}$.

... and so on...

resample W points from $x_{[\omega]} \longrightarrow x^{*b}_{[\omega]}$.

resampled data is $x^{*b} = (x^{*b}_{[1]}, \ldots, x^{*b}_{[\omega]})^T$.

evaluate bootstrap statistic $s(x^{*b})$.

End for b.

The resample distribution can then be used to produce either a moment or seductive confidence interval.

AN EXAMPLE

We use image **I3**, which we hope is a realisation of an m.r.f. with $\beta = 0.05$. If the technique works well for the true image, it may be of use in highlighting the effects of blur and/or noise on the pseudo–likelihood estimator, and so we apply the technique to the true image, to the true image after it has been blurred, to the true image with added Gaussian noise but no blur, and to the true image with blur *and* noise. The blur level is **B1** and the noise level is **N2**; see Section 2.2.1 for more details. These four images can be seen in Figure 2.11.

We carried out 1000 bootstrap replications in each case, and split the image into 12×12 sub–blocks, in an overlapping manner. Thus each of the pixels in the image was included in one of the sub–blocks, but none of the pixels in each sub–block was within 12 pixels of the others, either in the north–south or east–west directions.

After the bootstrapping we drew pictures of the bootstrap distributions (see Figure 2.12) and produced 95 per cent. seductive and moment intervals for β, as outlined above. The table below lists the values of the interval estimates.

Bootstrap interval estimates for the pseudo–likelihood statistic

image	percentile interval	moment interval
I3	(0.0464,0.0540)	(0.0463,0.0541)
I3B1	(0.1204,0.1367)	(0.1196,0.1370)
I3N2	(0.0045,0.0053)	(0.0044,0.0053)
I3B1N2	(0.0050,0.0058)	(0.0050,0.0058)

DISCUSSION

From examining Figure 2.12, we see that although the estimates of β are very different, each is quite symmetrical. This is reflected in the interval estimates, which show close agreement between the percentile and moment approaches. The

interval for the statistic calculated on the undistorted image I3 does indeed appear to sit almost symmetrically around the target value of 0.05, so we continue to view I3 as though it were a true realisation from the desired m.r.f. The other three intervals, however, indicate that the pseudo–likelihood estimator will fail, drastically, in the presence of distortion. For I3B1, the intervals are too high, and for the images with noise added, the intervals are too low. A glance at Figure 2.11 explains why: the blurring has emphasised the neighbourhood structure, producing an image which intuitively one would identify as a realisation from an m.r.f. with a much larger "attraction" parameter than 0.05. On the other hand, the images with noise added are completely swamped by the distortion; to the naked eye there is no evidence of the local dependency structure at all. While this is hardly surprising, it is a *caveat* against the use of pseudo–likelihood as an estimator of m.r.f. parameters in the presence of large scale distortions.

In summary, it appears that this adapted block–bootstrap procedure may be of use in the examination of the output from Monte Carlo Markov Chain simulations.

2.3.3 Results for Algorithm *vague*

Again, 4 test images were employed; I5 and I6 as before, but in addition:

I3: a simulation using the Metropolis–Hastings algorithm of a second order m.r.f. with $\beta = 0.05$.

I4: as I3 save for $\beta = 0.20$.

The degradation conditions were as for Algorithm **normal**, as was the convergence criterion. The starting value for $\hat{x}$ was always taken to be the data, y.

Results:

I3

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I3B1N1	0.05	4.0	0.28	3.43	1.73	2
I3B1N2	0.05	25.0	1.44	23.91	2.12	3
I3B2N1	0.05	4.0	0.42	3.63	1.85	2
I3B2N2	0.05	25.0	2.55	24.15	2.14	3

I4

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I4B1N1	0.20	4.0	0.367	3.366	0.648	2
I4B1N2	0.20	25.0	1.643	23.768	0.782	3
I4B2N1	0.20	4.0	0.517	3.614	0.676	2
I4B2N2	0.20	25.0	2.719	24.105	0.789	3

I5

Data	β_{true}	ϕ_{true}	$\hat{\beta}^{(1)}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I5B1N1	–	4.0	0.876	3.87	21.71	2
I5B1N2	–	25.0	1.04	22.2	32.33	3
I5B2N1	–	4.0	1.33	4.11	34.21	2
I5B2N2	–	25.0	1.47	23.72	44.84	3

(1) the values in this column are $\times 10^{-2}$.

I6

Data	β_{true}	ϕ_{true}	$\hat{\beta}^{(2)}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I6B1N1	–	4.0	0.973	5.79	15.77	2
I6B1N2	–	25.0	1.06	23.74	17.66	3
I6B2N1	–	4.0	0.955	8.43	30.56	2
I6B2N2	–	25.0	1.02	27.52	31.23	3

(2) the values in this column are $\times 10^{-2}$.

Discussion:

Here we see good– though consistently under– estimation of ϕ, not only for the simulated images but also for artificial I5 and I6 (recall that Algorithm **normal** was incapable of this). The remarkable "image–m.s.e.$(\hat{x}, x)$" effect noticed in Algorithm **normal** is not so evident here: perhaps there is evidence of a noise effect in the simulated image examples.

We may directly compare the results of Algorithm **vague** with Algorithm **normal** for artificial images I5,I6 where the starting point was taken to be the data, y. It is clear that there is a vast improvement in every measured respect: much "better" estimates of β, ϕ (i.e. the estimates of β are smaller and the estimates of ϕ are quite accurate) leading to a reduction in the number of iterations required to reach convergence, and a huge reduction in the m.s.e. between the final image estimate and the truth. Average m.s.e. discrepancies between estimate and truth for Algorithm **normal** are 15476.5,19086.4 for I5 and I6 respectively; for Algorithm **vague** these figures fall to 33.27 and 23.81.

Some restorations effected by this algorithm can be seen in Figure 2.13.

Notwithstanding this greater success, it remains that for those cases where we are permitted to make a decision about how close $\hat{\beta}$ is to the correct value, it appears that once again the algorithm is producing estimates that are too large – although the more realistic image–prior may have helped to reduce the estimates from the huge values noted in Algorithm **normal**. We turn our attention to this issue.

The behaviour of $\hat{\beta}$

First of all, let us consider a very simple set–up, keeping our argument as clear as possible. Setting $H = A \equiv I$ we may write the log–posterior density of x, β, ϕ given y as

$$\frac{n+1}{2}\log\beta - \frac{n+1}{2}\log\phi - \beta\sum x_i^2 - \frac{1}{2\phi}\sum(y_i - x_i)^2. \tag{2.28}$$

If this is differentiated w.r.t. x_m and set to zero, we see the m.a.p. estimate of x_m is

$$\hat{x}_m = \frac{y_m}{1+2\phi\beta}, m = 1, \ldots, n. \tag{2.29}$$

Now we substitute (2.29) into (2.28), to see that the "profile" log posterior for (ϕ, β) is

$$\frac{n+1}{2}\log\beta - \frac{n+1}{2}\log\phi - \frac{\beta\sum y_i^2}{1+2\phi\beta}. \tag{2.30}$$

Now let $\beta \to \infty$ in (2.30). Clearly, (2.30) will be dominated by $\log\beta$ as β increases in size. Since $\log\beta \to \infty$ as $\beta \to \infty$, we see that $\hat{\beta} = \infty$ *is* a maximiser of the log posterior density.

It remains to be seen, however, if the result holds for general H, A. The general form of (2.28) is (2.21), and the m.a.p. estimator of x is given in (2.24). Substitution as before yields

$$\begin{aligned}
\log p(\hat{x}, \phi, \beta \mid y) &= \frac{r+1}{2}\log\beta - \frac{n+1}{2}\log\phi \\
&\quad -\beta\{W^{-1}H^T y\}^T A\{W^{-1}H^T y\} \\
&\quad -\frac{1}{2\phi}\{y - HW^{-1}H^T y\}^T\{y - HW^{-1}H^T y\} \\
&= \frac{r+1}{2}\log\beta - \frac{n+1}{2}\log\phi
\end{aligned}$$

$$-y^T\{\beta HW^{-T}AW^{-1}H^T + \frac{1}{2\phi}I \tag{2.31}$$

$$-\frac{1}{\phi}HW^{-1}H^T + \frac{1}{2\phi}HW^{-T}H^THW^{-1}H^T\}y \tag{2.32}$$

$$\tag{2.33}$$

where $W = H^TH + 2\phi\beta A$ and so $W^{-1} = H^{-1}(I + 2\phi\beta C)^{-1}H^{-T}$ and $C = H^{-T}AH^{-1}$.

Now examining the term in braces in (2.31,2.32):

$$\begin{aligned}
\{\ldots\} &= \frac{1}{2\phi}(I + 2\phi\beta C)^{-1}(I + 2\phi\beta C)(I + 2\phi\beta C)^{-1} - \frac{1}{\phi}(I + 2\phi\beta C)^{-1} + \frac{1}{2\phi}I \\
&= -\frac{1}{2\phi}(I + 2\phi\beta C)^{-1} + \frac{1}{2\phi}I \\
&= \frac{1}{2\phi}(I + 2\phi\beta C)^{-1}I + 2\phi\beta C - I \\
&= \beta C(I + 2\phi\beta C)^{-1}.
\end{aligned}$$

So (2.33) has the simplified form

$$\frac{r+1}{2}\log\beta - \frac{n+1}{2}\log\phi - \beta y^TC(I + 2\phi\beta C)^{-1}y, \tag{2.34}$$

and, as with the simpler example, it is straightforward to see that this expression will be dominated by $\log\beta$ as $\beta \longrightarrow \infty$.

Besides explaining some of the large β estimates we have observed, this result carries the worrying implication that the posterior distribution is improper, i.e. it encloses a total volume greater than unity. The highly improper prior we have employed for β seems to be the main cause of this trouble.

In our next algorithm we take cognisance of this possibility and introduce a stronger prior distribution for the β hyper–parameter, which in effect places an

upper bound on the set of possible $\hat{\beta}$ values. However a little thought shows that the problem of posterior impropriety may well remain, despite the limits to the range of β.

2.4 Algorithm *gamma*

More realistic prior distributions for the m.r.f. parameter

We now take advantage of the general Gamma prior we specified for β in the preamble to Algorithm **vague**. Suppose

$$\beta \sim Ga(k, d)$$

that is, β follows a Gamma distribution, with associated p.d.f., defined by parameters k, d, given by

$$p(\beta) \propto \beta^{d-1} \exp(-\beta/k), \tag{2.35}$$

with $\mathrm{E}(\beta) = dk$ and $\mathrm{var}(\beta) = dk^2$.

If we retain the same priors for ϕ and x as before, then the posterior distribution becomes

$$p(x, \phi, \beta \mid y) = \beta^{\frac{r+2(d-1)}{2}} \phi^{-\frac{n+1}{2}} \exp\{-\frac{1}{2\phi}\|y - Hx\|^2 - \beta x^T A x - \beta k^{-1}\}. \tag{2.36}$$

On taking logs and differentiating, we arrive at the same stationarity equations for ϕ and x as in Section (2.3), but the estimate for β changes:

$$\hat{\beta}_{\mathrm{gamma}} = \frac{r + 2(d-1)}{2(x^T A x + k^{-1})}. \tag{2.37}$$

We thus have a third algorithm:

Algorithm **gamma**

–exactly the same as Algorithm **vague**, except replace vague-prior formula (2.26) for $\hat{\beta}$ with that of equation (2.37). □

Comparison of $\hat{\beta}_{\text{gamma}}$ and $\hat{\beta}_{\text{vague}}$

Comparing (2.37) with (2.26), we can see the mechanical implications of our desire to discourage large values of $\hat{\beta}$. Given that (as a glance at the formulae in the algorithms will confirm) as $\hat{\beta}$ increases, then $\hat{x}A^T\hat{x}$ decreases, it would be possible for iterative Algorithm **vague** to cause spiralling estimates of β and ever–shrinking–towards–zero estimates of x. However, the imposition of the tighter prior in Algorithm **gamma** means that, even if $\hat{x}A^T\hat{x} = 0$, our "worst–case" scenario for the m.r.f. parameter estimate is $\hat{\beta} = \frac{k}{2}(r + 2(d-1))$. For example, for a 64×64 image assumed to be the realisation of a 2nd order m.r.f., and choosing $k = 0.1, d = 0.5$, then the worst–case value for $\hat{\beta}_{\text{gamma}}$ is 192.15 – higher than any of the recorded values in Algorithm **vague**, but a lot better than some of the results in Algorithm **normal**. Of course, the price we must pay for this increased guarantee in the range of the m.r.f. parameter estimator is the generation of a further 2 hyper–parameters –namely k, d. However, as we hope the results below indicate, the procedure seems fairly robust to parameter mis–specification.

2.4.1 Results for Algorithm *gamma*

Only the images that are being considered as realisations of m.r.f.'s were employed at this stage. Other conditions are identical to Algorithm **vague**. Three different

parameterisations of the Gamma distribution used for the prior of the β parameter were used. If we write $\beta \sim Ga(k, d)$ then $\mathrm{E}(\beta) = kd$. For each image, we chose three values of k, d, so that (1) the expected value of β was equal to the true value, but the Gamma distribution was diffuse; (2) the expected value equalled the true value, and the distribution was tightly positioned around the true β, and (3) the expected value of the distribution was *not* equal to the true value of the m.r.f. parameter.

Results:

I3

Data	k	d	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I3B1N1	0.1	0.5	0.05	4.0	0.277	3.427	1.728	2
	0.001	50.0	0.05	4.0	0.251	3.431	1.719	2
	0.01	9.0	0.05	4.0	0.275	3.428	1.727	2
I3B1N2	0.1	0.5	0.05	25.0	1.430	23.912	2.120	3
	0.001	50.0	0.05	25.0	0.850	23.919	2.110	3
	0.01	9.0	0.05	25.0	1.352	23.915	2.119	3
I3B2N1	0.1	0.5	0.05	4.0	0.419	3.631	1.852	2
	0.001	50.0	0.05	4.0	0.357	3.632	1.841	2
	0.01	9.0	0.05	4.0	0.414	3.631	1.852	2
I3B2N2	0.1	0.5	0.05	25.0	2.516	24.150	2.143	3
	0.001	50.0	0.05	25.0	1.123	24.149	2.134	3
	0.01	9.0	0.05	25.0	2.268	24.151	2.142	3

I4

Data	k	d	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I4B1N1	0.1	2.0	0.20	4.0	0.366	3.366	0.648	2
	0.001	200.0	0.20	4.0	0.361	3.384	0.648	2
	1.5	3.0	0.20	4.0	0.362	3.367	0.648	2
I4B1N2	0.1	2.0	0.20	25.0	1.630	23.768	0.782	3
	0.001	200.0	0.20	25.0	1.049	23.843	0.787	3
	1.5	3.0	0.20	25.0	1.525	23.770	0.783	3
I4B2N1	0.1	2.0	0.20	4.0	0.516	3.614	0.676	2
	0.001	200.0	0.20	25.0	0.476	3.622	0.675	2
	1.5	3.0	0.20	4.0	0.507	3.614	0.675	2
I4B2N2	0.1	2.0	0.20	25.0	2.682	24.105	0.789	3
	0.001	200.0	0.20	25.0	1.298	24.141	0.792	3
	1.5	3.0	0.20	25.0	2.400	24.106	0.789	3

Discussion:

The results are more similar to those of Algorithm **vague** than expected! The ϕ estimates continue to be biased but good; since the formula is the same as before this is reassuring, given the iterative nature of these algorithms. The estimates of β, although still too large, are much smaller than the "worst–case" scenario outlined above. Indeed, we may point out that although these experiments suggest roughly similar values for the estimates of β, Algorithm **gamma** should ensure a finite result, while Algorithm **vague** cannot.

In Section 2.6, we carry out a simulation exercise to study more deeply the effect of (k, d) mis-specification, and to compare further Algorithms **vague** and **gamma**.

2.5 Algorithm *pseudo*

A pseudo–likelihood estimator for β

As our estimates for ϕ appear to be reasonable, we turn our attention now to

a different estimator for β, based on an adaption of a likelihood–function.

Many statisticians are attracted to the notion that all inference should proceed with respect to the likelihood function. Such people have attempted to deal with the dependency in the image prior, which makes estimation of β so difficult, by using a technique known as *maximum pseudo–likelihood estimation* (see, most famously, [6]). We have already used this estimator, in our section on bootstrapping and MCMC.

The likelihood function for a set of random variables $\{X_i : i = 1, 2, ..., n\}$, each with p.d.f. indexed by parameter $\theta \in \Theta$, is straightforward when there is stochastic independence among the $\{X_i\}$. This enables us to write the joint density of the random variables as the multiple of the individual densities (we are here assuming, in a non–Bayesian manner, that our only assumptions regarding θ are that it is fixed and unknown):

$$p(X \mid \theta) = \prod_i p(X_i \mid \theta).$$

If we then regard a set of data $x = \{x_i\}$ as being a set of realisations from $p(X \mid \theta)$, regard $p(x \mid \theta)$ as a function of θ rather than x and maximise this w.r.t. θ, the solution is called the *maximum likelihood estimator*. Note that the form of this estimator is equivalent to the Bayesian m.a.p. estimator, given a uniform prior distribution over θ's range.

However, in image analysis we cannot make the simplifying assumption that our $\{x_i\}$ are independent – in fact we feel confident to assert that there ought to be strong dependence among image elements in our data. A pseudo–likelihood function attempts to overcome this difficulty by conditioning on the values of the

m.r.f. neighbours of pixel i:

$$\mathrm{psl}(x;\beta) = \prod_{i=1}^{n} p(x_i \mid x_{\delta_i}, \beta), \tag{2.38}$$

where δ_i denotes the m.r.f. neighbourhood of pixel i, as described in Chapter 1.

It remains to construct the product terms on the R.H.S. of (2.38); given that we know, up to a constant of proportionality,

$$p(x \mid \beta) \;\propto\; \beta^{r/2} \exp\{-\beta \sum_{i\sim j} (x_i - x_j)^2\},$$

then we see that (initially omitting reference to β for clarity)

$$\begin{aligned}
p(x_j \mid x_{\delta_j}) &= p(x_j, x_{\delta_j})/p(x_{\delta_j}) \\
&= p(x)/p(x_{\delta_j}) \\
&\propto \exp\{-\beta \sum_{k\in\delta_j} (x_k - x_j)^2\} \\
&\propto \exp\{-\beta(n_j x_j^2 - 2x_j \sum_k x_k + \sum_k x_k^2)\} \\
&\propto \exp\{-\beta n_j (x_j - \bar{x}_{\delta_j})^2\},
\end{aligned}$$

since terms involving k are constants. In the above, $n_j = \mid \delta_j \mid$, $\bar{x}_{\delta_j} = \frac{1}{n_k} \sum_{k\in\delta_j} x_k$, and so we can see by inspection that

$$p(x_j \mid x_{\delta_j}) \equiv N(\bar{x}_{\delta_j}, \frac{1}{2n_j\beta}).$$

Substituting this information in (2.38), we see that

$$\begin{aligned}
\mathrm{psl}(x;\beta) &= \prod_{i=1}^{n} \beta^{1/2} \exp\{-n_i\beta(x_i - \bar{x_{\delta_i}})^2\} \\
&= \beta^{n/2} \exp\{-\beta \sum_i n_i (x_i - \bar{x_{\delta_i}})^2\}.
\end{aligned}$$

Clearly this function is maximised w.r.t. β when $\beta = \hat{\beta}_{psl}$ where

$$\hat{\beta}_{psl} = n/2[\sum_{i=1}^{n} n_i(x_i - \bar{x_{\delta_i}})^2]. \tag{2.39}$$

We must be aware that in practice our pseudo–likelihood estimator for β will be doubly approximate, since all x in (2.39) will of course be estimated.

However, the use of the (2.39) leads to a further image restoration algorithm:

Algorithm **pseudo**

– exactly the same as for Algorithm **vague**, except we replace the formula for $\hat{\beta}$ in (2.26) with that of $\hat{\beta}_{psl}$ in (2.39). □.

NB: Although we are using a point–estimate of β here, our aim is still that the iterative process should converge to the most probable m.a.p. estimates for ϕ and x. We return to this point in the next chapter.

2.5.1 Results for Algorithm *pseudo*

Both artificial images and m.r.f. images are used here; other conditions are as for Algorithm **vague**.

Results:

I3

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I3B1N1	0.05	4.0	75.275	4.233	2.152	3
I3B1N2	0.05	25.0	40.495	24.374	2.183	3
I3B2N1	0.05	4.0	264.852	4.102	2.179	3
I3B2N1	0.05	25.0	138.745	24.294	2.193	3

I4

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I4B1N1	0.20	4.0	0.988	3.356	0.662	2
I4B1N2	0.20	25.0	40.383	24.174	0.845	3
I4B2N1	0.20	4.0	301.096	3.985	0.853	3
I4B2N1	0.20	25.0	128.613	24.413	0.870	3

I5

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I5B1N1	–	4.0	1152998.837	655.067	708.512	14
I5B1N2	–	25.0	246253.579	674.421	708.472	11
I5B2N1	–	4.0	46744772.249	614.097	708.524	15
I5B2B2	–	25.0	189245.834	627.954	708.452	11

I6

Data	β_{true}	ϕ_{true}	$\hat{\beta}$	$\hat{\phi}$	mse$(\hat{x}, x)$	no. iters
I6B1N1	–	4.0	0.143	9.223	16.544	4
I6B1N2	–	25.0	0.352	36.261	25.609	5
I6B2N1	–	4.0	0.203	11.323	26.595	4
I6B2N2	–	25.0	0.436	46.901	32.802	5

Discussion:

The retreat from the fully Bayesian approach has produced results similar to those for the over naive Algorithm **normal**: inflation of the estimates of β; good estimation of ϕ for I3 and I4, the m.r.f. simulations, but poor estimation of ϕ for the artificial images. The image restorations from this algorithm are undoubtedly the worst in this thesis.

Clearly the estimation of β and ϕ is more successful the fewer edges there are in the underlying image. (This is reflected by comparing the results for sharp–edged I5 with continuous I6). This is not unexpected since our model for the true image (2.17) assumes there are no sharp discontinuities present. The value of $\hat{\beta}$, whether the pseudo–likelihood estimator (2.39) or the earlier Bayesian ones, will

be decreased in the presence of edges, since all involve calculation of a sum–of–squares between the value of each pixel and its neighbours.

Some of the reconstructions from Algorithms **gamma** and **pseudo** can be seen in Figure 2.14.

2.6 Simulation exercise

More detailed comparison between Algorithms *vague* and *gamma*

Above, we noted that the vague prior and the Gamma prior algorithms appeared to produce estimates of the m.r.f. parameter that were similar, although in theory the vague prior method is capable of producing much larger estimates of β.

Here, we carry out a simulation exercise to investigate the behaviour of the two estimators further. If there is no major difference between the results of the two algorithms, common sense would suggest the use of Algorithm **vague**, since there are fewer parameters involved.

Details

We use m.r.f. images I3 ($\beta_{\text{true}} = 0.05$) and I4 ($\beta_{\text{true}} = 0.20$), corrupted with blur level B1 and noise level N1 ($\phi_{\text{true}} = 4.0$).

As starting point for the image estimate we take $\hat{x}^{(0)} = y$, the data, and also $\hat{x}^{(0)} = x_{\text{true}}$.

We attempted to simulate 2nd order m.r.f.'s with the Metropolis–Hastings algorithm; here we examine the effects of m.r.f. order mis–specification by assuming

that our image is of the first and then of the second order.

Each trial is allowed to run for a maximum of 120 iterations, or until $| \hat{x}_i^{\text{iter}} - \hat{x}_i^{\text{iter}-1} | \leq 0.5$, for all i.

We assume that β has (1) the vague prior detailed in Algorithm **vague**, and (2) each of the 3 gamma priors for each image as specified in Algorithm **gamma**.

There are 1000 simulations carried out for each of the possible combinations of starting points.

Results

In the following, a "$\bar{}$" represents the mean value of a set of simulations, and "e.s.e." stands for the estimated standard error. "V" indicates that the vague prior of Algorithm **vague** was assumed, while "g" indicates a Gamma prior, with the following parameters:

image	code	k	d
I3	g1	0.1	0.5
	g2	0.001	50.0
	g3	1.5	3.0
I4	g1	0.1	2.0
	g2	0.001	200.0
	g3	1.5	3.0

I3 : $\beta_{\mathbf{true}} = 0.05$; $\phi_{\mathbf{true}} = 4.0$.

$\hat{x}^{(0)}$	m.r.f.	prior	$\bar{\hat{\beta}}$	e.s.e.$(\hat{\beta})$	$\bar{\hat{\phi}}$	e.s.e.$(\hat{\phi})$	m.$\bar{\hat{\text{s}}}$.e.	e.s.e.(m.$\hat{\text{s}}$.e.)
truth	1	V	0.837	0.037	3.514	0.085	0.244	0.006
		g1	0.833	0.037	3.514	0.085	0.244	0.006
		g2	0.597	0.019	3.510	0.085	0.189	0.005
		g3	0.838	0.037	3.514	0.085	0.244	0.006
	2	V	0.731	0.039	3.794	0.089	0.257	0.009
		g1	0.727	0.038	3.794	0.089	0.257	0.009
		g2	0.543	0.021	3.794	0.089	0.217	0.008
		g3	0.731	0.039	3.794	0.089	0.257	0.009
data	1	V	1.015	0.681	3.481	0.302	0.315	0.169
		g1	0.987	0.671	3.474	0.303	0.319	0.170
		g2	0.283	0.051	3.218	0.096	0.459	0.034
		g3	1.016	0.682	3.482	0.302	0.315	0.169
	2	V	0.281	0.013	3.505	0.081	0.441	0.016
		g1	0.280	0.013	3.504	0.081	0.440	0.016
		g2	0.254	0.010	3.509	0.081	0.416	0.016
		g3	0.281	0.013	3.505	0.081	0.441	0.016

I4 : $\beta_{\mathbf{true}} = 0.20$; $\phi_{\mathbf{true}} = 4.0$.

$\hat{x}^{(0)}$	m.r.f.	prior	$\bar{\hat{\beta}}$	e.s.e.$(\hat{\beta})$	$\bar{\hat{\phi}}$	e.s.e.$(\hat{\phi})$	m.$\bar{\hat{\text{s}}}$.e.	e.s.e.(m.$\hat{\text{s}}$.e.)
truth	1	V	2.945	0.164	3.645	0.085	0.122	0.004
		g1	2.902	0.159	3.644	0.085	0.121	0.004
		g2	1.287	0.029	3.638	0.085	0.062	0.002
		g3	2.948	0.164	3.645	0.085	0.122	0.004
	2	V	2.663	0.188	3.825	0.087	0.121	0.007
		g1	2.626	0.183	3.825	0.087	0.121	0.007
		g2	1.244	0.036	3.830	0.087	0.077	0.005
		g3	2.644	0.188	3.825	0.087	0.121	0.007
data	1	V	0.429	0.309	3.192	0.104	0.462	0.043
		g1	0.422	0.280	3.191	0.101	0.462	0.040
		g2	0.381	0.014	3.202	0.076	0.424	0.015
		g3	0.430	0.309	3.192	0.104	0.462	0.043
	2	V	0.402	0.020	3.445	0.079	0.341	0.014
		g1	0.401	0.020	3.445	0.079	0.341	0.014
		g2	0.390	0.016	3.463	0.079	0.314	0.014
		g3	0.402	0.020	3.445	0.079	0.341	0.014

Discussion:

First let us discuss estimation of β. The best results for I3 and I4 are predictably those for which the Gamma hyper–prior is tightly positioned around the true value. Not only are the estimates and s.e.'s of β smaller, but $mse(\hat{x}, x)$ is also significantly reduced. This is, at least, reassuring.

There is *very* little difference between the results assuming a vague prior, diffuse Gamma prior with expected value the true value of β, and wrongly positioned Gamma prior. The conclusion we would draw is that unless one has highly accurate information concerning the β parameter, there is little to be gained from the computational expense of imposing a Gamma, rather than a vague, hyper–prior.

With respect to ϕ, there seems little to choose between any of the results, except that on two occasions the use of the tight β hyper–prior seems to cause smaller estimates, both of ϕ and of its standard error. ϕ is consistently under–estimated by all hyper–prior/starting value for x/m.r.f. order combinations.

2.7 Summary

Four restoration/estimation algorithms were proposed. The very simplest produced bad parameter estimates for both the m.r.f parameter, β, and the noise level, ϕ. Two more successful algorithms used a Gibbs prior for the unknown image and placed hyper–priors of different complexity on β: a simulation exercise indicated that the less complicated algorithm was preferable, despite our fears of the possible ever–increasing behaviour of $\hat{\beta}$ under that scheme. The fourth algorithm, which estimated β by pseudo–likelihood at each iteration, was highly

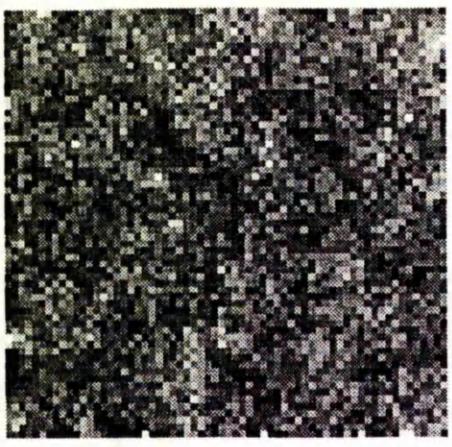
unsuccessful, partly due to the cyclical nature of the algorithm, and partly because the m.r.f. realisations, to which the algorithm was applied, are swamped by the degradation process.

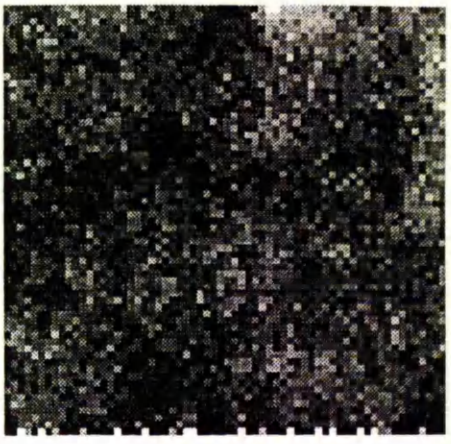
All the algorithms seem to be implementable from a practical point of view, given that there is little difference between results which start with the truth as the initial estimate of x, and those which use the data in this role.

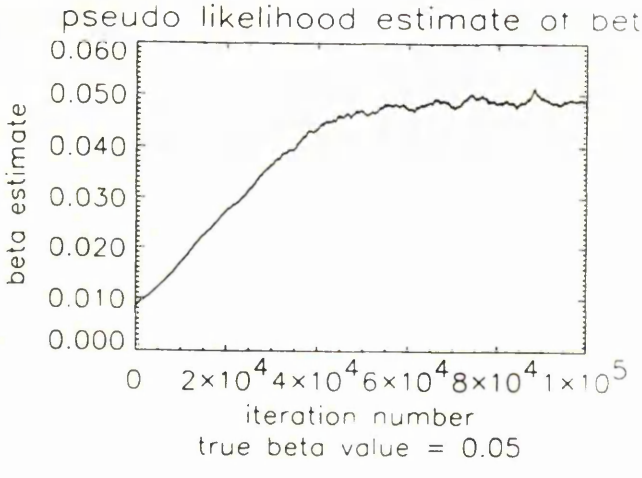
We attempted to simulate true realisations of m.r.f.s using the Metropolis–Hastings algorithm, and proposed a resampling technique to help judge when equilibrium is reached.

Finally, we should emphasise that none of the methods produces estimates of β close to the correct value. According to Ripley ([77]), estimation of β, even from the true image, undistorted by blur or noise, is liable to be unsuccesful, not only because of the computational complexity due to the normalisation constant, which we have already mentioned, but also because the parameter is measuring the conditional variance of a pixel, *given* its m.r.f. neighbours. Therefore our estimates of β are unable to explain the large scale variability observed in typical images – further explanation of why the methods are more successful on the simulated m.r.f. images, and also on the test image with no sharp discontinuities.

The same author makes the point which we mentioned earlier: "Perhaps when a model is only a means to another end, its inadequacies are only of second–order importance", i.e. if we are observing good image restoration, then the parameter estimation method, though not optimal in some senses, is certainly performing sufficiently well in another.







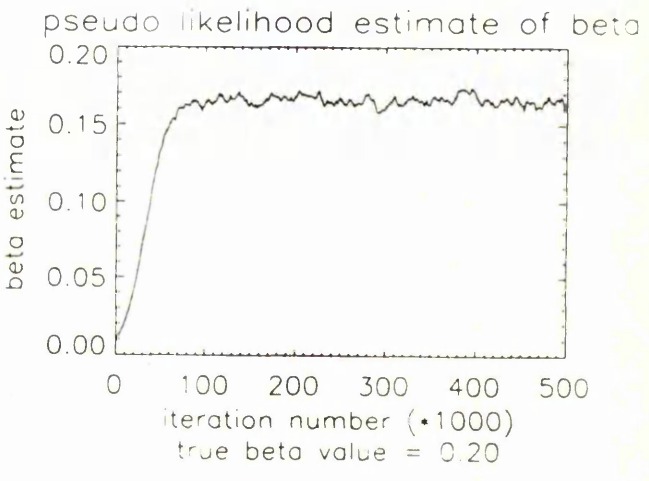


Figure 2.10: Figures (i) and (ii) show images 13, 14 simulated from prior model (2.27) with $\beta = 0.05$ and $\beta = 0.20$ respectively. Plots (iii) and (iv) of the pseudo-likelihood estimator of β *vs* iteration number indicate that it is safe to assume convergence.

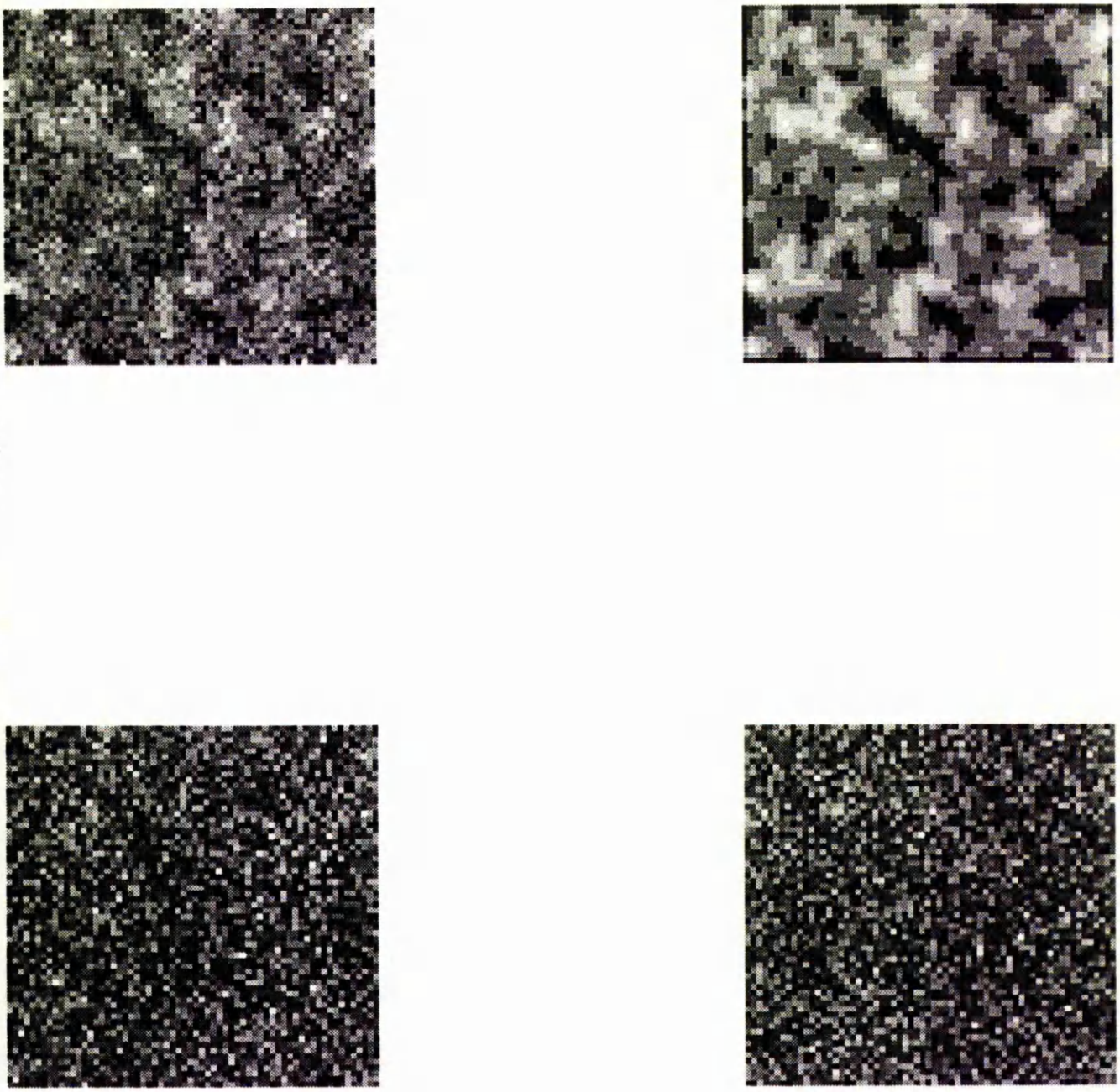


Figure 2.11: The m.r.f. realisation and its degraded forms. From top left to bottom right, we have (i) I3, (ii) I3B1, (iii) I3N2, and (iv) I3B1N2.

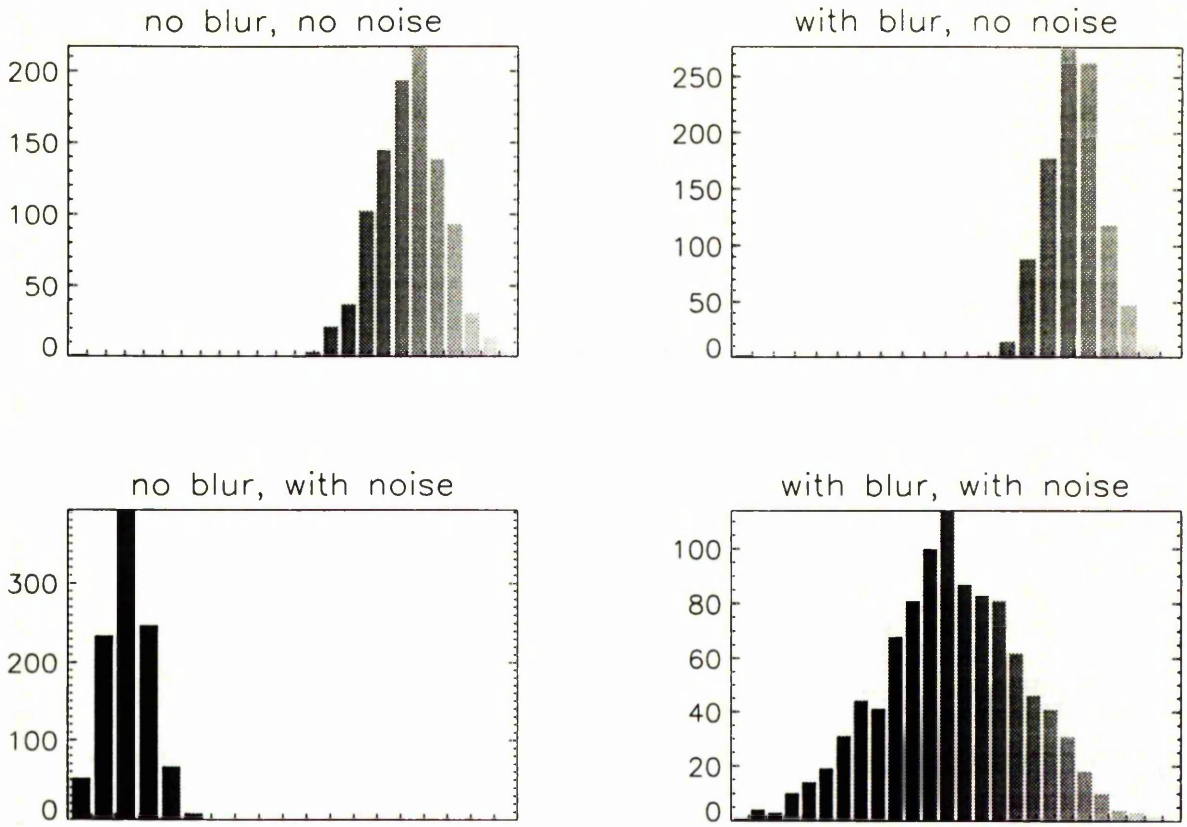


Figure 2.12: Histograms of four resample distributions of the pseudolikelihood statistic applied to various distortions of true image **I3**.

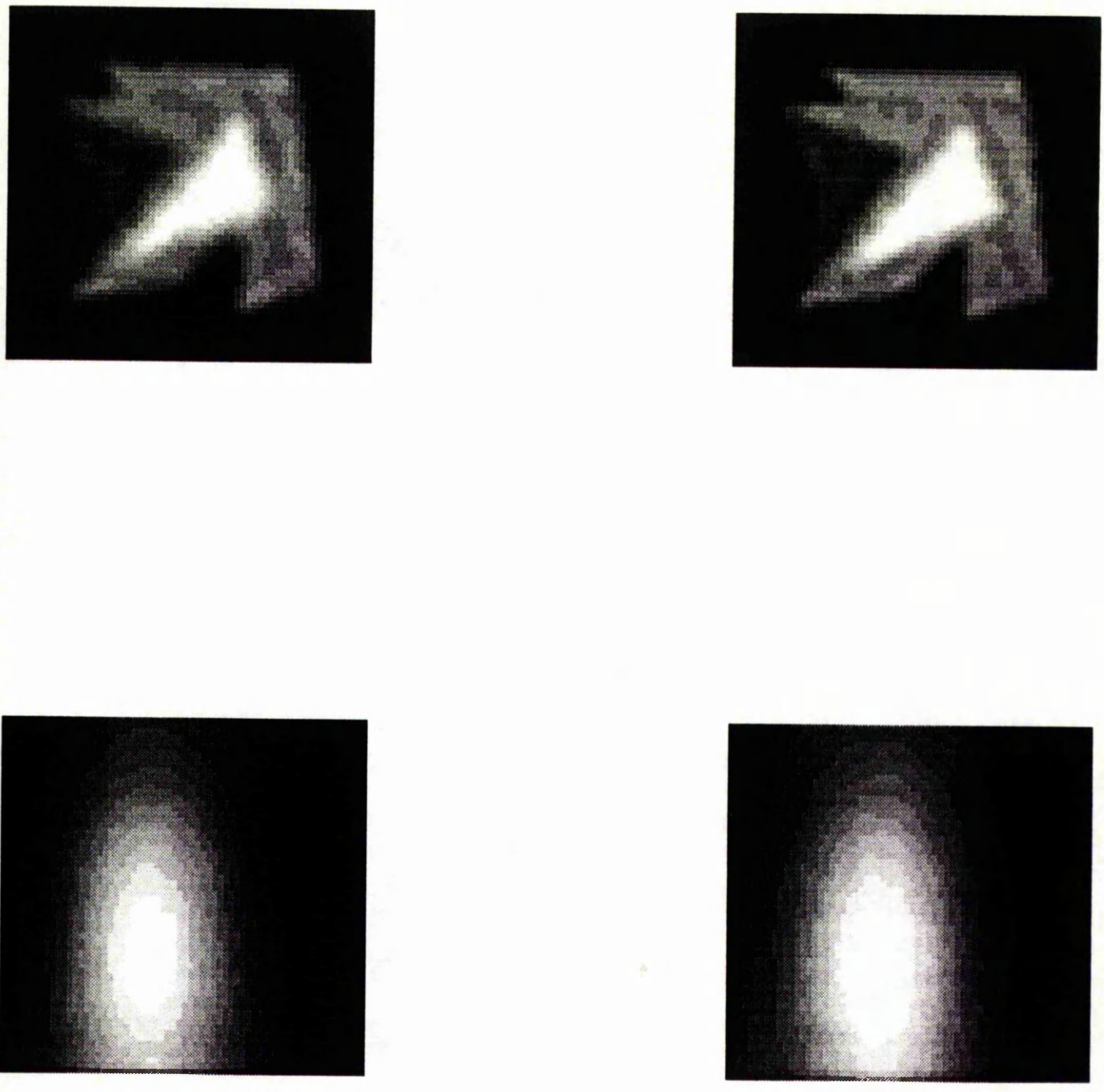


Figure 2.13: Some restorations from Algorithm **vague**. From top left to bottom right, we have (i) I5B2N2, (ii) I5B2N1, (iii) I6B1N2, and (iv) I6B1N2.

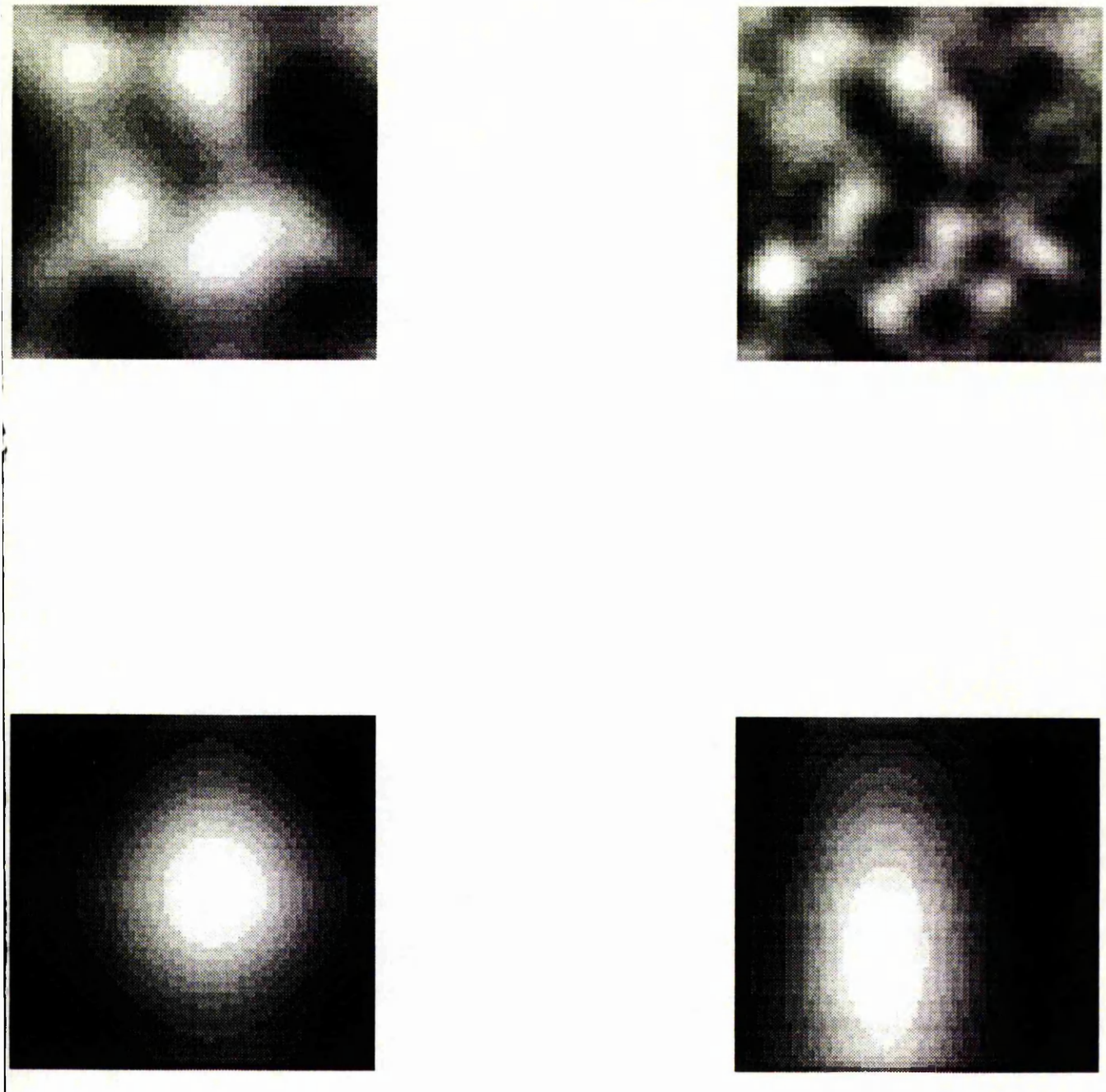


Figure 2.14: Some reconstructions from Algorithms **gamma** and **pseudo**. Top: Algorithm **gamma** - (i) I3B2N2, first gamma prior, (ii) I3B2N1, third gamma prior; and bottom: Algorithm **pseudo** - (iii) I5B1N2, and (iv) I6B1N1.

Chapter 3

Empirical Bayesian estimators, and other "plug–in" approaches

3.1 Introduction

In the previous chapter, we derived and implemented various hierarchical, iterative procedures for the estimation of the true image and unknown model parameters. Here we investigate a conceptually different method, a compromise between Bayesian and Maximum Likelihood estimation. A likelihood function is used to estimate the parameters (β, ϕ); then these m.l.e.'s are "plugged–in" to a formula for the m.a.p. estimate of x.

It is the reliance on the likelihood function of the data which leads to this method being known as the *empirical* Bayesian approach.

This two stage approach is implicitly advocated in [48], where the authers assert that the "normative" tool for parameter estimation is the likelihood function, and the "normative" tool for image restoration is the posterior density of x given y.

Various other plug–in estimates, many from the regularisation literature, will be compared with this technique.

3.2 The "plug–in" approach

As before, $\{p(x)\}$ is assumed to be a realisation of an m.r.f., $p(\beta), p(\phi)$ are assumed vague, with form given in (2.7),(2.9), and the records $\{y_i, i = 1, \ldots, n\}$ are formed by the same Gaussian degradation as in Chapter 2. The correct posterior distribution for inference is again (2.5) :

$$p(x, \phi, \beta \mid y) \propto p(y \mid x, \phi) \times p(x \mid \beta) \times p(\phi) \times p(\beta). \tag{3.1}$$

Since it is x, the true scene, which in practice will usually be of greatest interest, it is perhaps more natural to regard (ϕ, β) as nuisance parameters and integrate them out from the posterior, leaving:

$$p(x \mid y) = \int p(x, \phi, \beta \mid y) d\phi d\beta \tag{3.2}$$

to be maximised w.r.t. x at $x = \hat{x}_m$, the *marginal mode.*

The integration in (3.2) is not generally straightforward (however, see Section 3.4) and a common technique is to approximate the marginal density by

$$p(x, \hat{\phi}, \hat{\beta} \mid y) \tag{3.3}$$

where $(\hat{\phi}, \hat{\beta})$ are picked in some suitable manner. The approximation to the marginal mode is then (see (2.24) in Chapter 2.):

$$\begin{aligned} \hat{x}_m &= \text{argmax}_x \{p(y \mid x, \hat{\phi}) p(x \mid \hat{\beta})\} \\ &= (H^T H + 2\hat{\phi}\hat{\beta} A)^{-1} H^T y. \end{aligned}$$

3.3 Empirical Bayesian parameter estimates

Clearly we require good estimates for β and ϕ. Here, $(\hat{\phi}, \hat{\beta})$ are chosen to be m.l.e.'s based on the data y. The relevant likelihood function is $p(y \mid \phi, \beta)$, computed thus:

$$\begin{aligned} p(y \mid \phi, \beta) &= \int p(x, y \mid \phi, \beta) dx \\ &\propto \int p(y \mid x, \phi) p(x \mid \beta) p(\beta) p(\phi) dx \\ &= \int \phi^{-\frac{n+1}{2}} \beta^{\frac{r+1}{2}} \exp\{-\beta x^T A x - \frac{1}{2\phi} \|y - Hx\|^2\} dx \\ &= \phi^{-\frac{n+1}{2}} \beta^{\frac{r+1}{2}} \int \exp\{-\frac{1}{2}[2\beta x^T A x + (1/\phi)\|y - Hx\|^2]\} dx. \end{aligned}$$

In fact the $p(\beta) \times p(\phi)$ factor is included erroneously on line two of the above formulation, which has various effects on the subsequent discussion. The distribution $p(y \mid \phi, \beta)$ is improper (if $r < n$) and so the maximum likelihood estimate of β (i.e. ∞) is not helpful! This problem can be obviated by restricting the range of $p(y \mid \phi, \beta)$ to ensure a proper distribution.

If we manipulate the exponential term, easing matters by writing $\alpha = (1/\phi)$,$\gamma = 2\beta$, we see that

$$\begin{aligned} & \alpha\|y - Hx\|^2 + \gamma x^T A x \\ = \; & \alpha y^T y - 2\alpha y^T H x + \alpha x^T H^T H x + \gamma x^T A x \\ = \; & x^T[\alpha H^T H + \gamma A]x - 2\alpha y^T H x + \alpha y^T y. \end{aligned} \tag{3.4}$$

We would like to express (3.4) in quadratic form

$$(x - m)^T W (x - m) + b, \tag{3.5}$$

with b a scalar, in order to avail ourselves of results dealing with the integrals of such forms.

Now, $(3.5) = x^T W x - 2m^T W x + m^T W m + b$, so by identification with (3.4):

$$W = \alpha H^T H + \gamma A. \tag{3.6}$$

Bearing in mind the symmetry of H and A and therefore of W, $m^T W = \alpha y^T H$, giving $W^T m = \alpha H^T y = W m$, so that

$$m = \alpha(\alpha H^T H + \gamma A)^{-1} H^T y. \tag{3.7}$$

Finally, $m^T W m + b = \alpha y^T y$, so

$$\begin{aligned} b &= \alpha y^T y - m^T W m \\ &= \alpha y^T y - \alpha^2 y^T H W^{-1} H^T y \\ &= \alpha y^T y - \alpha^2 y^T H[\alpha H^T H + \gamma A]^{-1} H^T y. \end{aligned} \tag{3.8}$$

Neither the expression for m nor that for b involves x, so we may combine (3.6),(3.7),(3.8) to write the likelihood as

$$\begin{aligned} p(y \mid \phi, \beta) &= \phi^{-\frac{n+1}{2}} \beta^{\frac{r+1}{2}} e^{(-b/2)} \int \exp\{(-1/2)(x-m)^T W (x-m)\} dx \\ &\propto \phi^{-\frac{n+1}{2}} \beta^{\frac{r+1}{2}} e^{(-b/2)} \mid W \mid^{-1/2}, \end{aligned}$$

where $W = [(1/\phi) H^T H + 2\beta A]$ and $b = (1/\phi) y^T y - (1/\phi)^2 y^T H W^{-1} H^T y$.

The log–likelihood function for ϕ, β is therefore

$$l(\phi, \beta; y) \equiv \log\{p(y \mid \phi, \beta)\}.$$

Again, we appeal to the block–circulant approximation to the Toeplitz matrices H, A to see that we can write the log likelihood in the more tractable form:

$$\begin{aligned} l(\phi,\beta;y) &= -((n+1)/2)\log\phi + ((r+1)/2)\log\beta \\ &\quad -(1/2)\sum_{i=1}^{n}\log\{(1/\phi) \mid h_i \mid^2 +2\beta a_i\} - (1/2)\sum_{i=1}^{n}\frac{(1/\phi)2\beta a_i w_i^2}{(1/\phi) \mid h_i \mid^2 +2\beta a_i} \\ &= -((n+1)/2)\log\phi + ((r+1)/2)\log\beta \\ &\quad -(1/2)\sum_{i=1}^{n}\log\{(1/\phi) \mid h_i \mid^2 +2\beta a_i\} - \sum_{i=1}^{n}\frac{\beta a_i w_i^2}{\mid h_i \mid^2 +2\phi\beta a_i}, \end{aligned} \tag{3.9}$$

where $\{h_i\}, \{a_i\}$ are the eigenvalues of H, A respectively, obtained by carrying out a discrete f.f.t. on their first rows, and $\{w_i\}$ are the components of the discrete f.f.t. of the data y. The expression (3.9) will still require numerical maximisation. The estimates of the parameters thus found are labelled "e.b." (for Empirical Bayes) due to their reliance on the data likelihood.

Algorithm **eb**

1. Obtain $(\hat{\phi}_{eb}, \hat{\beta}_{eb}) = \text{argmax}_{\phi,\beta} l(\phi, \beta; y)$.

2. Calculate $\hat{x}_{eb} = \text{argmax}_x p(x \mid y, \hat{\phi}_{eb}, \hat{\beta}_{eb}) = (H^T H + \lambda_{eb} A)^{-1} H^T y$, where $\lambda_{eb} = 2\hat{\phi}_{eb}, \hat{\beta}_{eb}$. □

Numerical equivalence of Empirical Bayes and Marginal Modes estimates

We note here that we obtain the same numerical results from this e.b. motivation as would have been this case had we attempted to find the "marginal mode"

estimates for (ϕ, β), had we not included the $p(\beta) \times p(\phi)$ factor in calculating $p(y \mid \phi, \beta)$, through joint maximisation of

$$\begin{aligned} p(\phi, \beta \mid y) &= \int p(x, \phi, \beta \mid y) dx \\ &\propto p(y \mid \phi, \beta). \end{aligned}$$

Alternatively, we could have decided to form *separate* marginal densities for the two parameters, choosing $\phi = \hat{\phi}$ to maximise

$$p(\phi \mid y) = \int p(\phi, \beta \mid y) d\beta, \tag{3.10}$$

and $\beta = \hat{\beta}$ to maximise

$$p(\beta \mid y) = \int p(\phi, \beta \mid y) d\phi. \tag{3.11}$$

It is not possible to produce an estimate for the image directly from these two marginal densities. We make the transformation $(\phi, \beta) \longrightarrow (\phi, \lambda)$, where $\lambda = 2\phi\beta$. Transforming the joint marginal density of (ϕ, β) we obtain:

$$\begin{aligned} p(\phi, \lambda \mid y) &= \phi^{-(r+4)/2} \lambda^{(r+1)/2} \mid H^T H + \lambda A) \mid^{-1/2} \\ &\quad \times \exp[-(1/2\phi) y^T I - H(H^T H + \lambda A)^{-1} H^T y]. \end{aligned} \tag{3.12}$$

For fixed λ, this is maximised at

$$\hat{\phi}(\lambda) = (y^T \{I - H(H^T H + \lambda A)^{-1} H^T\} y)/(r + 4). \tag{3.13}$$

Then the posterior modes of (ϕ, λ) could be obtained by substituting (3.13) into (3.12), maximising numerically to find $\hat{\lambda}$, and substituting back into (3.13) to obtain $\hat{\phi}(\hat{\lambda})$.

Further, we could integrate ϕ from (3.12), to obtain

$$\begin{aligned} p(\lambda \mid y) &\propto \lambda^{(r+1)/2} \mid H^T H + \lambda A \mid^{-1/2} \\ &\quad [y^T \{I - H(H^T H + \lambda A)^{-1} H^T\} y]^{-\frac{r+2}{2}}, \end{aligned}$$

which could be maximised numerically to provide the marginal mode of λ. However, an explicit formula for $p(\phi \mid y)$ does not seem attainable, and so this procedure seems unworkable without excessive numerical computation.

3.4 Iteration from e.b. to m.a.p.

It will be recalled that motivation for Algorithm **eb** arose from the desire to obviate the need for integration in (3.2). To enable the evaluation of the approximation, we now consider a method for obtaining iteratively the modal estimate of x, i.e. the $\hat{x}_m$ which maximises (3.2). Well,

$$\begin{aligned} p(x \mid y) &= \int p(x, \phi, \beta \mid y) d\phi d\beta \\ &= \int p(x, \phi, \beta, y)/p(y) d\phi d\beta \\ &= (1/p(y)) \int p(y \mid x, \phi) p(\phi) d\phi \int p(x \mid \beta) p(\beta) d\beta \\ &= (1/p(y)) I_A I_B, \text{ say.} \end{aligned}$$

In what follows, we shall assume that $p(\phi) \propto c$, and $p(\beta) \propto d$, where c, d are constants.

$$\begin{aligned} I_A &= \int p(y \mid x, \phi) p(\phi) d\phi \\ &= c \int_0^\infty \phi^{-n/2} e^{(-1/2\phi)\|y - Hx\|^2} d\phi \\ &\propto \int_0^\infty \phi^{-n/2} e^{-z\phi^{-1}} d\phi, \end{aligned}$$

where $z = (1/2)\|y - Hx\|^2$. Now put $\eta = z/\phi$, so that $d\phi = -z\eta^{-2} d\eta$ and I_A becomes

$$\begin{aligned} I_A &\propto -\int_\infty^0 z^{-n/2} z \eta^{n/2} \eta^{-2} e^{-\eta} d\eta \\ &\propto -z^{-(n/2)+1} \int_\infty^0 \eta^{(n/2)-2} e^{-\eta} d\eta \\ &\propto z^{-((n/2)-1)} \int_0^\infty \eta^{(n/2)-2} e^{-\eta} d\eta \\ &= z^{-((n/2)-1)} \Gamma(p), \end{aligned}$$

where $p = (n/2) - 1$. For square or rectangular images, n will be even, so p will be an integer, and I_A is easily evaluated since $\Gamma(p) = (p-1)!$ in such cases.

Using a similar argument to the above, and writing $w = x^T A x$, it is easily seen that $I_B \propto w^{-q} \Gamma(q)$, for $q = (r/2) + 1$. We combine these results to see that

$$\begin{aligned} p(x \mid y) &\propto z^{-((n/2)-1)} w^{-((r/2)+1)} \\ &= \{(1/2)\|y - Hx\|^2\}^{-((n/2)-1)} \{x^T A x\}^{-((r/2)+1)}. \end{aligned}$$

The most probable estimate of the true image is the $x = \hat{x}_m$ which maximises

$L = \log p(x \mid y)$. Differentiating L w.r.t. x, we have

$$\frac{\partial L}{\partial x} = -\frac{r+2}{x^T A x} Ax - \frac{n-2}{2\|y - Hx\|^2}[2H^T Hx - 2(H^T y)^T], \tag{3.14}$$

and (3.14) is equal to zero when

$$\begin{aligned} \frac{2+r}{x^T A x} Ax + \frac{n-2}{\|y - Hx\|^2} H^T Hx &= \frac{n-2}{\|y - Hx\|^2} H^T y \\ \text{i.e. } \frac{(2+r)\|y - Hx\|^2}{(x^T Ax)(n-2)} Ax + H^T Hx &= H^T y \\ \text{i.e. } (H^T H + \lambda A)x &= H^T y. \end{aligned}$$

That is,

$$\hat{x}_m = (H^T H + \lambda A)^{-1} H^T y \tag{3.15}$$

where

$$\begin{aligned} \lambda &= \{(2+r)\|y - Hx\|^2\}/\{x^T Ax(n-2)\} \\ &\equiv \lambda(x). \end{aligned}$$

In (2.25), we defined $\lambda = 2\phi\beta$. Later, we shall define certain estimators of ϕ, β as $\phi_T = (\|y - Hx\|^2)/(n+1)$ and $\beta_T = (r+1)/(2x^T Ax)$. For large r, n (as is the case in image analysis), $\lambda(x)$ can therefore be seen to be nearly equivalent to $2\phi_T\beta_T$.

If we define $\hat{x}^{(s+1)} = (H^T H + \lambda(\hat{x}^{(s)})A)^{-1} H^T y$ and $\lambda(\hat{x}^{(s)}) = \{(2+r)\|y - H\hat{x}^{(s)}\|^2\}/\{\hat{x}^{(s)T} A\hat{x}^{(s)}(n-2)\}$, then the following iterative algorithm is suggested for estimation of the modal $\hat{x}_m$:

Algorithm **map**

1. Choose $\hat{x}^{(0)}$, and set $s := 0$.

2. Evaluate $\lambda(\hat{x}^{(s)})$.

3. Compute $\hat{x}^{(s+1)}$.

4. Check convergence of λ: if

   YES $\longrightarrow$ **STOP**

   NO $\longrightarrow$ set $s := s + 1$ and go to step (2). $\square$

In our numerical experiments we always chose the **eb** estimates as the starting point for this algorithm. As noted, we hope that this algorithm converges to the modal estimate of x. If it converges at all, it will at least be a solution of stationarity equation 3.14.

3.5 Discussion of the e.b., m.a.p. and hierarchical methods

Above, we developed an empirical Bayesian and a maximum a posteriori approach to our problem of blind image restoration. The e.b. approach integrates x over the true image space to form a likelihood from which we may estimate our two parameters. The m.a.p. approach, instead, integrates out the hyper–parameters and then maximises the marginal posterior probability distribution of interest.

Intuitively it may seem that the m.a.p. approach should perform better than e.b.; after all, we are really only co–incidentally interested in the values of (ϕ, β)

in order to have a good estimate of x. However, the results of our experiments (see Section 3.7) seem to contradict this intuition. Further, in the literature there is evidence that this should not be entirely unexpected.

In [67], Mackay presents a lively confrontation between the two approaches. Although he approximates posteriors with Gaussian distributions rather than employing the iterative methodology we have used, and considers the case of ϕ fully known, so that β is the only hyper–parameter, he concludes that "in severely ill–posed problems [such as blind image restoration] ... significant biases arise in the [m.a.p.] method".

He concludes that the estimator $\hat{x}_m$ is more regularised than $\hat{x}_{eb}$, and demonstrates the difference between $\hat{\beta}_m$ and $\hat{\beta}_{eb}$, analogous to the difference between the two estimators of the standard deviation of a normal distribution. The estimator $\hat{\sigma}_n^2 = (\sum_i (x_i - \bar{x})^2)/n$ is biased, while $\tilde{\sigma}_n^2 = (\sum_i (x_i - \bar{x})^2)/(n-1)$ is not. The novelty in his work is that he motivates the unbiased $\tilde{\sigma}_n^2$ as the empirical Bayesian estimate obtained after one has integrated over the nuisance parameter (the mean of the distribution).

In any case, a mode may be representative of its distribution in the case of a tight, symmetrical Gaussian, but, for a skewed, multimodal distribution, it is hard to see how it can be other than misleading. (This would be particularly relevant were we interested in prediction.)

Further support that our results in Chapter 2 (which were generally superior to those of Chapter 3, e.b. or m.a.p.) are not atypical but perhaps to be expected, can be found in [54]. Again, ϕ is assumed fully known, and again different methods are used in the estimation of x. The authors report good results from a

hierarchical procedure for the estimation of β, in a situation where they were unable to see how to proceed in an e.b. based manner. However, they also highlight the difficulties with all iterative procedures (interestingly, they discuss pseudo–likelihood, which for us performed worst), which base parameter estimates on current estimates of the image as though they were the truth.

In three related papers, Skilling and Gull ([83],[49], [84]) develop what is essentially an empirical Bayesian approach justified through the use of an entropic prior distribution, which they view as deductively unavoidable. They point out that faulty parameter values are at least as likely to be the fault of an incorrectly specified model as they are to be with any particular estimation procedure. The Maximum Entropy (MaxEnt) approach can be summarised (ignoring the philosophical considerations) as:

$$\text{argmax}_x\{-\sum_i x_i \log x_i\} \text{ subject to } \|y - Kx\|^2 \leq S^2, \tag{3.16}$$

or, equivalently,

$$\hat{x}_{me} = \text{argmin}_x\{\|y - Hx\|^2 + 2\lambda \sum_i x_i \log x_i\}. \tag{3.17}$$

(See [22]) Thus, MaxEnt can be seen as essentially equivalent to our formulation, save for the alteration of

$$\sum_{i\sim j}(x_i - x_j)^2 \longrightarrow \sum_i x_i \log x_i. \tag{3.18}$$

According to the authors in [22], the MaxEnt advantages are that the solution of (3.17) must be non–negative, and that the solution is non–linear in y. However, there is a disadvantage in terms of computational expense, and the restorations in the Gull and Skilling papers appear qualitatively no better than those we have presented above, and in Section 3.7.

Further, we remain attracted to the justification for the m.r.f. specification of the prior distribution of images – that pixels closer together should have a tendency towards the same value. While the MaxEnt authors cited have clearly demonstrated that *if* there is a consistent prior on the space of all images, *then* that distribution must be of the form $\exp\{\beta S(x, M)\}$, where $S(x, M)$ is the entropy of x relative to model M, the assertion that there must exist such a consistent prior for all images has not been demonstrated.

3.6 Connections with regularisation

The form of the Bayes estimate for the image, (2.24):

$$\hat{x} = (H^T H + \lambda A)^{-1} H^T y, \tag{3.19}$$

will be recognised as being equivalent to the ridge regression estimator of x. This *regularised* estimate is proposed in the regression literature when the least–squares estimate (LSE):

$$\hat{x}_{LS} = (H^T H)^{-1} H^T y \tag{3.20}$$

is unsuccessful, due to the ill–posedness of the problem. (The ill–posedness is due to the near linear dependence between columns of H, leading to instability of $(H^T H)^{-1}$, which in turn renders the LSE "spiky" (see [58], also [82], where the authors propose the combination of a smoothing step and the EM algorithm to regularise the estimate)).

Here we will discuss some of the history of the regularisation approach and examine some of the techniques developed for ill–posed regressions, such as we

deal with in image restoration. An excellent summary of many of the methods discussed here can be found in [93].

However, before proceeding with the similarities between the Bayesian and regularisation approaches, let us emphasise their difference. Providing one is prepared to form beliefs in a well–defined rational manner, the Bayesian paradigm for statistical inference is complete and coherent. On the other hand, regularisation of LSEs was proposed as an adhoc "fix" to ill–posed problems and as such regularised least–squares estimators (RLSEs) satisfy many various criteria. Indeed the least–squares approach itself, defended as it usually is on the three grounds of intuitive plausibility, equivalence with maximum likelihood estimates (MLEs) under the normal linear model (NLM), and the fact that LSEs are UMVUE (that is, of uniformly minimum variance within the class of unbiased estimators), has recently been exposed to serious Bayesian criticism ([97]). Thus, while a Bayesian may say :"$\hat{x}_m$ is the most probable estimate of x given y and our prior belief", classical statisticians, who deny themsleves the opportunity to refer to λ as a random variable, rely on various asymptotic *predictive* criteria when selecting an appropriate value for their parameter.

Philosophical worries apart, it remains that the MAP estimate of x and the RLSE are equivalent, in the sense that both select the x which solves:

$$\text{argmin}_x\{\Delta(x,y) + \lambda\Phi(x)\}, \tag{3.21}$$

where

1. Δ measures a "distance" between the data and the true image;

2. Φ is a regularisation functional which measures the roughness of the image -

this can take the entropic form discussed above, or the finite difference form which we employ;

3. λ, the smoothing parameter, measures the "trade-off" between Δ and Φ.

$\lambda = 0$, for a classical statistician, implies that the least–squares solution is acceptable; for a Bayesian it corresponds to the rather unlikely notion that we are *certain* that $\beta = 0$ in the m.r.f. prior.

Before proceeding, we introduce some helpful notation:

Notation

To emphasise the dependence of an image estimate on λ, write:

$$\hat{x}(\lambda) = (H^T H + \lambda A)^{-1} H^T y. \tag{3.22}$$

Also, we define the set of fitted values, or predictors, of y, as

$$\begin{aligned} y(\lambda) &= H\hat{x}(\lambda) \\ &= H(H^T H + \lambda A)^{-1} H^T y \\ &= N(\lambda) y, \text{ say,} \end{aligned}$$

and the residual sums–of–squares is defined obviously as:

$$RSS(\lambda) = \|y - y(\lambda)\|^2. \tag{3.23}$$

Hall and Titterington ([50]) pointed out that the main techniques of image regularisations are equivalent to simple linear regularisations: that is, the estimate of x_i is a smoothed version of the LSE of x_i, this smoothing being carried out by averaging over the (m.r.f.) neighbouring pixels. The same authors, in [51],

compared some popular choices of λ. First, λ_{risk}, defined as:

$$\lambda_{risk} = \text{argmin}_{\lambda} \mathrm{E}[\|\hat{x}(\lambda) - x\|^2]; \tag{3.24}$$

next, λ_{resid}, the value of λ such that

$$RSS(\lambda) = n\phi. \tag{3.25}$$

Clearly, in its present form, λ_{risk} is not practicable; although it minimises the mean squared error (MSE) between x and $\hat{x}$, it requires knowledge of x so to do. However, a cross–validatory estimate of the smoothing parameter, λ_{gcv}, has been proposed, in [39], which is asymptotically equivalent to λ_{risk}.

λ_{resid} requires either knowledge, or a very good estimate, of ϕ. Furthermore, since $y(\lambda)$ is biased for Hx, λ_{resid} will probably oversmooth the data. This led Wahba ([102]) to define the "equivalent degrees of freedom" as:

$$EDF(\lambda) = n - \mathrm{tr}N(\lambda), \tag{3.26}$$

and to estimate ϕ via

$$\hat{\phi}_{edf} = RSS(\lambda_{risk})/EDF(\lambda_{risk}). \tag{3.27}$$

Then a third choice for λ is λ_{edf}, defined as the solution to:

$$\mathrm{E}[RSS(\lambda_{edf})] = \hat{\phi}_{edf} EDF(\lambda_{edf}). \tag{3.28}$$

Hall and Titterington found that λ_{risk} and λ_{edf} were often equivalent, while λ_{risk} often led to over–smoothing, as expected.

The complete data–based estimate λ_{gcv} therefore became popular; however problems were found to exist with it. In [90], it was found that the cross–validatory function which is minimised to find $\lambda_{gcv}, G(\lambda)$:

$$G(\lambda) = RSS(\lambda)/[n - \mathrm{tr}N(\lambda)]^2, \tag{3.29}$$

could suffer through having multiple minima, or no minimum with $\lambda > 0$; or indeed a global minimum could exist, but which produced unsatisfactory results.

These problems had been noted in [91]: λ_{gcv} was prone to smallness and thus to under–smoothing, while λ_{resid} (more often known as λ_{chi}, due to the expected Chi–squared distribution of the residual sums–of–squares), over–smoothed, as expected.

In [60], a neighbourhood noise based estimator is proposed for ϕ and used in conjunction with these 3 methods. In an example using each of $\lambda_{gcv}, \lambda_{resid}, \lambda_{edf}$, all the estimators produced over–smooth $\hat{x}$. Theoretical considerations of signal–to–noise ratios in this paper, by Kay, show again why we may expect λ_{resid} to over–smooth. Further asymptotic work by Kay , in [61], shows that, relative to the mean–squared prediction error choice of λ_{tp}:

$$\lambda_{tp} = \text{argmin}_{\lambda} \mathrm{E}[\|H\hat{x}(\lambda) - Hx\|^2], \tag{3.30}$$

λ_{gcv} is asymptotically optimal, and $\lambda_{edf} < \lambda_{resid}$. However, since image restoration is a question of estimation rather than prediction, Kay points out that comparison with λ_{tp} is not the only possible criterion. However, similar empirical results to those in the previous cited works were noted.

In [29], despite confusion over the designation of posterior distributions and likelihood functions, it is shown that $\hat{x}(\lambda)$ improves the MSE properties of $\hat{x}$, relative to $\hat{x}(0) = \hat{x}_{LS}$. The claim of these authors, that "no rigorous justification of this assumption [that is, of the superiority of a regularised over a least–squares estimate] has appeared in the image restoration literature" is odd, given that most image restoration is carried out within the Bayesian paradigm, for which

very rigorous justification exists, and that within that paradigm it would be the specification of $\lambda = 0$ which would require explanation. Further theoretical work leads Galatsanos and Katsaggelos to conclude that the variance of $\hat{x}(\lambda)$ is a monotonically decreasing function of λ, and that $\text{var}[\hat{x}(\infty)] = 0$. Again, for a Bayesian, this is no more than what is required from the definition of λ (and hence β).

In the following section on numerical work, we will compare our hierarchical experiments from Chapter 2, and the **eb** and **map** methods from this chapter, with the three methods which seem most common from the literature: λ_{chi} (that is, λ_{resid}), λ_{edf} (the "fix" to λ_{chi}), and λ_{gcv} (the totally data–based estimate, asymptotically optimal relative to λ_{tp}).

3.7 Numerical work

We have detailed how to find estimates of λ, the smoothing parameter, through a variety of methods : **eb, map** (the Bayesian plug–in choices), CHI, EDF, GCV (some classical regularisation choices). Here we present the results of some numerical work to compare the different approaches.

To facilitate this comparison, we require some measure for the "success" or otherwise of a particular restoration, so we first turn attention to this issue.

3.7.1 Optimal choices for λ

If x were known, then natural (for statisticians) estimates of (ϕ, β) are given by

$$\hat{\phi}_T = \|y - Hx\|^2/(n+1)$$

$$\hat{\beta}_T \quad = \quad (2+r)/2x^T Ax$$

which combine to give λ_T and thus $\hat{x}_T$. Since this estimate requires knowledge of the true image (hence the "T" for "truth" subscript), it is rather impractical to say the least, but we include it here for comparison.

A standard measure of success in image restoration is the m.s.e. between an estimate and the truth. So if we define λ_{MSE} as

$$\lambda_{MSE} \quad = \quad \text{argmin}_\lambda \|x - \hat{x}(\lambda)\|^2$$

then $\hat{x}_{MSE}$ formed from such a λ_{MSE} will again provide a useful comparator. Note that this is essentially the λ_{risk} of the previous Section 3.6.

In Figure 3.1 we display some plots of λ against m.s.e., for some of our test images.

3.7.2 Error criterion

We define the m.s.e. per pixel between a target image x and its estimate $\hat{x}$ as

$$mse1 = mse(x, \hat{x}) = n^{-1} \sum_i (x_i - \hat{x}_i)^2. \tag{3.31}$$

The m.s.e. between x and noisy, blurred y is

$$mse2 = mse(x, y) = n^{-1} \sum_i (x_i - y_i)^2. \tag{3.32}$$

In the experiments which follow, we judge a restoration to be a success in the cases when $mse1 \leq mse2$. Although this is far from nonsensical, we must remember that m.s.e. is just one from an infinite range of error criteria. Not least among

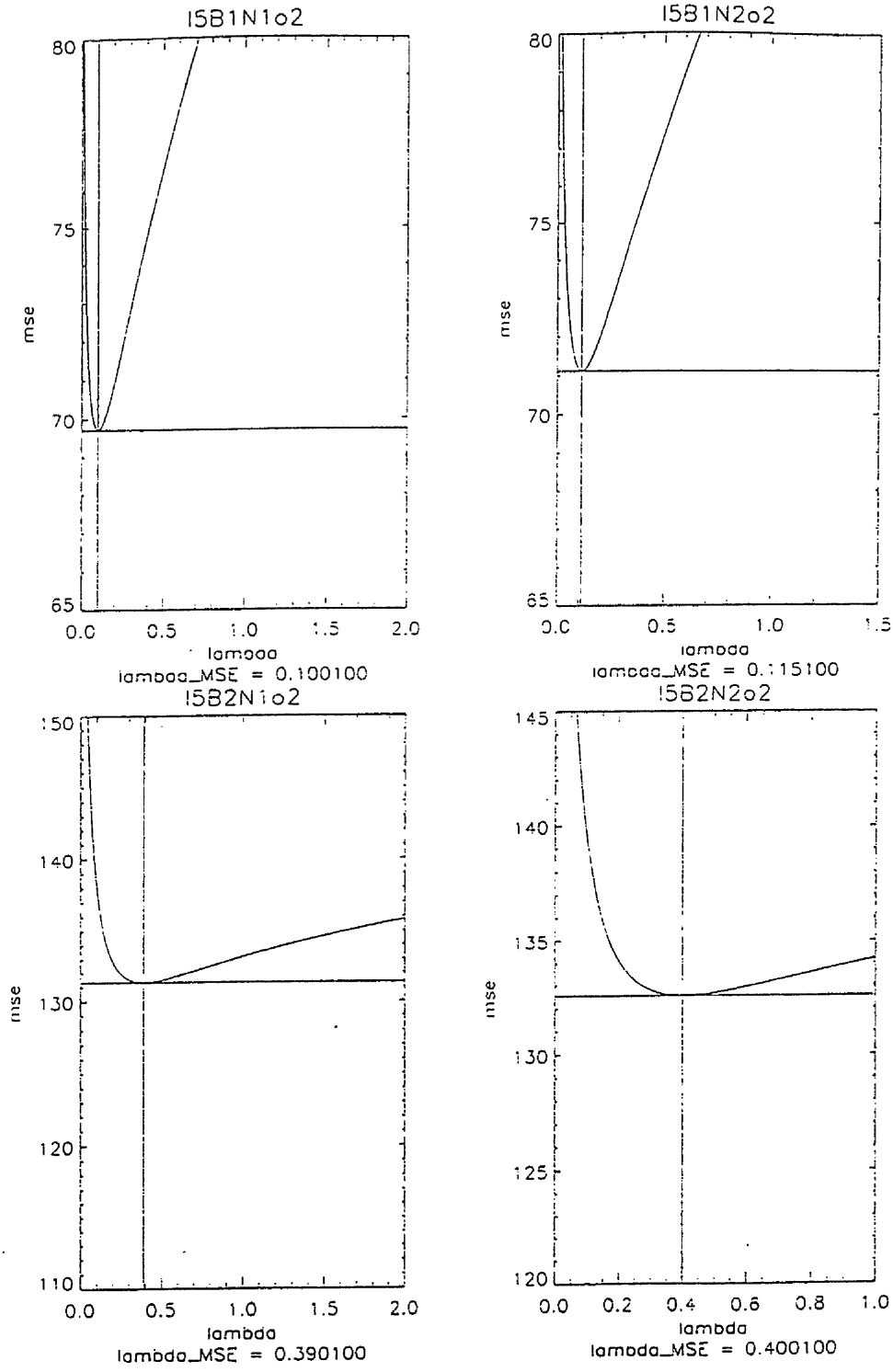


Figure 3.1: Plots of lambda against m.s.e.

the rest of the class is our own aesthetic judgement of a particular restoration: e.g. $\hat{x}_1$ may have higher $mse1$ than $\hat{x}_2$; yet we prefer $\hat{x}_1$, perhaps because the underlying discontinuities in the true x are more clearly reflected in the latter restoration.

3.7.3 Experimental details

4 test images: **I3, I4, I5, I6** were employed (see Chapter 2 : **I3,I4** are ostensibly realisations of m.r.f.'s, with $\beta = 0.05, 0.20$ respectively). Each image was degraded using blur levels **B1,B2** and noise levels **N1,N2**.

For each **IiBbNn**, i= $3,4,5,6$; **b,n**= $1,2$, two restorations for each of the methods was carried out: one where the m.r.f. was presumed to be of order 1, and another where the assumption was order 2. Recall that **I3,I4** were simulations from a 2nd order m.r.f.

3.7.4 Results

Parameter estimates:

eb and "truth" results for I3

y	order	β	ϕ	$\hat{\beta}_{eb}$	$\hat{\phi}_{eb}$	λ_{eb}	$\hat{\beta}_T$	$\hat{\phi}_T$	λ_T
B1N1	1	0.05	4.0	0.0992	3.9481	0.7832	0.0902	4.0003	0.7216
B1N1	2	0.05	4.0	0.0045	3.8352	0.0348	0.0437	4.0003	0.3493
B1N2	1	0.05	25.0	0.0395	24.0922	1.9045	0.0902	25.0016	4.5102
B1N2	2	0.05	25.0	0.0012	23.5542	0.0557	0.0437	25.0016	2.1829
B2N1	1	0.05	4.0	0.0679	3.8972	0.5291	0.0902	4.0003	0.7216
B2N1	2	0.05	4.0	0.0016	3.8302	0.0122	0.0437	4.0003	0.3493
B2N2	1	0.05	25.0	0.0212	24.1868	1.0236	0.0902	25.0016	4.5102
B2N2	2	0.05	25.0	0.0003	23.8558	0.0156	0.0437	25.0016	2.1829

other results for I3

y	order	β	ϕ	λ_{map}	λ_{chi}	λ_{edf}	λ_{gcv}	λ_{mse}
B1N1	1	0.05	4.0	0.1700	2.6201	0.3880	1.0754	1.100
B1N1	2	0.05	4.0	0.1873	0.0527	0.1080	0.1244	0.501
B1N2	1	0.05	25.0	1.1618	49.9300	6.1630	170.156	55.100
B1N2	2	0.05	25.0	1.2821	0.7668	13.99	18.4605	6.001
B2N1	1	0.05	4.0	0.1939	6.666	0.5543	4.9853	2.600
B2N1	2	0.05	4.0	0.1988	0.0460	0.32225	0.3629	0.501
B2N2	1	0.05	25.0	1.2625	585.5	92.66	800.0	409.100
B2N2	2	0.05	25.0	1.3317	24.10	3.04	32.5069	45.001

eb and "truth" results for I4

y	order	β	ϕ	$\hat{\beta}_{eb}$	$\hat{\phi}_{eb}$	λ_{eb}	$\hat{\beta}_T$	$\hat{\phi}_T$	λ_T
B1N1	1	0.20	4.0	0.1744	3.8879	1.3564	0.2531	4.0003	2.0249
B1N1	2	0.20	4.0	0.0063	3.7929	0.0478	0.1264	4.0003	1.0115
B1N2	1	0.20	25.0	0.0453	23.9874	2.1717	0.2531	25.0016	12.6556
B1N2	2	0.20	25.0	0.0013	23.4881	0.0593	0.1264	25.0016	6.3221
B2N1	1	0.20	4.0	0.0918	3.8869	0.7137	0.2531	4.0003	2.0249
B2N1	2	0.20	4.0	0.0018	3.8249	0.0137	0.1264	4.0003	1.0115
B2N2	1	0.20	25.0	0.0225	24.1740	1.0871	0.2531	25.0016	12.6556
B2N2	2	0.20	25.0	0.0003	23.8534	0.0159	0.1264	25.0016	6.3221

other results for I4

y	order	β	ϕ	λ_{map}	λ_{chi}	λ_{edf}	λ_{gcv}	λ_{mse}
B1N1	1	0.20	4.0	0.1666	22.01	4.526	8.985	13.100
B1N1	2	0.20	4.0	0.1821	0.5509	0.9752	1.0176	1.501
B1N2	1	0.20	25.0	1.1253	521.4	131.2	192.404	389.600
B1N2	2	0.20	25.0	1.1816	15.00	21.17	21.405	43.501
B2N1	1	0.20	4.0	0.1901	34.82	6.793	11.837	43.600
B2N1	2	0.20	4.0	0.1951	0.7666	1.081	1.0962	5.001
B2N2	1	0.20	25.0	1.2229	898.2	241.6	234.043	579.100
B2N2	2	0.20	25.0	1.2839	27.25	26.10	26.079	64.500

eb and "truth" results for I5

y	order	β	ϕ	$\hat{\beta}_{eb}^{(*)}$	$\hat{\phi}_{eb}$	$\lambda_{eb}^{(*)}$	$\hat{\beta}_T^{(*)}$	$\hat{\phi}_T$	λ_T
B1N1	1	–	4.0	3.3591	10.1750	68.359	30.7543	4.0003	0.0246
B1N1	2	–	4.0	0.3646	10.2958	7.5070	11.7464	4.0003	0.0094
B1N2	1	–	25.0	5.8134	33.6637	391.40	30.7543	25.0016	0.1538
B1N2	2	–	25.0	0.5223	32.7560	34.213	11.7464	25.0016	0.0587
B2N1	1	–	4.0	1.2163	3.8247	9.3038	30.7543	4.0003	0.0246
B2N1	2	–	4.0	0.1178	3.6325	0.8555	11.7464	4.0003	0.0094
B2N2	1	–	25.0	2.6282	26.0784	137.08	30.7543	25.0016	0.1538
B2N2	2	–	25.0	0.1694	24.9504	8.4522	11.7464	25.0016	0.0587

(*) The values in this column are $\times 10^{-4}$.

other results for I5

y	order	β	ϕ	$\lambda_{map}^{(*)}$	$\lambda_{chi}^{(*)}$	$\lambda_{edf}^{(*)}$	$\lambda_{gcv}^{(*)}$	λ_{mse}
B1N1	1	–	4.0	11.195	686.0	172.5	0116.1	0.9201
B1N1	2	–	4.0	12.457	74.64	18.18	14.9	0.1001
B1N2	1	–	25.0	18.785	3225.0	714.4	749.7	1.04510
B1N2	2	–	25.0	21.767	401.2	98.62	95.0	0.1151
B2N1	1	–	4.0	22.258	682.5	113.7	116.4	3.51510
B2N1	2	–	4.0	22.572	82.67	13.19	14.2	0.3901
B2N2	1	–	25.0	32.506	4898.0	857.2	900.3	3.6101
B2N2	2	–	25.0	34.207	282.7	32.31	58.7	0.4001

(*) The values in this column are $\times 10^{-4}$.

eb and "truth" results for I6

y	order	β	ϕ	$\hat{\beta}_{eb}^{(*)}$	$\hat{\phi}_{eb}$	$\lambda_{eb}^{(*)}$	$\hat{\beta}_T$	$\hat{\phi}_T$	λ_T
B1N1	1	–	4.0	4.4224	10.4217	92.178	0.0159	4.0003	0.1268
B1N1	2	–	4.0	0.4587	10.4708	9.6059	0.0052	4.0003	0.0418
B1N2	1	–	25.0	9.2891	34.2218	567.33	0.0159	25.0016	0.7926
B1N2	2	–	25.0	0.6879	33.1548	45.6121	0.0052	25.0016	0.2610
B2N1	1	–	4.0	0.5801	3.7595	4.3625	0.0159	4.0003	0.1268
B2N1	2	–	4.0	0.1000	4.1867	0.8373	0.0052	4.0003	0.0418
B2N2	1	–	25.0	2.5671	28.8946	148.35	0.0159	25.0016	0.7926
B2N2	2	–	25.0	0.1413	27.0900	7.6565	0.0052	25.0016	0.2610

(*) The values in this column are $\times 10^{-4}$

other results for I6

y	order	β	ϕ	$\lambda^{(*)}_{map}$	$\lambda^{(*)}_{chi}$	$\lambda^{(*)}_{edf}$	$\lambda^{(*)}_{gcv}$	λ_{mse}
B1N1	1	–	4.0	8.0136	12.44	7.63	6.87	4.0051
B1N1	2	–	4.0	8.8821	7.63	0.2134	1.62	0.4451
B1N2	1	–	25.0	14.540	15.26	252.5	1155.7	4.6401
B1N2	2	–	25.0	17.0827	7.63	38.97	158.7	0.5201
B2N1	1	–	4.0	19.631	7.64	2.282	2.101	41.00
B2N1	2	–	4.0	17.115	7.64	0.274	0.233	4.5551
B2N2	1	–	25.0	27.681	15.27	9.92	9.304	43.5
B2N2	2	–	25.0	26.986	7.639	1.204	1.094	4.8501

(*) The values in this column are $\times 10^{-4}$

Mean square errors:

$mse(x, \hat{x})$ **for I3**

y	order	$mse(x,y)$	eb	map	chi	edf	gcv	mse	truth
B1N1	1	5.141	1.667	1.908	1.657	1.673	1.623	1.622	1.626
B1N1	2	5.141	1.786	1.649	1.661	1.619	1.618	1.690	1.661
B1N2	1	26.018	2.726	3.053	2.129	2.275	2.139	2.128	2.338
B1N2	2	26.018	4.036	2.167	2.258	2.133	2.137	2.127	2.149
B2N1	1	5.440	2.014	2.301	1.897	1.957	1.887	1.876	1.926
B2N1	2	5.440	2.653	1.883	2.015	1.874	1.875	1.882	1.875
B2N2	1	26.310	3.503	3.308	2.189	2.203	2.191	2.188	2.555
B2N2	2	26.310	7.796	2.305	2.190	2.251	2.188	2.188	2.280

$mse(x, \hat{x})$ **for I4**

y	order	$mse(x,y)$	eb	map	chi	edf	gcv	mse	truth
B1N1	1	4.445	0.780	1.245	0.687	0.700	0.686	0.685	0.739
B1N1	2	4.445	0.986	0.765	0.698	0.686	0.686	0.684	0.686
B1N2	1	25.191	1.730	2.222	0.817	0.833	0.823	0.816	1.057
B1N2	2	25.191	3.179	1.097	0.832	0.823	0.823	0.816	0.874
B2N1	1	4.572	1.047	1.492	0.763	0.794	0.776	0.763	0.873
B2N1	2	4.572	1.795	0.898	0.794	0.782	0.781	0.762	0.784
B2N2	1	25.319	2.495	2.376	0.848	0.859	0.860	0.845	1.157
B2N2	2	25.319	6.939	1.177	0.858	0.859	0.859	0.845	0.941

$mse(x, \hat{x})$ **for I5**

y	order	$mse(x,y)$	eb	map	chi	edf	gcv	mse	truth
B1N1	1	85.613	133.049	244.196	85.062	108.858	118.565	70.241	101.375
B1N1	2	85.613	134.531	118.584	84.838	110.096	115.042	69.719	82.080
B1N2	1	106.015	101.222	277.219	75.050	90.893	90.125	71.567	80.905
B1N2	2	106.015	107.983	120.901	74.052	87.803	88.373	71.119	72.346
B2N1	1	137.247	456.593	364.786	191.908	258.292	257.149	131.594	225.326
B2N1	2	137.247	497.188	231.867	189.210	256.485	252.937	131.345	185.642
B2N2	1	158.108	270.645	408.935	148.696	191.827	190.277	132.788	174.650
B2N2	2	158.108	319.363	230.174	161.859	233.920	208.922	132.549	147.227

$mse(x, \hat{x})$ **for I6**

y	order	$mse(x,y)$	eb	map	chi	edf	gcv	mse	truth
B1N1	1	47.442	91.395	254.187	200.488	253.388	266.913	32.395	46.526
B1N1	2	47.442	94.771	97.369	102.930	628.077	197.989	32.394	38.561
B1N2	1	68.402	62.643	291.064	273.272	80.235	52.404	33.446	37.041
B1N2	2	68.402	70.273	98.565	140.997	74.466	50.445	33.460	33.971
B2N1	1	72.183	583.947	328.448	474.283	714.782	732.327	58.180	115.620
B2N1	2	72.183	496.854	178.872	219.903	780.094	833.344	58.173	95.103
B2N2	1	93.547	201.562	382.751	527.734	675.578	701.136	58.911	86.359
B2N2	2	93.547	266.208	177.926	267.371	718.885	765.428	58.906	73.813

3.7.5 Discussion of results

Methods which estimate β.

Method "T", which assumed knowledge of the true image, performed very well on I3 when the m.r.f. assumption was – correctly– that of a second order neighbourhood. However, for I4 the first order estimates are over biased, while the second order ones are underbiased, although the estimates are of the correct order of magnitude.

For I3 and I4, "e.b." estimates of β are much smaller than those produced by hierarchical Algorithms **vague, gamma**. However, for I5 and I6, $\hat{\beta}$ tends to be so small that one suspects the resultant image to be undersmoothed.

Gray, Kay and Titterington, in [42], had found that specification of m.r.f. order was of crucial importance in obtaining estimates of the parameter in a binary Ising model. Here, we note clear evidence of the same effect, for a prior model more complicated than Ising. Changing order specification from "one" to "two" does indeed decrease the value of $\hat{\beta}$; almost exactly halving it in the case of method "T". The effect, if anything, is even more marked for the artificial images which contain edges. Since the "T" method has an explicit formula for $\hat{\beta}$ it is not difficult to see why this should be the case for that procedure, but the **eb** results rely on numerical maximisation and so we find them more surprising and interesting.

Methods which estimate ϕ.

Not surprisingly, "T" estimation of ϕ performs extremely well across the range of test conditions. Estimation by the "e.b." method also produced good results, although once again we observe that variance is better estimated for the m.r.f. simulations than for the artificial images. There seems to be an increase in the precision of $\hat{\phi}$ the larger the p.s.f. and the larger the value of ϕ_{true}.

Estimation of λ.

As suspected, λ_{eb} is smaller in every instance than λ_{mse}, the λ which minimised $mse(x, \hat{x})$.

We would also conclude that λ_{chi} does indeed tend to oversmooth, as expected, and certainly when compared to the two Bayesian estimators λ_{eb} and λ_{map}.

The behaviour of the three "standard" estimators λ_{chi}, λ_{edf} and λ_{gcv} seems too erratic to generalise. For I3 and I4 the estimates were usually of the same order of magnitude as the "optimal" λ_{mse}, while λ_{eb} and λ_{map} were much smaller. For I5, I6 we see again that, in all but two of the tests, λ_{chi} was larger than either λ_{edf} or λ_{gcv}.

Mean Square Error between truth and restoration

Examining the m.s.e.'s for the m.r.f. simulations – and remembering that defining "success" by $mse(x, \hat{x}) < mse(x, y)$ is only one such measure – we see that every method was in this respect successful in every test condition. As examination of the λ estimates indicated, however, m.s.e. values tend to be slightly higher for methods "e.b." and "m.a.p." than for the other methods, particularly for I4.

With respect to the two artificial images, none of the methods can be judged a success. Strangely, all seemed to perform best under the same blur/noise combination: B1N2.

Some restorations can be seen in Figures 3.2 – 3.4.

The likelihood surfaces

Since our e.b. parameter estimates are maximum likelihood estimates, it could be revealing to examine the joint likelihood surface for some typical data.

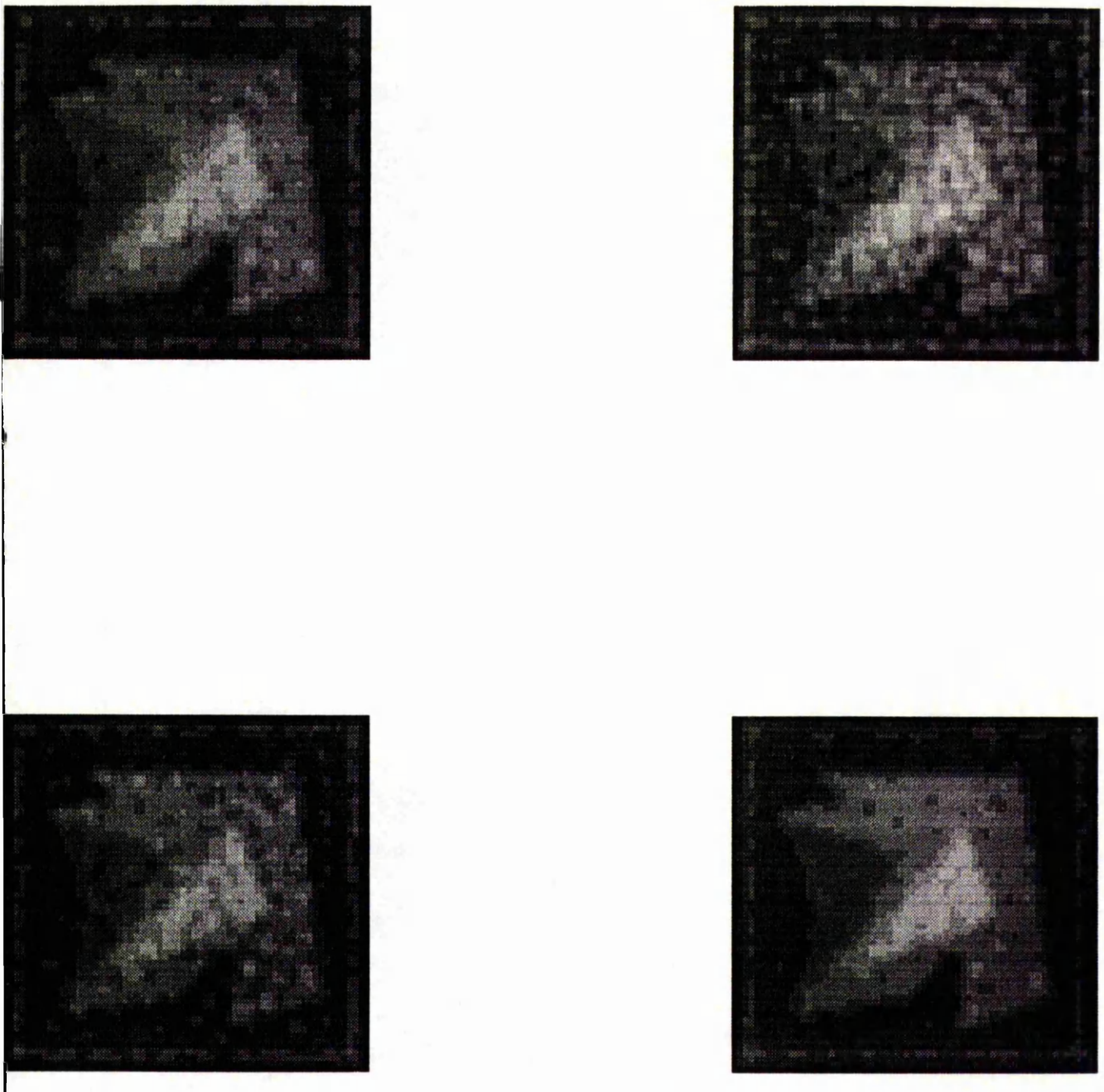


Figure 3.2: Some restorations from the algorithms in this chapter. From top left to bottom right: (i) I5B2N2, assumed mrf order 1, Algorithm **eb**, (ii), I5B2N2, mrf order 1, Algorithm **map**, (iii) I5B2N2, mrf order 2, Algorithm **eb**, and (iv) I5B2N2, mrf order 2, Algorithm **map**.

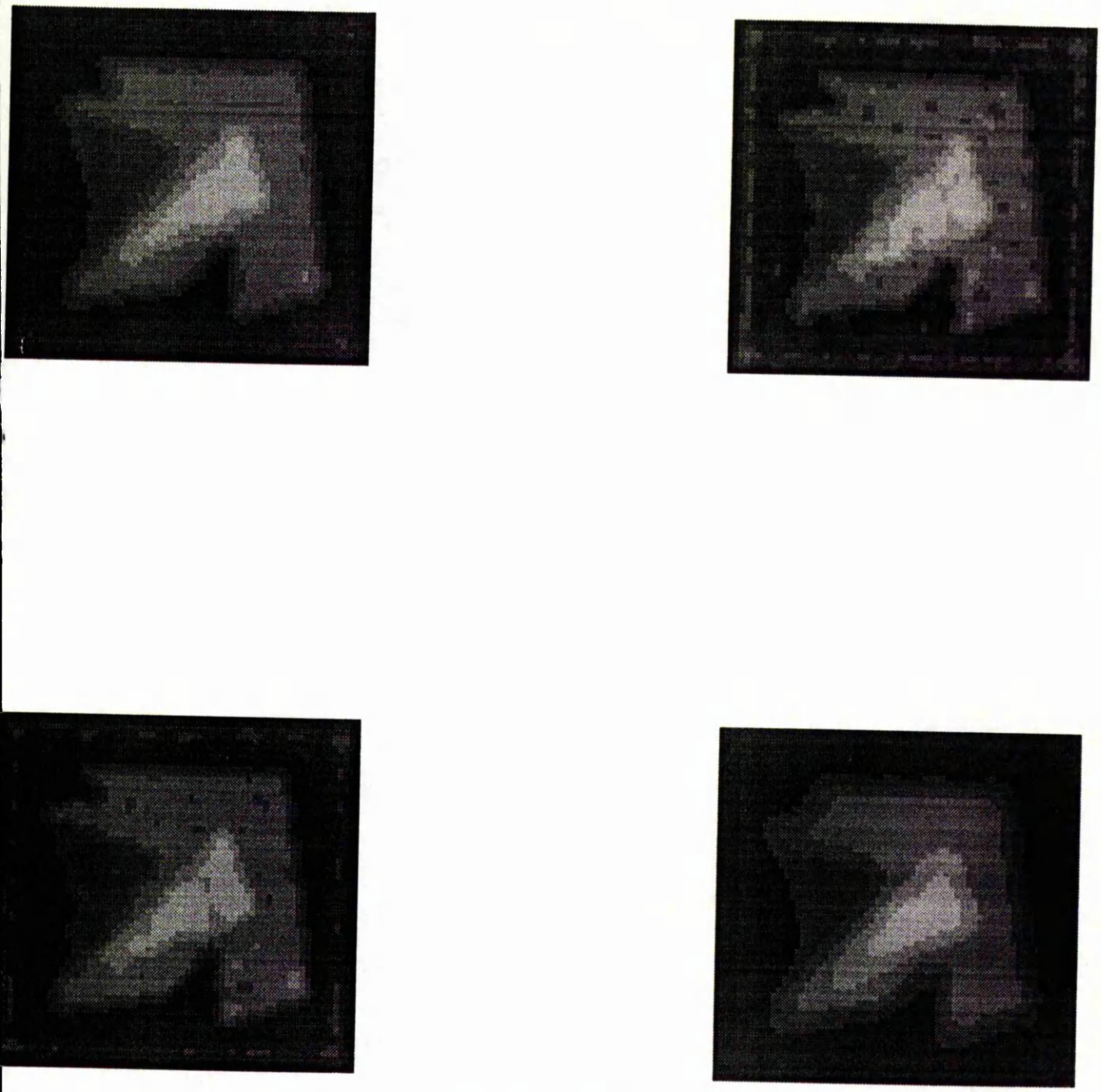


Figure 3.3: More restorations from the algorithms in this chapter. All data is image I5B2N2, true image is assumed to be mrf order 2. From top left to bottom right: (i) CHI, (ii) EDF, (iii) GCV and (iv) MSE.

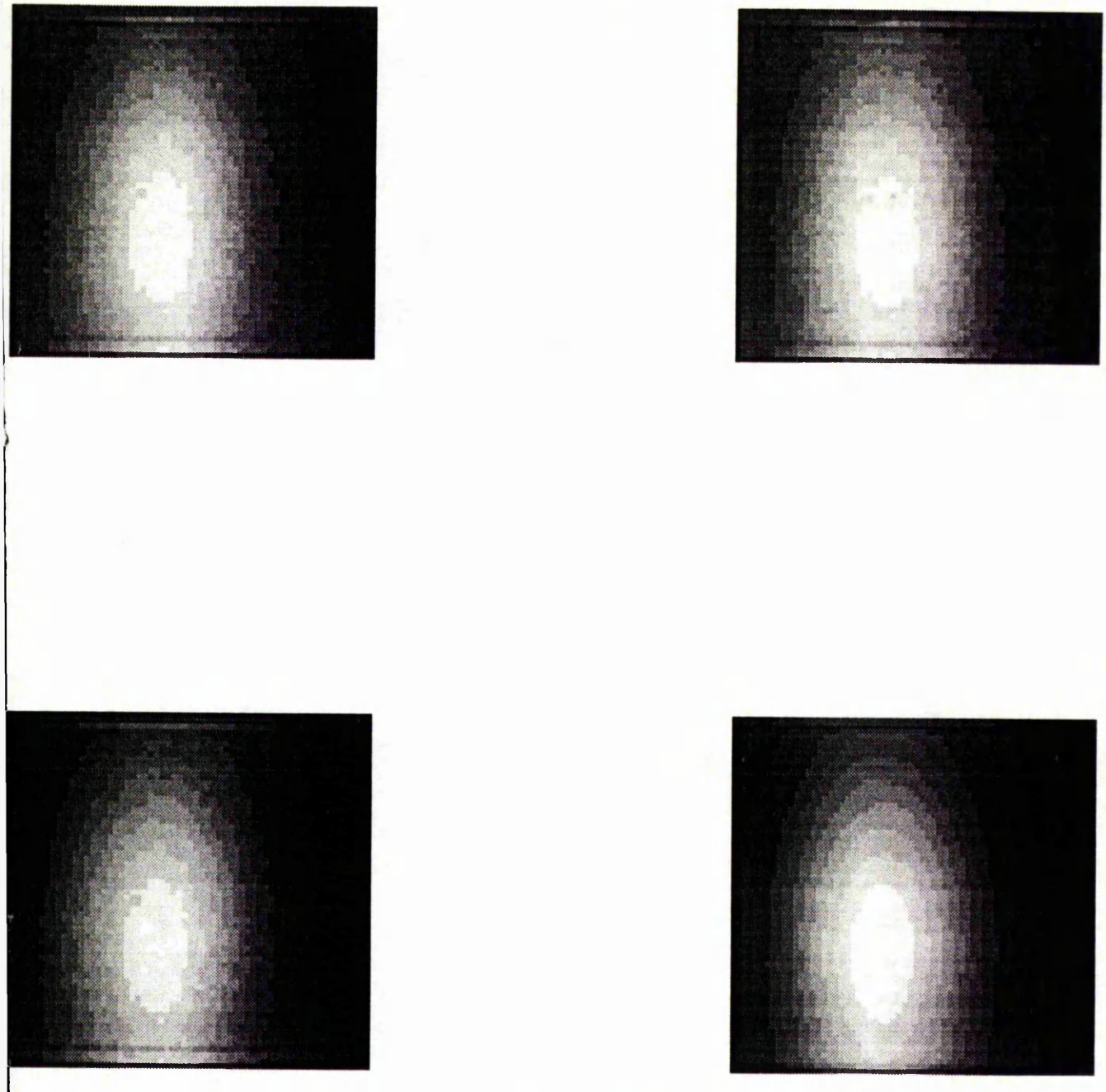


Figure 3.4: More restorations from the algorithms in this chapter. All data is I6B1N1, true image assumed to be an mrf of order 2. From top left to bottom right: (i) Algorithm **map**, (ii) CHI, (iii) Algorithm **eb**, and (iv) T.

In Figures 3.5 – 3.8 we present not only the joint likelihood function for a range of ϕ, β for two sets of data, but also some profile likelihood plots; it is clearly of interest to examine if the maximum of one profile is strongly affected by mis-specification of the other parameter.

Figure 3.5 shows the joint likelihood surface for image I5B2N2, assumed here to be an m.r.f. of order 1. Figure 3.6 displays the same for image I3B1N1, an m.r.f. of order 2. These images are very different in quality and distortion, and the likelihood surfaces reflect this. In the β axis in I3, there is a clear maximum - albeit at a much higher value than we would wish. No such maximum is visible in the I5 surface.

What is less clear however is that both have maxima along the ϕ axis. These become much clearer in the profile plots. A profile likelihood is the univariate likelihood obtained by fixing one parameter at a constant value and allowing the other to vary. For example, a profile likelihood function of ϕ is obtained from evaluating $p(\phi, \beta = \text{constant} \mid y)$. Plotting profile likelihood functions of ϕ against various values of β helps us determine if mis–estimating one parameter can lead to bad estimation of the other. For β, this does not seem to be the case; however,the shape of the ϕ profile alters dramatically depending on the value of β chosen. The value of β selected in the I5 profiles (Figure 3.7) is the estimate of β found by the e.b. algorithm. As was found in the experiments, this leads to a clear maximum of ϕ close to the correct value. With the I3 image (Figure 3.8) , we select the "correct" value of 0.05 for β (assuming our simulated image really *is* an observation of an m.r.f.) and are rewarded with a profile of ϕ with a clear maximum at the correct value of 4.0.

The message of these plots of likelihood surfaces would seem to be: correct likelihood estimation of β seems to be out of the question. However, given a "workable" value of $\hat{\beta}$, good estimates of ϕ are forthcoming.

It could be that the low–level Gibbs distribution prior is not able to capture well enough the features of a real image. A possible avenue of progress is to build higher–level priors, which model the features of an image, rather than the pixel values themselves. Such work is carried out in [79] and [89]. The vector of feature parameters would be called x, and a representation in terms of pixels, $z(x)$, could then be constructed, and compared with the data y, using Bayes' theorem.

3.8 Summary

We presented two Bayesian algorithms for image restoration: one where parameters are first estimated by maximum likelihood and an approximation to the modal estimate of x is found; and one where such a modal approximation is iterated towards the true m.a.p. estimate. We compared these methods with three plug–in choices from the regularisation literature, and two optimal choices. We found that the estimates of the smoothing parameter were nearly always smaller than the value which was optimal w.r.t. the mean–squared–error between x and $\hat{x}$. Examination of some likelihood surfaces shed some light on the reasons for difficulties w.r.t β estimation.

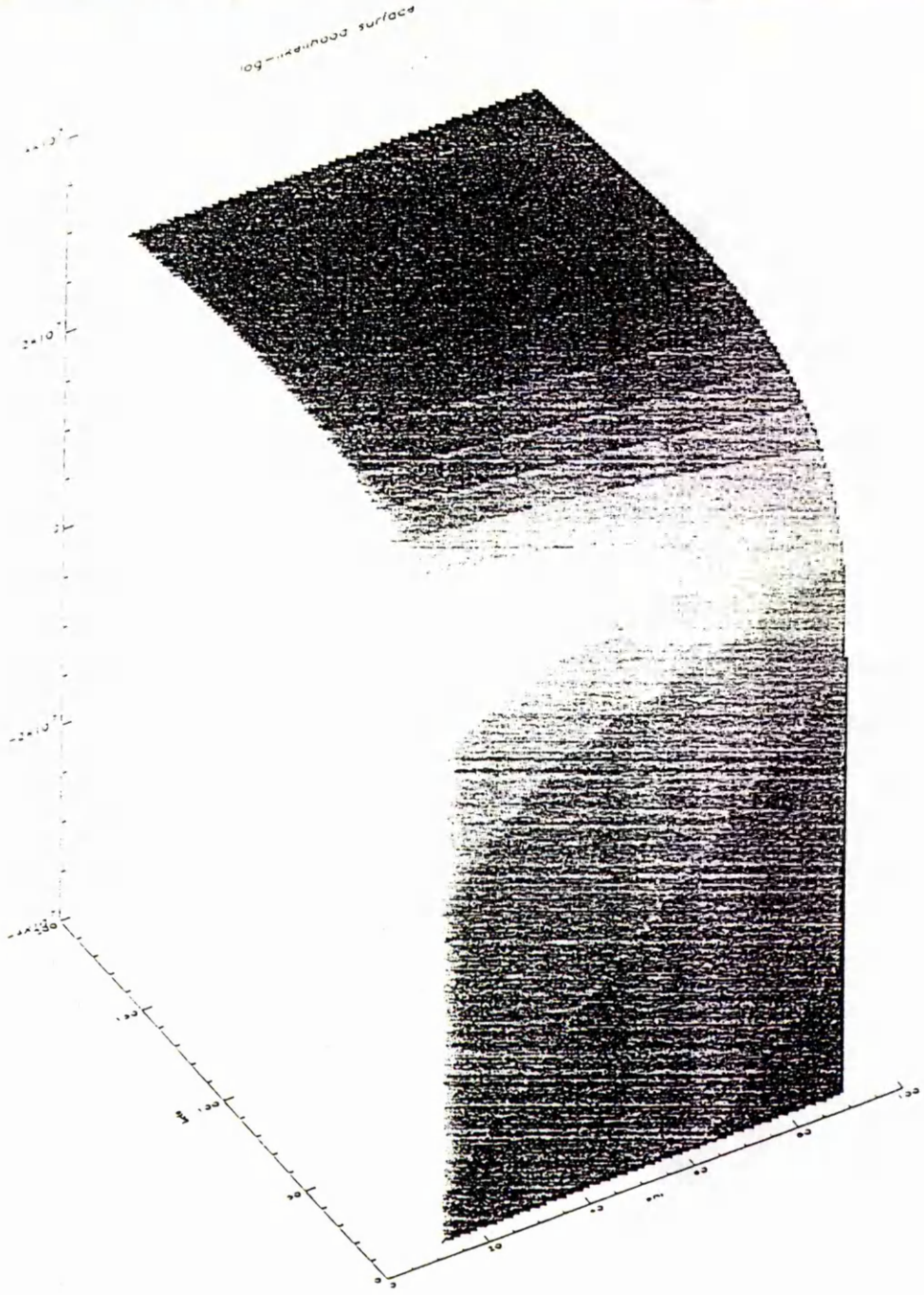


Figure 3.5: The likelihood surface for image I5B2N2

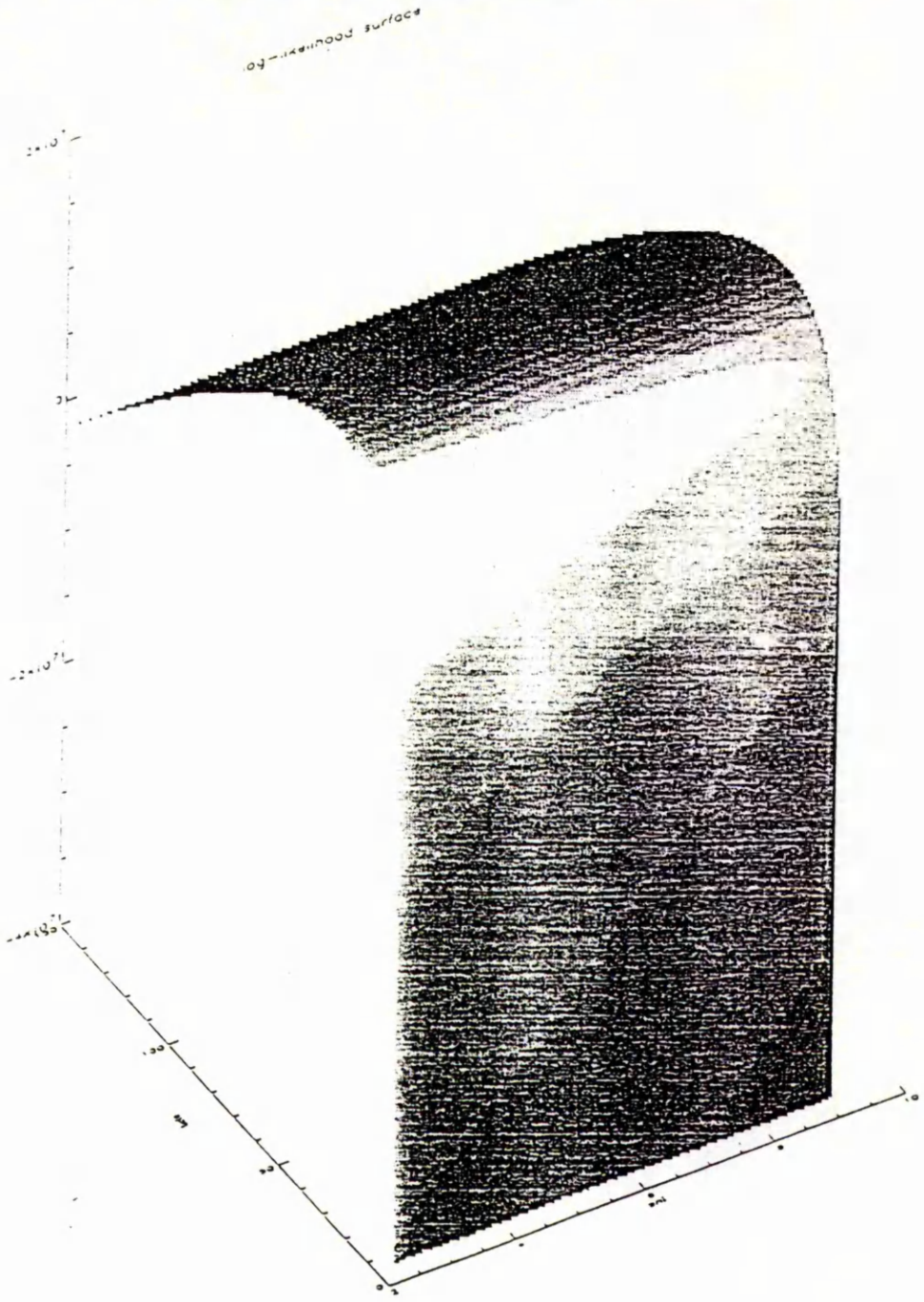


Figure 3.6: The likelihood surface for image I3B1N1

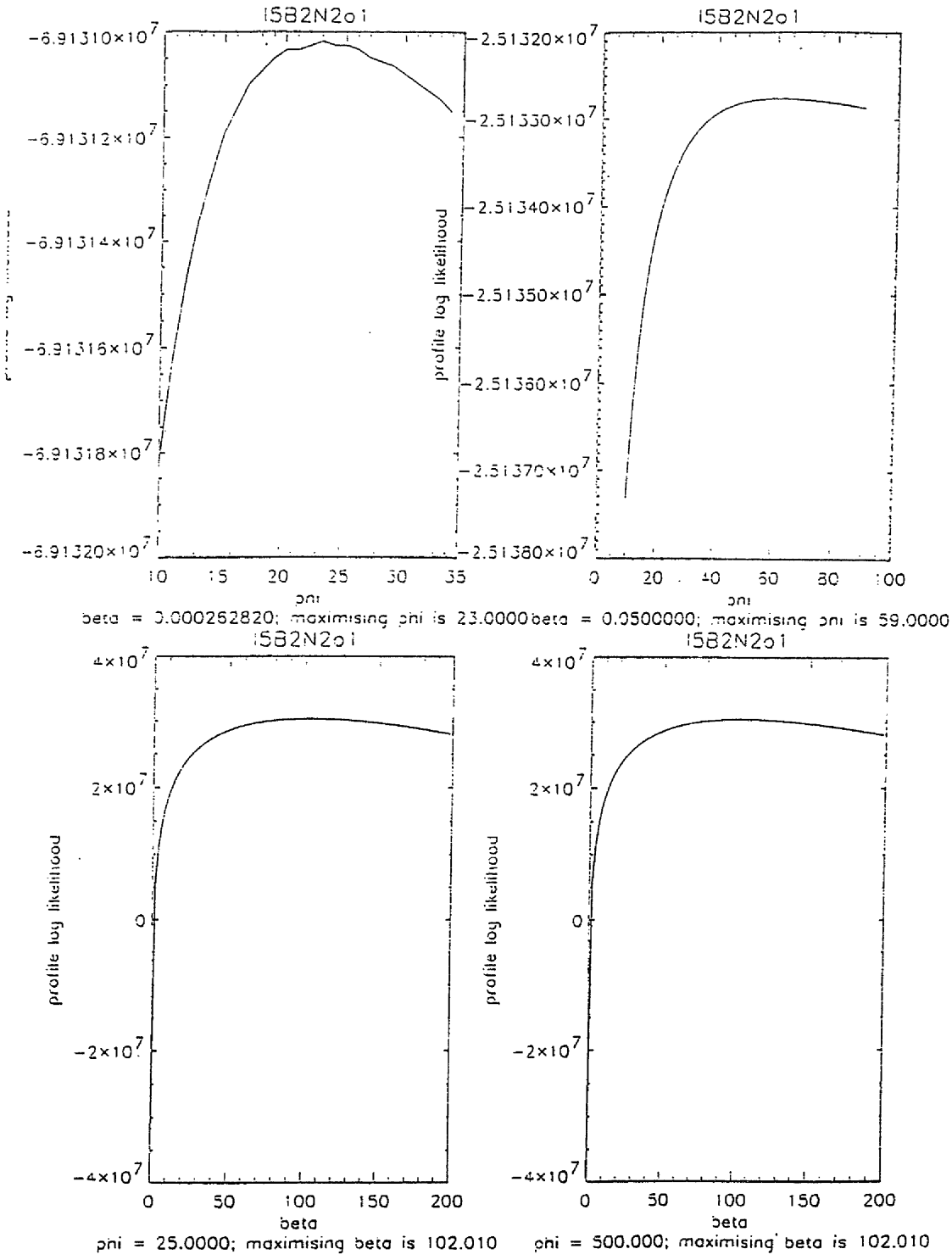


Figure 3.7: Some profile likelihoods from the I5 data

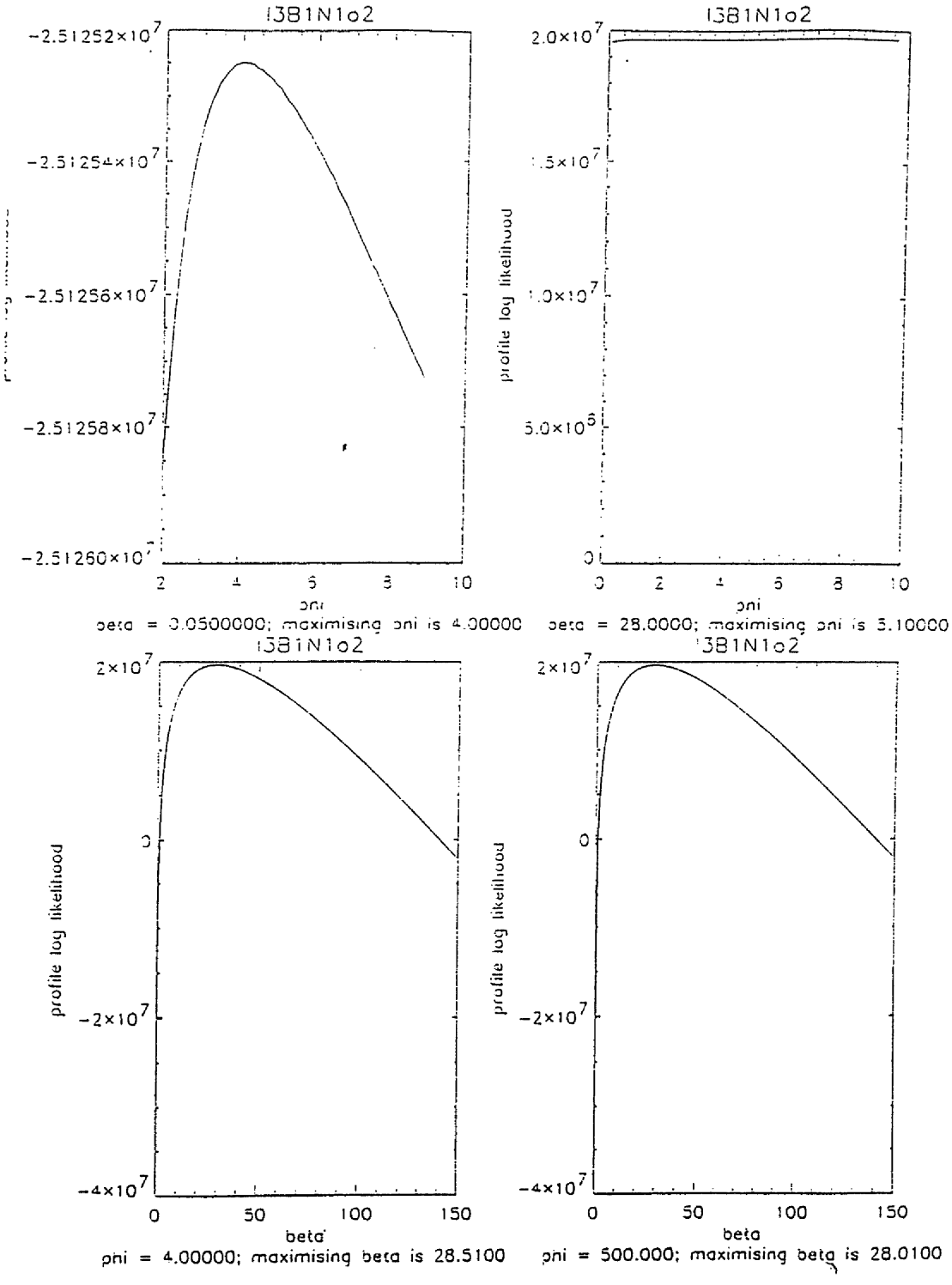


Figure 3.8: Some profile likelihoods from the I3 data

Chapter 4

The iterative Image Space Reconstruction Algorithm (ISRA)

4.1 Introduction

In the previous two chapters, we have tackled the problem of blind image restoration via a series of Bayesian parameter estimation methods. Different hierarchical and empirical approaches were investigated, but the general procedure was to: (1) estimate parameters within image and data models, and (2) use these estimates for some iterative or regularisation method of image restoration.

In this chapter we motivate a fundamentally different approach to the problem: viz, we address the issue of image restoration head–on, via the image space reconstruction algorithm (ISRA). The ISRA is motivated by a least–squares (LS) argument, and so requires no distributional assumptions, and therefore no parameter estimation.

We have already mentioned that there are many possible reasons why a LS solution may not be acceptable (see Section 3.6 and Citation [97]); however, the sum–of–squares distance *is* a commonly used metric in standard statistical

methodology and so it is arguably of interest to examine whether it is of practical use with the linear models commonly employed for image analysis. Furthermore, we seek to make no inference about the estimates of the image so found, so strict adherence to the Bayesian paradigm is perhaps not vital.

The structure of this chapter is as follows: since we will be comparing the ISRA with the better–known EMA ("Expectation – Maximisation Algorithm", [17]), we begin with a discussion of that algorithm. In particular we review the work of Vardi and Lee ([98]), hereafter referred to as "VL", who recently presented work that vastly increased the scope of the EMA. We examine the relationship between the two algorithms, and we propose that the ISRA be used for solving a range of LININPOS (linear inverse problems with positivity restrictions) as defined by VL. Image restoration is an example of a LININPOS, and we demonstrate both algorithms on examples of images distorted by linear motion blur.

We will discuss modifications possible for the ISRA (including the application of a Bayesian framework) . In particular, we discuss recent work on speeding up convergence of the EM algorithm ([36],[56],[71]) and apply similar methods to the ISRA. We discuss convergence properties of the ISRA, and show the difficulties involved in proving convergence for the altered ISRA.

The chapter concludes with a discussion of the results and some ideas for further work.

4.2 The EM algorithm

Since we wish to compare the ISRA to the EM algorithm, we here outline the latter's methodology, and the extensions of VL which make it a general solution to LININPOS.

The standard EM ("Expectation–Maximisation") algorithm runs as follows: suppose we wish to estimate a parameter $\theta \in \Theta$, which specifies the p.d.f. of a set of r.v.'s Z, viz, $f(z \mid \theta)$. Were we in possession of a set of realisations $\{z\}$, then estimation of θ, all other things being equal, would be straightforward. However, we suppose that Z is decomposed into $Z = (X, Y)$, where Y are observed random variables, and the X are somehow "missing". EM is an iterative procedure, with each iteration consisting of two stages:

E–step. Compute

$$Q^{(m)}(\theta) = \mathrm{E}\{\log f(X, Y \mid \theta) \mid y, \theta^{(m)}\}. \tag{4.1}$$

M–step. Evaluate

$$\theta^{(m+1)} = \mathrm{argmax}_{\theta} Q^{(m)}(\theta). \tag{4.2}$$

Initialise with some $\theta^{(0)} > 0$, and the sequence of estimates for $m = 1, 2, 3, \ldots$, can be shown to converge to at least a local maximum of the likelihood function of z, and the total likelihood increases at every iteration.

The algorithm has been applied in many statistical arenas and commonly in image analysis; see, for example, [75], [76] and [92].

The work of VL was to extend this algorithm to solve any LININPOS, which

may be written as

$$y(w) = \int_{D_x} h(v,w)x(v)dv, w \in D_y, \tag{4.3}$$

where D_x, D_y are the domains of functions x, y, which are both non–negative and real–valued. The model (4.3) describes the distortion of a signal, x, by a function h: the LININPOS is to recover x given h, y.

Equation (4.3) is also the formula for a mixture of probability densities, as well as being the law of iterated expectations: in fact, it can be generalised to cover a vast range of mathematical and statistical set–ups if we write it as

$$Y(w) = \int_{D_x} X(dv)H(v,w), w \in D_y \tag{4.4}$$

where $Y(\cdot), H(v,\cdot), v \in D_x$, are measures on D_y and X is a measure on D_x. This general model (4.4) contains the following discrete representation of a noise–free version of the image degradation model we have been considering:

$$y_j = \sum_{i=1}^{M} x_i h_{ij}, j \in D_y, \tag{4.5}$$

where $D_x = \{1, 2, \ldots, M\}, D_y = \{1, 2, \ldots, N\}$. Note that, in the case where $\{h_{ij}\}$ is the p.s.f. of a blurring process, the sum on the right hand side of (4.5) need only be over the members of B_j, where B_j is the blurring neighbourhood as defined in section (2.2). (This definition of the blurring process leads to a blurring matrix which is the transpose of that used in earlier chapters. We adopt this formulation to ease comparison of our work with that of Vardi and Lee ([98]).)

VL's EM solution to the recovery of the signal $x(v)$ (i.e. the vector of values of the true image x – we have not previously distinguished between the vector of pixel values and the image itself, but do so here to emphasise the generality of the argument) is to consider $\{(v_j, w_j) : j = 1, 2, \ldots, N\}$ as a set of complete data

with joint density $x(v)h(v,w)$ on $D_x \times D_y$. Of course we observe only $\{w_j\}$. The nonparametric maximum likelihood estimate of the distribution function of x is

$$\hat{X}(A) = N^{-1} \sum_j 1_A(V_j), A \subseteq D_x, \tag{4.6}$$

where 1_A is the indicator function applied to region A. This takes the part of the M–step. The E–step involves replacing $1_A(V_j)$ with its conditional expectation $\Pr(V_j \in A \mid W_j, x) = \int f_{V|W}(v \mid W_j)dv$. Combining the E– and M–steps by substituting the conditional expectation within the formula for $\hat{X}$ we obtain the iterative formula

$$X^{(m)}(A) = \int_A [N^{-1} \sum_j \frac{h(v,W_j)x^{(m-1)}(v)}{\int_{D_x} h(s,W_j)x^{(m-1)}(s)ds}]dv. \tag{4.7}$$

We take derivatives of (4.7) with respect to the Lebesgue measure to obtain the probability densities

$$x^{(m)}(v) = N^{-1}x^{(m-1)}(v) \sum_j [h(v,W_j)/\int_{D_x} h(s,W_j)x^{(m-1)}(s)ds], \tag{4.8}$$

to which we may apply the strong law of large numbers to see that, as $N \longrightarrow \infty$,

$$x^{(m)}(v) = x^{(m-1)}(v) \int_{D_y} \frac{h(v,w)}{\int_{D_x} h(s,W_j)x^{(m-1)}(s)ds} y(w)dw. \tag{4.9}$$

The discrete version of (4.9), suitable for image restoration, is

The EMA

1. Choose $\hat{x}^{(0)} > 0$.

2. For $m = 1, 2, \ldots$, compute

$$x_i^{(m)} = x_i^{(m-1)} \sum_{j=1}^{N} (h_{ij}/\sum_{s=1}^{M} x_s^{(m-1)}h_{sj})y_j, \tag{4.10}$$

for $i = 1, 2, \ldots, M$. □.

VL show that the solution of (4.5) computed via (4.10) will minimise the Kullback–Leibler divergence between the left– and right–hand sides of (4.5).

The EMA has had a long and popular history within statistical image analysis. In [99] a statistical model for emission tomography, which we shall use in our next section to motivate the ISRA, was implemented via the EMA. However, Little and Rubin, in [66], warn of the dangers of equating missing data (which are r.v.'s) with fixed parameters and caution against the over–usage of EM procedures: they argue that parameters should be estimated from an integrated likelihood function, i.e. $L(\theta \mid y) = \int f(x, y \mid \theta) dx$, essentially the **eb** approach of the previous chapter.

4.3 The ISRA

4.3.1 Development of the discrete ISRA

The ISRA was first proposed by Daube–Witherspoon and Muehllehner ([15]) as a method of image reconstruction for emission computed tomography (ECT) data. In fact they proposed it as an improvement of an EM algorithm ([80]) commonly used in such cases, and found in some experiments that it seemed to converge, initially, more rapidly than EM, and required fewer computations, although they offered no theoretical convergence results. Titterington ([94]) showed that the ISRA can be motivated as, and proven to converge to, the least–squares solution of the image restoration problem. In the ECT context, x_i labels the number of emissions from the i'th of M sources, $\{y_j\}$ are the data collected by the set of N detectors, and the $\{h_{ij}\}$ are numbers such that $\Pr\{\text{an emission from } i \text{ is detected at } j\} = h_{ij}$. Thus $\sum_j h_{ij} = 1$ and $\mathrm{E}(y_j) = \sum_j x_i h_{ij}$. It is assumed that the $\{h_{ij}\}$ are

known *a priori*.

Consider minimisation of $y_j = \sum_i x_i h_{ij}$, for $j = 1, \ldots, N$, with respect to the sum–of–squares function:

$$\begin{aligned} S(x) &= \|y - Hx\|^2 \\ &= \sum_{j=1}^{N} (y_j - \sum_{i=1}^{M} x_i h_{ij})^2. \end{aligned} \tag{4.11}$$

Any minimiser $\hat{x}$ of $S(x)$ will satisfy

$$\begin{aligned} HH^T \hat{x} &= Hy \\ \text{i.e. } \sum_j h_{ij} (\sum_{s=1}^{M} \hat{x}_s h_{sj}) &= \sum_j h_{ij} y_j, \end{aligned}$$

for $i = 1, \ldots, M$. Thus

$$1 = (\sum_j h_{ij} y_j) / [\sum_j h_{ij} (\sum_s \hat{x}_s h_{sj})], \tag{4.12}$$

for $i = 1, \ldots, M$. Multiplying both sides of (4.12) by $\hat{x}_i$ clearly motivates the following algorithm, which is the discrete ISRA:

The discrete iterative ISRA

1. Choose $\hat{x}^{(0)} > 0$.

2. For $m = 1, 2, \ldots$, compute

$$\hat{x}_i^{(m)} = \hat{x}_i^{(m-1)} [\frac{\sum_j h_{ij} y_j}{\sum_j h_{ij} \sum_s \hat{x}_s^{(m-1)} h_{sj}}], \tag{4.13}$$

for $i = 1, \ldots, M$. □.

4.3.2 The continuous ISRA

Examination of the continuous general models for LININPOS, (4.3, 4.4), leads to the conclusion that the ISRA can be extended to deal with such problems in the same manner as the EMA. For the continuous ISRA, the appropriate sum–of–squares function is

$$S(x) = \int \{y(w) - \int x(s)h(s,w)ds\}^2 dw, \tag{4.14}$$

and application of the calculus of variations provides stationarity equations

$$\int h(v,w) \int x(s)h(s,w)dsdw = \int h(v,w)y(w)dw. \tag{4.15}$$

In the same manner that the discrete ISRA evolved from (4.12), so (4.15) leads directly to the following iteration:

$$x^{(m)}(v) = x^{(m-1)}(v) \frac{\int_{D_y} h(v,w)y(w)dw}{\int_{D_y} h(v,w)[\int_{D_x} x^{(m-1)}(s)h(s,w)ds]dw}, v \in D_x, \tag{4.16}$$

initialised by $x^{(0)} > 0$, for all $v \in D_x$.

Both the discrete and continuous formulae for the ISRA hold whether or not the measures $X(\cdot)$ and $\{H(v,\cdot) : v \in D_x\}$ are probability measures, whereas the VL formulae require normalisation modifications in order to cover the case of general, non–negative measures.

Clearly, in both the discrete and continuous cases, if the initial $x^{(0)} > 0$, then $x^{(m)} \geq 0$ for all m, in view of the non–negativity of $Y(\cdot)$ and $\{H(v,\cdot)\}$. Even if the latter are all probability measures, there is no guarantee that $x^{(m)}$ is a probability measure for any m, except possibly for $m = 0$ by design, but, if the algorithms converge to a solution of (4.3) or (4.5) then the limit will be a probability measure if the same is true for $Y(\cdot)$ and $\{H(v,\cdot)\}$. If only the $\{H(v,\cdot)\}$ are probability

measures, then the algorithm converges to a solution with total measure equal to that of Y. In principle, we could constrain the total measure associated with X using a Lagrange multiplier (c.f. Section 4.5.2 of [95]), but this seems an unnecessary complication in view of the limiting behaviour.

An interesting point about both the ISRA and the EMA is that, although here we are concerned with the statistical estimation of distorted images, they are both in fact applicable in a vast range of non–statistical applications.

4.3.3 Convergence of the ISRA

Convergence aspects of the discrete ISRA are discussed by Titterington ([94]) and De Pierro ([18],[19]), but a more general discussion is presented by Eggermont ([26]) and it is this argument we summarise below.

Define $D_r = \text{diag}\{x_i^{(r)}/HH^T x_i^{(r)}, i = 1,\ldots,M\}$. Then, by Lemma 6.1 of Eggermont ([26]),

$$S(x^{(m-1)}) - S(x^{(m)}) \geq (x^{(m-1)} - x^{(m)})^T D_{m-1}^{-1}(x^{(m-1)} - x^{(m)}), \tag{4.17}$$

where $S(x)$ is the sum-of-squares function as defined in (4.11). So at each iteration the sum–of–squares function decreases and since, for every m, $\{x \geq 0 : S(x) \leq S(x^{(0)})\}$ is a compact set, $\{x^{(m)}\}$ is bounded and every subsequence itself contains a convergent subsequence.

Now let $\hat{x}$ be any point of accumulation of $\{x^{(m)}\}$, let $\Lambda = \text{diag}\{(Hy)_i, i = 1,\ldots,M\}$, let

$$\Delta_{KL}(v;w) = \sum_{i=1}^{M} v_i \log(v_i/w_i) + w_i - v_i, \tag{4.18}$$

the Kullback–Leibler directed divergence for $v, w > 0$, and let

$$e(\hat{x};x) = \Delta_{KL}(\Lambda\hat{x};\Lambda x) + S(x) - S(\hat{x}). \tag{4.19}$$

Then Lemma 6.2 of Eggermont ([26]) shows that

$$e(\hat{x}; x^{(m-1)}) \geq e(\hat{x}; x^{(m)}). \tag{4.20}$$

The convergence properties of $\{x^{(m)}\}$ imply that $e(\hat{x}; x^{(m)}) \longrightarrow 0$ as $m \longrightarrow \infty$, and so, therefore, does $\Delta_{KL}(\Lambda\hat{x}; \Lambda x^{(m)})$. The convergence of the Kullback–Leibler distance shows that $\{x^{(m)}\} \longrightarrow \hat{x}$, and since $\hat{x}$ can be shown to be a minimiser of $S(x)$, subject to $x \geq 0$, we have enough to state the following theorem:

Theorem : ISRA convergence

The ISRA, defined by (4.5) and initialised by $x^{(0)} > 0$, generates a sequence $\{x^{(m)}\}$ that converges to an $\hat{x}$ which minimises $S(x)$ subject to $x \geq 0$. □.

Extension beyond the discrete case proceeds along similar lines to those in Section 3.4 of VL: the main steps are sketched below.

The case of finite D_x

Suppose $D_x = \{1, \ldots, M\}$, that

$$y(w) = \sum_{i=1}^{M} h_i(w) x_i, w \in D_y \tag{4.21}$$

and that $\{x^{(m)}\}$ are generated according to

$$x_i^{(m)} = x_i^{(m-1)} \frac{\int_{D_y} h_i(w) y(w) dw}{\int_{D_y} h_i(w) [\sum_{s=1}^{M} x_s^{(r)} h_s(w)] dw}, i = 1, \ldots, M, \tag{4.22}$$

starting from $x^{(0)} > 0$. Then (4.17) and (4.19) hold with

$$S(x) = \int_{D_y} (y(w) - \sum_i h_i(w) x_i)^2 dw, \tag{4.23}$$

$$D_r = \text{ diag}\{x_i^{(r)} / [\int_{D_y} h_i(w) (\sum_{s=1}^{M} x_s^{(r-1)} h_s(w)) dw], i = 1, \ldots, M\}. \tag{4.24}$$

and $\Lambda = \text{diag}\{\int_{D_y} h_i(w)y(w)dw, i = 1, \ldots, M\}$. The argument of our theorem on the ISRA convergence then confirms that $\{x^{(m)}\}$ converges to an $\hat{x}$ that minimises $S(x)$, as defined by (4.23), subject to $x \geq 0$. □.

The case of continuous D_x.

As in the case of Section 3.4.2 of VL, we approach this by way of a discretization method. Suppose $y(\cdot)$ and $h(v, \cdot)$ are non–negative, integrable functions on D_y and suppose

$$y(w) = \int_{D_x} h(v, w)x(v)dv, v \in D_x \tag{4.25}$$

has a non–negative solution that is piecewise constant over the measurable partition $\{B_1, \ldots, B_{M'}\}$ of D_x ($\mu(B_i) > 0$, all i). Then, for any refinement $\{A_1, \ldots, A_M\}$ of $\{B_1, \ldots, B_{M'}\}$ ($\mu(A_i) > 0$, all i), the ISRA

$$\lambda_i^{(m)} = \lambda_i^{(m-1)} \frac{\int H_2(A_i, w)y(w)dw}{\int_{D_y} H_2(A_i, w)(\sum_s \lambda_s^{(m-1)} H_2(A_s, w))dw}, \tag{4.26}$$

$i = 1, \ldots, M, m = 1, 2, \ldots$, initialised by $\lambda^{(0)} > 0$, converges to a limiting value $\lambda^*(\geq 0)$, and

$$x_M^*(v) = \sum_{i=1}^{M} \lambda_i^* 1_{A_i}(v), v \in D_x, \tag{4.27}$$

is a solution of (4.25). In (4.26), $H_2(A_i, w) = \int_{A_i} h(v, w)dv$, all i, and in (4.27), $1_{A_i}(\cdot)$ is the indicator function for A_i, all i. This result follows from the results obtained in this section, and the final steps to the continuous case are parallel to those in Section 3.5 of VL. □.

4.4 Examples of possible applications

In this section we list briefly versions of the ISRA and the EMA for a variety of cases, drawing heavily from the examples in VL, since we aim for a comparison between the two algorithms. In particular we compare the ISRA and the EMA with respect to image restoration, for the case when linear motion blur has distorted an image, as discussed in VL.

4.4.1 Inversion of Simple Linear Equations

This example, though trivial, illustrates elementary behaviour of the ISRA and EMA in solving under–determined linear systems.

Case 1. One equation in two unknowns

Suppose we wish to find x_1, x_2 to solve

$$y = h_1 x_1 + h_2 x_2. \tag{4.28}$$

The ISRA can easily be shown to be

$$x_i^{(m)} = x_i^{(m-1)} y / [h_1 x_1^{(m-1)} + h_2 x_2^{(m-1)}], i = 1, 2. \tag{4.29}$$

Thus, for any $x^{(0)}, h_1 x_1^{(1)} + h_2 x_2^{(1)} = y$, so that the ISRA converges in one step. The same is easily shown of the EM algorithm.

Case 2. Two equations in three unknowns

In this case, neither algorithm converges at once. For illustrative purposes, consider the equations

$$y_1 = x_1 + x_2$$

$$y_2 = x_2 + x_3$$

with $y_1 = y_2 = 2$. Clearly, any x of the form $x^T = (x_1, 2 - x_1, x_1)$ solves these equations. The ISRA is

$$\begin{aligned} x_1^{(m)} &= 2x_1^{(m-1)}/(x_1^{(m-1)} + x_2^{(m-1)}) \\ x_2^{(m)} &= 4x_2^{(m-1)}/(x_1^{(m-1)} + 2x_2^{(m-1)} + x_3^{(m-1)}) \\ x_3^{(m)} &= 2x_3^{(m-1)}/(x_2^{(m-1)} + x_3^{(m-1)}). \end{aligned}$$

The first and third equations for the EM algorithm are the same as those for the ISRA, whereas the second one is

$$x_2^{(m)} = x_2^{(m-1)}[(x_1^{(m-1)} + x_2^{(m-1)})^{-1} + (x_2^{(m-1)} + x_3^{(m-1)})^{-1}].$$

Table (4.4.1) compares the algorithms in terms of the number of iterations required to obtain the limiting point correct to three decimal places in each component of $\hat{x}$. If $x^{(0)} = (a, a, a)$, for any a, each algorithm converges at once to $\hat{x} = (1, 1, 1)$. If $x^{(0)} = (a, b, a)$, for $b \neq a$, each algorithm also converges at once, to the same $\hat{x}$ for both algorithms. Otherwise, the algorithms converge to different, but similar, $\hat{x}$'s, in roughly the same number of iterations as each other.

Table 4.4.1

$x^{(0)}$	$(\hat{x}_1, \hat{x}_2)_{\text{ISRA}}$	$(\hat{x}_1, \hat{x}_2)_{\text{EM}}$	Iters (ISRA)	Iters (EM)
(0.5,1.0,0.5)	(0.951,1.049)	(0.906,1.094)	12	13
(0.5,1.5,1.0)	(0.646,1.354)	(0.636,1.364)	18	18
(1.5,0.5,1.0)	(1.425,0.575)	(1.416,0.584)	6	7
(0.1,1.0,9.9)	(0.648,1.352)	(0.664,1.336)	10	20
(1.0,0.1,9.9)	(1.963,0.037)	(1.899,0.101)	3	3
(1.0,9.9,0.1)	(0.063,1.937)	(0.061,1.939)	190	182

Case 3. Three equations in four unknowns

In this case, we discovered an example where the ISRA performs well and the EMA badly. Consider the equations

$$\begin{aligned} y_1 &= x_1 + x_2 + x_4 \\ y_2 &= x_2 + x_3 + x_4 \\ y_3 &= x_1 + x_4 \end{aligned}$$

If values $(x_1, x_2, x_3, x_4) = (2, 4, 5, 9)$ are selected, then $y_1 = 15, y_2 = 18, y_3 = 11$. Using these values of y to ensure a solution exists, we applied the discrete EMA and ISRA of (4.10) and (4.13) respectively, with $N = 3, M = 4$. We used different starting values for $\hat{x}$ and allowed the algorithms to run until third decimal place convergence was achieved. We also calculated the fitted values, $\hat{y} = H\hat{x}^{(final)}$. We chose three sets of starting values for both algorithms; set 1 was $(2.5, 3.5, 4.5, 8.5)$, set 2 was $(1.0, 100.0, 100.0, 50.0)$ and set 3 was $(50.0, 75.0, 20.0, 10.0)$.

The ISRA performed well over a wide range of starting values, while the EMA could not find a good solution at all. Table 4.4.2 summarises the results for both algorithms.

Table 4.4.2

Alg.	$x^{(0)}$	$\hat{x}$	$\hat{y}$	Iters to converge
ISRA	set 1	(2.316, 3.980, 5.327, 8.694)	(14.990, 18.002, 11.010)	48
	set 2	(0.809, 4.020, 3.796, 10.182)	(15.010, 17.997, 10.991)	101
	set 3	(8.301, 4.019, 11.290, 2.689)	(15.009, 17.999, 10.990)	107
EM	set 1	(0, 0.002, 0, 43.998)	(44.0, 44.0, 43.998)	30
	set 2	(0, 0.003, 0, 43.997)	(44.0, 44.0, 43.997)	32
	set 3	(0, 0.002, 0, 43.998)	(44.0, 44.0, 43.998)	37

4.4.2 Portfolio Optimisation

This subject is dealt with in VL, Section 3.1. VL use a different notation, which is in fact unnecessary: for each i, let x_i be the proportion of total assets to be invested in stock i , the j'th column of H contains one of N possible sets of returns from the M stocks, and y_i is the probability that the j'th set of returns will materialise, $j = 1, \ldots, N$. Clearly, the ISRA is precisely that of (4.13). The approach of VL is to maximise, in our notation,

$$W(x) = \sum_j y_j \log(\sum_i x_i h_{ij}). \tag{4.30}$$

In fact, any maximising $\hat{x}$ satisfies $y = H^T\hat{x}$, the basic set of linear equations.

This example is basically the same as the example of grouped data considered in Section 3.2 of VL.

4.4.3 Emission Tomography

(VL: Section 3.3, and see also [80].) Again, (4.13) is the relevant ISRA, if we use the notation we declared in Section 4.3.1. This problem is similar to estimating the mixing weights of a mixture of M known multinomials, each defined on a sample space of N categories. In that case, x is the set of mixing weights, y is

the proportion of observations that fall into category j, and the i'th row of H contains the i'th "pure" multinomial distribution.

Titterington and Rossi ([96]) noticed the relationship between these two problems in the context of the EMA, building on the earlier work of Di Gesu and Maccarone ([21]). Woodward et al. ([104]) compare an EMA for estimating the mixing proportions with a minimum distance technique, which selects an estimate to minimise some distance between the empirical c.d.f. and the chosen family of theoretical c.d.f.'s. The EMA outperformed minimum distance in cases of Gaussian mixes, but was shown to suffer from a lack of robustness.

4.4.4 Mixtures

(VL: Section 3.4.)

The special case of mixtures of multinomials is dealt with in Section 4.4.3. For more general finite mixtures, VL distinguish between two cases related to the "estimation" of the mixing weights x_i corresponding to the following version of (4.4):

$$Y(\cdot) = \sum_i x_i H(i, \cdot). \tag{4.31}$$

Suppose $h_i(\cdot)$ denotes the density associated with $H(i, \cdot)$. If the problem is the statistical estimation of the $\{x_i\}$ from a random sample $W_1, \cdots, W_N$ from the mixture, and if Y_N denotes the corresponding empirical distribution, then the ISRA is

$$x_i^{(m)} = x_i^{(m-1)} \frac{\int h_i(w) dY_N(w)}{\int h_i(w)[\sum_s x_s^{(m-1)} h_s(w)]dw}. \tag{4.32}$$

If, on the other hand, one is simply inverting (4.31), then the ISRA is (4.32), with Y_N replaced by Y.

4.4.5 Convolutions and Motion Blurring

(VL: Sections 4 and 5.)

In this example, $h(v,w) \equiv h(w-v)$. We follow VL in concentrating, for simplicity, on the case of one–dimensional images, so that $x(\cdot), y(\cdot)$ denote, respectively, the unblurred and blurred images, and $\Gamma = \{\gamma(t), 0 \leq t \leq T\}$ describes the blur in terms of the path followed by an origin of the co–ordinate system during the exposure interval $[0,T]$ of the photograph that produced y. Thus, instead of (4.4) we have, for all w,

$$y(w) = \int_0^T x(w-\gamma(t))dt = \int x(v)\frac{1_\Gamma(w-v)}{\mid \gamma'\{\gamma^{-1}(w-v)\} \mid}dv, \tag{4.33}$$

where the limits of integration are defined by the indicator function. In terms of (4.4),

$$h(v,w) \equiv 1_\Gamma(w-v)/ \mid \gamma'\{\gamma^{-1}(w-v)\} \mid . \tag{4.34}$$

Particular cases of this are listed below.

Continuous case

In general (c.f. VL: Section 4.8),

$$\begin{aligned} x^{(m)}(v) &= x^{(m-1)}(v)\int_{v+\Gamma} \mid \gamma'(\gamma^{-1}(w-v)) \mid^{-1} y(w)dw/ \\ &\int_{v+\Gamma} \mid \gamma'(\gamma^{-1}(w-v)) \mid^{-1} \{\int_{w-\Gamma} \frac{x^{(m-1)}(s)}{\mid \gamma'(\gamma^{-1}(w-v)) \mid^{-1}} ds\} du \end{aligned} \tag{4.35}$$

Special cases include the following:

(i) Constant speed linear motion

Here, $\gamma(t) = at, 0 \leq t \leq T$, for some $a > 0$. Then (4.35) becomes (c.f. (4.10)

in VL)

$$x^{(m)}(v) = x^{(m-1)}(v)a \int_v^{v+aT} y(w)dw / [\int_v^{v+aT} \{\int_{w-aT}^{w} x^{(m-1)}(s)ds\}dw]. \quad (4.36)$$

(ii) Constant acceleration from rest along a straight line

This time, $\gamma(t) = at^2, 0 \leq t \leq T$, for some $a > 0$, giving (c.f. (4.11) in VL)

$$x^{(m)}(v) = \frac{x^{(m-1)}(v)2\sqrt{(a)} \int_v^{v+aT^2} (w-v)^{-1/2} y(w)dw}{\int_v^{v+aT^2} (w-v)^{-1/2} \{\int_{w-aT^2}^{w} x^{(m-1)}(s)(w-s)^{-1/2}ds\}dw}. \quad (4.37)$$

Discrete case

The general form of the algorithm is given by (4.13).

(i) Constant–speed linear motion

In this case, $h_{ij} = a1_{\{0,\ldots,b\}}(j-1)$, for some $a > 0$, and (4.13) becomes (c.f. (5.6) of VL)

$$x_i^{(m)} = x_i^{(m-1)} a^{-1} (\sum_{j=1}^{i+b} y_j) / \{\sum_{j=1}^{i+b} \sum_{s=1 \vee (j-b)}^{M \wedge j} x_s^{(m-1)}\}. \quad (4.38)$$

(ii) Constant acceleration from rest along a straight line

Now, $h_{ij} = a \mid j-i \mid^{1/2} 1_{\{1,\ldots,b\}}(j-i)$, for some $a > 0$, so that (c.f. (5.11) of VL)

$$x_i^{(m)} = \frac{x_i^{(m-1)} a^{-1} \sum_{j=i+1}^{i+b} (j-1)^{-1/2} y_j}{\sum_{j=i+1}^{i+b} (j-i)^{-1/2} \{\sum_{s=1 \vee (j-b)}^{M \wedge (j-1)} x_s^{(m-1)} (j-s)^{-1/2}\}}. \quad (4.39)$$

4.5 Image restorations

We applied the algorithms to the "cart" example that constitutes Experiment 2 of VL. We followed their procedure as closely as possible and ran both the EMA and ISRA for 106 iterations. (We are grateful to Professor P. J. Green for helpful

discussions, and the supply of code, which enabled us to gain access to these data).

Two details of the image were considered, each of which was a 250×250 pixel scene ($M = 62500$). It was assumed, as in VL, that a motion blur of 106 pixels had been imposed and, in analysing the two sub–images, data were used from adjoining strips of widths 106 pixels. The ISRA was therefore based on (4.38).

Figure 4.1 shows the first sub–image, which comprises the area of the whole image near a blurred wheel. Figures 4.1 (a) and (d) show, respectively, the results of applying the EMA and the ISRA for 106 iterations, and Figure 4.1 (c) shows the difference–image between EM and ISRA. The only substantial differences occur near the bottom left edge of the wheel. In both Figures (b) and (d) there is evidence of vertical artefacts, mentioned in VL, that are, interestingly, less evident in Figure 4.2. Figure 4.2 is the 40–iterations equivalent of Figure 4.1; at that stage, the EMA and the ISRA results are very similar.

Figure 4.3 plots the pixel intensities corresponding to Figures 4.1(b),(d),(c) in Figures 4.3(a), (b),(c) respectively, where the pixels are numbered in raster–scan beginning at the top left–hand corner. Figure 4.4 gives some idea of the convergence behaviour of the two algorithms, in terms of the per–pixel mean–squared difference between successive iterates, i.e. $M^{-1} \sum_{i=1}^{M} (x_i^{(m)} - x_i^{(m-1)})^2$. Both Figure 4.4(a) and Figure 4.4(b), which is on the log–scale, indicate that the behaviour is very similar, with the EMA consistently taking slightly larger steps.

The other part of the image examined includes the letters "RPO" from the word "AIRPORT". Figure 4.5 is the 106–iterate version (c.f. Figure 4.1) and Figure 4.6 corresponds to 40 iterations (c.f. Figure 4.2). Again, the two algorithms

produced very similar results, and the 40–iterate restorations seem to be at least as appealing as those after 106 iterations. It seems likely that, as emphasised later in Section 4.8, the inverse problem is somewhat ill–posed, and that stopping at, say, 40 iterations imposes beneficial regularisation.

We also examined the ubiquitous "lena" image, imposing a linear blur of 30 pixels and examining the EMA and ISRA restorations after 40 iterations. Figure 4.7(a) displays the true 256×256 image; we applied the algorithms to the internal 256×196 sub–image (allowing a border of 30 pixels at either side) and Figures 4.7 (b),(c), (d) display, respectively, the data, EM reconstruction and ISRA reconstruction. Again, there seems very little to choose between the performance of the two algorithms. Figure 4.8(a) shows the logarithm of the m.s.e. between successive iterations – again, the EMA takes slightly larger steps than the ISRA. Figure 4.8(b), a plot of the m.s.e. between the true image and current reconstruction, shows that this may, however, have an adverse effect on image restoration however; EMA is always slightly further away w.r.t. this criterion than the ISRA. This is certainly not contradicted by the results we observed in the solutions of small sets of linear equations (see Section 4.4.1).

4.6 Discussion of results

The illustrations in Section 4.5 provide information about the comparison between the ISRA and the EMA. In general, both algorithms are prone to slow convergence. In the context of PET, Daube–Witherspoon and Muehllehner ([15]) provide further empirical evidence of this, but point out that the ISRA requires

far fewer operations per iteration than does EM. Both algorithms can be accelerated, either by applying Aitken's Δ^2 procedure, or by adding a linear search embellishment. Lewitt and Muehllener ([64]) implement this in the case of the EMA for ECT, and De Pierro ([18]) points out for the ISRA both that the optimal step–length in the proposed direction is easily computed and that the link with Chahine's ([12]) algorithm makes available further improvement mechanisms developed in other branches of the inverse–problems literature. In the next section, Section 4.7, we introduce some new methods for acceleration and compare their effects on both algorithms.

An alternative general approach to the inversion of linear equations is through the Fourier domain. In the situation when there are missing data, Ollinger and Karp ([73]) compare the ISRA with two such methods, finding that the ISRA is slow in comparison but admitting that it could be accelerated.

The (very limited) evidence from Table 4.4.1 is that the convergence rates of the EMA and the ISRA are similar. The local (near the limit point) convergence behaviour in the case of x with finite domain D_x is dependent on Ostrowski's Theorem; see, for instance, Ortega and Rheinboldt ([74], p.300). Consider an iterative algorithm of the form

$$x_i^{(m)} = \phi_i(x^{(m-1)}), i = 1, \ldots, M, n = 1, 2, \ldots, \tag{4.40}$$

and suppose that $\hat{x}$ is the limit of $\{x^{(m)}\}$. Define the matrix $U(x) = \{U_{is}(x)\}$ by $U_{is}(x) = \partial\phi_i(x)/\partial x_s$, for $i, s = 1, \ldots, M$. Then the rate of local convergence to $\hat{x}$ is dictated by the spectral radius of $U(\hat{x})$.

Consider now the versions of $U(x)$ corresponding to the discrete ISRA given

by (4.13) and the discrete EMA defined by

$$x_i^{(m)} = x_i^{(m-1)}(\sum_{j=1}^{N} h_{ij})^{-1} \sum_{j=1}^{N}(\frac{h_{ij}}{\sum_{s=1}^{M} x_s^{(m-1)} h_{sj}})y_j, i = 1, \ldots, M. \tag{4.41}$$

This is the version of (4.10) corresponding to the more general case when $\sum_j h_{ij} \neq 1$ for all i. We can now evaluate expressions for the matrices $U(\hat{x}_{EM})$ and $U(\hat{x}_{ISRA})$. For the ISRA,

$$\phi_i(x) = x_i \frac{\sum_j h_{ij} y_j}{\sum_s x_s \sum_j h_{ij} h_{sj}}, i = 1, \ldots, M, \tag{4.42}$$

so that, for $i, s = 1, \ldots, M$,

$$U_{is}(x) = \delta_{is} \frac{\sum_j h_{ij} y_j}{\sum_k x_k (\sum_j h_{ij} h_{kj})} - x_i \frac{(\sum_j h_{ij} y_j)(\sum_j h_{ij} h_{sj})}{(\sum_k x_k \sum_j h_{ij} h_{kj})^2}; \tag{4.43}$$

δ is the Kronecker delta. At $\hat{x}$, $\sum_j h_{ij} y_j / (\sum_s \hat{x}_s (\sum_j h_{ij} h_{sj})) = 1$, so that

$$U_{is}(\hat{x}) = \delta_{is} - \hat{x}_i \frac{\sum_j h_{ij} h_{sj}}{\sum_j h_{ij} y_j}. \tag{4.44}$$

For the EMA,

$$\phi_i(x) = x_i (\sum_j h_{ij})^{-1} \sum_j (h_{ij} / (\sum_s x_s h_{sj})) y_j, i = 1, \ldots, M, \tag{4.45}$$

giving, for $i, s = 1, \ldots, M$,

$$U_{is}(x) = \delta_{is} (\sum_j h_{ij})^{-1} \sum_j (\frac{h_{ij}}{\sum_k x_k h_{kj}}) y_j - x_i (\sum_j h_{ij})^{-1} \sum_j (\frac{h_{ij} h_{sj}}{(\sum_k x_k h_{kj})^2}) y_j. \tag{4.46}$$

However, at $\hat{x}$, $\sum_k \hat{x}_k h_{kj} = y_j$, for all j, so that

$$U_{is}(\hat{x}) = \delta_{is} - \hat{x}_i (\sum_j h_{ij})^{-1} \sum_j \frac{h_{ij} h_{sj}}{y_j}. \tag{4.47}$$

For both (4.44) and (4.47), the methods of Titterington ([94]) show that the eigenvalues of $U(\hat{x})$ are non–negative and strictly less than unity. Comparison of the maximum eigenvalues in particular cases would complete the comparison of

local convergence properties. In the case of the illustration in Section 4.4.1, part (ii), where $N = 2$ and $y_1 = y_2$, (4.44) and (4.47) are identical.

4.7 Ordered Subsets

We have hitherto demonstrated a large range of applications for the EMA and ISRA. We have shown that both algorithms will, in suitable conditions, converge to produce estimates of x that are optimal, in different but well–defined senses. We turn our attention now to the problem, mentioned earlier, of increasing the rate of convergence. There exist many established methods (see Section 4.6 and also [59] or [36]) which attempt to speed up the EMA, but a recent paper ([56]) has offered empirical evidence that a new method could increase rate of convergence by an order of magnitude. We will detail that method, offer an adaption of it for the ISRA, display an example of the new algorithm, and finally discuss convergence issues.

4.7.1 Hudson and Larkin's Ordered Subsets EMA

In [56], Hudson and Larkin attempt to increase the rate of convergence for the EMA, in the context of single photon emission computed tomography (SPECT) by splitting the data into blocks, or ordered subsets (OS) and processing each block sequentially. If there are K OS, we say "OS level = K". A complete iteration of the algorithm, called OS–EMA, is a single pass through all of the OS.

For SPECT we use the same notation developed for emission tomography in Section 4.3.1; $\{x_i : i = 1, \ldots, M\}$ are the unknown values of the emissions from the sources, $\{y_j : j = 1, \ldots, N\}$ are the observed counts at the detectors, and

$h_{ij} = \Pr\{\text{emission from } i \text{ is observed at } j\}$. The data are divided into ordered subsets $S_1, \ldots, S_K$. Using this notation , Hudson and Larkin's OS–EM algorithm is as follows.

The discrete iterative OS–EMA

1. $m = 0$. Choose $x^{(m)} \geq 0$.

2. $x^1 := x^{(m)}$.

3. $m := m + 1$.

4. For subsets $k = 1, \ldots, K$:

$$x_i^{k+1} = x_i^k \sum_{j \in S_k} [y_j h_{ij} / \sum_{i=1}^{M} h_{ij} x_i^k], i = 1, \ldots, M. \tag{4.48}$$

N.B. if the divisor in (4.48) is zero, set $x_i^{k+1} := x_i^k$.

5. $x^{(m)} := x^{K+1}$.

6. Check for convergence of $\{x^{(m)}\}$. If

NO $\longrightarrow$ step 2

YES $\longrightarrow$ STOP. $\square$.

The obvious extension of Hudson and Larkin's OS–EMA to provide a new ISRA is to replace step 4 with:

4′ For subsets $k = 1, \ldots, K$:

$$x_i^{k+1} = x_i^k [\sum_{j \in S_k} y_j h_{ij}] / [\sum_{j \in S_k} h_{ij} (\sum_{i=1}^{M} h_{ij} x_i^k)], i = 1, \ldots, M. \tag{4.49}$$

4.7.2 Choice of ordered subsets

For SPECT or PET there is a natural geometrical ordering for the subsets, corresponding to groups of projections. In this chapter we have been considering the

restoration of images degraded by linear motion blur, and here we find the choice of subset to be less obvious.

Initally, we selected each row as a subset, i.e. $S_k = \{\text{row } k \text{ of data}\}$; or a cumulative set was chosen, so that $S_k = \{\text{the first } k \text{ rows of the image}\}$. These produced results that were better than standard ISRA for the same number of *complete* iterations; however, a little thought showed that we were in fact merely applying the standard ISRA k times as often. For define

$$\begin{aligned} T_k &= \{i : \sum_{j \in S_k} h_{ij} \neq 0\} \\ &= \{i : \text{ denominator of (4.48) is non-zero }\} \\ &= \{i : x_i^k \text{ is changed }\}. \end{aligned}$$

Then if $S_k = \{k\text{'th row (or group of rows)}\}$, $T_k = S_k$, and $T_k \bigcap T_l = \emptyset, k \neq l$. So for linear motion blur with row ordered subsets, OS–ISRA $\equiv$ ISRA, and OS–EMA $\equiv$ EMA.

However, if $S_k = \{k\text{'th column (or group of columns)}\}$ then $T_k \bigcap T_l \neq \emptyset$ since $T_k = \{S_k + \text{ some more of } x\}$. So, for our experiment, we used $S_k = \{k\text{'th column}\}$, and also $S_k = \{k\text{'th and } (k+1)\text{'th column}\}$.

4.7.3 Example

For our example we worked with artifical image I1, to which we added linear motion blur of size 10 pixels (i.e. parameter $b = 10$ in equation (4.38)). Reconstructions were carried out using the OS–ISRA assuming, first, that $S_k = k$'th column, and then that $S_k = k$'th and $(k+1)$'th columns. The true image, the data, and the one–column and two–column reconstructions can be seen in Figure

4.9. In a second trial, Gaussian noise of s.d. 2.0 was added to the data before reconstruction commenced – Figure 4.10 contains the results. It seems clear that both one–column and two–column reconstructions are very similar to the data, a fact which is born out by examining the m.s.e. between the data and the truth, and comparing it with the m.s.e. between the data and the reconstruction. The m.s.e.'s between the data and the true image were 220.9321 and 223.9807 for the noise–free and Gaussian noise cases, respectively. Table 4.7.3 displays the m.s.e.'s between the truth and the reconstruction after 10 iterations, when each of the algorithms was stopped.

Table 4.7.3: m.s.e. between the truth and reconstruction

	1 column OS	2 column OS
no noise	220.8776	383.2672
$\sigma = 2.0$	224.1884	398.5564

Figure 4.11 displays the value of m.s.e. between successive reconstructions as iteration number increased. It is clear that a large initial drop in m.s.e. is followed by changes so small as to be almost inconsequential (this is why we ran the algorithms for only 10 iterations and did not use m.s.e. as a stopping criterion as previously). Removing the large, initial first value shows that this m.s.e. figure can increase as well as decrease between iterations (see Figure 4.12), perhaps raising empirical doubts about our unproved hope that the OS–ISRA will converge to a useful solution (see following Section 4.7.4). The experimental evidence indicates that it converges to a different solution than the standard ISRA.

4.7.4 Convergence

We have not been able to prove convergence of the OS–ISRA, but here we present details of how we feel the argument should proceed. Hudson and Larkin in [56] were able to prove convergence of the OS–EMA estimates to the m.l.e. in the noise–free case, but were unable to extend the proof to the case where noise is present.

We attempt to adapt the argument of Eggermont ([26]), in particular his Lemma 6.1, which was used to demonstrate convergence of the ordinary ISRA.

Define $D_k = \text{diag}\{x_i^k / \sum_{j\in S_k} h_{ij} y_j^k\}$, where

$$y_j^k = \begin{cases} \sum_{s=1}^M h_{sj} x_s^k & \text{if } j \in S_k \\ y_j & \text{otherwise} \end{cases} \tag{4.50}$$

and also set $\lambda_i = \sum_{j\in S_k} h_{ij} y_j = (Hy)_i$. $\Lambda = \text{diag}\{\lambda_i : i = 1, \ldots, M\}$. Let I_k be a diagonal $N \times N$ matrix with 1's in the rows corresponding to $j \in S_k$ and zeros elsewhere.Then

$$\sum_{j\in S_k} h_{ij} d_j = \{(I_k H^T)^T d\}_i = \{H I_k d\}_i \tag{4.51}$$

for any M–vector d. So $\sum_{j\in S_k} h_{ij} y_j = \{H I_k y\}_i$ and $\sum_{j\in S_k} h_{ij} (\sum_{s=1}^M h_{sj} x_s^k) = \{H I_k H^T x^k\}_i$.

Our aim, following Eggermont, is to show that

$$e(x^*, x^k) \geq e(x^*, x^{k+1}) \tag{4.52}$$

where

$$e(x^*, x) = \Delta_{KL}(\Lambda x^*, \Lambda x) + S(x) - S(x^*). \tag{4.53}$$

Here, x^* is any point of accumulation of the series of x estimates, $S(x)$ is the sum–of squares function defined previously in (4.11), and Δ_{KL} is the Kullback–Leibler

(KL) directed divergence, which we can write as

$$\Delta_{KL}(\Lambda x^*, \Lambda x) = \sum_{i=1}^{M} \lambda_i x_i^* \log(x_i^*/x_i) + \sum_{i=1}^{M} \lambda_i (x_i - x_i^*). \tag{4.54}$$

We examine the difference in KL distances between Λx^* and two successive OS estimates of x:

$$\begin{aligned} \Delta_{KL}(\Lambda x^*, \Lambda x^k) - \Delta_{KL}(\Lambda x^*, \Lambda x^{k+1}) &= \sum_i \frac{\lambda_i}{x_i^{k+1}} (x_i^k - x_i^{k+1})(x_i^{k+1} - x_i^*) \\ &= (x^k - x^{k+1})^T D_k^{-1} (x^{k+1} - x^*) \\ &= -(x^k - x^{k+1})^T D_k^{-1} (x^k - x^{k+1}) \\ &+ (x^k - x^{k+1})^T D_k^{-1} (x^k - x^*) \end{aligned} \tag{4.55}$$

Now, $(x^k - x^{k+1}) D_k^{-1} = Hy^k - Hy = H(y^k - y)$, so that the second term in (4.55) can be written as

$$(x^k - x^*)^T H(y^k - y) = (H^T x^k - H^T x^*)^T (y^k - y). \tag{4.56}$$

But in view of the definition of y^k and I_k, we can write $(y^k - y) = I_k(H^T x^k - y)$, so that

$$(H^T x^k - H^T x^*)^T (y^k - y) = (H^T x^k - H^T x^*)^T I_k (H^T x^k - y), \tag{4.57}$$

which, by convexity, dominates

$$\frac{1}{2}(H^T x^k - y)^T I_k (H^T x^k - y) - \frac{1}{2}(H^T x^* - y)^T I_k (H^T x^* - y). \tag{4.58}$$

We would like to use (4.58) to prove an equivalent of Lemma 6.1 in Eggermont ([26]):

Lemma

$$(H^T x^k - y)^T I_k (H^T x^k - y) - (H^T x^{k+1} - y)^T I_k (H^T x^{k+1} - y) \geq (x^k - x^{k+1}) D_k^{-1} (x^k - x^{k+1}). \tag{4.59}$$

Unfortunately we have not been able to prove this Lemma. Also, it may not be of direct help in proving convergence since it deals not with the standard sum–of–squares function, but contains a troublesome I_k.

4.7.5 Related work

One other approach for increasing the rate of convergence for the EMA, not unrelated to that of the previous section, is that of Neal and Hinton, in [71]. They propose that the EMA be viewed as a problem of maximising a joint function $F(f, \theta)$, where, as in Section 4.2, $f = f(x, y)$ is the p.d.f. of the observed (y) and unobserved (x) variables, and θ are the unknown parameters we would like to estimate. The E–step maximises F w.r.t. the distribution over the unobserved variables, the M–step w.r.t. the parameters. The function F is defined by

$$F(f, \theta) = \mathrm{E}_f[\log f(x, y, \mid \theta)] + G(f),$$

where $G(f) = -\mathrm{E}_f[\log f(x)]$, the entropy of the distribution. Neal and Hinton prove that a maximum of F at (f^*, θ^*) implies that θ^* is also a maximum of $f(y \mid \theta) = \sum_x f(x, y, \mid \theta)$. This justifies the use of incremental methods, where we seek only to increase, rather than maximise, the value of the likelihood, at each stage of the algorithm. Assuming the $\{x_i\}$ are independent, so that $F(f, \theta) = \sum_i F_i(f_i, \theta)$, where

$$F_i(f_i, \theta) = \mathrm{E}[\log f(x_i, y_i \mid \theta)] + G(f_i),$$

Neal and Hinton propose the following incremental EMA:

Neal and Hinton's incremental EMA

E–step Choose a data point i to be updated. Set:

$$\begin{aligned} f_j^{(m)} &= f_j^{(m-1)} \text{ for } j \neq i \\ f_i^{(m)} &= \text{argmax}_{f_i} F_i(f_i, \theta^{(m-1)}), \end{aligned}$$

where $f_i^{(m)} = f_i^{(m)}(x_i) = f(x_i \mid y_i, \theta^{(m-1)})$.

M–step Set $\theta^{(m)}$ to θ which maximises $F(f^{(m)}, \theta)$, or, equivalently, which maximises $\mathrm{E}_f[\log f(x, y \mid \theta)]$. □.

An ISRA version of this algorithm would be as follows:

$$y_j^k = \begin{cases} \sum_{s=1}^M h_{sj} x_s^k & \text{if } j \in S_k \\ y_j & \text{otherwise} \end{cases} \tag{4.60}$$

Then the incremental ISRA (INC–ISRA) would be the same as OS–ISRA defined above, but with the following alteration to (4.49):

$$x_i^{k+1} = x_i^k \sum_{j \in S_k} h_{ij} y_j / \sum_{j \in S_k} h_{ij} y_j^k, \tag{4.61}$$

for $i = 1, \ldots, M$. Clearly it would be of interest to implement this algorithm and compare with OS–ISRA.

Finally, the work of Meng ([68]), and Meng and Rubin ([69]), is related to these ordered subsets approaches. They have proposed an algorithm called ECM, for Expectation/Conditional Maximisation. The M–step of (4.2) in Section 4.2 is replaced by a sequence of CM steps, each of which maximises $Q(\theta)$ w.r.t. a different subset of θ. For example, if $\theta = (\theta_1, \ldots, \theta_S)$, then the s'th CM step

might maximise $Q(\theta)$ w.r.t. θ_s, all other θ–parameters being held constant at their current value.

4.8 Regularising the ISRA and the EMA

At the beginning of this chapter we noted that the ISRA and the EMA were outside the Bayesian framework adopted in previous chapters. In some of their applications to image restoration we have seen evidence that the problems are ill–posed, in the sense discussed in Chapter 3. This is particularly likely in the case of images further degraded by noise. In Bayesian work, image estimates are regularised, and the ill–posedness removed, by the use of the prior distribution which acts as a roughness penalty function on the sum–of–squares or likelihood function which is to be optimised. This approach is possible with not only the EMA but also the ISRA.

Consider first the discrete case. Instead of $S(x)$ in (4.11) we could have

$$S_{\beta,A}(x) = \|y - H^T x\|^2 + \beta x^T A x, \tag{4.62}$$

where $\beta > 0$ and A is non–negative definite. We have seen in previous chapters that (4.62) is equivalent to the logarithm of a normal posterior density for x, given y, so that an EMA for seeking the posterior mode (the m.a.p. estimate of x) can be constructed along the lines of Section 4.5 of the EMA paper by Dempster et al. ([17]).

So far as the ISRA is concerned, any minimiser of $S_{\beta,A}(x)$ satisfies

$$Kx = \nu,$$

where $K = (HH^T + \beta A)$ and $\nu = Hy$, stimulating the algorithm

$$x_i^{(m)} = x_i^{(m-1)} \nu_i / \{\sum_s k_{is} x_s^{(m-1)}\}, i = 1, \ldots, M. \tag{4.63}$$

In this case it is typical that some of the elements of A are negative, so that it is not automatic that $x^{(m)} > 0$ for all m.

Silverman et al. ([82]) proposed the addition of a simple smoothing step after every M–step in EMA as a solution to the "spiky" m.l. solutions that result from ill–posed problems in PET. Of course the convergence–to–m.l.e. property of EM was then lost, but Nychka, in [72], shows a connection between smoothed EM and maximisation of penalised likelihoods, such as (4.62). This was made more explicit, and set in a Bayesian framework, by Green, in [44] and [45], who used an adaption of the EMA to estimate the m.a.p. rather than the m.l. estimate of x. He incorporated a Gibbs prior to explain prior knowledge, as we have employed in Chapters 2 and 3.

Turning to the case of a continuous function, the usual kind of penalty function is based on derivatives of x, a common one being

$$\lambda \int \{x''(v)\}^2 dv, \tag{4.64}$$

where $x(\cdot)$ is the density associated with the measure $X(\cdot)$. However, it is common to restrict the choice of x to some space spanned by a certain class of basis functions, and (4.64) reduces to a quadratic form in a transformed, finite vector of parameters; see Silverman ([81]), for example. The corresponding $S(x)$ is similarly transformed and the problem reverts to one of the form (4.62). Penalised versions of the EMA are studied by Byrne in [11].

4.9 A wider class of algorithms

In this final section we refer again to Eggermont ([26]), who considers wider classes of algorithms which include the EMA and ISRA as special cases. Consider the (discrete case) problem of minimising $S(x)$ subject to $x \geq 0$, where $S(\cdot)$ is a convex, continuously differentiable function on $\mathcal{R}^{\mathrm{M}}$, with compact level sets and locally Lipschitz continuous gradient. Eggermont considers three classes of multiplicative, iterative algorithms, of the following forms:

$$x_i^{(m)} = x_i^{(m-1)}[1 - w_m\{\Delta S(x^{(m-1)})\}_i], i = 1, \ldots, M, \tag{4.65}$$

$$x_i^{(m)}[1 + w_m\{\Delta S(x^{(m)})\}_i] = x_i^{(m-1)}, i = 1, \ldots, M, \tag{4.66}$$

and

$$x_i^{(m)} = x_i^{(m-1)}/[1 + w_m\{\Delta S(x^{(m-1)})\}_i], i = 1, \ldots, M. \tag{4.67}$$

In (4.65)-(4.67), w_m is a step parameter. Algorithm (4.66) is called an *implicit* algorithm and (4.67) *explicit*. For appropriate choices of S, $w_m \equiv 1$ in (4.65) gives the EMA in PET and (4.67) gives the ISRA. In discussing (4.67), Eggermont develops the convergence properties for the ISRA as described above in Section 4.3.3, and establishes the convergence properties of the implicit algorithm (4.66) by a similar but slightly simpler argument.

It would be of interest to investigate other versions of these algorithms, for different choices of S, for various choices of step–length $\{w_m\}$, and in the context of the versions appropriate for solving integral equations.

4.10 Summary

We have presented and investigated an algorithm, the ISRA, which can be used to solve the large class of LININPOS problems. We have compared it with the EMA of Vardi and Lee ([98]) and find it is as useful as that algorithm for restoring images distorted by linear motion blur. Our experiments with images found that, although the EMA converged more quickly, it was perhaps converging to a solution further from the truth than was the ISRA.

Our attempts to adapt the ISRA to speed up its rate of convergence, by applying it to subsets of the data in turn, were less successful. We were not able to prove convergence of the adapted ISRA, and found disappointing restorations of the degraded images.

Finally, we have explained how both the ISRA and the EMA may be placed within a Bayesian/regularisation framework.

We feel that the ISRA has shown itself a useful alternative to the more common EMA as a tool for the solution of this wide range of problems, particularly in the area of blind image restoration.

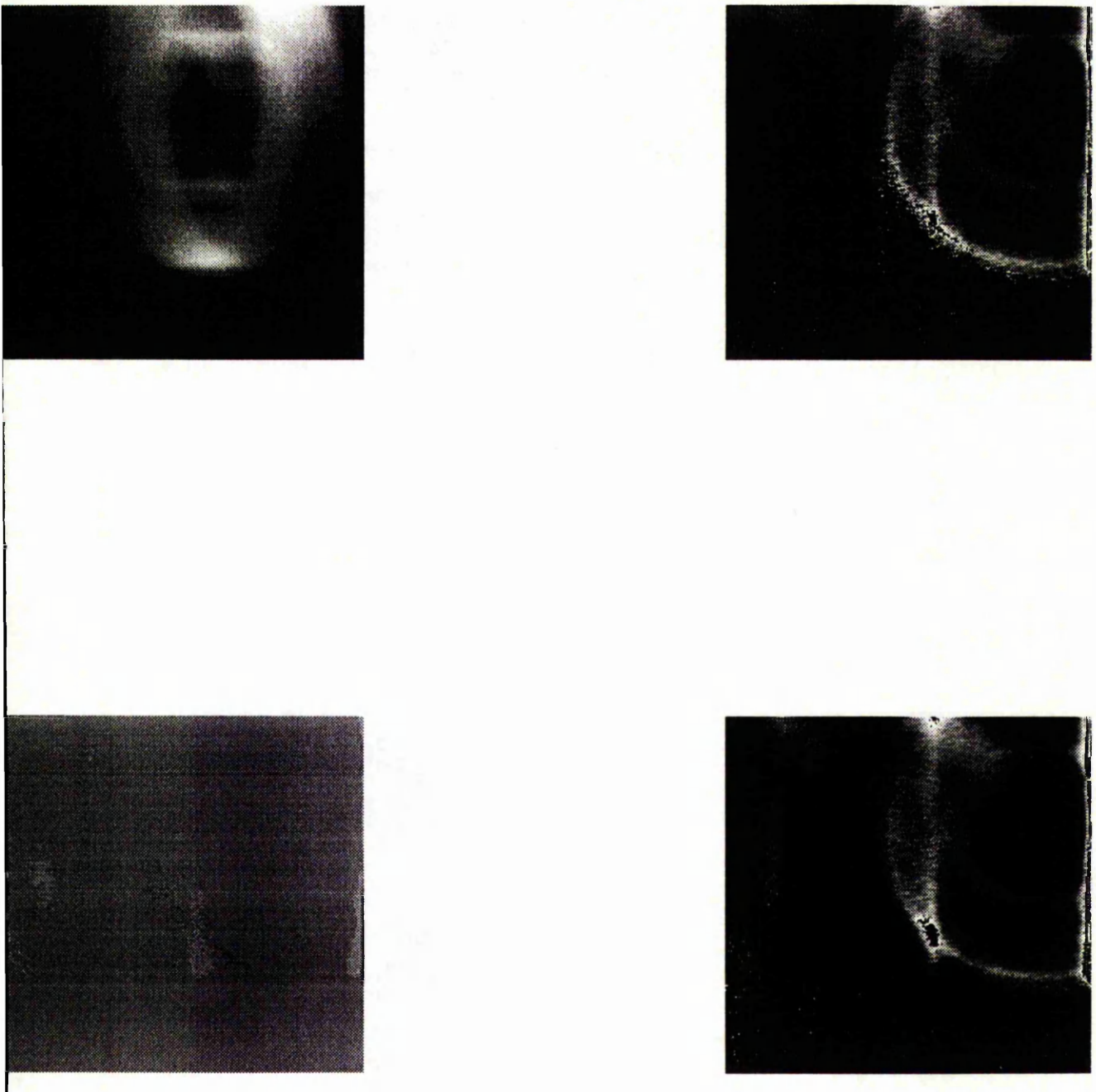
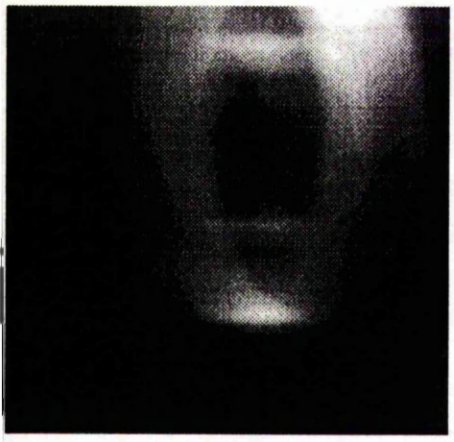
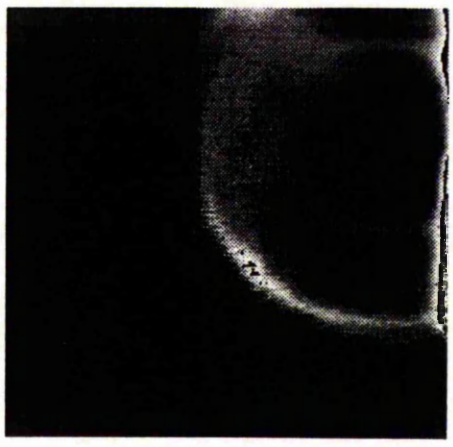


Figure 4.1: from top left to bottom right: (a) "wheels" from "cart" image; (b) EMA reconstruction after 106 iterations; (c) difference between EMA and ISRA reconstruction; (d) ISRA reconstruction after 106 iterations.





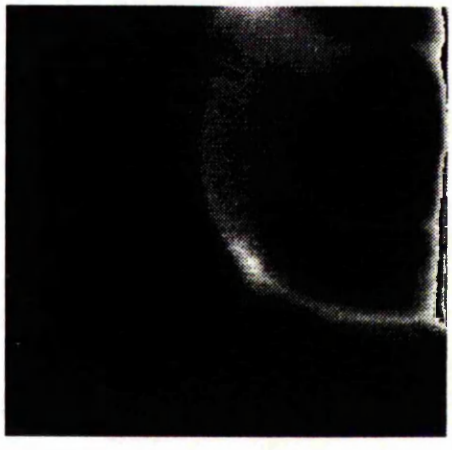


Figure 4.2: from top left to bottom right: (a) "wheels" from "cart" image; (b) EMA reconstruction after 40 iterations; (c) difference between EMA and ISRA reconstruction; (d) ISRA reconstruction after 40 iterations.

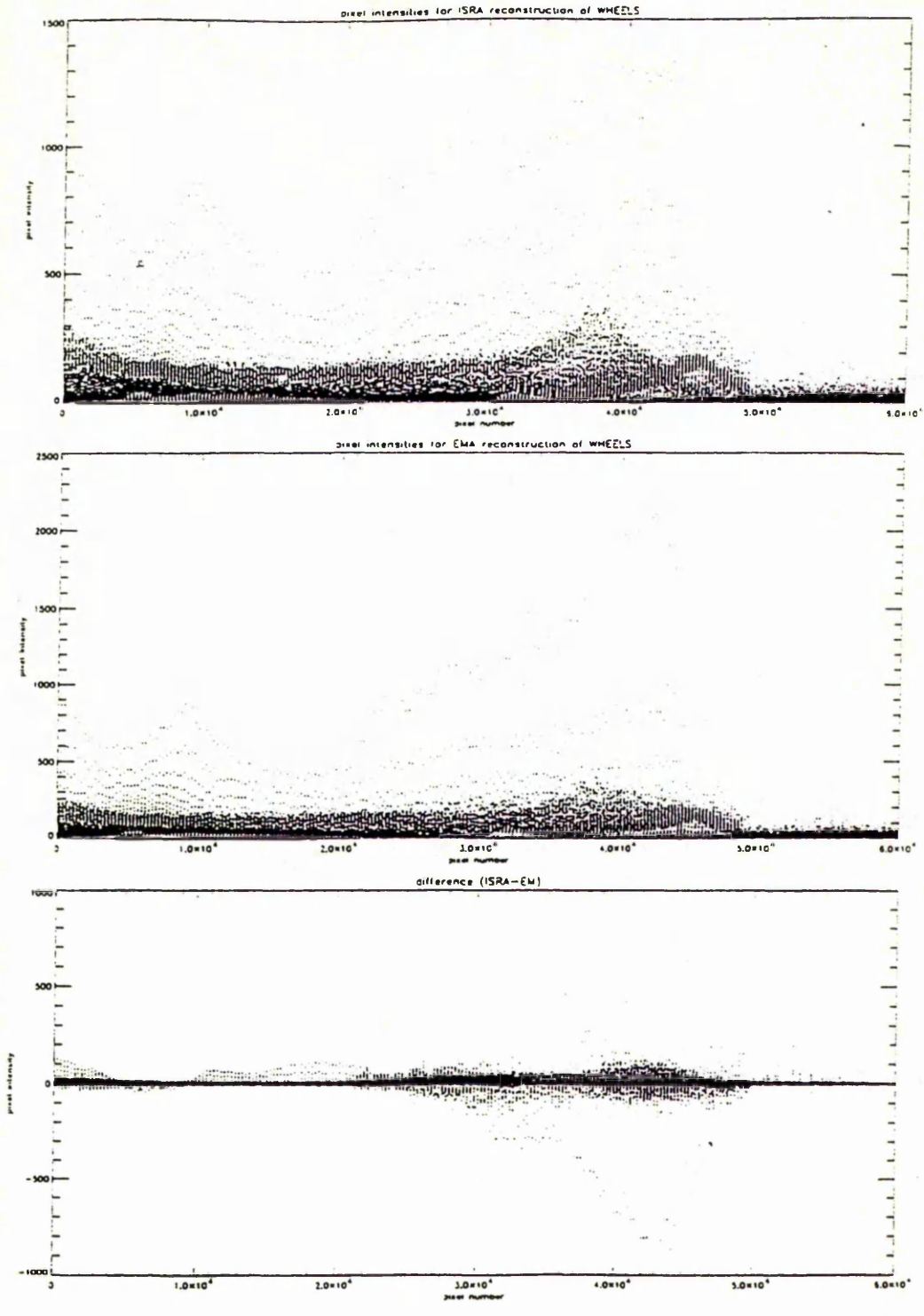


Figure 4.3: from top to bottom: (a) pixel values for EMA at 106 iterations; (b) pixel values for ISRA at 106 iterations; (c) difference between (a) and (b).

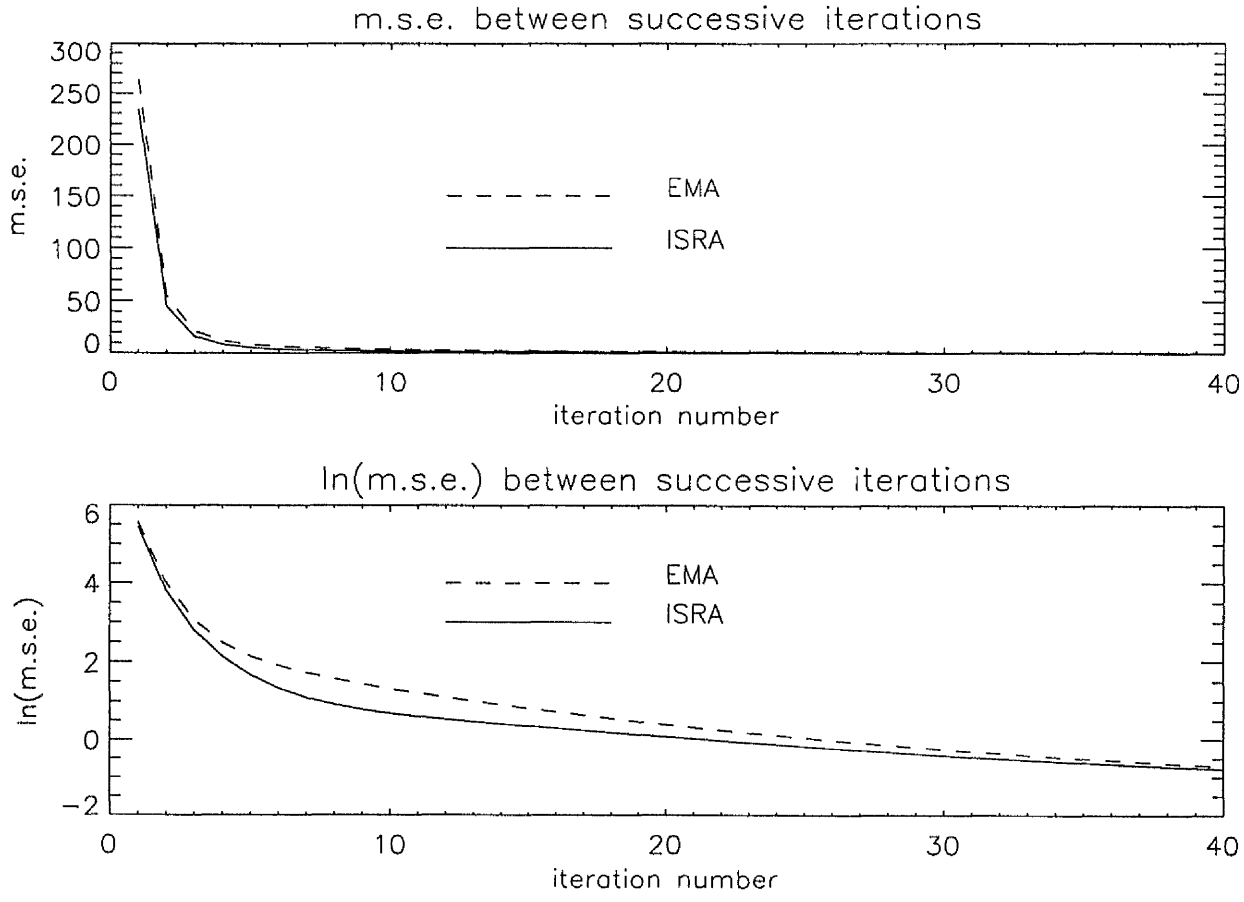
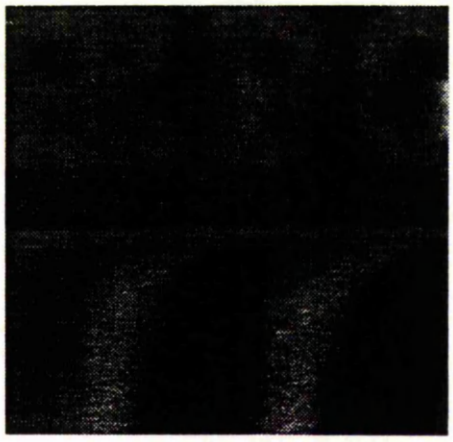
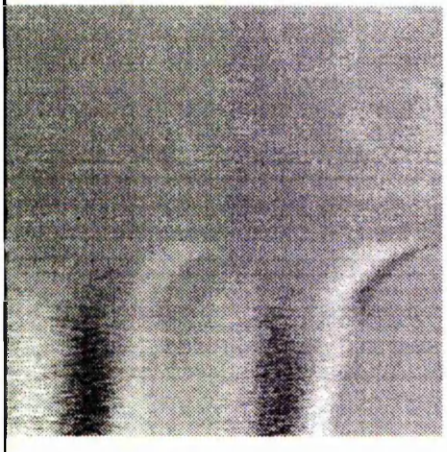


Figure 4.4: from top to bottom: (a) m.s.e. between successive iterations; (b) logarithm of m.s.e. between successive iterations.





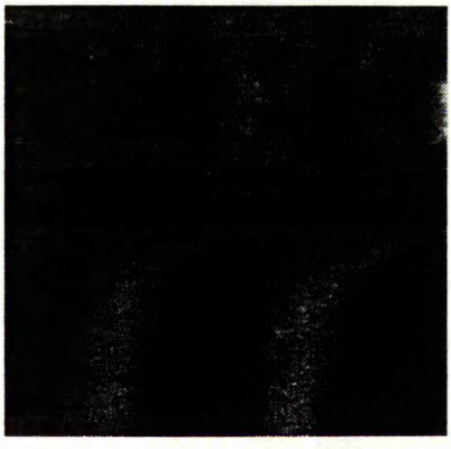


Figure 4.5: from top left to bottom right: (a) "rpo" from "cart" image; (b) EMA reconstruction after 106 iterations; (c) difference between EMA and ISRA reconstruction; (d) ISRA reconstruction after 106 iterations.

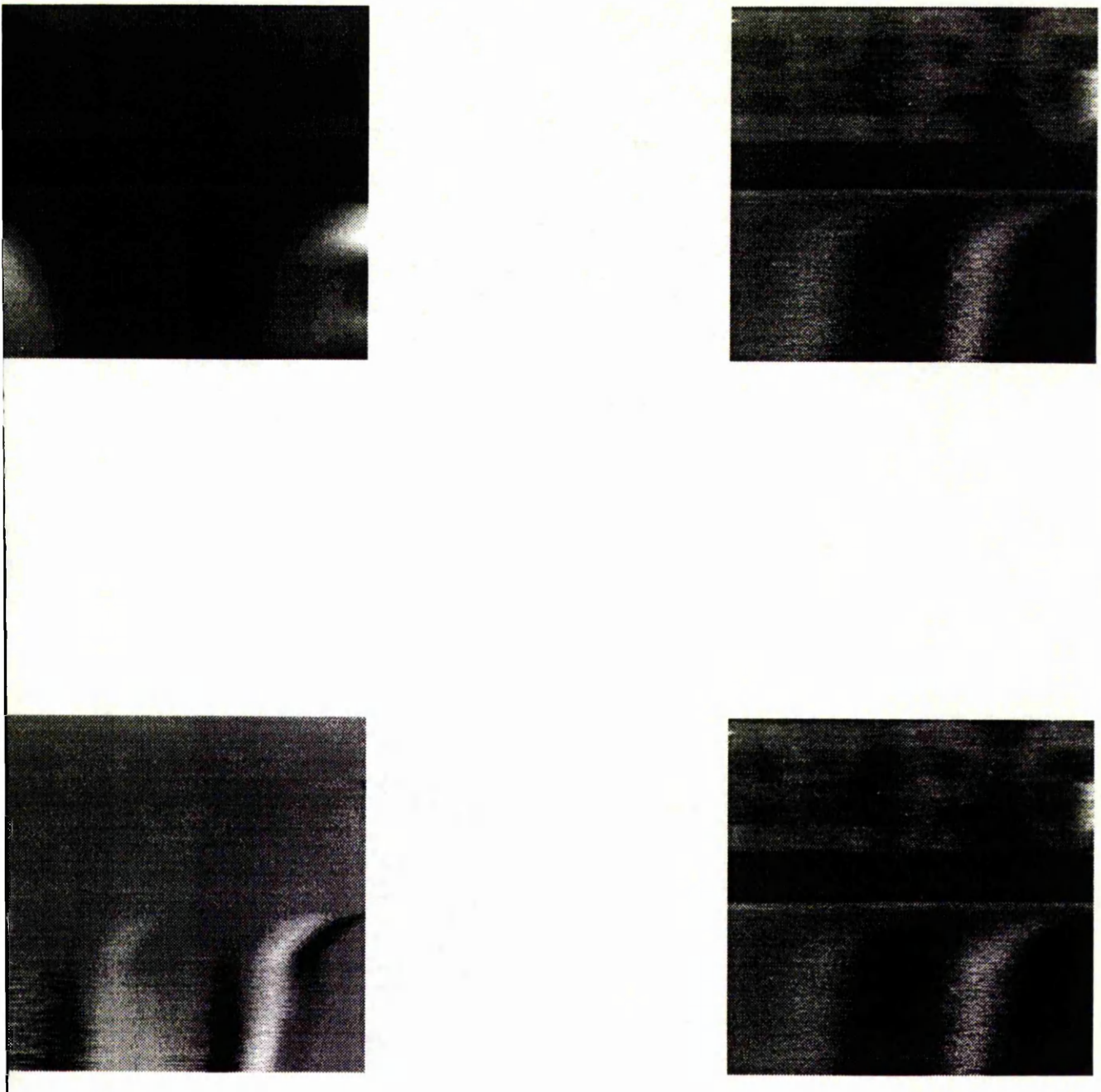


Figure 4.6: from top left to bottom right: (a) "rpo" from "cart" image; (b) EMA reconstruction after 40 iterations; (c) difference between EMA and ISRA reconstruction; (d) ISRA reconstruction after 40 iterations.

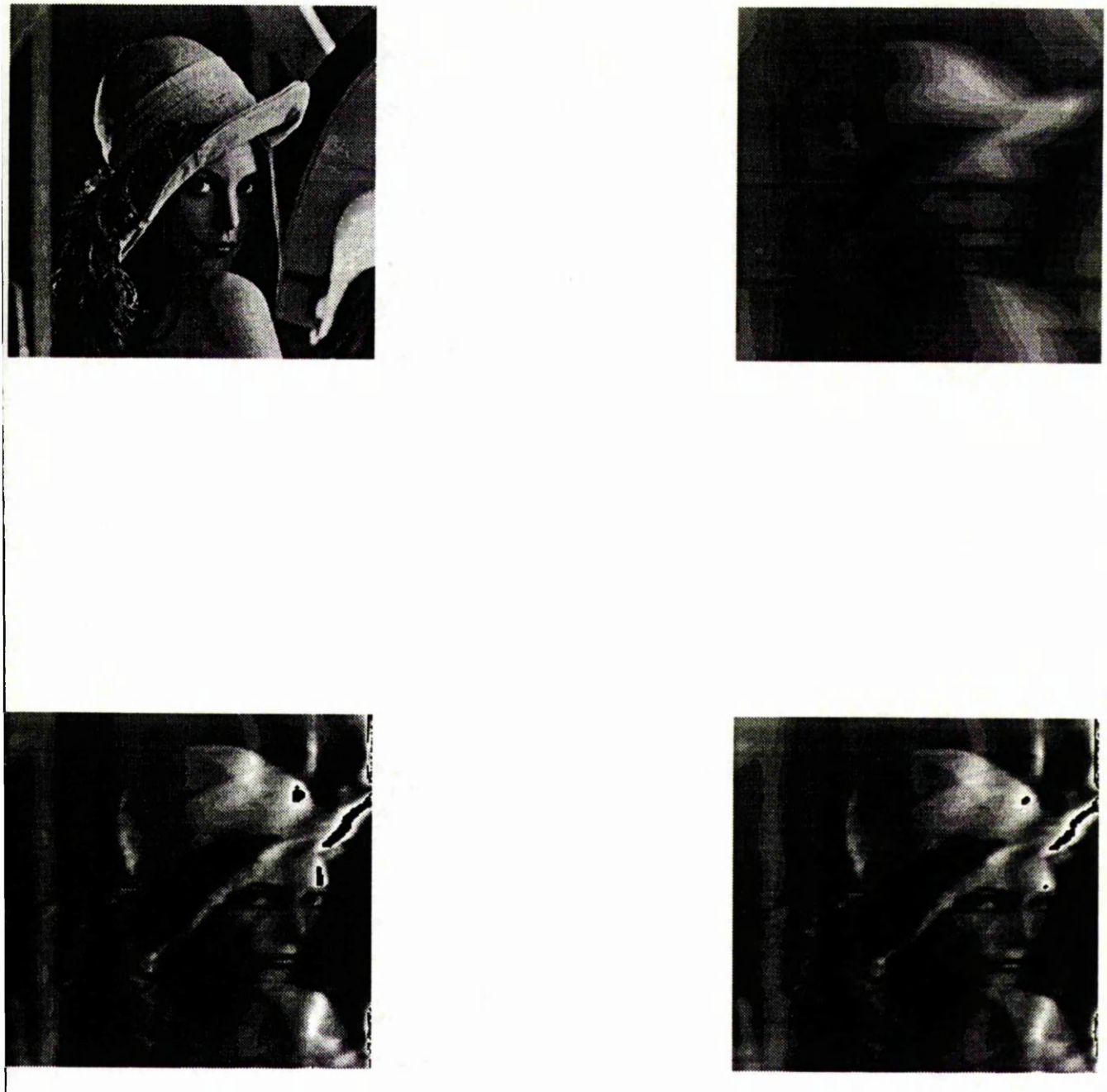


Figure 4.7: from top left to bottom right: (a) "lena" image; (b) image with linear motion blur; (c) EMA reconstruction; (d) ISRA reconstruction.

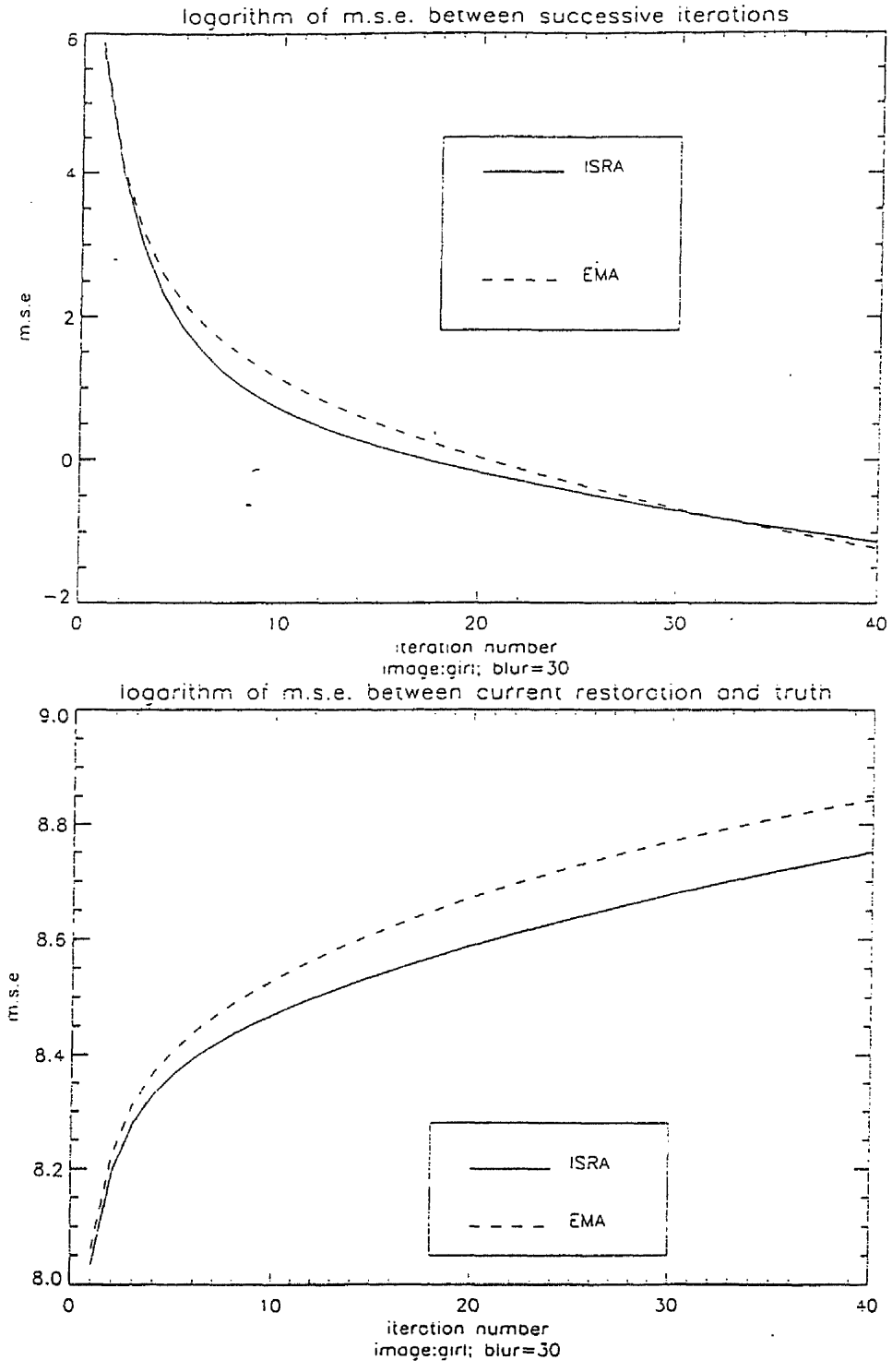


Figure 4.8: from top to bottom: (a) logarithm of m.s.e. between successive iterations for "lena" image; (b) logarithm of m.s.e. between current iteration and true image.

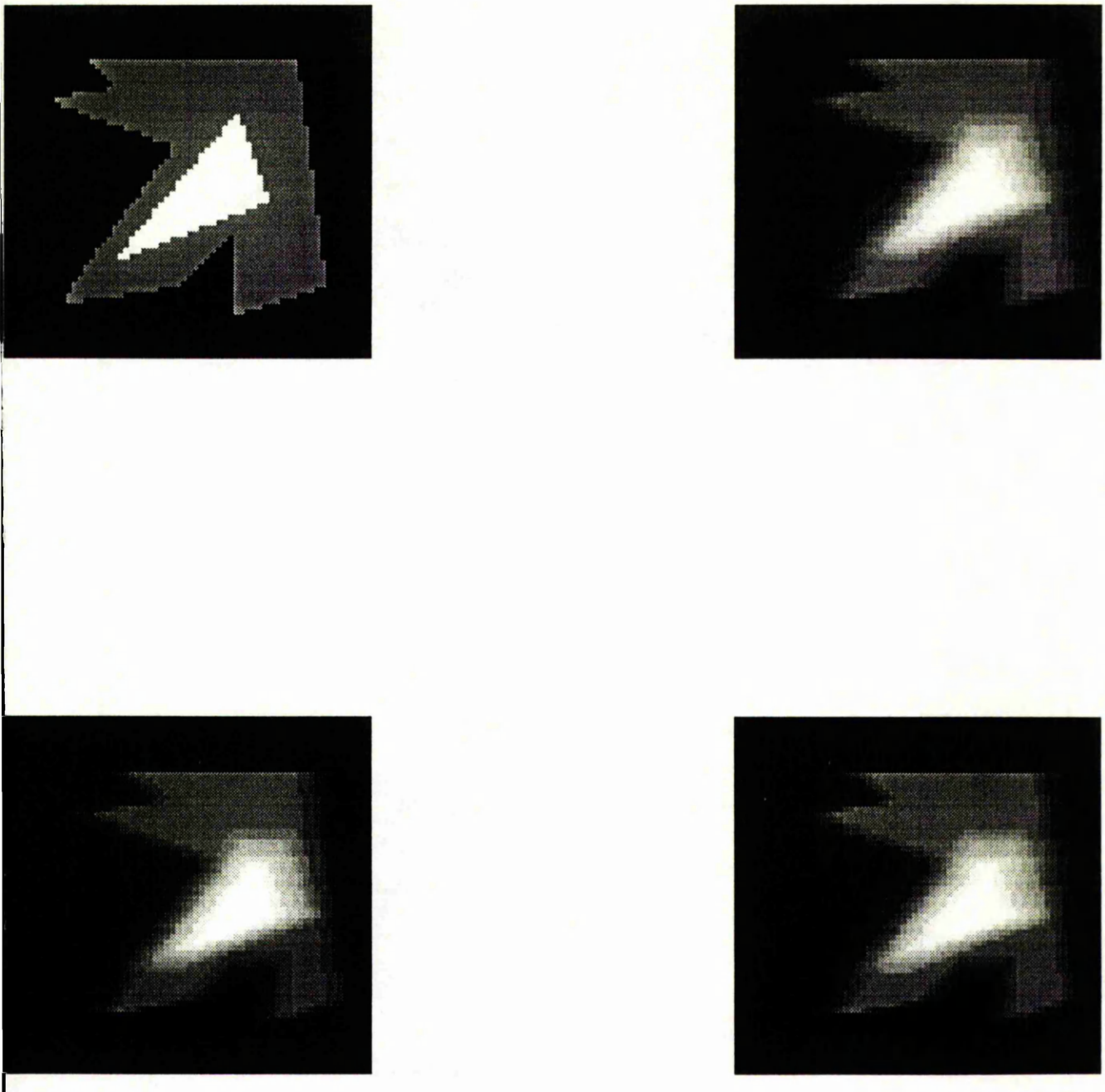


Figure 4.9: From top left to bottom right: the true image H; the data (H with linear motion blur); the one-column reconstruction; the two-column reconstruction.

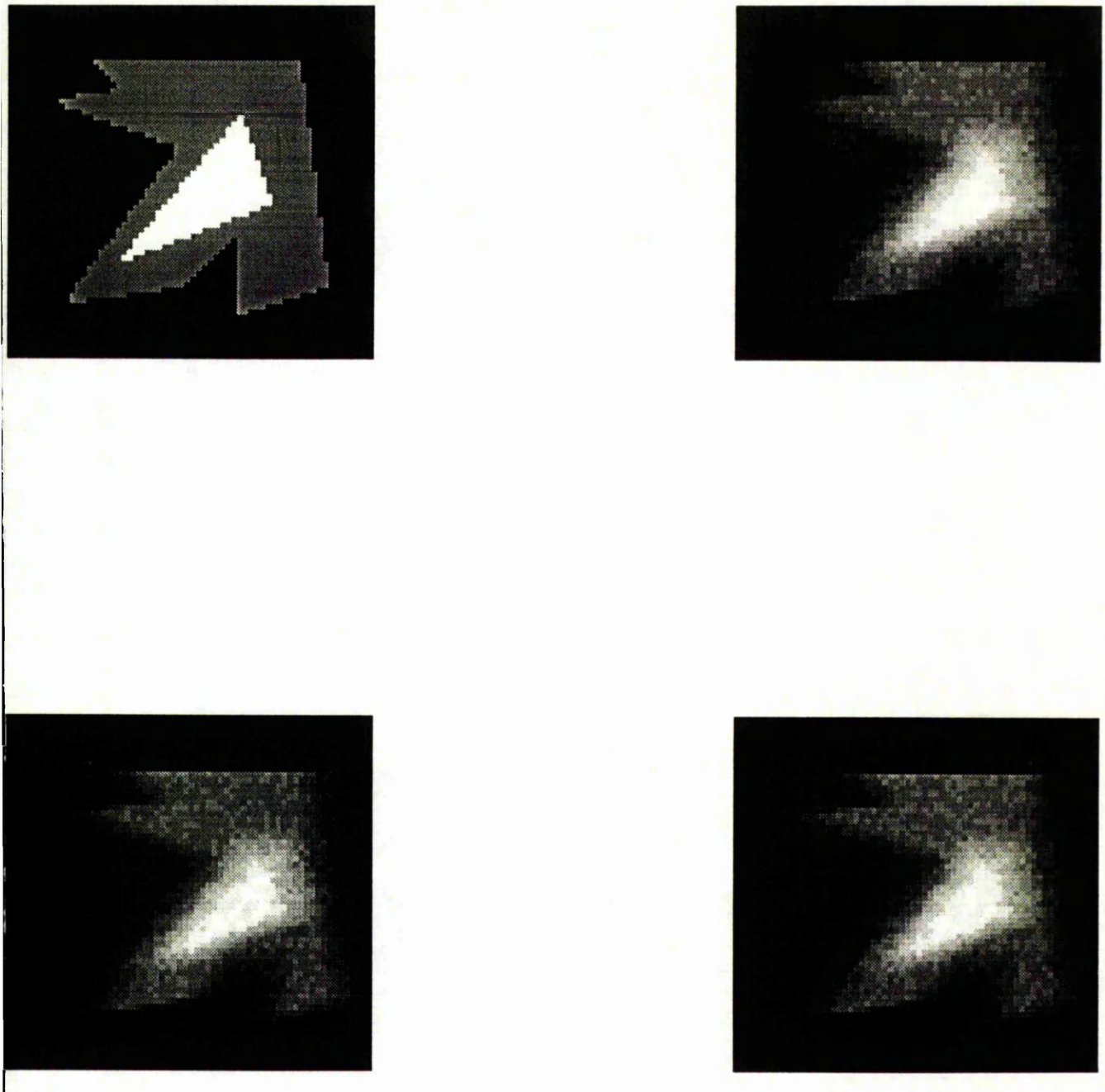


Figure 4.10: From top left to bottom right: the true image H; the data (H with linear motion blur and Gaussian noise with s.d.=2.0); the one-column reconstruction; the two-column reconstruction.

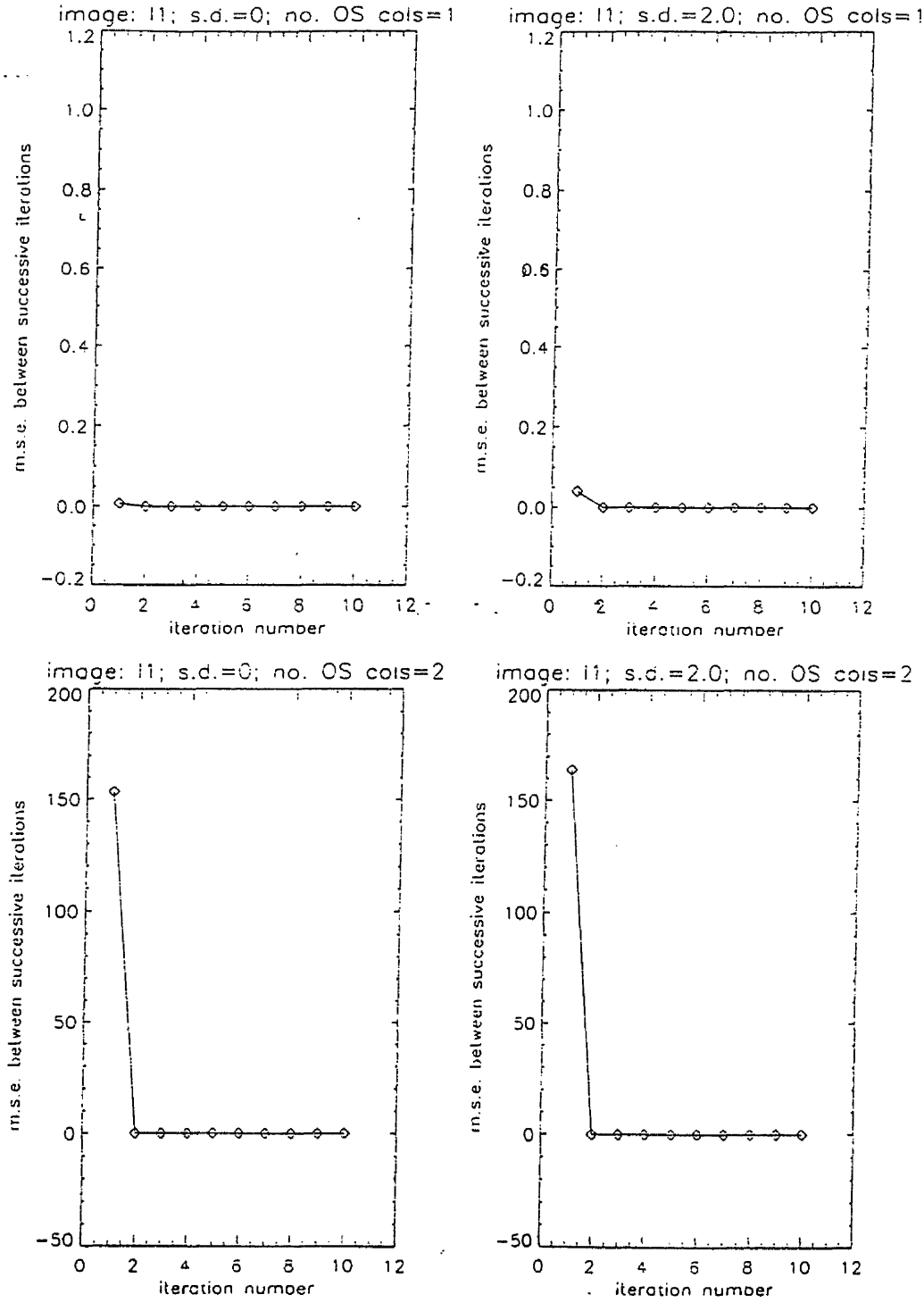


Figure 4.11: Mean squared error values calculated on successive reconstructions.

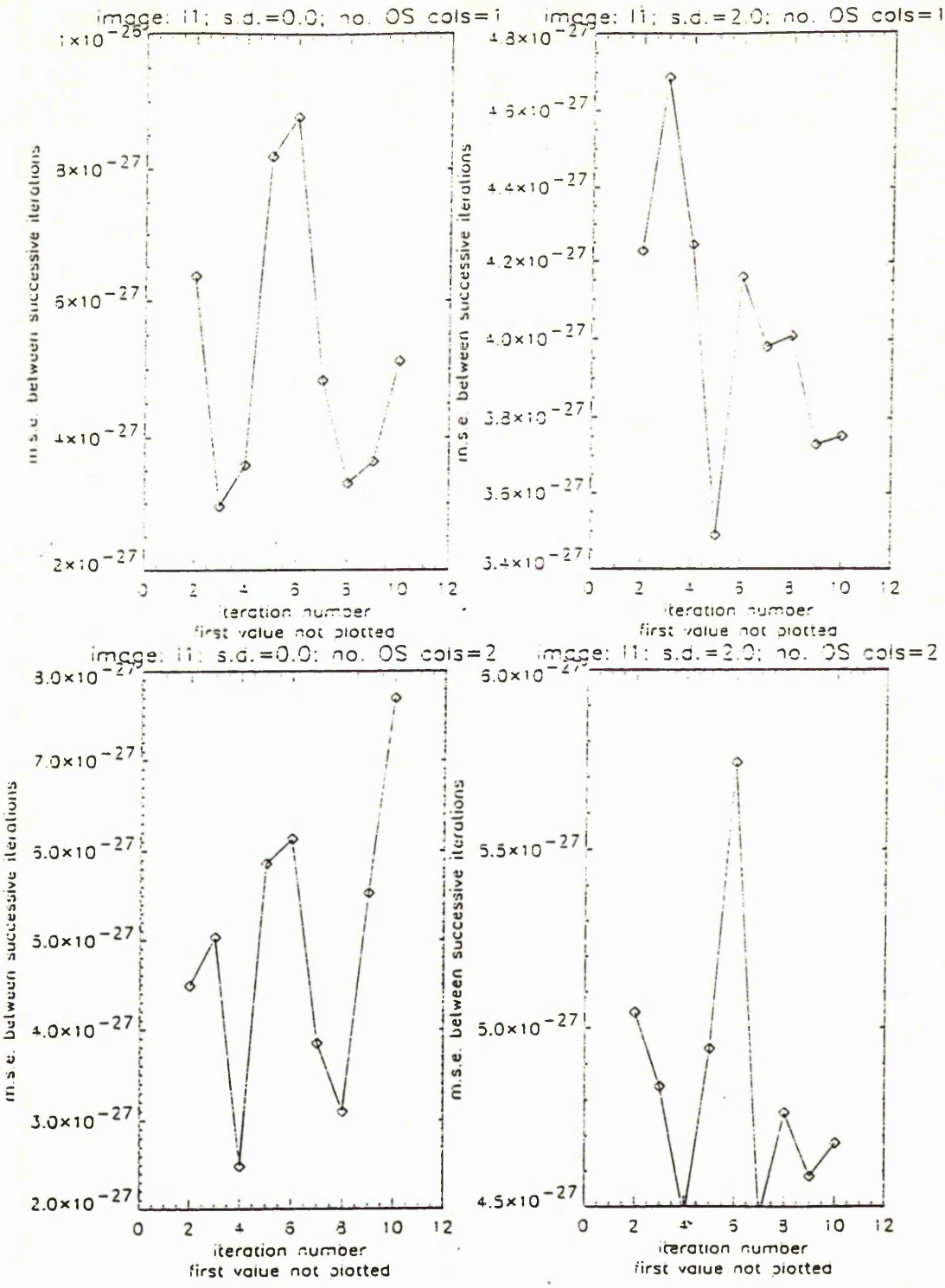


Figure 4.12: Mean squared error values calculated on successive reconstructions, with the large decrease at the first iteration excluded.

Chapter 5

Edge preserving image restoration

5.1 Introduction

In this chapter we turn our attention to the Gibbs prior distribution we have used to describe the underlying true image (or, rather, which we have used to describe the probability distribution of the pixel values which in turn describe the true image). In particular, we introduce an adaption of it which, while preserving the useful property of inducing a local Markov dependency structure, has the aim of capturing edges and preventing sharp discontinuities from being over–smoothed.

Such techniques could be useful in the field of medical imaging, for example, if they could help mark out more clearly areas of interest in remotely sensed data.

The work in this chapter is a direct extension to the work of Abdallah and Kay (see [1]). We attempt to fully automate their technique. Blake and Zisserman ([9]) also used a similar concept, that of building into the continuity–constrained prior model a smaller penalty for allowing an edge.

In this chapter, as in the previous, we focus on the problem of image restoration. Although parameters β and ϕ still require estimation, we will not concern

ourselves too greatly with their accuracy, if we can use the estimates to good effect.

The plan of the chapter is as follows. In the next section we discuss problems with the Gibbs prior we have been using, and suggest a possible remedy. Then we detail how parameters will be estimated, and present the algorithm for edge preserving image restoration. In the numerical section, we present results of the algorithm and compare them with the results obtained by Abdallah and Kay, as well as comparing different versions of the same process. We then carry out some simulation studies to see how sensitive the algorithm is to changes in its input parameters. Finally we present our conclusions.

5.2 The Gibbs distribution prior

Our desire that the probability distribution for x should display a local Markov dependency structure has dictated that we choose a Gibbs form for $p(x)$:

$$p(x) = Z(\beta)^{-1} \exp\{-\beta U(x)\}.$$

$U(\cdot)$ is the energy function, describing how the pixels interact with one another. We choose to specify pairwise interactions, so we can write

$$U(x) = \sum_{i \sim j} \Psi(x_i - x_j),$$

$i \sim j$ indicating, as usual, that pixels i and j are neighbours, as defined by the m.r.f. structure.

Until now we have been content to specify $\Psi(u) = u^2$ – a simple quadratic prior. Although simple to manipulate, this prior can be criticised for heavily

penalising large differences between neighbouring pixels, large differences which could reasonably be expected to occur at areas of sharp discontinuity.

In the literature, this problem seems to have been tackled in two distinct ways; either by adapting the prior to include a "line process", which attempts to accommodate edges by modelling them explicitly in the energy function U, or by changing the prior merely to less heavily penalise large pixel differences. The former is the method of Blake and Zisserman ([9]) and of the Gemans in their seminal paper on image restoration ([32]). The latter method is explored by Bouman and Sauer in [10], who suggest modifying the prior, by setting

$$\Psi(x_i - x_j) = \rho(\lambda \mid x_i - x_j \mid),$$

where λ is a scaling parameter, and $\rho(\cdot)$ is a monotone increasing non–convex function. Chosen appropriately, such a function would show quadratic behaviour near the origin – i.e., where pixels i and j are close in value – and flat, non-punitive behaviour for larger values.

In [30], two line processes are incorporated into the prior model, one horizontal, the other vertical. The novelty in this approach is that the m.r.f. is then approximated by a deterministic structure, which is used to estimate the value of the partition function Z, from which line process effects are averaged out.

A different approach is suggested by the authors of [33], who implemented a binary edge detection scheme (edges are either "on" or "off") by classifying groups of pixels as "alike" or "unalike" using a Kolmogorov–Smirnov distance measure.

Other choices suggested for $\Psi(\cdot)$ include that of Geman and McClure ([34]):

$$\Psi(u) = (1 + u^{-2})^{-1},$$

while Peter Green ([45]) proposes

$$\Psi(u) = c_1 \log \cosh c_2(u),$$

for certain values of c_1, c_2, chosen to make the behaviour of these two priors practically identical.

These choices of prior distribution certainly succeed in damping down the edge penalty. However, they are perhaps a little difficult to motivate, and the choice of parameterisation is arbitrary. In an effort to obviate these difficutlies, Abdallah and Kay, in [1], suggest the following form, which models the edges explicitly:

$$\Psi(x_i - x_j) = (x_i - x_j)^2(1 - e_{ij}). \tag{5.1}$$

The $\{e_{ij}\}$ denote our belief in the existence or not of an edge between pixels i and j; if we think there is no edge present, set $e_{ij} = 0$, otherwise, if there is an edge present, set $e_{ij} = 1$. A simple "on–off" scheme, where e_{ij} is *either* 0 *or* 1, we call a "discrete edge structure". As we may not wish to quantify our belief about an edge as strictly as this, we shall also examine a "continuous edge structure", that is, one where $e_{ij} \in [0, 1]$ for all i, j. Of course, we also preserve symmetry so that $e_{ij} = e_{ji}$.

There remains the problem of estimating these edge variables. If we for the moment ignore the fact that we do not know what value x takes, then a plausible scheme could be as follows. If it is assumed that the probability of an edge between two pixels increases as the absolute value of their difference increases,

then estimation of the edges may naturally be based upon consideration of the *contrast* variables $\{c_{ij}\}$, defined as

$$c_{ij} = \mid \bar{x}_i^{P_j} - \bar{x}_j^{P_i} \mid, \tag{5.2}$$

where $\bar{x}_i^{P_j}$ is the average value of the pixels i and the P pixels in the opposite direction to pixel j, where $j \in \delta_i$, the neighbourhood of i, and P is specified in advance (we take $P = 2$ in our experiments). This definition is more easily understood by examining Figure 5.1.

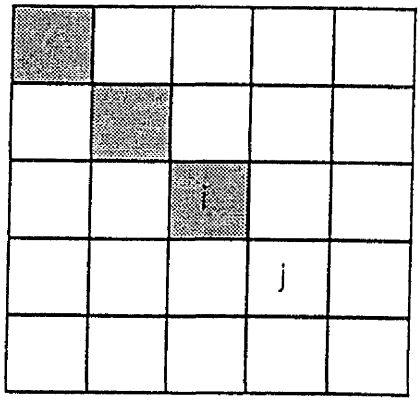


Figure 5.1: Explanation of the $\bar{x}_i^{P_j}$ notation

For images which contain edges, a likely form for the distribution of $c = \{c_{ij}\}$ to take is for it to be multimodal, with a number of local maxima corresponding to the edges, and an overall maximum, corresponding to the smooth part (most of) the image. We could therefore use the quantiles of c to make some sort of decision about e_{ij}. We compare three different quantile manipulation rules in our numerical work:

Continuous edges

$$e_{ij} = \begin{cases} 0 & c_{ij} \leq q_1 \\ \frac{c_{ij}-q_1}{q_2-q_1} & q_1 < c_{ij} < q_2 \\ 1 & q_2 \geq c_{ij} \end{cases}$$

Here, q_1, q_2 are the p_1, p_2 quantiles of the c distribution. Abdallah and Kay supplied p_1, p_2 intuitively, finding that the quality of the restored image seemed fairly robust against a range of p_1 values, but that selection of p_2 was critical. $1-p_2$ can be equated with the percentage of the image that is "edge", thus giving the user a physical parameter to guess at. We wish, however, to fully automate the process, and, bearing in mind the postulated structure of the c distribution, we specify q_1 and q_2 to be the c_{ij} values at either side of the largest gap in the $\{c_{ij}\}$ order statistic.

Discrete edges – version 1

Using the same definition of q_1, q_2, a straightforward rule for forming edges is

$$e_{ij} = \begin{cases} 0 & c_{ij} < \frac{q_1+q_2}{2} \\ 1 & c_{ij} \geq \frac{q_1+q_2}{2} \end{cases}$$

Discrete edges – version 2

A slightly more involved method to estimate the edges is to judge whether or not the corresponding contrast is "significant" in some pseudo–t–value manner. Thus, let $ese(c)$ be the estimated standard error of the c values; $ese = \sqrt{\frac{\sum_{i,j}(c_{ij}-\bar{c})^2}{n(n-1)}}$, where n is the number of contrasts in the image and $\bar{c}$ is their arithmetic mean. Then

$$e_{ij} = \begin{cases} 0 & \frac{c_{ij}}{ese(c)} < k \\ 1 & \frac{c_{ij}}{ese(c)} \geq k \end{cases} \tag{5.3}$$

Here, k is a user–supplied constant. In keeping with the pseudo–significance approach, we chose $k = 2$ in our experiments. This method does not require us to inspect or estimate quantiles of the contrast distribution.

Here then are three methods of estimating the images. We have assumed throughout the above that we know the true image x. In reality, this is precisely what we don't have, and before we can estimate x, we need to estimate or supply reasonable values for the other parameters in the model.

We note that there is a certain lack of clarity about the three quantities x,$\{e_{ij}\}$, $\{c_{ij}\}$, since each one is being used to define the others. One may ask, are the edges *fixed* and unknown parameters? This seems initially feasible. Like β, they are a part of the image prior, set in advance at a particular level which lead to the generation of the true x. Or, are they themselves random variables? This seems inescapable since we base their estimation on the contrast variables, which are themselves a function of the indubitably random x.

Some of this conundrum we later attempt to resolve, by adapting the hierarchical Bayesian argument of Chapter 2. Meanwhile we proceed by (falsely) imagining that at each step of the following procedures, the unknown quantities are in fact supplied by some external agent, without reference to other factors of the system.

5.3 Parameter estimation and image restoration

The estimator of ϕ we employ is the residual sums–of–squares divided by degrees of freedom that has performed well in previous chapters. The prior parameter β we will estimate by pseudo–likelihood ([6]), slightly more complicated than in Section 2.5 due to the inclusion of the edge term. Once again, for the moment we assume knowledge of the true image. Then, if we write $e = \{e_{ij}\}$, and $S \setminus i$ to denote the set of pixels with the i'th removed, the pseudo–likelihood function of β is

$$\begin{aligned} psl(x;\beta) &= \prod_{i=1}^{N} p(x_i \mid x_{\delta_i}, \beta, e) \\ &= \prod_{i=1}^{N} p(x_i \mid x_{S\setminus i}, \beta, e) \\ &= \prod_{i=1}^{N} \frac{p(x \mid \beta, e)}{\int_R p(x \mid \beta, e) dx_i} \\ &= \prod_{i=1}^{N} \frac{\exp[-\beta \sum_{i\sim j}(x_i - x_j)^2(1-e_{ij})]}{\exp[-\beta \sum_{\{(r,s):(r,s)\notin\{\delta_i \cup i\}\}}(x_r - x_s)^2(1-e_{rs})] I_i} \\ &= \prod_{i=1}^{N} \frac{\exp[-\beta \sum_{j\in\delta_i}(x_i - x_j)^2(1-e_{ij})]}{I_i}, \end{aligned} \tag{5.4}$$

where $I_i = \int_R \exp[-\beta \sum_{j\in\delta_i}(x_i - x_j)^2(1-e_{ij})]dx_i$, and R denotes the real line. If we expand the square in I_i, and move all terms not involving x_i outside the integral, we can write

$$I_i = \exp[-\beta \sum_{j\in\delta_i} x_j^2(1-e_{ij})] \int_R \exp[-\beta\{Ax_i^2 - 2x_i B\}]dx_i, \tag{5.5}$$

where $A = \sum_{j\in\delta_i}(1-e_{ij})$ and $B = \sum_{j\in\delta_i} x_j(1-e_{ij})$. Next we complete the square in the exponential term within the integral and rewrite (5.5) as

$$
\begin{aligned}
I_i &= \exp[-\beta \sum_j x_j^2(1-e_{ij})] \\
&\times \int_R \exp[-A\beta[(x_i - A^{-1}B)^2 - A^{-2}B^2]]dx_i \\
&= \exp[\beta\{\frac{[\sum_j x_j(1-e_{ij})]^2}{\sum_j(1-e_{ij})} - \sum_j x_j^2(1-e_{ij})\}] \\
&\times \int_R \exp[-A\beta(x_i - A^{-1}B)^2]dx_i. \qquad (5.6)
\end{aligned}
$$

The integral in (5.6) can be identified as a Gaussian c.d.f. with mean $A^{-1}B$ and variance $(2A\beta)^{-1}$, allowing us, finally, to write

$$
\begin{aligned}
I_i &\propto (2\beta \sum_j (1-e_{ij}))^{-1/2} \\
&\times \exp[\beta\{\frac{[\sum_j x_j(1-e_{ij})]^2}{\sum_j(1-e_{ij})} - \sum_j x_j^2(1-e_{ij})\}]. \qquad (5.7)
\end{aligned}
$$

The pseudo–likelihood estimator of β can now be found by substituting (5.7) into (5.4), taking logs, differentiating and maximising, to find

$$
\hat{\beta}_{psl} = \frac{N}{2\sum_{i=1}^N \{\sum_{j\in\delta_i}(1-e_{ij})[x_i - \frac{\sum_j x_j(1-e_{ij})}{\sum_j(1-e_{ij})}]^2\}}.
$$

We have now detailed how we aim to estimate the edges, and the m.r.f. parameter β. However, both these estimators have assumed knowledge of the true image x. In practice, of course, we will need to provide a good estimate of x. Here, we use an approximant to the modal image estimate , Besag's **Iterated Conditional Modes** ([6]), which at each step of the recursive procedure picks as an estimate for pixel m the most likely value given the data, other parameter estimates, and the present values of the pixels in the m.r.f. neighourhood. That

is,

$$\hat{x}_m^{new} = \text{argmax}_x p(x_m \mid \hat{x}_{\delta_m}^{old}, y, \hat{e}, \hat{\beta}),$$

for $m = 1, \ldots, N$. So we need to compute

$$p(x_m \mid x_{\delta_m}, e, y, \beta) = \frac{p(x \mid e, y, \beta)}{\int_R p(x \mid e, y, \beta) dx_m}. \tag{5.8}$$

We assume the data is of Gaussian form, as in (2.8), and through an argument similar to the early stages of the computation of the pseudo–likelihood, (5.8) can be seen to be proportional to

$$\exp[(-1/2\phi) \sum_{i \in \{B_m \cup m\}} (y_i - \sum_{j \in B_i} h_{ij} x_j)^2 - \beta \sum_{j \in \delta_m} (x_m - x_j)^2 (1 - e_{mj})].$$

Here, B_m represents the "blurring" neighbourhood of pixel m, that is, those pixels which are covered by the point spread function (discussed in Section 2.2) associated with matrix H, when it is centred on pixel m.

Some more algebra (taking logs, differentiating, solving) produces the following estimate of x_m

$$\begin{aligned} \hat{x}_m^{new} &= (\sum_{i \in B_m} (y_i - \sum_{j \in B_i} h_{ij} x_j^{old}) + 2\phi\beta \sum_{j \in \delta_m} x_j (1 - e_{mj})) \\ &/ \ (\sum_{i \in B_m} h_{im}^2 + 2\phi\beta \sum_{j \in \delta_m} (1 - e_{mj})), \end{aligned} \tag{5.9}$$

for $m = 1, \ldots, N$.

5.4 The Edge Preserving Image Restoration Algorithm

Assembling together the estimators for the edges, other parameters and the image itself, we can construct this edge–preserving parameter estimation and image restoration algorithm.

Algorithm **edge**

1. Choose $\hat{x}^{old}$; set $e_{ij} = 0$ for all i, j.

2. Estimate ϕ, β from y, $\hat{x}^{old}$ and $\{e_{ij}\}$, using the standard formula for ϕ and the pseudolikelihood formula for β in (5.3).

3. Use formula (5.9) to update $\hat{x}$.

4. Estimate the edges using one of the methods described above.

5. Check for convergence of $\hat{x}$:

   YES $\longrightarrow$ **STOP**

   NO $\longrightarrow$ set $\hat{x}^{new} := \hat{x}^{old}$ and go to step 2. □

This algorithm, although automatic, will probably suffer from the lack of consistency we discussed above. In particular we have no fixed point justification for the algorithm, due to the fact that the updating equations are derived from disparate sources and principles.

5.5 Numerical Work

For our tests of this algorithm, we used two artifical images, both containing sharp discontinuities. They were **I1**, which we have used in many other experiments, and **I7**, an image containing several circles. The same Gaussian noise, of s.d. 5.0, was added to each image, after a geometrical blurring had been carried out whose p.s.f. matrix was of size 7×7. Since initial experiments and nearly all our previous work suggest that the estimator of ϕ works well, we assumed in this study that the correct value of ϕ was known. The definition of the contrast

variables requires taking an average over a pre–specified number of pixels. We set this number to 2, which could perhaps be criticised for being too low.

We tested all three methods of estimating the edges ("cts" in the tables below signifies that they were assumed to be continuous, "ave" means that they are discrete and based on the quantile average, while "ese" indicates they are discrete and based on the estimated standard errors of the contrasts) and used three techniques to specify the values of q_1 and q_2. First, we specified the values of (p_1, p_2) in advance ("user" in the table of results); following Abdallah and Kay we set $p_1 = 0.1, p_2 = 0.85$. Next, we examined fully automatic estimation of (q_1, q_2) ("auto" in the tables); that is, at each iteration we chose them to be either end of the largest gap in the $\{c_{ij}\}$ order statistic. Finally, as a compromise between full automation and user–interference, we combined both methods, allowing user–specification of p_1, p_2 for the first seven iterations, then switching to the automatic method (labelled "semi" below). Unfortunately only one of the experiments ran for enough iterations for the switchover to occur.

Convergence was assumed when the mean square error between successive iterations' image estimates fell to less than or equal to 0.5, where we define m.s.e., as usual, as $mse(\hat{x}^{iter}, \hat{x}^{iter-1}) = N^{-1} \sum_{i=1}^{N} (\hat{x}_i^{iter} - \hat{x}_i^{iter-1})^2$.

5.5.1 Results

image: I1

edge method	quantile method	$\hat{\beta}$	$\hat{q}_1$	$\hat{q}_2$	no. iters
cts	user	0.7604	0.1	11.3	5
	auto	0.1525	27.8	29.9	3
ave	user	0.6727	0.0	12.5	12
	auto	0.1522	28.1	29.8	3
	semi	0.1659	27.6	29.8	11
ese	N/A	0.4031	—	—	4

image: I7

edge method	quantile method	$\hat{\beta}$	$\hat{q}_1$	$\hat{q}_2$	no. iters
cts	user	1.1043	0.1	3.5	4
	auto	0.1558	22.6	23.9	2
ave	user	0.7619	0.2	4.2	4
	auto	0.1552	22.7	23.9	2
ese	N/A	0.7045	—	—	3

Discussion

The results suggest that, if we are using one of the quantile methods to estimate the edges, the estimate of β is unaffected by the assumption of either continuous or discrete edges. However, supplying "sensible" values for p_1, p_2 leads to very different results for $\hat{\beta}$ than allowing the data to specify q_1, q_2.

The quantiles, when estimated from the data, are much larger than when specified directly. In Figure 5.2 we present histograms of the contrast variables for the true images (undistorted by blur and noise) and for the raw data. The true images display the clear multimodality corresponding to the different edges that we expected, but the data shows no such "nice" clustering, and so our hope that the largest gap in the $\{c_{ij}\}$ values would lead to a useful cut–off point to decide "edge" from "not–edge" looks overly optimistic.

The one experiment which ran for enough iterations to switch from user–supplied to automatic quantile estimation instantly exhibited behaviour typical of those experiments where q_1, q_2 were automatically selected from iteration one.

The "ese" method of edge detection seems to have been a compromise between the "user" and "auto" methods, producing estimates of β smaller than the former and larger than the latter.

There remains the important consideration of the image restorations themselves. Figure 5.3 displays the true images and the raw data; Figures 5.4 and 5.5 show various restorations.

With respect to image I1, the user–supplied quantiles restoration does appear superior to the three automatic attempts, in that much sharper edges are clearly evident. For I7, however, the two user–supplied restorations, although again more sharply defined, have spurious artifacts in areas of the image we would expect to be uniform. Perhaps image I7 requires specification of different values of p_1, p_2 – precisely the problem with this method.

A final point is that the contrast variables were calculated as the average over three pixels only (the original pixel and two others); this may not have extended far enough into the image to find enough of a difference to weight the contrast for that direction highly enough.

5.6 Simulation study

We have already expressed the worry that each of the edge–detection methods which are based on the distribution of the contrast variables (Equation (5.2))

may be sensitive to specification of parameter P. Recall that P is the number of pixels used in the construction of the contrast between every pair of pixels. We may ask, how important is the value of P? Would an alteration in its value from that of 2, which we used throughout Section 5.5, significantly affect results?

For the method of edge estimation based on the estimated standard error of the $\{c_{ij}\}$ (Equation (5.3)), there is a further parameter of interest – the pseudo t–value, k, which in the experiments in Section 5.5 was set to 2.0, and which is clearly of crucial importance in deciding the presence or absence of an edge. We chose $k = 2.0$ as a nod towards the practice of significance testing (a standard normal random variable has probability 0.05 of being larger than 1.96 or less than -1.96), but since the $\{c_{ij}\}$ are neither normally distributed nor independent, it is hard to justify this choice.

We therefore carried out two simulation studies to investigate the effect of these parameters.

5.6.1 Methodology

We used image **I7**, with the same blur/noise degradation used in Section 5.5, and the same convergence criterion for each run of Algorithm **edge**. Edges were estimated using the discrete e.s.e approach.

The estimate of β and the m.s.e. between the restored image and the true image were used to judge the effect of the varying parameters.

Each simulation run was carried out 1000 times.

In Set 1, we varied the value of k, and held P constant (and equal to 2). The threshold value was allowed to vary over $k = 1.5, 2.5, 3.5, 4.5$. In Set 2, we held k

constant (at 2.0) and varied the value of P, choosing $P = 2, 3, 4, 5$.

5.6.2 Results

Tables 5.6.1 and 5.6.2 display the results obtained by varying parameters k and P, respectively. We display the average value of the statistics over each simulation run, and the estimated standard errors (denoted in the tables by "e(·)").

Table 5.6.1 : Results from Set 1

k	$\bar{\hat{\beta}}$	e($\bar{\hat{\beta}}$)	$m.\bar{s}.e.$	e($m.\bar{s}.e.$)
1.5	0.6990	0.0013	8539.93	0.4423
2.5	0.5506	0.0011	8539.68	0.4535
3.5	0.4473	0.0012	8539.28	0.4562
4.5	0.5160	0.0052	8537.98	0.4603

Table 5.6.2 : Results from Set 2

P	$\bar{\hat{\beta}}$	e($\bar{\hat{\beta}}$)	$m.\bar{s}.e.$	e($m.\bar{s}.e.$)
2	0.6186	0.0012	8539.76	0.4507
3	0.5159	0.0011	8539.31	0.4553
4	0.4392	0.0021	8538.83	0.4544
5	0.5854	0.0049	8537.22	0.4602

5.6.3 Discussion

Regarding β estimation, the value of both k and P is crucial. It appears that $\hat{\beta}$ tends to *decrease* as both k, P *increase*, until a changepoint is reached, after which the estimates increase.

Restoration of the image, however, seems very robust across the range of both parameters, with respect to the measured $m.s.e.$. Since we have already stated that we were more concerned with restoration of the image, rather than

with parameter estimation – since these images are certainly not realisations of Markov random fields – we are encouraged by these results.

In Figure 5.6, we display some of the histograms of the estimates of β and *m.s.e.* for different values of k and P. It is clear that the distributions of β are strongly (k, P) dependent, while those of *m.s.e.* are not.

5.7 Further work

It may be possible to marry the "edge–preserving" prior with a more formal hierarchical–modelling approach. Recall that this requires us to specify a probability distribution for all unknown quantities, and then use Bayes' theorem to form a posterior distribution in the quantities in which we are interested. We could approach this in the following manner.

For simplicity, assume that $p(\beta) \propto \beta^{1/2}$, and $p(\phi) \propto \phi^{-1/2}$; that the likelihood of the data is Gaussian:

$$p(y \mid x, \phi) \propto \phi^{-N/2} \exp\{-\frac{1}{2\phi}\|y - Hx\|^2\},$$

and that the prior for x is as specified in this chapter, in Equation (5.1). We will need a hyperprior for the edges, $\{e_{ij}\}$.

We could define the *total edge strength* associated with pixel i as

$$e_i = \sum_{j \in \delta_i} e_{ij}^{\gamma}.$$

To keep our discussion as simple as possible, we will assume $\gamma = 2$, although theoretically it too could be estimated in the hierarchical framework.

Then we could assume that the prior distribution of the edges is

$$p(e) \propto \exp[-\theta \sum_i \sum_{j \in \delta_i} e_{ij}^2]. \tag{5.10}$$

For the sake of estimability, we need to specify a prior for θ; again, to keep notation simple, set $p(\theta) \propto \theta^{1/2}$.

Then we can form the log of the posterior distribution of the unknown quantities, given the data, as

$$\begin{aligned} \log[p(x, \phi, \beta, e, \theta \mid y)] \quad \propto \quad & \frac{-(N+1)}{2} \log \phi + \frac{1}{2} \log \beta \\ & -\frac{1}{2\phi} \|y - Hx\|^2 - \beta \sum_{i \sim j} (x_i - x_j)^2 (1 - e_{ij}) \\ & + \frac{1}{2} \log \theta - \theta \sum_i \sum_{j \in \delta_i} e_{ij}^2. \end{aligned} \tag{5.11}$$

This seems formulaically similar to the ideas of Blake and Zisserman ([9]). Now by taking the derivative of $L(y) = \log[p(x, \phi, \beta, e, \theta \mid y)]$ with respect to x, β, ϕ, θ, estimators can be found which lead to an iterative algorithm for finding the m.a.p. estimates, in the same manner as in Chapter 2. For example,

$$\hat{\beta} = \frac{1}{2} \sum_{i \sim j} (x_i - x_j)^2 (1 - e_{ij}),$$

$$\hat{\phi} = \frac{\|y - Hx\|^2}{N+1},$$

$$\hat{\theta} = \frac{1}{2 \sum_i \sum_{j \in \delta_i} e_{ij}^2},$$

and

$$\hat{x}_m = \frac{\sum_i h_{im} (y_i - \sum_{j \in \delta_i} h_{ij} x_j) + \sum_{j \in \delta_m} x_j (1 - e_{mj})}{(2\phi\beta) \sum_{j \in \delta_m} (1 - e_{mj})},$$

for $m = 1, \ldots, N$. However, the normal equation for the edge variable is not so easily solved, since

$$\frac{\delta L}{\delta e_{kl}} = \beta \sum_{k \sim l} (x_k - x_l)^2 - 2\theta \sum_k \sum_{l \in \delta_k} e_{kl},$$

which equals zero when

$$\sum_k \sum_{l \in \delta_k} e_{kl} = \frac{\beta}{2\theta} \sum_{k \sim l} (x_k - x_l)^2. \tag{5.12}$$

The solution to (5.12) would itself require an iterative procedure, within each iteration of the algorithm.

Alternatively, in a pseudo–empirical Bayes manner, at each step in the algorithm the $\{e_{ij}\}$ could be estimated in one of the ways we investigated earlier.

5.8 Summary

We have presented a fully automatic restoration algorithm which seeks to preserve image discontinuities, by building an edge term into the image prior distribution. A more interactive form of the algorithm, where the user supplies the (p_1, p_2) quantiles of an edge-related variable's distribution, seems a sensible approach, if specification of p_1, p_2 is non–problematic. If this does present a problem, then any of the automatic methods seem capable of producing restorations which are almost as good, and which are certainly an improvement on the data.

A simulation study indicated that the other parameters which must be set by the user, although strongly influential on the estimation of the β parameter, have practically no effect on the restoration of the image. Edges in the true image are more sharply distinguished using this adapted prior than was the case

in Chapters 2 and 3, which utilised the simple quadratic prior.

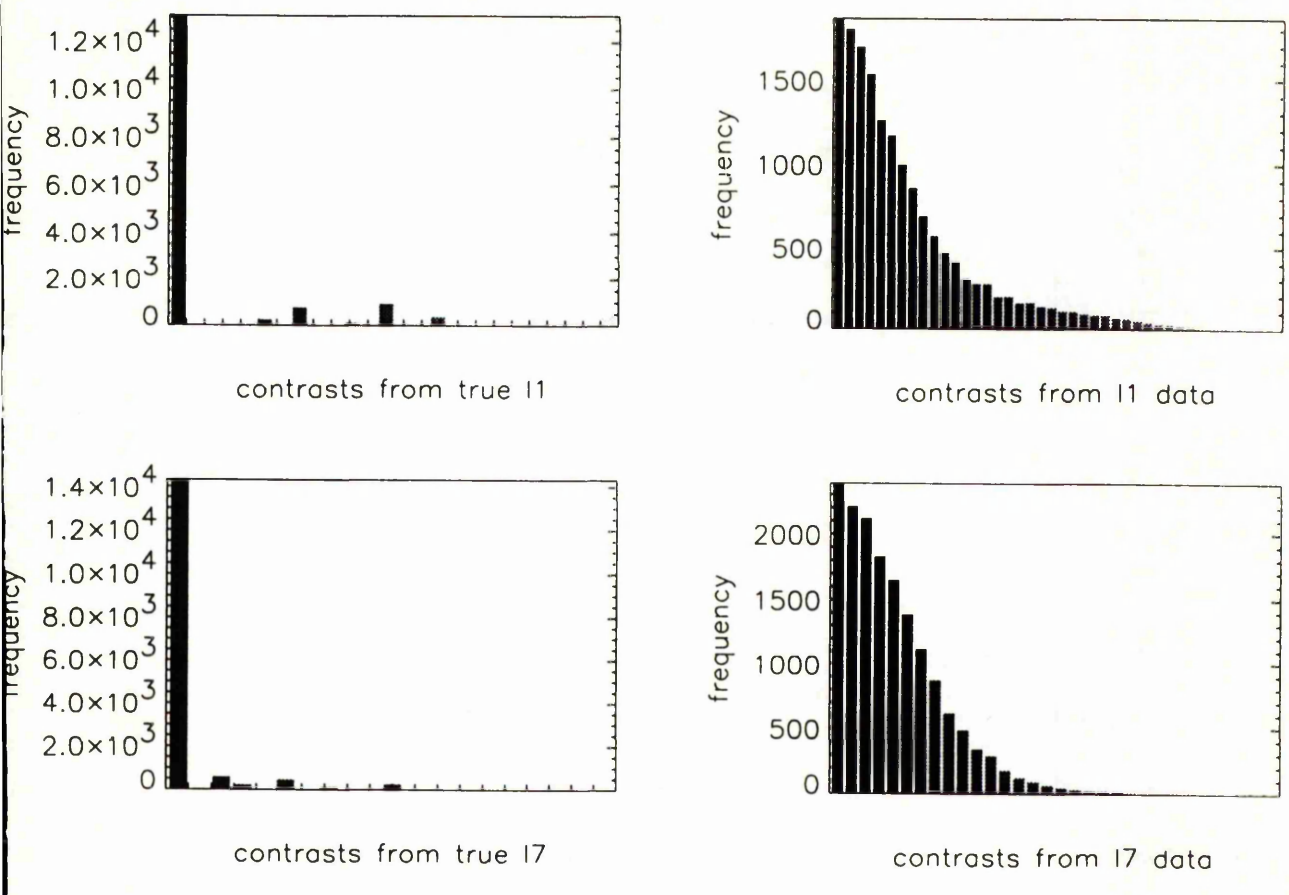


Figure 5.2: Histograms of the contrast variables.

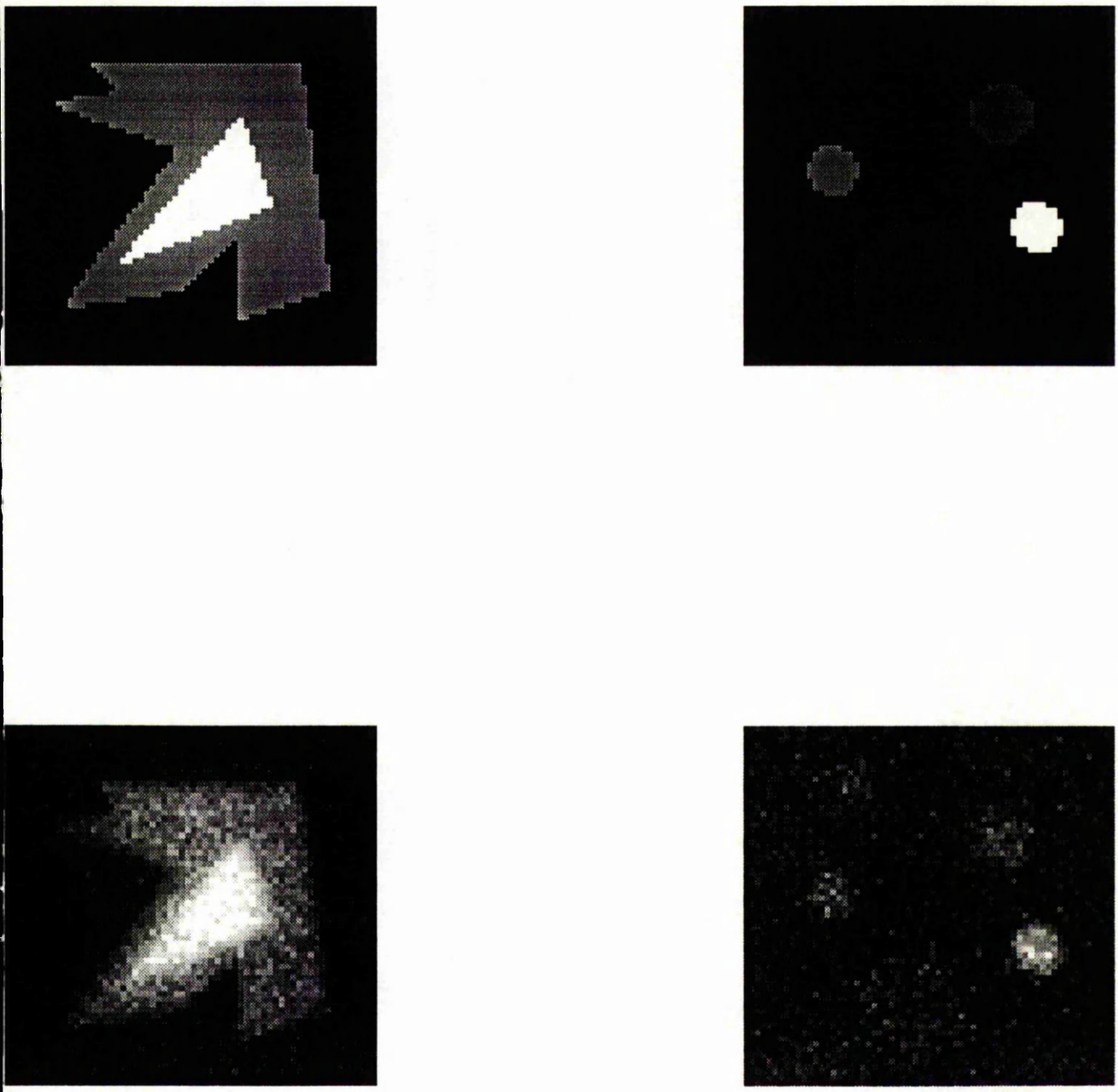


Figure 5.3: The true images and the data. From top left to bottom right: I1, I7, data(I1), data(I7).

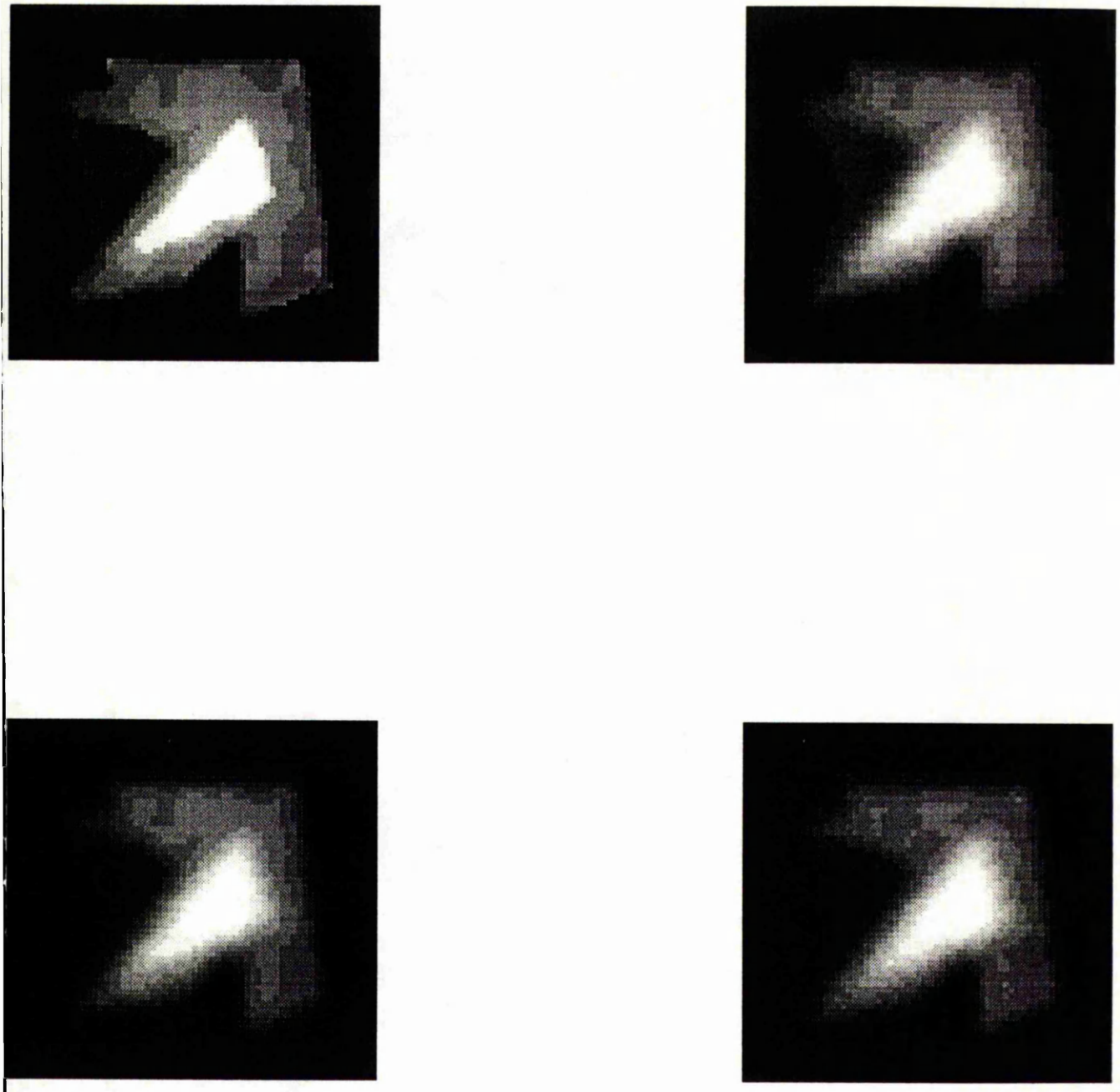


Figure 5.4: Some restorations of image II. From top left to bottom right, we have (i) user-supplied quantiles, continuous edges; (ii) automatically selected quantiles, continuous edges; (iii) automatically selected quantiles, discrete edges; (iv) discrete edges based on the e.s.e. of the contrasts.

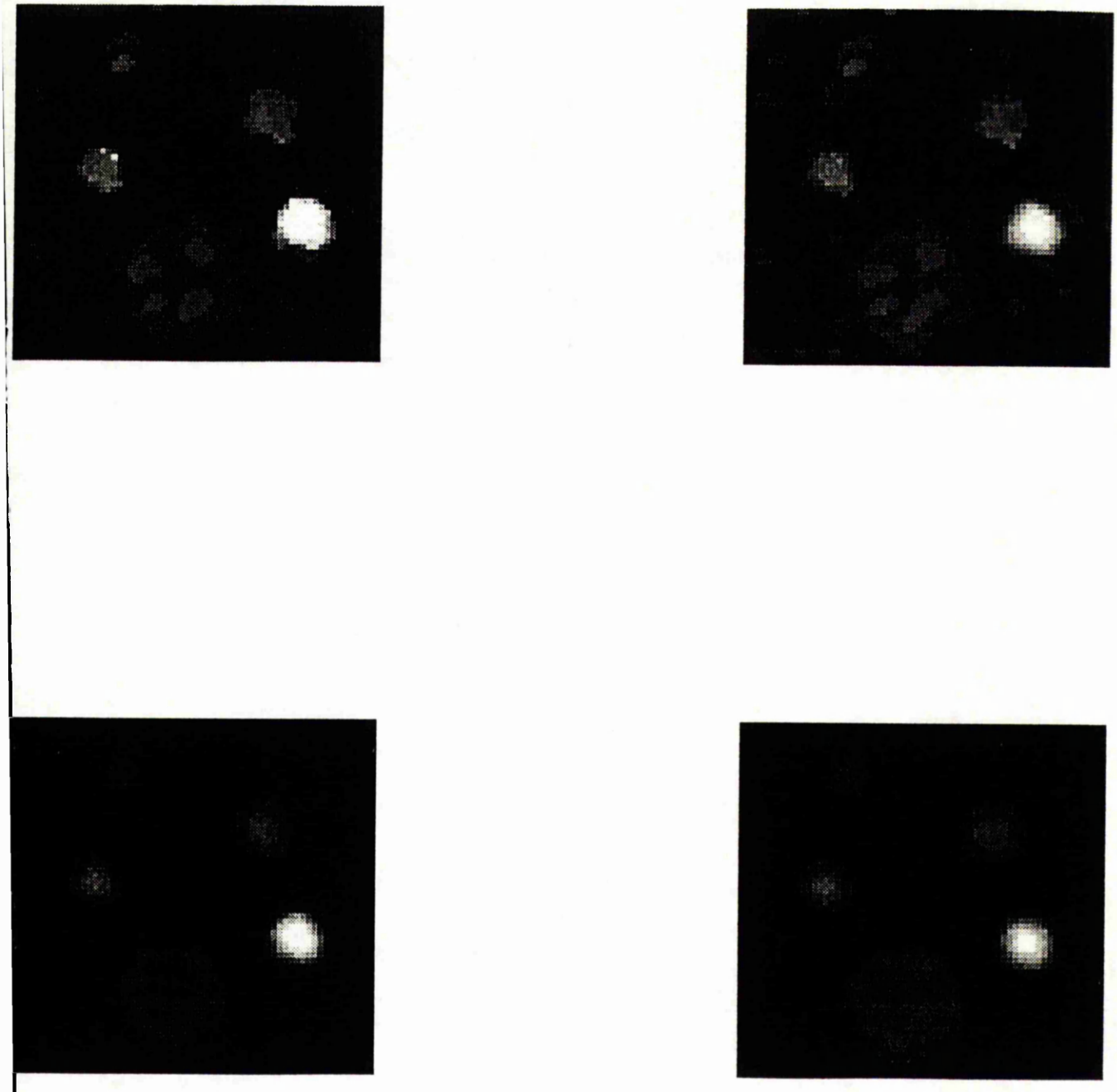


Figure 5.5: Some restorations of image 17. From top left to bottom right, we have (i) user–supplied quantiles, continuous edges; (ii) user–supplied quantiles, discrete edges; (iii) automatically selected quantiles, discrete edges; (iv) discrete edges based on the e.s.e. of the contrasts.

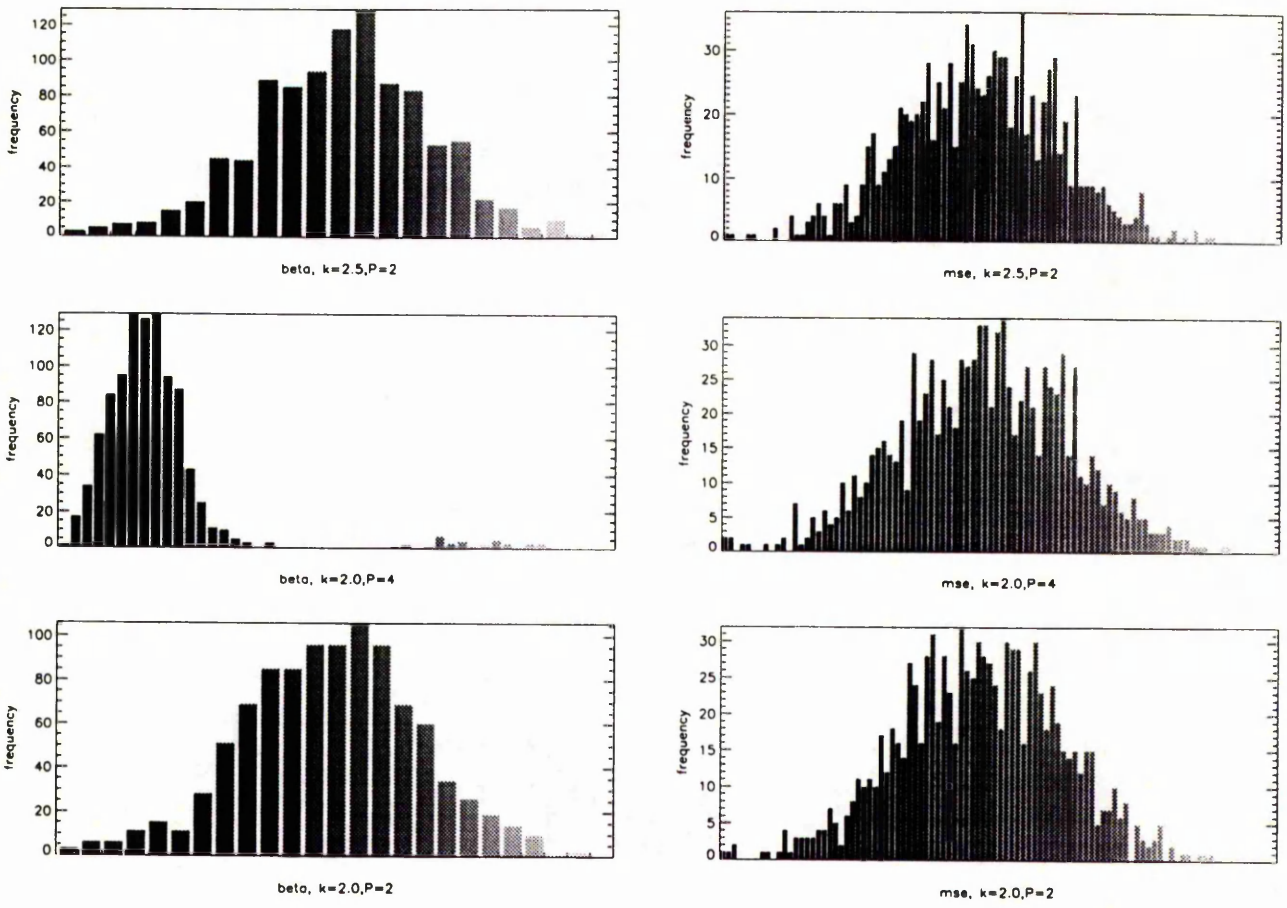


Figure 5.6: Histograms of the sample distributions of statistics β and $m.s.e.$, for different values of the parameters k and P.

Bibliography

[1] Abdallah, M. and Kay, J.W. : Edge Preserving Image Restoration , in *Lecture Notes In Statistics: Stochastic Models, Statistical Methods, and Algorithms in Image Analysis*, 1992, editors: P.Barone, A. Frigessi and M. Piccioni.

[2] Archer,G.E.B. and Titterington,D.M. : On some Bayesian/regularization methods for image restoration, **IEEE Trans. Im. Proc.**, to appear.

[3] Archer,G.E.B. and Titterington,D.M. : On the iterative image space reconstruction algorithm for solving positive linear inverse problems, **Statistica Sinica**, to appear.

[4] Aykroyd, R.G. and Green, P.J. : Global and local priors, and the location of lesions using Gamma camera imagery, **Phil. Trans. R. Soc. Lond. A.**, 1991, 337, pp323–342.

[5] Besag, J. : Spatial Interaction and the statistical analysis of lattice systems, **J.Royal Statist. Soc., Series B**, 1974, 36, No.2, pp192–236.

[6] Besag, J. : On the statistical analysis of dirty pictures, **J. Royal Statist. Soc., Series B**, 1986 , No.3, pp259-302.

[7] Besag, J. and Green, P.J.: Spatial Statistics and Bayesian Computation, **J. Royal Statist. Soc., Series B**, 1993, Vol. 55, No.1, pp25–37.

[8] Besag, J., York,J. and Mollie', A.: Bayesian Image restoration, with two applications in Spatial Statistics, **Ann. Inst. Statist. Maths. (with discussion)**, 1991, 43, pp1–59.

[9] Blake, A. and Zisserman, A.: Localising discontinuities using weak continuity constraints, **Pattern recognition Letters**,1987, Vol.6,No.1,pp51–59.

[10] Bouman, C. and Sauer, K.: A generalized Gaussian Image Model for edge–preserving m.a.p. restoration,**IEEE Trans. Im. Proc.**, 1993, Vol.2, No.3, pp296–310.

[11] Byrne, C.L.: Iterative Image Reconstruction Algorithms based on cross–entropy minimization, **IEEE Trans. Im. Proc.**, 1993, Vol.2, No.1, pp91–103.

[12] Chu, W.P. : Convergence of Chahine's nonlinear relaxation inversion method used for limb viewing remote sensing, 1970, **Applied Optics**, Vol. 24, pp445–447.

[13] Clifford, P : Discussion on the meeting on the Gibbs Sampler and other Markov Chain Monte Carlo methods, **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.1, pp53–102.

[14] Cressie,N.A.C,: *Statistics for Spatial Data*, 1993, John Wiley and Sons.

[15] Daube–Witherspoon, M.E. and Muehllehner, G.: An iterative image space reconstruction algorithm suitable for volume ECT, **IEEE Trans. Med. Imag.**, 1986, Vol. M1–5, No.2, pp61–66.

[16] De Groot, M. H. *Probability and Statistics*, Addison–Wesley, USA, 1986.

[17] Dempster, A.P., Laird,N.M. and Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM algorithm, **J. Royal Statist. Soc., Series B**, 1977, Vol.39, pp1–38.

[18] De Pierro, A.R. : On the convergence of the iterative Image Space Reconstruction Algorithm for volume ECT, 1987, **IEEE Trans. Med. Imag**, 1987, Vol. 6, pp174–175.

[19] De Pierro, A.R. : Nonlinear relaxation methods for solving symmetric linear complementary problems, **J. Optim. Theory Appl.**, 1990, Vol. 64, pp87–99.

[20] Derin,H. and Elliott, H.: Modelling and Segmentation of noisy and textured images using Gibbs Random Fields, **IEEE Trans. Patt. Anal. Mach. Intel.**, 1987,Vol.9, No.1, pp39–55.

[21] Di Gesu, V. and Maccarone, M.C. : The Bayesian direct deconvolution method: properties and applications, **Signal Processing**, 1984, Vol. 6, pp201–211.

[22] Donoho, Johnstone, Hoch, Stern : Maximum Entropy and the Nearly Black Object, **J. Royal Statist. Soc., Series B**, 1991, 53, no.3

[23] Dubes, R.C. and Jain, A.K.: Random field models in image analysis, **J. App. Stats.**, 1989, Vol.16, No.2, pp131–164.

[24] Efron, B.: *The jackknife, the bootstrap, and other resampling plans*, 1982, Philadelphia: The Society for Industrial and Applied Mathematics.

[25] Efron, B.: Jackknife after bootstrap and other resampling plans, **J. Royal Statist. Soc., Series B**, 1992, Vol.54, pp83–127.

[26] Eggermont, P.P.B.: Multiplicative Iterative Algorithms for convex programming, **Linear Algebra and its applications** , 1990, No.130, pp25–42.

[27] de Finetti, B. : *Probability, Induction and Statistics: the Art of Guessing*, 1972, John Wiley and Sons Ltd.

[28] Frigessi, A., di Stefano,P., Hwang, C.–R. and Shell,S.–J.: Convergence rates of the Gibbs Sampler, the Metropolis Algorithm and other single–site updating programs, **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.1, pp205–219.

[29] Galatsanos,N.P. and Katsaggelos,A.K. : Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation, **IEEE Trans. Imag. Proc.**, 1992, Vol.1, No.3, pp332–336.

[30] Geiger, D. and Girosi, F.: Parallel and Deterministic Algorithms from m.r.f.'s: surface reconstruction, **IEEE Trans. Patt. Anal. Mach. Intel.**, 1991, Vol.13, No.5, pp401–412.

[31] Gelman, A.: Iterative and non–iterative simulation algorithms, in *Proceedings of the 24th Symposium on the Interface between Computing Science and Statistics* , Vol.24, ed. H. J. Newton.

[32] Geman, D. and Geman, S. : Stochastic Relaxation, Gibbs Distributions and the Bayesian restoration of images, **IEEE Trans. Patt. Anal. Mach. Intell.**, 1984, Vol.6, pp721–741.

[33] Geman,D., Geman, S., Graffigne, C. and Dong, P. : Boundary Detection by Constrained Optimisation, **IEEE Trans. Patt. Anal. Mach. Intell.**, 1990, Vol. 12, No.7, pp609–628.

[34] Geman, S. and McClure, D. : Statistical Methods for tomographic image reconstruction, **Bull. Int. Stat. Inst.** ,1987, Vol. LII, No.4, pp5–21.

[35] Geyer, C.J. and Thompson, E.A. : Constrained Monte Carlo Maximum Likelihood for Dependent Data, **J. Royal Statist. Soc., Series B**, 1992, Vol. 54, No.3, pp657–699.

[36] Gidas, B. and Hudson, H.M. : A two stage EM algorithm with applications in emission tomography, *Brown University Complex System Report*, 1991, No.39.

[37] Gilks, W.R., Clayton, D.G., Spiegelhalter, D.J., Best, N.G., McNeil,A.J., Sharples, L.D. and Kirby, A.J.: Modelling complexity: Applications of Gibbs Sampling in Medicine, **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.1, pp39–52.

[38] Gilks, W.R., Thomas, A. and Spiegelhalter, D.J.: Software for the Gibbs Sampler, in *Proceedings of the 24th Symposium on the Interface between Computing Science and Statistics*, Vol.24, ed. H. J. Newton.

[39] Golub,G., Heath,M. and Wahba, G.: Generalised cross–validation as a method for choosing a good ridge parameter, **Technometrics**, 1979, vol.21, pp215–223.

[40] Gonzalez, R.C. and Woods, R.E.: *Digital Image Processing*, 1992, Addison–Wesley.

[41] Gray, A.J. : Simulating Posterior Gibbs Distributions: a comparison of the Swendsen–Wang and Gibbs Sampler methods, preprint.

[42] Gray, A.J., Kay, J.W. and Titterington. D.M.: On the estimation of noisy binary markov random fields, **Pattern Recognition**, 1992, Vol.25, pp749–768.

[43] Gray, R.M. : On the asymptotic eigenvalue distribution of Toeplitz matrices, **IEEE Trans. on Inf. Th.**, 1972, Vol IT-18, No.6, pp725–729.

[44] Green, P.J.: On the use of the EM algorithm for penalized likelihood estimation, **J. Royal Statist. Soc., Series B**, 1990, Vol.52, No.3, pp443–452.

[45] Green, P.J.: Bayesian reconstructions from Emission Tomography data using a modified EM algorithm, 1990, **IEEE Trans. Med. Imag.**, Vol.9, No.1, pp84–94.

[46] Green, P : discussion to Vardi, Lee : "From image deblurring to optimal investments: maximum likelihood solutions for positive linear inverse problems", **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.3.

[47] Green, P.J., Han, X-L. : Metropolis Methods, Gaussian proposals and antithetic variables, in *Lecture notes in Statistics: Stochastic models, statistical methods and algorithms in image analysis*, Springer, Berlin, editors: P.Barone, A.Frigessi and M.Piccioni.

[48] Green,P.J. and Titterington, D.M.: Recursive methods in image processing, invited paper to the 46th session of the I.S.I.

[49] Gull, S.: Developments in Maximum Entropy Data Analysis, in *Maximum Entropy and Bayesian Methods* , Cambridge 1988, pp 53–71, ed. J. Skilling

[50] Hall, P. and Titterington, D.M. : On some smoothing techniques used in image restoration, **J. Royal Statist. Soc., Series B**, 1986, Vol.48, No.3, pp330–343.

[51] Hall, P. and Titterington, D.M. : Common Structure of Techniques for choosing Smoothing Parameters in Regression Problems, **J. Royal Statist. Soc., Series B**, 1987, Vol.49, No.2, pp184–198.

[52] Hanna, S.R. : Confidence Limits for Air Quality Model evaluations, as estimated by bootstrap and jackknife resampling methods, **Atmospheric Environment**, 1989, Vol.23, pp1385–1398.

[53] Hastings,W.K. : Monte Carlo sampling methods using Markov Chains and their applications, **Biometrika**, 1970, Vol.57, No.1, pp97–109.

[54] Heikkinen, J. and Hogmander, H. : Fully Bayesian Approach to Image Restoration, Preprint from the Department of Statistics, University of Jykvaskyla, June 1993.

[55] Howson, C. and Urbach, P. : *Scientific Reasoning : the Bayesian Approach*, 1989, Open Court Publishing Company.

[56] Hudson, H.M. and Larkin, R.S.: Accelerated Image Reconstruction using Ordered Subsets of Projection Data, **J. Nucl. Med.**, 1994, Vol. MI–13, No.4, pp1–9.

[57] Hume, D. : *An Inquiry Concerning Human Understanding*, 1777, ed. L.A. Selby-Bigge. Oxford: The Clarendon Press.

[58] Hunt, B.R. : The application of constrained least squares estimation to image restoration by digital computer, **IEEE Trans. on Computers**, 1973, Vol. C-22, No.9, pp805–812.

[59] Kaufman, L.: Implementing and accelerating the EM algorithm for positron emission tomography, 1987, **IEEE Trans. Med. Imag.**, Vol. 1, pp37–51.

[60] Kay, J.W. : On the choice of regularization parameter in image restoration, in *Pattern Recognition*, 1988, ed. J. Killer, pub. Springer, New York.

[61] Kay, J.W. : A Comparison of Smoothing Parameter Choices in Image Restoration, in *Spatial Statistics and Imaging*, ed. A. Possolo, pub. Inst. of Math. Stats.

[62] Kirkland, M.: Simulating Markov Random Fields, 1989, PhD thesis, University of Strathclyde.

[63] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P.: Optimization by Simulated Annealing, **Science**, 1983, Vol.220, No.4598, pp671–680.

[64] Lewitt, R.M. and Muehllehner, G. : Accelerated iterative reconstruction for positron emission tomography based on the EM algorithm for maximum likelihood estimators, **IEEE Trans. Med. Imag.**, 1986, Vol. 5, pp16–22.

[65] Lindley, D.V. and Smith, A.F.M.: Bayes estimates for the Linear model,**J. Royal Statist. Soc., Series B**, 1973, Vol.1, pp1–41.

[66] Little, R.J.A. and Rubin, D.B.: On jointly estimating parameters and missing data by maximising the complete data likelihood, **Am. Stat.**, 1983, Vol.37, No.3, pp218–220.

[67] Mackay, David J.C. : Hyperparameters : optimise or integrate out? (preprint).

[68] Meng, X.–L.: On the rate of convergence of the ECM algorithm, **Ann. Statist.**, to appear.

[69] Meng,X.–L. and Rubin, D.B.: Maximum Likelihood estimation via the ECM algorithm: a general framework, **Biometrika**, 1993, Vol.80, No.2.

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

[71] Neal, R.M. and Hinton, G.E.: A new view of the EM algorithm that justifies incremental and other variants, **Biometrika**, submitted.

[72] Nychka, D. : Some properties of adding a smoothing step to the EM algorithm, **Statistics and Probability letters**, 1990, Vol. 9, pp187–193.

[73] Ollinger, J.M. and Karp, J.S. : An evaluation of three algorithms for reconstructing images from data with missing projections, **IEEE Trans. on Nuclear Science**, 1988, Vol. 35, pp629–634.

[74] Ortega, J.M. and Rheinboldt, W.C. : *Iterative solution of nonlinear equations in several variables*,1970, New York: Academic Press.

[75] Qian,W. and Titterington, D.M.: On the use of Gibbs Markov Chain models in the analysis of images based on second order pairwise interactive distributions, **J. App. Stats.**, 1989, Vol.16, pp267–281.

[76] Qian,W. and Titterington, D.M.: Parameter Estimation for Hidden Gibbs Chains. **Statistics and Probability Letters**, 1990, Vol.10, pp49–58.

[77] Ripley, B.D.: The use of spatial models as image priors, in *Spatial Statistics and Imaging*, ed. A. Possolo, pub. Inst. of Math. Stats.

[78] Ripley, B.D.: *Stochastic Simulation*, 1987, John Wiley and Sons Ltd.

[79] Ripley, B.D. and Sutherland, A.I.: Finding spiral structures in images of galaxies, **Phil. Trans. R. Soc. Lond. ,Series A**, 1990, Vol 332, pp477–485.

[80] Shepp, L.A. and Vardi, Y.: Maximum Likelihood Reconstruction for Emission Tomography, **IEEE Trans. Med. Imag.**, 1982, Vol.MI–1, No.2, pp113–122.

[81] Silverman, B.W. : Some aspects of the spline smoothing approach to non–parametric regression curve fitting (with discussion), 1985, **J. Royal Statist. Soc., Series B**, Vol. 47, pp1–52.

[82] Silverman, B.W. , Nychka, D.W., Jones, M.C. and Wilson, J.D.: A smoothed EM approach to indirect estimation problems, with particular reference to stereology and emission tomography, **J. Royal Statist. Soc., Series B**, 1990, Vol.52, No.2, pp271–324.

[83] Skilling, J.: Classic Maximum Entropy, in *Maximum Entropy and Bayesian Methods*, Cambridge 1988, pp 45–52, ed.: J. Skilling

[84] Skilling, J. and Gull, S. : Bayesian Maximum Entropy Image Reconstruction, in *Spatial Statistics and Imaging*, ed. A. Possolo, pub. Inst. of Math. Stats.

[85] Smith,A.F.M.: Bayes estimates in one–way and two–way models, **Biometrika**, 1973, Vol.60, No.2, pp319–329.

[86] Smith, A.F.M. : Bayesian computational methods,**Phil. Trans. Roy. Soc. Lond., Series A.**, 1991, Vol.337, pp369–386.

[87] Smith, A.F.M. and Roberts, G.O. : Bayesian Computation and related Monte Carlo Markov Chain methods, **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.1, pp3–23.

[88] Spiegelhalter, D.J., Dawid, A.P., Hutchinson, T.A. and Cowell,R.G.: Probabilistic Expert Systems and graphical modelling: a case study in drug safety, **Phil. Trans. R. Soc. Lond., Series A**, 1991, vol.337, pp387–405.

[89] Sutherland, A.I. and Titterington, D.M.: Fitting deformable template priors to image sequences, preprint.

[90] Thompson,A.M., Kay,J.W. and Titterington,D.M. : A Cautionary Note about Cross–Validatory Choice, **J.Statist.Comput.Simul.**, Vol.33, pp199–216.

[91] Thompson, A.M., Kay, J.W., Brown, J.C. and Titterington, D.M. : A Study of methods of choosing the smoothing parameter in image restoration by regularization, **IEEE Trans. Patt. Anal. Mach. Intell.**, 1991, Vol.13, No.4, pp326–339.

[92] Titterington, D.M.: Recursive parameter estimation using incomplete data, **J. Royal Statist. Soc., Series B**, 1984, Vol.46, No.2, pp256–267.

[93] Titterington, D.M. : Choosing the regularization parameter in image restoration, in *Spatial Statistics and Imaging*, 1991, ed. A. Possolo, pub. Inst. of Math. Stats.

[94] Titterington, D.M.: On the iterative image space reconstruction algorithm for ECT, **IEEE Trans. Im. Proc.**, 1987,Vol. MI–6, No.1,pp52–56.

[95] Titterington, D.M., Smith, A.F.M. and Makov, U.E. : *Statistical Analysis of Finite Mixture Distributions*, 1985, Chichester, Wiley.

[96] Titterington, D.M. and Rossi, C. : Another look at a Bayesian direct deconvolution method, **Signal Processing**, 1985, Vol. 9, pp. 101–106.

[97] Urbach,P.: Regression analysis, classical and Bayesian, **Brit. J. Phil. Sci.**, 1992, Vol.43, pp311-342.

[98] Vardi, Y. and Lee, D. : From image deblurring to optimal investments: maximum likelihood solutions for positive linear inverse problems, **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.3.

[99] Vardi, Y., Shepp, L.A. and Kaufman, L.: A statistical model for Positron Emission Tomography, **J. Am. Statist. Soc.**, 1985, Vol.80, No.389, pp8–33.

[100] Various : Discussion on the meeting on the Gibbs Sampler and other Markov Chain Monte Carlo methods, **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.1, pp53–102.

[101] Vovk,V.G. and V'Yugin, V.V.: On the empirical validity of the Bayesian method, **J. Royal Statist. Soc., Series B**, 1993, Vol.55, No.1, pp235–236.

[102] Wahba, G.: Bayesian "Confidence Intervals" for the Cross–validated Smoothing Spline, **J. Royal Statist. Soc., Series B** 1983, Vol.45, pp133–150.

[103] Whittaker, J. : *Graphical models in applied multivariate statistics*, 1990, John Wiley and Sons: New York.

[104] Woodward, W.A., Parr, W.C., Schucany, W.R., Lindsey, H.: A comparison of minimum distance and maximum likelihood estimation of a mixture population, **J. Am. Statist. Assoc**, 1984, Vol. 79, pp590–598.

[105] Zhang, J. : The Mean Field Theory in EM procedures for Blind Markov Random Field Image Restoration, **IEEE Trans. Im. Proc.**, 1993, Vol.2, No.1, pp 27–40.