



HAL
open science

Unsupervised joint deconvolution and segmentation method for textured images: a Bayesian approach and an advanced sampling algorithm

Cornelia Vacar, Jean-François Giovannelli

► **To cite this version:**

Cornelia Vacar, Jean-François Giovannelli. Unsupervised joint deconvolution and segmentation method for textured images: a Bayesian approach and an advanced sampling algorithm. EURASIP Journal on Advances in Signal Processing, 2019, 10.1186/s13634-018-0597-x . hal-02472301

HAL Id: hal-02472301

<https://hal.science/hal-02472301>

Submitted on 10 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RESEARCH

Open Access



Unsupervised joint deconvolution and segmentation method for textured images: a Bayesian approach and an advanced sampling algorithm

Cornelia Vacar and Jean-François Giovannelli*

Abstract

The paper tackles the problem of joint deconvolution and segmentation of textured images. The images are composed of regions containing a patch of texture that belongs to a set of K possible classes. Each class is described by a Gaussian random field with parametric power spectral density whose parameters are unknown. The class labels are modelled by a Potts field driven by a granularity coefficient that is also unknown. The method relies on a hierarchical model and a Bayesian strategy to jointly estimate the labels, the K textured images in addition to hyperparameters: the signal and the noise levels as well as the texture parameters and the granularity coefficient. The capability to estimate the latter is an important feature of the paper. The estimates are designed in an optimal manner as a risk minimizer that yields the marginal posterior maximizer for the labels and the posterior mean for the rest of the unknowns. They are computed based on a convergent procedure from samples of the posterior obtained through an advanced MCMC algorithm: Perturbation-Optimization step and Fisher Metropolis-Hastings step within a Gibbs loop. Various numerical evaluations provide encouraging results despite the strong difficulty of the problem.

Keywords: Segmentation, Deconvolution, Texture, Bayes, Unsupervised learning, Potts, Sampling, Optimization

1 Introduction: motivation and state of the art

This paper addresses the complex problem of textured image segmentation that is a subject of importance in various applications [1, 2] (see also [3, 4]). In practice, observations are often affected by blur (due to finite resolution of observation systems) and by noise (due to various sources of error). However, existing approaches do not account for these issues and focus only on segmentation. On the contrary, this paper addresses the problem of textured image segmentation from indirect (blurred and noisy) observations. It tackles the problem of joint deconvolution-segmentation of textured images, and to the best of our knowledge, no other paper has done any work in this area.

Image segmentation is a computer vision/image processing problem [5] consisting in partitioning an image

into groups of adjacent pixels that have a certain homogeneity property (grey level, colour or texture) or that compose an object of interest. Since the problem has been of great interest for decades, the literature is extensive.

The most straightforward *segmentation method* is thresholding; however, it is seldom applicable, since it is only adapted for piecewise constant images, not affected by blur or noise. In the class of region growing methods, Zhang et al. [6] present a seeded image segmentation based on a heat diffusion process, Ugarriza et al. [7] describe an unsupervised region growing and multiresolution merging algorithm and Alpert et al. [8] present a bottom-up aggregation approach. Partial differential equation-based techniques have also been employed. For instance, Chan and Mulet [9] introduce an active contour without edges method for object segmentation, based on level sets, curve evolution, and the Mumford-Shah model. As for the watershed approach, Malik et al. [10] present a normalized cut approach relying on a local measure of similarity of the textural features in a neighbourhood

*Correspondence: Giova@IMS-Bordeaux.fr
IMS (Univ. Bordeaux, CNRS, BINP), Cours de la Libération, 33400 Talence, France

of the pixel, while Grady [11] uses a small number of predefined labels and computes a probability for each unlabelled pixel. The final label is the one maximizing this probability. Sinop and Grady [12] unify the graph cuts and the random walker methods in a common framework, based on L_q norms minimization for seeded image segmentation.

One of the first approaches for *textured image segmentation* [13] is based on using as texture features the moments of the image, computed on small windows. The more recent method in [14] consists in computing features based on the Discrete Wavelet Transform of blocks of the image, evaluating the difference between these features on adjacent blocks and processing to obtain a one-pixel thick contours. This method does not provide a label field; thus, it gives no information about which texture belongs to which region. Another method providing texture edges [15] uses active contours and the patch-based approach for texture analysis. Textured image segmentation is also achieved in [16], based on features extracted from the Fourier transform of the learning textures. A significant method for image segmentation based on both grey level (intervening contour framework) and texture (textons) is presented in [10]. The segmented image is obtained using a normalized cut approach. Mobahi et al. [17] model a homogeneous textured region of an image by a Gaussian density and the region boundaries by adaptive chain codes. The segmentation is obtained using a clustering process. Another approach devoted to strongly resembling textures is given in [18]. The goal is to accurately characterize the textures, and this is achieved by combining a collection of statistics and filter responses. This local information is then used in an aggregation process to determine the segmentation. A three-stage segmentation method is presented in [19] and relies on characterizing both textured and non-textured regions using local spectral histograms. Texel-based image segmentation is achieved in [20] by identifying the modes in the probability density function of region properties.

A very significant class of segmentation methods relies on a *probabilistic* model-based formulation. Geman et al. [21] present an approach for image partitioning into homogeneous regions and for locating edges based on disparity measures. In [22], an image segmentation method is developed based on Monte Carlo Markov Chain (MCMC) and the K adventurers algorithm by integrating clustering and edge detection in the proposal probabilities. Deng and Clausi [23] introduced a weighted Markov random field model that estimates the model parameters and thus performs unsupervised image segmentation. Among the probabilistic methods, the graph partitioning approach is very popular. Felzenszwalb and Huttenlocher [24] uses a graph-based image model and measures the evidence for a boundary between two regions, while Boykov and

Funka-Lea [25] describe the basic framework for efficient object extraction from multi-dimensional image data using graph cuts. One of the most commonly used models for the labels in the probabilistic approaches is the Potts model to favour homogeneous regions. The pixels that belong to different regions are considered independent of each other (given the labels). Within a region, the pixels are either independent or in a Markovian dependency, most often Gaussian or conditionally Gaussian. This type of image model is mostly used for piecewise constant or piecewise smooth images. It is explored by [26] (see also [27]) for image segmentation by introducing a site-dependent external field. Barbu and Zhu [28] present a method based on a generalized Swendsen-Wang form. It is based on an adjacency graph and computes probabilities for each edge and performs graph clustering and graph flipping (instead of single pixel flipping as in the case of the Gibbs sampler). Pereyra et al. [29] propose a method for jointly estimating the Potts parameter using a likelihood free Metropolis-Hastings algorithm.

However, none of the aforementioned segmentation approaches is formulated in the context of indirect observations. Interesting works [29–37] are the Bayesian methods for image segmentation from indirect data (inversion-segmentation) also based on Potts model for the labels. These developments have been an important source of inspiration but they are devoted to piecewise constant or piecewise smooth images and not adapted for textures. On the contrary, the present work tackles the question of textured image segmentation, from indirect (blurred and noisy) data. It proposes a solution for joint deconvolution-segmentation of textured images, and to the best of our knowledge, it is a first attempt to solve the problem. In addition, the approach also includes the estimation of the hyperparameters: the signal and the noise levels as well as the texture parameters and the Potts coefficient. The capability to estimate the latter is an important feature of the paper. The solution is designed by means of a Bayesian strategy, in an optimal scheme. It yields the decision/estimation as the posterior maximizer or mean depending on the type of variable. They are computed based on a convergent procedure from samples of the posterior obtained through an MCMC algorithm. These two properties, optimality and convergence, are also crucial features of the proposed method.

2 Method: probabilistic modelling

In this work, y represents the blurred and noisy observation of the original image z and ℓ represents the hidden label field. y , z and ℓ are column vectors of size P (the total number of pixels). The unobserved image z is composed of a small number of regions, each of these regions consisting in a patch of texture. The texture patches belong to one of K given texture classes.

Remark 1 *There is no constraint specifying that all the texture classes must be represented in the image. Consequently, K only represents an upper bound of the number of classes that will be present in the estimation.*

2.1 Label model

The label set $\ell = [\ell_p, p = 1, \dots, P]$ naturally takes its values in $\{1, \dots, K\}^P$ and is considered to follow a Potts model [38, 39] in order to favour compact regions. It is driven by the granularity coefficient $\beta \in \mathbb{R}_+$ that tunes the mean size of these regions. For a configuration ℓ , the probability reads:

$$\Pr[\ell|\beta] = \mathcal{Z}(\beta)^{-1} \exp\left[\beta \sum_{r \sim s} \delta(\ell_r, \ell_s)\right], \quad (1)$$

where \mathcal{Z} is the normalization constant (partition function), \sim stands for the neighbour relationship in a 4-connectivity system and δ is the Kronecker function, $\delta(k, k')$ is 1 if $k = k'$ and 0 otherwise.

Remark 2 *Let us note $\sigma(\ell) = \sum_{p \sim q} \delta(\ell_p; \ell_q)$. It is the number of pairs of neighbour pixels with identical label. The total number of pairs of neighbour pixels minus $\sigma(\ell)$ is hence the number of “active contours” and then the length of the contours of the label image. It is also the zero-norm of a “gradient” of the label image.*

An important feature of the proposed method is the capability to estimate the parameter β . To this end, the partition function \mathcal{Z} is a crucial function since it is involved in the likelihood of β attached to any configuration. Its analytical expression is unknown¹ and it is a huge summation over the K^P possible configurations. However, based on stochastic simulations, we have precomputed it for several numbers of pixel P and numbers of class K (see Appendix A and our previous papers [40, 41]). The reader is invited to consult papers such as [29, 42] for alternatives. See also [43–46] for complementary results.

2.2 Texture model

The textured images $\mathbf{x}_k \in \mathbb{C}^P$, for $k = 1, \dots, K$ are modelled as zero-mean stationary Gaussian random fields with covariance \mathbf{R}_k :

$$f(\mathbf{x}_k|\mathbf{R}_k) = (2\pi)^{-P} \det(\mathbf{R}_k)^{-1} \exp\left(-\|\mathbf{x}_k\|_{\mathbf{R}_k^{-1}}^2\right).$$

Remark 3 *We address the case of textured images having grey level with the same mean and similar variance since it is particularly challenging. However, the method is also suited for textured images having different mean and variance grey levels.*

For notational convenience, \mathbf{R}_k is defined through a scale parameter γ_k and a structure matrix $\mathbf{\Lambda}_k$, that is to say $\mathbf{R}_k^{-1} = \gamma_k \mathbf{\Lambda}_k$. Since \mathbf{x}_k is a stationary field, \mathbf{R}_k is a Toeplitz-block-Toeplitz (TbT) matrix and by Whittle approximation, it becomes Circulant-block-Circulant (CbC), meaning that the previous expression becomes separable in the Fourier domain:

$$f(\mathbf{x}_k|\mathbf{R}_k) = \prod_{p=1}^P (2\pi)^{-1} \gamma_k \lambda_{k,p} \exp\left[-\gamma_k \lambda_{k,p} |\hat{x}_{k,p}|^2\right] \quad (2)$$

where the $\hat{x}_{k,p}$ for $p = 1, \dots, P$ are the Fourier coefficients of the image \mathbf{x}_k and the $\lambda_{k,p}$ for $p = 1, \dots, P$ are the eigenvalues of $\mathbf{\Lambda}_k$. Thus, as a physical interpretation, $\lambda_{k,p}^{-1}$ describes the Power Spectral Density (PSD) of \mathbf{x}_k in discrete form. More specifically, $\gamma_k \lambda_{k,p}$ is the inverse variance of $\hat{x}_{k,p}$.

We have chosen a parametric model for the PSD, of Lorentzian form:

$$\lambda^{-1}(v_x, v_y, \theta) = \pi^2 u_x u_y [1 + S_x^2] [1 + S_y^2] \quad (3)$$

with

$$S_x = (v_x - v_x^0) / u_x \text{ and } S_y = (v_y - v_y^0) / u_y$$

where v_x / v_y are the horizontal/vertical frequency and $\theta = [v_x^0, v_y^0, u_x, u_y]$ is the shape parameter. The parameters v_x^0, v_y^0 are the central frequencies and u_x, u_y are the PSD widths. Nevertheless, any other parametric form can be used for the PSD, e.g., Gaussian and Laplacian.

Remark 4 *The variables $v_x, v_y \in [-0.5, 0.5]^2$ are the continuous reduced frequencies, while (v_m, v_n) are the discretized reduced frequencies. We associate the frequency pair (v_m, v_n) to index p . Then, $\lambda_p(\theta) = \lambda(v_m, v_n, \theta)$.*

2.3 Image model

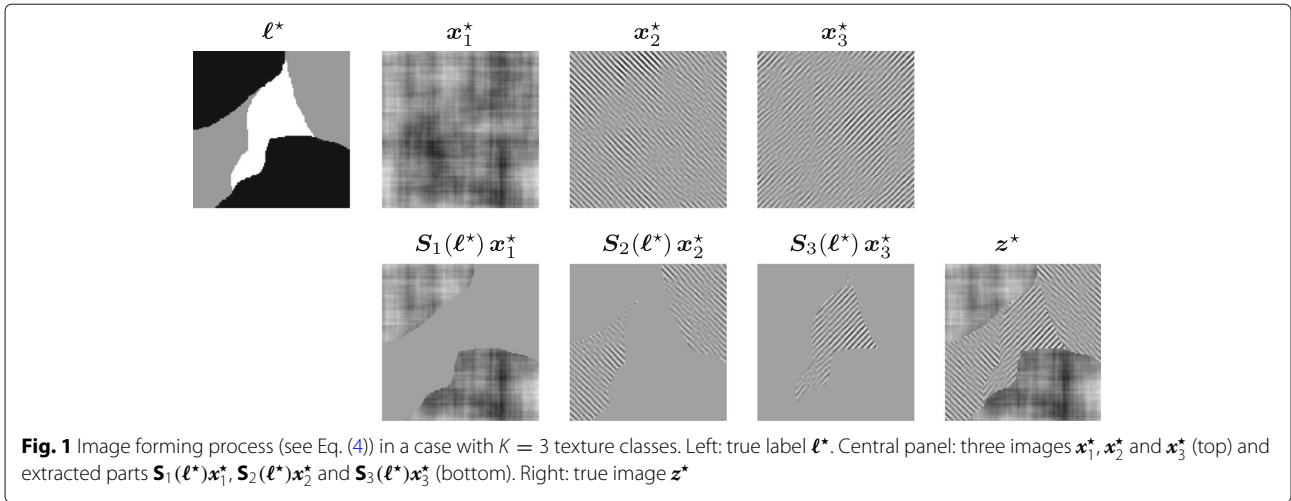
The process of obtaining the image \mathbf{z} containing the textured patches, starting from the full textured images $\mathbf{x}_k, k = 1, \dots, K$ and the labels ℓ , can be visualized by the schematic in Fig. 1. This image forming process is mathematically formalized as:

$$\mathbf{z} = \sum_{k=1}^K \mathbf{S}_k(\ell) \mathbf{x}_k \quad (4)$$

where $\mathbf{S}_k(\ell)$ are $P \times P$ diagonal binary matrices obtained based on the labels ℓ . These matrices extract from the textured image \mathbf{x}_k the pixels with label k and replace the other pixels with 0. They are zero-forcing matrices defined by:

$$\mathbf{S}_k(\ell) = \text{diag}\{\delta(\ell_p, k), p = 1, \dots, P\}$$

with entries 1 at pixel p when $\ell_p = k$ and 0 elsewhere.



Remark 5 Let us consider $\mathcal{I}_k = \{p \mid \ell_p = k\}$ the set of sites having label k . Then, these sets for $k = 1, \dots, K$ encode a repartition of the set of pixel indices, thus having the properties:

- Are disjoint, i.e., $\mathcal{I}_k \cap \mathcal{I}_l = \emptyset$, for $l \neq k$
- Cover the entire lattice, i.e., $\cup_k \mathcal{I}_k = \{1, \dots, P\}$
- May be empty

In terms of the extraction matrices, these properties are summarized by $\sum_{k=1}^K S_k = \mathbf{I}_P$, the identity matrix of size P .

2.4 Observation system model

Now we turn to the observation system that is modelled as a linear and invariant transform. It is accounted for through a $P \times P$ convolution matrix with a TbT structure denoted by \mathbf{H} . It becomes CbC by circulant approximation, and its eigenvalues are defined by the Fourier transfer function \hat{h}_p . Any function could be introduced (Gaussian, Lorentzian, Airy, ...), and the considered one is a Laplacian:

$$\hat{h}_{nm} = \exp[-w^{-1} (|v_n| + |v_m|) / 2]$$

centred in the $(0, 0)$ frequency with width w . This is only one of the countless models that can be used.

2.5 Noise model

The noise is considered to be additive, zero-mean, white, and Gaussian of inverse variance γ_n . Based on this model, the density of the data given the image z and the noise parameter γ_n , reads:

$$f(\mathbf{y}|\mathbf{z}, \gamma_n) = (2\pi)^{-P} \gamma_n^P \exp[-\gamma_n \|\mathbf{y} - \mathbf{H}\mathbf{z}\|^2] \tag{5}$$

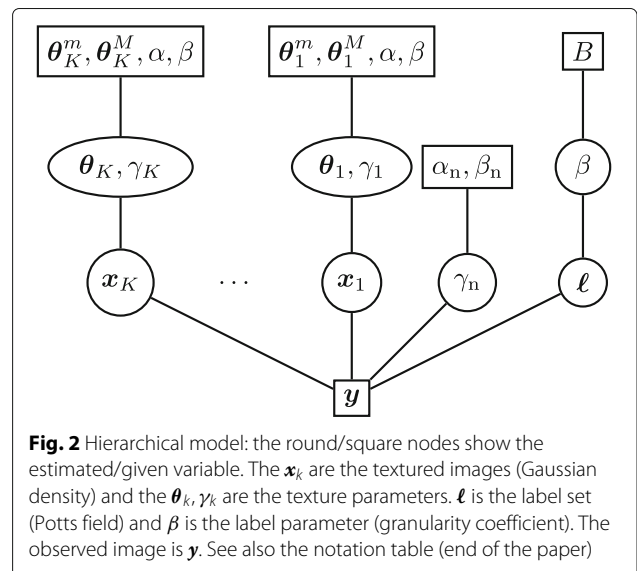
that is to say the likelihood.

2.6 Hierarchical model

Based on the variables above, the hierarchy for the model in preparation for the segmentation problem from blurred and noisy data can be established and it is graphically represented in Fig. 2. Based on the variable dependencies encoded in this figure, the joint distribution can be expressed:

$$f(\mathbf{y}, \ell, \mathbf{x}_{1..K}, \gamma_n, \gamma_{1..K}, \theta_{1..K}) = f(\mathbf{y}|\gamma_n, \ell, \mathbf{x}_{1..K}) \Pr[\ell|\beta] \prod_{k=1}^K f(\mathbf{x}_k|\theta_k, \gamma_k) \tag{6}$$

$$f(\beta) f(\gamma_n) \prod_{k=1}^K f(\theta_k) \prod_{k=1}^K f(\gamma_k).$$



In order to complete the probabilistic description, the next section introduces the hyperparameter densities.

2.7 Hyperparameter models

Regarding the precision parameters $\gamma_k, k = 1, \dots, K$ and γ_n , it can be noticed that in the model for the textured images \mathbf{x}_k (Eq. (2)) and for the observation \mathbf{y} (Eq. (5)), they intervene as precision parameters in Gaussian conditional densities; hence, the Gamma densities $\mathcal{G}(\gamma; \alpha^0, \beta^0)$ are conjugate forms. Furthermore, little prior information is available on these parameters, so uninformative Jeffreys prior are used, by setting $(\alpha^0, \beta^0) \rightarrow (0, 0)$.

Otherwise, the dependency of the likelihood w.r.t. the parameter θ_k is very complicated, meaning that there is no conjugate form. Moreover, the lack of prior information suggests the use of the uniform density between a minimum and a maximum value: $f(\theta_k) = \mathcal{U}_{[\theta_k^m, \theta_k^M]}(\theta_k)$.

When it comes to β , a conjugate prior is not available, given the expression of the partition function $\mathcal{Z}(\beta)$. A uniform prior on an interval $[0, B]$ is considered as a reasonable choice: $f(\beta) = \mathcal{U}_{[0, B]}(\beta)$ where B is defined as the maximum possible value of β .

3 Method: Bayesian formulation

3.1 Estimation

The Bayesian strategy designs an estimator based on a loss function that quantifies the discrepancy between the true value of a parameter and an estimated one. It then relies on a risk that is the mean value of the loss function, the mean being considered under the joint distribution (6) that is to say the distribution of the unknown parameters and the data. The optimal estimator is defined as the function of the data that minimizes the risk. It is naturally different for the various types of parameters and choices of loss function.

- Regarding the labels ℓ (discrete parameters), we resort to a binary loss function and the estimates are the *marginal posterior maximizers*.
- Regarding the continuous parameters β, γ_n , the γ_k , the θ_k and the \mathbf{x}_k , we resort to the quadratic loss function and the estimates are the *posterior means*.

Remark 6 *A specificity of the chosen loss functions is separability, resulting in marginal estimates. It allows for relatively fast computations but with possible limitations regarding image quality. Alternatives could rely on non-separable loss function and non-marginal estimates, for instance (joint) posterior maximizer. Numerical implementation could then rely on non-guaranteed optimization algorithm (e.g. block iterative conditional mode) or on computationally intensive algorithm (e.g. simulated annealing).*

An estimate $\hat{\mathbf{z}}$ of the image \mathbf{z} can be obtained based on the estimated labels $\hat{\ell}$ and textured images $\hat{\mathbf{x}}_k$ based on Eq. (4) as follows:

$$\hat{\mathbf{z}} = \sum_k \mathbf{S}_k(\hat{\ell}) \hat{\mathbf{x}}_k \tag{7}$$

each extraction matrix being based on the label estimate $\hat{\ell}$.

3.2 Posterior

The posterior is proportional to the joint distribution (6) and is fully specified based on the formation model (4) for the image \mathbf{z} , the model (2) for the textured images \mathbf{x}_k , the Potts model (1) for the labels ℓ , the model (5) for the observation \mathbf{y} and the priors above for β , for γ_n , for the γ_k and θ_k .

$$\begin{aligned} f(\ell, \mathbf{x}_{1\dots K}, \gamma_n, \gamma_{1\dots K}, \theta_{1\dots K}, \beta | \mathbf{y}) \propto & \\ & \exp \left[-\gamma_n \|\mathbf{y} - \mathbf{H} \sum_k \mathbf{S}_k(\ell) \mathbf{x}_k\|^2 \right] \\ & \mathcal{Z}(\beta)^{-1} \exp \left[\beta \sum_{r \sim s} \delta(\ell_r, \ell_s) \right] \\ & \prod_k \left[\det(\mathbf{\Lambda}_k(\theta_k))^{-1} \exp \left(-\gamma_k \|\mathbf{x}_k\|_{\mathbf{\Lambda}_k(\theta_k)}^2 \right) \right] \\ & \prod_k \left[\gamma_k^{\alpha_k + P - 1} \exp(-\gamma_k \beta_k) \right] \\ & \gamma_n^{\alpha_n + P - 1} \exp(-\gamma_n \beta_n) \mathcal{U}_{[\theta_k^m, \theta_k^M]}(\theta_k) \mathcal{U}_{[0, B]}(\beta). \end{aligned} \tag{8}$$

This distribution summarizes all the information about the unknowns contained by the data and the prior models.

3.3 Computing—posterior conditionals

Due to the sophisticated form of the posterior, the estimates (marginal posterior maximizers or means) cannot be calculated; consequently, they will be numerically extracted. Stochastic samplers seem adequate and the literature on the subject is abundant and varied [47–50]. More specifically, a (block) Gibbs loop is particularly appealing since it enables to split the global sophisticated problem in several far simpler sub-problems. It requires to sequentially sample each variable, under its conditional posterior. These distributions are described in the next section.

4 Algorithm: sampling aspects

This section describes the conditional posterior for each unknown parameter in order to implement a Gibbs sampler. In particular, it details the cumbersome task of sampling the full textured images (Section 4.4) and the labels (Section 4.5). These two sampling processes represent the major algorithmic challenges of our approach.

Each conditional posterior can be deduced from the joint posterior (8) by picking the factors that are function of the considered parameter.

4.1 Precision parameters

Regarding the noise parameter γ_n and the texture scale parameters γ_k , from (8), we have:

$$\begin{aligned} \gamma_n &\sim \gamma_n^{\alpha_n^0 + P - 1} \exp -\gamma_n [\beta_n^0 + \|\mathbf{y} - \mathbf{H}\mathbf{z}\|^2], \\ \gamma_k &\sim \gamma_k^{\alpha_k^0 + P - 1} \exp -\gamma_k [\beta_k^0 + \|\mathbf{x}_k\|_{\Lambda_k(\theta_k)}^2]. \end{aligned}$$

They must be sampled under Gamma densities $\mathcal{G}(\gamma; \alpha, \beta)$ with respective parameters:

$$\alpha = \alpha_n^0 + P \text{ and } \beta = \beta_n^0 + \|\mathbf{y} - \mathbf{H}\mathbf{z}\|^2$$

for the noise parameter γ_n and

$$\alpha = \alpha_k^0 + P \text{ and } \beta = \beta_k^0 + \|\mathbf{x}_k\|_{\Lambda_k(\theta_k)}^2$$

for the texture parameters γ_k . As Gamma variables, they can be straightforwardly sampled. In addition, given the hierarchical structure (see Fig. 2), they are mutually (a posteriori) independent.

4.2 Shape texture parameters

Regarding the shape parameters θ_k of the textured image PSD, the problem is made far more complicated by the intricate relation between the density, the PSD and the parameter θ_k ; see Eqs. (2) and (3). As a consequence, the conditional posterior has a non-standard form:

$$\theta_k \sim \mathcal{U}_{[\theta_k^m, \theta_k^M]}(\theta_k) \prod_p \lambda_p(\theta_k) \exp -\gamma_k \lambda_p(\theta_k) |\hat{x}_{k,p}|^2$$

nevertheless, it can be sampled using a Metropolis-Hastings (MH) step². Basically, it consists in drawing a proposal based on a proposition law, evaluating an acceptance probability, and then, at random according to this probability, setting the new value as the proposal (acceptation) or as the current value (duplication). There are numerous options in order to formulate a proposition law, and both the convergence rate and the mixing properties are influenced by its adequacy to the (conditional) posterior. Thus, designing a proposition law that embeds information about the posterior will significantly enhance the performances. In this context, the directional Random Walk MH (RWMH) algorithm taking advantage of first- or second-order derivatives of the posterior seems relevant. A standard case is the Metropolis-adjusted Langevin algorithm (MALA) [51], which takes advantage of the posterior derivative. The preconditioned MALA [52] and the quasi-Newton proposals [53] exploit the posterior curvature. More advanced versions rely on the Fisher matrix (instead of the Hessian) and leads to an efficient sampler

called the Fisher-RWMH: [54] proposes a general statement and our previous paper [55] (see also [56]) focuses on texture parameters.

Explicitly, from the current value θ_c , the algorithm formulates the proposal θ_p :

$$\theta_p = \theta_c + \frac{1}{2} \varepsilon^2 \mathcal{I}^{-1}(\theta_c) \mathcal{L}'(\theta_c) + \varepsilon \mathcal{I}(\theta_c)^{-1/2} \mathbf{u}$$

where $\mathcal{I}(\theta)$ is the Fisher matrix, $\mathcal{L}(\theta)$ is the log of the conditional posterior and $\mathcal{L}'(\theta)$ its gradient, ε is a tuning parameter and $\mathbf{u} \sim \mathcal{N}(0, \mathbf{I})$ a standard Gaussian sample.

4.3 Potts parameter

The granularity coefficient β conditionally follows an intricate density also deduced from (8):

$$\beta \sim \mathcal{Z}(\beta)^{-1} \exp \left[\beta \sum_{p \sim q} \delta(\ell_p; \ell_q) \right] \mathcal{U}_{[0, B]}(\beta).$$

The sampling is a very difficult task first of all because the density does not have a standard form. Moreover, the major problem is that $\mathcal{Z}(\beta)$ is intractable, so the density cannot even be evaluated for a given value of β .

To overcome the obstacle, the partition function $\mathcal{Z}(\beta)$ has been precomputed on a fine grid of values for β , ranging from $\beta = 0$ to $\beta = B = 3$, with a step of 0.01, for several numbers of pixel P and numbers of class K . Details are given in Annex 6. It is therefore easy to compute the cumulative density function $F(\beta)$ by standard numerical integration / interpolation. Then, it suffices to sample a uniform variable u on $[0, 1]$ and to compute $\beta = F^{-1}(u)$ to obtain a desired sample. So, this step is inexpensive (since the table of values of $\mathcal{Z}(\beta)$ is precomputed).

Remark 7 *Although it allows for very efficient computations, this approach has a limitation: \mathcal{Z} must be precomputed for the considered number of pixel P and class K .*

The procedure is identical to the one presented in our previous papers [40, 41, 57]. The reader is invited to consult [29, 42–46] for alternatives and complementary results.

4.4 Textured image

Remark 8 *To improve the readability, in the following, we will use the simplified notation $\Lambda_k = \Lambda_k(\theta_k)$.*

The textured image \mathbf{x}_k has a Gaussian density, deduced from (8):

$$\mathbf{x}_k \sim \exp - \left[\gamma_n \|\mathbf{y} - \mathbf{H} \sum_l \mathbf{S}_l \mathbf{x}_l\|^2 + \gamma_k \|\mathbf{x}_k\|_{\Lambda_k}^2 \right] \quad (9)$$

and it is easy to show that the mean $\boldsymbol{\mu}_k$ and the covariance $\boldsymbol{\Sigma}_k$ write:

$$\begin{aligned} \Sigma_k^{-1} &= \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \Lambda_k \\ \boldsymbol{\mu}_k &= \gamma_n \Sigma_k \mathbf{S}_k^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}}_k \end{aligned}$$

where $\bar{\mathbf{y}}_k = \mathbf{y} - \mathbf{H} \sum_{l \neq k} \mathbf{S}_l \mathbf{x}_l$. This quantity is founded on the extraction of the contribution of the image \mathbf{x}_k from the data. More specifically, $\bar{\mathbf{y}}_k$ relies on the subtraction from the observations \mathbf{y} of the convolution of all the parts of the image \mathbf{z} that are not labelled k .

However, the practical sampling of this Gaussian density is a thorny issue due to the high dimension of the variable. Usually, sampling a Gaussian density requires handling the covariance or the precision, for instance factorization (e.g., Cholesky), diagonalisation, or inversion, which are impossible here. This could be possible for special structures, e.g., sparse or circulant. Here, Λ_k , \mathbf{H} and by extension $\mathbf{H}^\dagger \mathbf{H}$ are CbC; however, the presence of the \mathbf{S}_k breaks the circularity: Σ_k is not diagonalizable by FFT and, consequently, the sampling of \mathbf{x}_k cannot be performed efficiently in the Fourier domain.

Nevertheless, the literature accounts for alternatives based on the strong links between matrix factorization, diagonalization, inversion, linear system and optimization of quadratic criteria [58–62]. We resort here to our previous work [61] (see also [63]) based on a perturbation-optimization (PO) principle: adequate stochastic perturbation of a quadratic criterion and optimization of the perturbed criterion. It is shown that the criterion optimizer is a sample of the target density. It is applicable if the precision matrix and the mean can be written as a sum of the form:

$$\Sigma_k^{-1} = \sum_{n=1}^N \mathbf{M}_n^t \mathbf{C}_n^{-1} \mathbf{M}_n \quad \text{and} \quad \boldsymbol{\mu}_k = \Sigma_k \sum_{n=1}^N \mathbf{M}_n^t \mathbf{C}_n^{-1} \mathbf{m}_n$$

By identification, with $N = 2$:

$$\begin{cases} \mathbf{M}_1 &= \mathbf{H} \mathbf{S}_k \\ \mathbf{C}_1 &= \gamma_n^{-1} \mathbf{I}_p \\ \mathbf{m}_1 &= \bar{\mathbf{y}}_k \end{cases} \quad \begin{cases} \mathbf{M}_2 &= \mathbf{I}_p \\ \mathbf{C}_2 &= \gamma_k^{-1} \Lambda_k^{-1} \\ \mathbf{m}_2 &= \mathbf{O}_p \end{cases}$$

4.4.1 Perturbation

The perturbation phase of this algorithm consists in drawing the following Gaussian samples:

$$\boldsymbol{\xi}_1 \sim \mathcal{N}(\mathbf{m}_1, \mathbf{C}_1) \quad \text{and} \quad \boldsymbol{\xi}_2 \sim \mathcal{N}(\mathbf{m}_2, \mathbf{C}_2)$$

The cost of these sampling is not prohibitive: $\boldsymbol{\xi}_1$ is a realization of a white noise and $\boldsymbol{\xi}_2$ is a realization of the prior model for \mathbf{x}_k and it is computed by FFT.

4.4.2 Optimization

In order to obtain a sample of the image \mathbf{x}_k , the following criterion must be optimized w.r.t. \mathbf{x} :

$$J_k(\mathbf{x}) = \gamma_n \|\boldsymbol{\xi}_1 - \mathbf{H} \mathbf{S}_k \mathbf{x}\|^2 + \gamma_k \|\boldsymbol{\xi}_2 - \mathbf{x}\|_{\Lambda_k}^2.$$

For notational convenience, let us rewrite:

$$J_k(\mathbf{x}) = \mathbf{x}^\dagger \mathbf{Q}_k \mathbf{x} - 2\mathbf{x}^\dagger \mathbf{q}_k + J_k(0)$$

where the matrix $\mathbf{Q}_k = \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{S}_k + \gamma_k \Lambda_k$ is half the Hessian (and the precision matrix) and the vector $\mathbf{q}_k = \gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1 + \gamma_k \Lambda_k^{-1} \boldsymbol{\xi}_2$ is the opposite of the gradient at the origin. The gradient at \mathbf{x} itself is: $\mathbf{g}_k = 2(\mathbf{Q}_k \mathbf{x} - \mathbf{q}_k)$.

Theoretically, there is no constraint on the optimization technique to be used and the literature on the subject is abundant [64–66]. We have only considered algorithms that are guaranteed to converge (to the unique minimizer) and among them the basic directions:

- Gradient descent,
- Conjugate gradient descent.

We have first used the conjugate gradient direction, since it is more efficient especially for a high-dimension problem and a quadratic criterion. However, we have experienced convergence difficulties, making the overall algorithm very slow. In practice, the step length at each iteration was extremely small, probably due to conditioning issues. Consequently, the differences between the iterates were almost insignificant. The solution relies on a preconditioner. It has been defined as a CbC approximation of the inverse Hessian of J_k :

$$\mathbf{\Pi}_k = \left(\gamma_n \mathbf{H}^\dagger \mathbf{H} + \gamma_k \Lambda_k \right)^{-1} / 2 \tag{10}$$

obtained by eliminating the \mathbf{S}_k matrix from \mathbf{Q}_k and chosen for computational efficiency. It is used for both of the aforementioned directions:

- Preconditioned gradient descent,
- Preconditioned conjugate gradient descent.

In this context, the two methods have yielded similar results, and finally, we have focused on the *preconditioned gradient*.

The second ingredient that is necessary is the step length s in the considered direction, at each iteration. Here again, a variety of strategies is available. We have used an *optimal step* that is explicitly given:

$$s = \frac{\mathbf{g}_k^\dagger \mathbf{\Pi}_k^\dagger \mathbf{g}_k}{\mathbf{g}_k^\dagger \mathbf{\Pi}_k^\dagger \mathbf{Q}_k \mathbf{\Pi}_k \mathbf{g}_k}$$

and efficiently computable.

4.4.3 Practical implementation

The algorithm requires at each iteration the computation of the preconditioned gradient and the step length. Finally, the required computations for performing the optimization are the vector \mathbf{q}_k and the products of a vector by the matrices $\mathbf{\Pi}_k$ and \mathbf{Q}_k .

- The vector \mathbf{q}_k writes:

$$\mathbf{q}_k = \underbrace{\gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \boldsymbol{\xi}_1}_{\text{FFT}} + \underbrace{\gamma_k \mathbf{\Lambda}_k^{-1} \boldsymbol{\xi}_2}_{\text{FFT}} \quad (11)$$

and thus efficiently computed through a FFT and zero-forcing (ZF).

- The product $\mathbf{Q}_k \mathbf{x}$ writes:

$$\mathbf{Q}_k \mathbf{x} = \underbrace{\gamma_n \mathbf{S}_k^\dagger \mathbf{H}^\dagger \mathbf{H}}_{\text{FFT}} \underbrace{\mathbf{S}_k \mathbf{x}}_{\text{ZF}} + \underbrace{\gamma_k \mathbf{\Lambda}_k}_{\text{FFT}}$$

and thus also efficiently computed through a series of FFT and ZF.

- Regarding $\mathbf{\Pi}_k \mathbf{g}_k$, since the matrix $\mathbf{\Pi}_k$ is CbC, the product can also be efficiently computed by FFT.

The zero-forcing process is achieved in the spatial domain (it amounts to setting to zero some pixels of images), while the costly products by matrices are performed in the Fourier domain (all of them by FFT).

4.5 Labels

The label set has a multidimensional categorical distribution:

$$\ell \sim \exp \left[\beta \sum_{r \sim s} \delta(\ell_r, \ell_s) - \gamma_n \|\mathbf{y} - \mathbf{H} \sum_k \mathbf{S}_k(\ell) \mathbf{x}_k\|^2 \right]$$

and it is a non-separable and non-standard form, so its sampling is not an easy task. A solution is to sample the ℓ_p one by one conditionally on the others and on the rest of the variables, in a Gibbs scheme.

To this end, let us introduce the notation \mathbf{z}_k^p for the image with all its pixels identical to \mathbf{z} except for pixel p . The pixel p in \mathbf{z}_k^p is the pixel p from \mathbf{x}_k . Let us note $\mathcal{E}_{p,k} = \|\mathbf{y} - \mathbf{H} \mathbf{z}_k^p\|^2$. This error quantifies the discrepancy between the data and the class k regarding pixel p .

Sampling a label ℓ_{p_0} requires its conditional probability. A precise analysis of the conditional distribution for ℓ_{p_0} yields:

$$\Pr(\ell_{p_0} = k | \star) \propto \exp \left[\beta \sum_{r \sim p_0} \delta(\ell_r, k) - \gamma_n \mathcal{E}_{p_0, k} \right]$$

for $k = 1, \dots, K$. This computation is performed up to a multiplicative constant, which can be determined knowing that the probabilities sum to 1.

To compute these probabilities, we must evaluate the two terms of the argument of the exponential function, at pixel p_0 . The first term is the contribution of the prior and it can be easily computed for each k by counting the neighbours of pixel p_0 having the label k . Let us now focus

on the second term, $\mathcal{E}_{p,k}$. To write this term in a more convenient form, we introduce:

- A vector $\mathbf{1}_p \in \mathbb{R}^P$: its p -th entry is 1 and the other is 0.
- A quantity $\Delta_{p,k} \in \mathbb{R}$ that records the difference between the p -th pixel of the image \mathbf{z} and the one of image \mathbf{x}_k : $\Delta_{p,k} = \mathbf{1}_p^\dagger (\mathbf{z} - \mathbf{x}_k)$.

We then have $\mathbf{z}_k^p = \mathbf{z} - \Delta_{p,k} \mathbf{1}_p$, so $\mathcal{E}_{p,k}$ writes:

$$\begin{aligned} \mathcal{E}_{p,k} &= \|\mathbf{y} - \mathbf{H}(\mathbf{z} - \Delta_{p,k} \mathbf{1}_p)\|^2 \\ &= \|(\mathbf{y} - \mathbf{H}\mathbf{z}) - \Delta_{p,k} \mathbf{H} \mathbf{1}_p\|^2 \\ &= \bar{\mathbf{y}}^\dagger \bar{\mathbf{y}} + \Delta_{p,k}^2 \mathbf{1}_p^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{1}_p - 2\Delta_{p,k} \mathbf{1}_p^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}} \end{aligned} \quad (12)$$

where $\bar{\mathbf{y}} = \mathbf{y} - \mathbf{H}\mathbf{z}$. Then, to complete the description, let us analyse each term.

1. The first term $\bar{\mathbf{y}}^\dagger \bar{\mathbf{y}}$ does not depend on p or k . Consequently, its value is not required in the sampling process and it can be included in a multiplicative factor.
2. The term $\mathbf{1}_p^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{1}_p = \|\mathbf{H} \mathbf{1}_p\|^2$ does not depend on p due to the CbC form of the \mathbf{H} matrix. Moreover, this norm only needs to be computed once for all, since the \mathbf{H} matrix does not change throughout the iterations. In fact, this norm amounts to the sum $\sum_q |\hat{h}_q|^2$.
3. Finally, in the third term $\mathbf{1}_p^\dagger \mathbf{H}^\dagger \bar{\mathbf{y}}$, the product $\mathbf{H}^\dagger \bar{\mathbf{y}}$ is a convolution efficiently computable by FFT and the product with $\mathbf{1}_p^\dagger$ selects the pixel p . Under this form, the computation would not be efficient since $\mathbf{H}^\dagger \bar{\mathbf{y}}$ should be recomputed at each iteration. A far better alternative is to update $\mathbf{H}^\dagger \bar{\mathbf{y}}$ after updating each label.

5 Results and discussion

The problem of texture segmentation has a considerable degree of difficulty, especially in the present case, where (i) the data are affected by blur and noise, (ii) the texture parameters are unknown and (iii) the granularity coefficient, the signal and the noise levels are also unknown. The previous sections provide a detailed description of our method, and this section presents numerical results, as follows.

1. First, implementation and practical considerations are described.
2. A study is then given for different image topologies in various combinations of blur and noise to assess the method versatility and identify the limitations.
3. Moreover, a posterior statistics analysis is given in order to evaluate the associated uncertainty.

5.1 Implementation and practical considerations

The method has been implemented³ as shown in Algorithm 1. Under different scenarios, the algorithm has been run several times from identical and different

Algorithm 1: Deconvolution-Segmentation of Textures

Input : Data \mathbf{y} , texture number K , impulse response

Output: Samples for labels $\ell^{(t)}$, textures $\mathbf{x}_k^{(t)}$, granularity $\beta^{(t)}$, noise parameter $\gamma_n^{(t)}$, texture parameters $(\theta_k^{(t)}, \gamma_k^{(t)})$

Initializations: $t = 0$, $\mathbf{z}^{(0)} = \mathbf{y}$, $\ell^{(0)} = \text{ceil}(K * \text{rand}(P))$;

while not convergence **do**

$t = t + 1$

$\gamma_n^{(t)} \sim f(\gamma_n | \mathbf{y}, \mathbf{x}_{1..K}^{(t-1)}, \ell^{(t-1)})$ [Sect. 4.1]

$\ell^{(t)} \sim \text{Pr}(\ell | \mathbf{y}, \gamma_n^{(t)}, \mathbf{x}_{1..K}^{(t-1)}, \beta^{(t-1)})$ [Sect. 4.5]

$\beta^{(t)} \sim f(\beta | \ell^{(t)})$ [Sect. 4.3]

for $k = 1$ to K **do**

$\gamma_k^{(t)} \sim f(\gamma_k | \mathbf{x}_k^{(t-1)}, \theta_k^{(t-1)})$ [Sect. 4.1]

$\theta_k^{(t)} \sim \text{F-RWMHtarget}f(\theta_k | \mathbf{x}_k^{(t-1)}, \gamma_k^{(t)})$
[Sect. 4.2]

$\mathbf{x}_k^{(t)} \sim f(\mathbf{x}_k | \mathbf{y}, \ell^{(t)}, \gamma_n^{(t)}, \gamma_k^{(t)}, \theta_k^{(t)}, \mathbf{x}_{l \neq k}^{(\cdot)})$
[Sect. 4.4]

end

% Count label occurrences and update segmentation

$\text{LabelOcc}(p, \ell_p) = \text{LabelOcc}(p, \ell_p) + 1, p = 1, \dots, P$;

$\hat{\ell} = \text{MaxOccurrenceLabel}(\text{LabelOcc})$;

% Update parameters;

$\hat{\gamma}_n = \text{UpdateAverage}(\gamma_n^{(\tau)}, \tau = T, \dots, t)$;

$\hat{\beta} = \text{UpdateAverage}(\beta^{(\tau)}, \tau = T, \dots, t)$;

for $k = 1$ to K **do**

$\hat{\mathbf{x}}_k = \text{UpdateAverage}(\mathbf{x}_k^{(\tau)}, \tau = T, \dots, t)$;

$\hat{\gamma}_k = \text{UpdateAverage}(\gamma_k^{(\tau)}, \tau = T, \dots, t)$

$\hat{\theta}_k = \text{UpdateAverage}(\theta_k^{(\tau)}, \tau = T, \dots, t)$;

end

% Reconstructed image;

$\hat{\mathbf{z}} = \text{BuildImage}(\hat{\ell}, \hat{\mathbf{x}}_{1..K})$; [Eq. (7)]

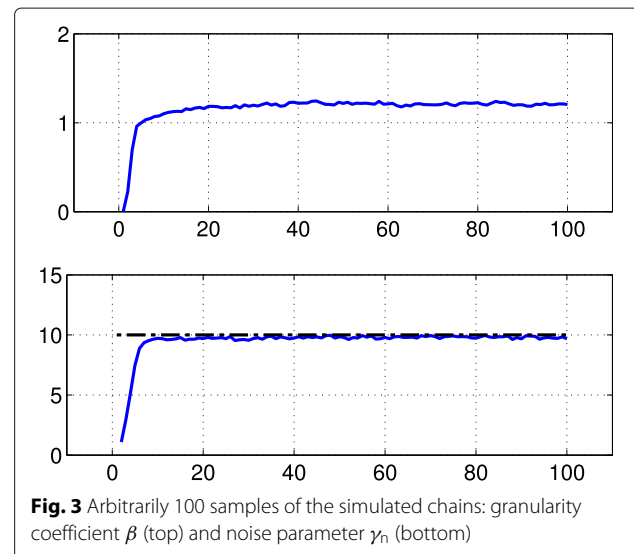
end

Table 1 The horizontal and vertical frequencies ν_x^0 and ν_y^0 are respectively given in the top and the bottom part of the table. For each parameter, it gives the true value, the prior interval and the estimated values. They are clearly very closed to the true ones

	Texture 1	Texture 2	Texture 3
True	0	0.2	0.2
Prior	[- 0.2; + 0.2]	[+ 0.1; + 0.5]	[+ 0.1; + 0.5]
Estimate	0.00011	0.20412	0.19687
True	0	- 0.2	0.2
Prior	[- 0.2; + 0.2]	[- 0.5; - 0.1]	[+ 0.1; + 0.5]
Estimate	0.00034	- 0.20134	0.20223

initializations, and it has shown consistent qualitative and quantitative behaviours. It has lead us to a series of practical considerations.

- The label set is initialized by a realization of a white noise with uniform probability in $\{1, \dots, K\}$. Our tests have shown a faster convergence as compared to other initialization (e.g. constant label field).
- An important practical point is the initialization of the texture parameters θ_k . Each frequency is set to the maximizer of the periodogram of the observed image \mathbf{y} over its prior interval.
- The preconditioned gradient and the preconditioned conjugate gradient directions have similar performances. Contrary, the non preconditioned versions are very slow.
- Stopping rule: the algorithm stops when the difference between successive updates of the image \mathbf{z} (see Eq. (7) and last line of Algorithm 1) becomes smaller than a given threshold s . Practically, we set $s = 10^{-3}$, the algorithm iterates usually about two hundred times and it takes about 4 min for a 256×256 image.



5.2 Evaluation of the method

The first example is given in Fig. 4. It consists in a simple image topology containing $K = 3$ classes of texture. The true values of the frequency parameters of the textured images are given in Table 1. The value of the spectral width is $u_x = u_y = 0.005$ for all the textures (and it is assumed to be known). These values produce two oriented textures and a low-frequency noise shown in Fig. 4 (and already given in Fig. 1). The observation scenario is with $w = 1/2$ (full width at half maximum is about 0.5) and $\gamma_n = 10$ (signal to noise ratio is about 5 dB).

For an illustrative plot, the algorithm has been iterated arbitrarily 100 times and Fig. 3 shows the simulated chains for the granularity coefficient β and for the noise parameter γ_n . It shows that the distributions are stable after about $T = 50$ iterations (burn-in period). The first T samples are then discarded. From the remaining samples, the decisions for the labels are computed as the empirical marginal posterior maximizers and the estimations for the other parameters are computed as empirical posterior averages.

The algorithm produces a label configuration (Fig. 4d) very similar to the true one (Fig. 4a), with only 0.90%

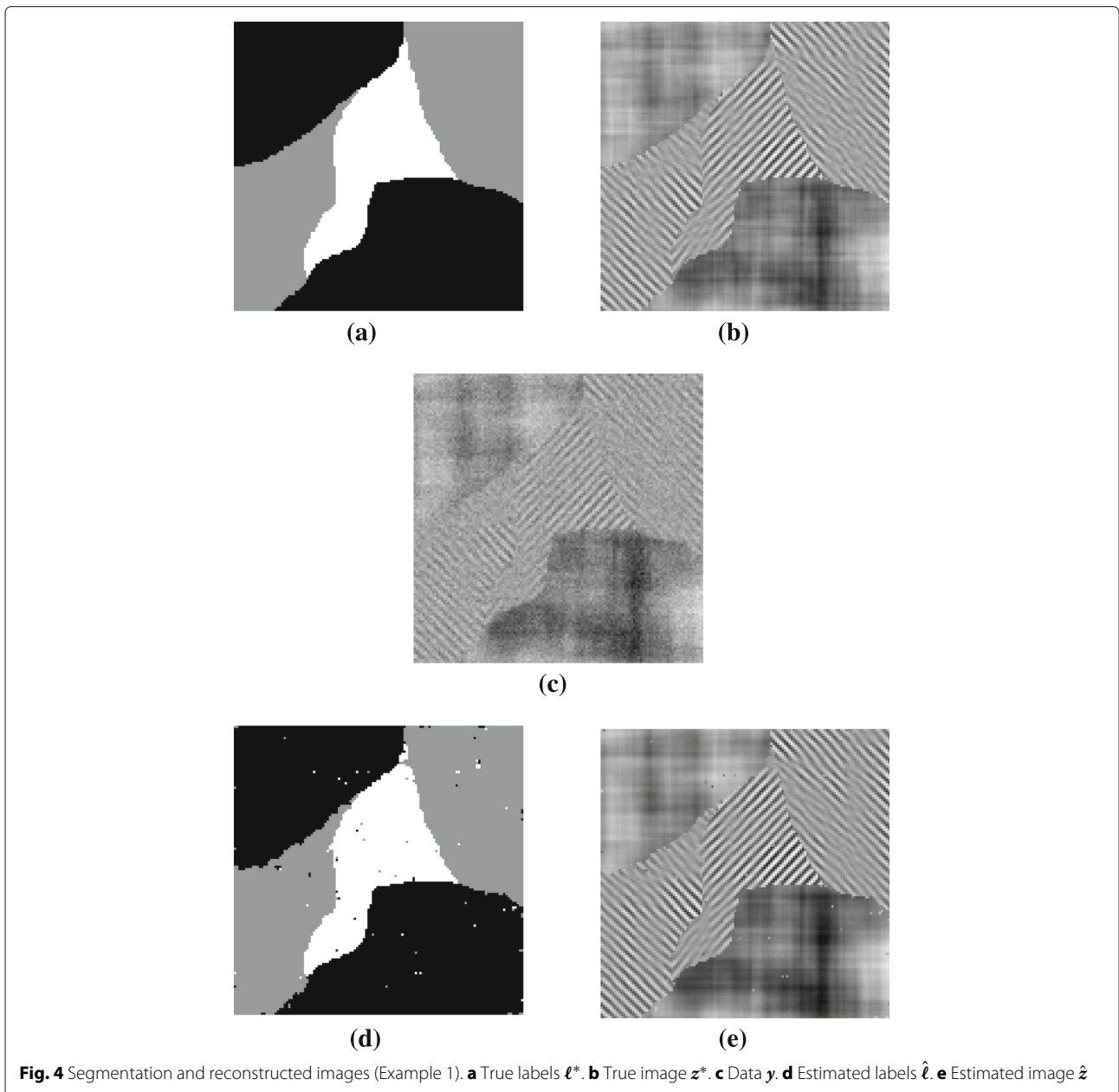


Fig. 4 Segmentation and reconstructed images (Example 1). **a** True labels ℓ^* . **b** True image z^* . **c** Data y . **d** Estimated labels $\hat{\ell}$. **e** Estimated image \hat{z}

of mislabelled pixels, despite the degradation of the image.

Remark 9 *The method is region-based, meaning that it provides closed contours, unlike a part of the existing works in texture segmentation.*

Moreover, the texture parameters estimation error is small, less than 10^{-2} , as mentioned in Table 1. The full textured images x_k are also accurately estimated, having the same characteristics as the original textured images. The blur and the noise are reduced in the resulting image (Fig. 4e) with respect to the data (Fig. 4c), and it strongly resembles the original image (Fig. 4b).

5.2.1 Label analysis: error and probability of error

One of the main advantages of probabilistic approaches is that they not only provide estimates for the unknowns, but also coherent uncertainties associated to these estimates. Figure 5 illustrates our analysis on the label estimates and their probability.

Figure 5a gives the empirical marginal probabilities for the three values of the label $\ell_p = 1, \ell_p = 2$ and $\ell_p = 3$ for each pixels $p = 1, \dots, P$. Figure 5b gives the probabilities of the selected labels (the one with the maximum probability). This maximum probability can have various values: a small value indicates a less reliable decision for the label. These probabilities are small (black or grey) at

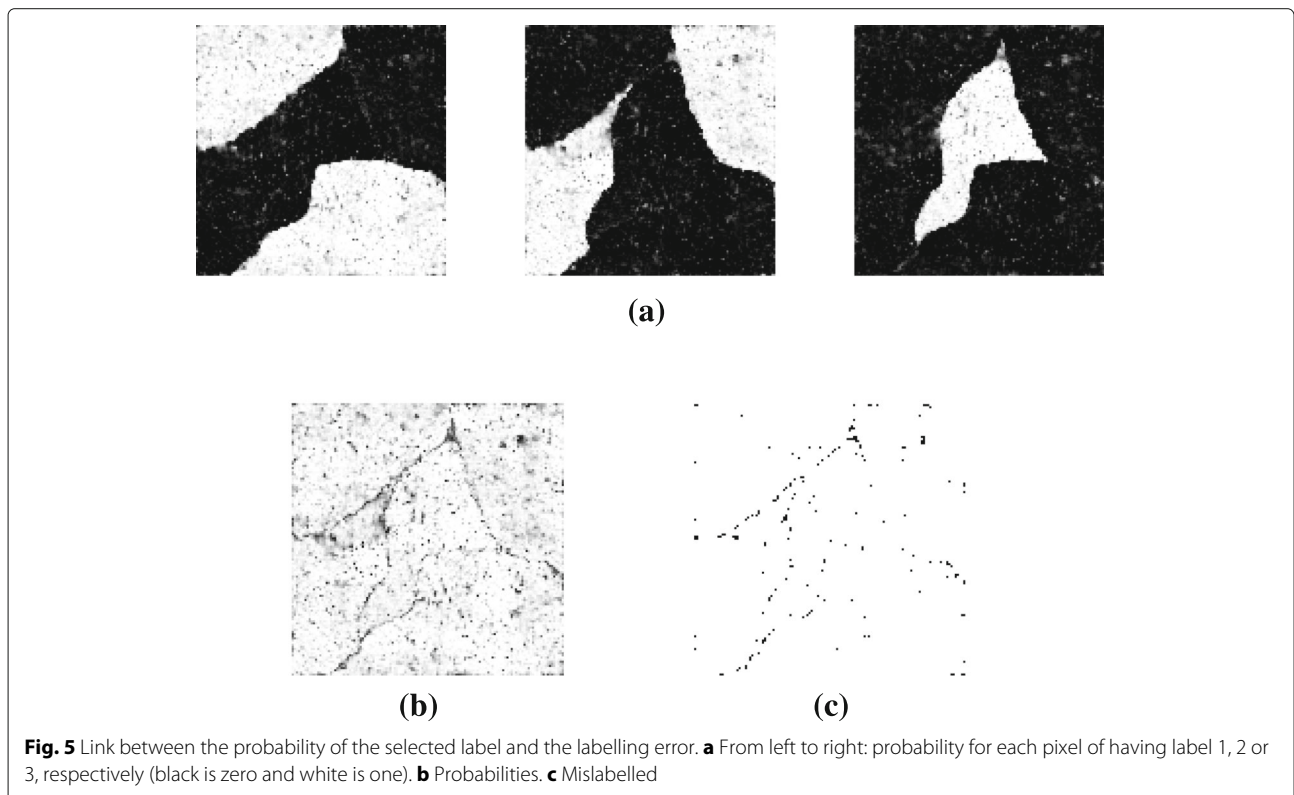
certain locations in the image Fig. 5b, and it is safe to assume that at these locations, there is a smaller chance of selecting a correct label.

This analysis can naturally be done even without the knowledge of the true labels. In order to verify if indeed we are more prone to error in the area with small posterior probability, we have compared the selected label configuration $\hat{\ell}$ to the true one ℓ^* . We can immediately notice in Fig. 5c that all of the mislabelled pixels are in fact positioned in the areas of weaker probability, shown in Fig. 5b. This reinforces our statement concerning the utility of the probabilistic approach, due to its ability to anticipate errors.

5.2.2 Other image topologies, blur and noise

In the case of the second image topology, given in Fig. 6, although the number of textures is reduced ($K = 2$), the task is more difficult due to the shape of the regions: the presence of a relatively thin, continuous structure makes the label decision hard. In addition, only a small patch of the texture associated with the “white” class is present and that complicates the texture parameter estimation. However, the results shown in Fig. 6 are remarkably correct, for both label and image. We only have 0.64% of mislabelled pixels.

Our third example is given in Fig. 7. Here again, the shape of some of the regions are relatively thin making the label decision hard and only a small patches of the



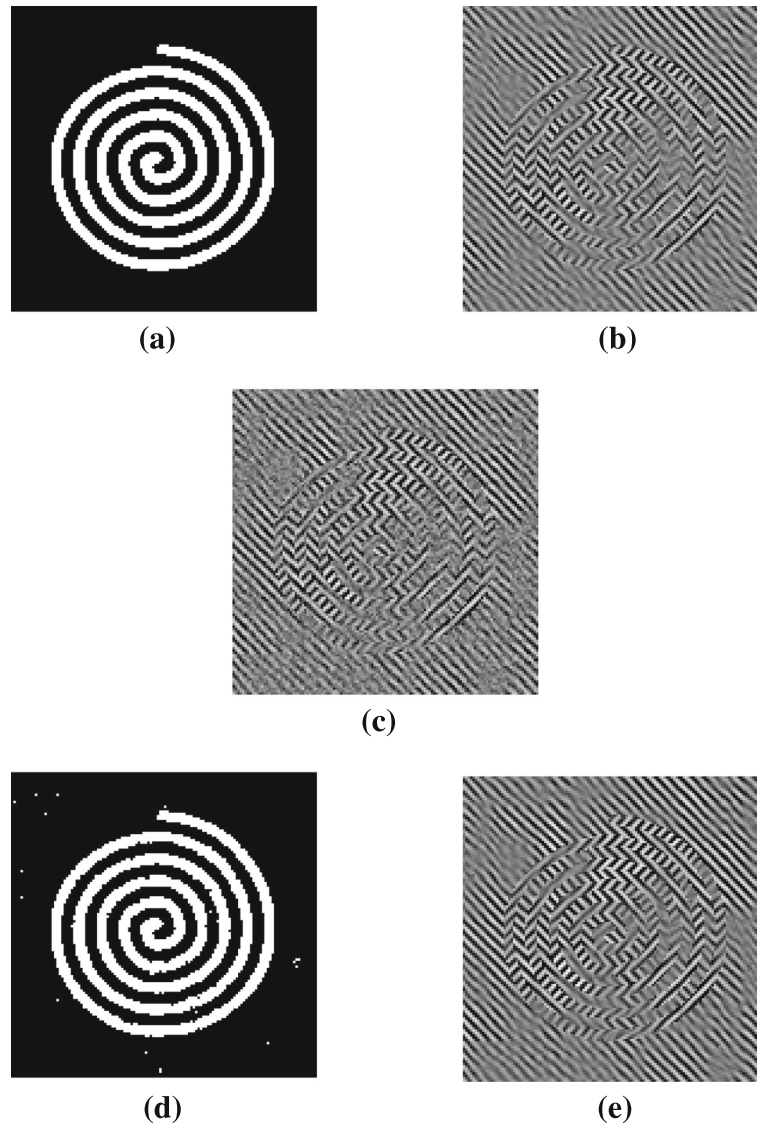


Fig. 6 Segmentation and reconstructed images (Example 2). **a** True labels ℓ^* . **b** True image z^* . **c** Data y . **d** Estimated labels $\hat{\ell}$. **e** Estimated image \hat{z}

second texture class is observed making texture parameter estimation difficult. Figure 7 illustrates the method performances in a weaker convolution case $w = 2$ and higher noise level $\gamma_n = 5$. The method performs very well in this case, the estimated label field being very close to the true labels (only 0.70% of miss-labelled pixels).

6 Conclusion and perspectives

The paper presents our method for joint deconvolution and segmentation, dedicated to textured images, with an emphasis on oriented structures. This is a very difficult task due to the large amount of unknowns and their complicated dependencies. The formulation of the

problem itself has demanded a careful consideration in order to design the best manner to accurately account for the hierarchical dependencies. In this context, the most adapted choice was to model K full images x_k corresponding to each class, rather than directly model the compound image z itself. This has allowed us to obtain an expression for the joint probability distribution in a relatively convenient form.

The proposed solution follows a Bayesian strategy that yields optimal functions in the sense of minimum risk for the decisions (labels) and for the estimations (continuous parameters). Both are founded on the posterior (marginal maximizer and mean). The

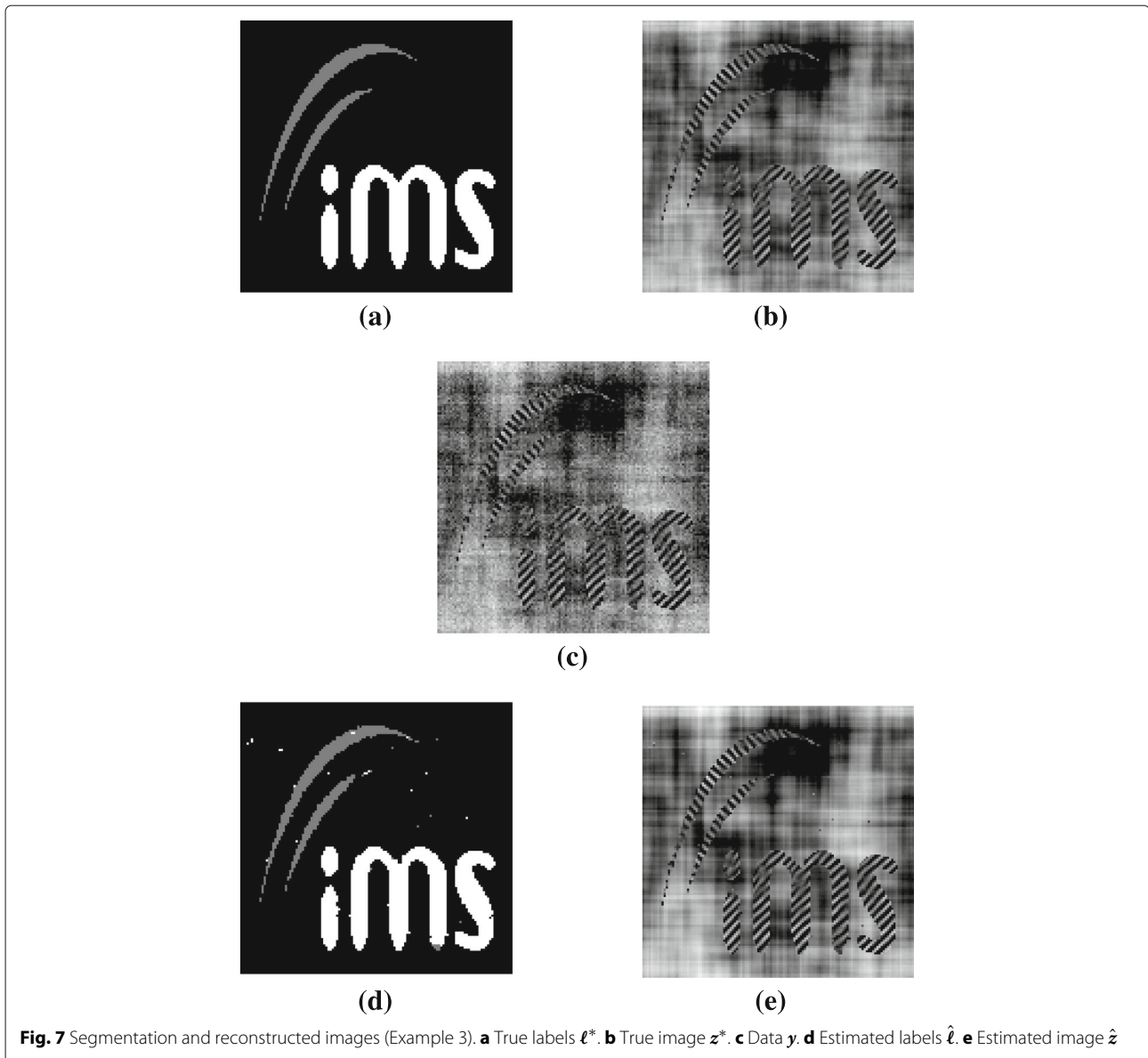


Fig. 7 Segmentation and reconstructed images (Example 3). **a** True labels ℓ^* . **b** True image z^* . **c** Data y . **d** Estimated labels $\hat{\ell}$. **e** Estimated image \hat{z}

intricate nature of the posterior distribution does not allow for an analytical expression for either the decisions or the estimates. A numerical approach is then used to explore the posterior, and the samples are subsequently used in computing them. The numerical scheme is guaranteed to converge: samples are asymptotically drawn under the posterior and empirical approximation converges towards the optimal decisions and estimates.

Nevertheless, the sampling process for the full set of variables has also proved to be challenging and has required advanced sampling approaches to overcome the impasses. We resort to a Gibbs sampler in order to split

the original problem for the full set of variables in several smaller problems for subsets of variables.

- (i) One of the steps requires the sampling of a Gaussian density in large dimension and we resort to recent developments on Perturbation-Optimization.
- (ii) The method includes the sampling of the granularity coefficient: it is itself a thorny question, hardly ever tackled. The proposed approach relies on the inverse cumulative density function and takes advantage of our precomputations of the partition function.
- (iii) Regarding the texture parameters, the algorithm resorts to a recent efficient directional

Metropolis-Hastings step within the Gibbs loop.

The proposed methodological aspects are original and have contributed to developing an approach that is both theoretically sound and practically efficient for the problem.

The previous section has presented the results of a series of numerical assessments performed on various convolution and noise conditions, for different image topologies. These results have shown that the method is able to accurately segment the image, provide a good estimation for the texture parameters as well as the hyperparameters and thus accurately restore the original image.

From a theoretical and modelling standpoint, the study leads us to several future developments.

- A future contribution is the use of a non-Gaussian model for the constituent textures, possibly based on latent variables and conditional Gaussian models [67]. This would add an extra layer of complexity to the model and to the sampling stage.
- The second future development aims at performing a myopic deconvolution [56, 68], i.e., considering that w , the width of the convolution filter, is unknown and estimating it along with the rest of the parameters.
- Thirdly, the problem of missing data (inpainting) will also be addressed. An extension of the present work to solve this problem would require to include a truncation matrix, say T , and substitute H by TH in (5).
- The fourth future contribution will deal with the problem of model selection, especially to choose the number of classes [69] (see also our previous works [70–72]). The difficulty would regard the computation of the evidences of the models.

The study also opens up new perspectives from a numerical standpoint, notably in order to reduce computation time.

- A future contribution will resort to the Swendsen-Wang algorithm in order to improve the sampling step of the label field [73] (see also [74]).
- The second future development in order to reduce computation time could rely on variational Bayes approaches [30, 32, 75] (see also [76, 77]).
- Thirdly, the problem of fast sampling will also be addressed through the abundant literature as already mentioned [47–54] and more recently [78].

As it can be seen from this brief listing of the perspectives, the work on this topic is far from being over. Nevertheless, even in its current form, the method

presented in this paper addresses a problem that had not been tackled so far (deconvolution-segmentation of textured images including hyperparameter and texture parameter estimation), while achieving very satisfactory results.

Endnotes

¹ Except for the Ising field ($K = 2$), see [79], also [80, 81].

² A unique step is used, in order to design a valid algorithm.

³ The algorithm is implemented within the computing environment Matlab on a standard PC with a 3 GHz CPU and 64 GB of RAM.

Appendix A: Potts partition

For the sake of self-containedness, we describe here the pre-computation of the partition function already given in our previous paper [40]. It is based on known properties [38, 39] for the partition function of the exponential family distributions.

Let us note $\sigma(\ell) = \sum_{p \sim q} \delta(\ell_p; \ell_q)$ the number of pair of adjacent pixels with identical label. The partition $\mathcal{Z}(\beta)$ normalizes the probability distribution (1), so it writes:

$$\mathcal{Z}(\beta) = \sum_{\ell} \exp[\beta \sigma(\ell)]$$

where the summation runs over all the configurations of the field $\ell \in \{1, \dots, K\}^P$. Numerically, it is a colossal summation over the K^P possible configurations and the exhaustive exploration of these configurations is impossible (except for minuscule images). The derivation w.r.t. β straightforwardly yields:

$$\mathcal{Z}'(\beta) = \sum_{\ell} \sigma(\ell) \exp[\beta \sigma(\ell)]$$

then dividing by $\mathcal{Z}(\beta)$ we have

$$\frac{\mathcal{Z}'(\beta)}{\mathcal{Z}(\beta)} = \sum_{\ell} \sigma(\ell) \mathcal{Z}(\beta)^{-1} \exp[\beta \sigma(\ell)] .$$

The left-hand side reads as the derivative of the log-partition $\tilde{\mathcal{Z}}(\beta) = \log \mathcal{Z}(\beta)$ and the right-hand side reads as an expectation:

$$\tilde{\mathcal{Z}}'(\beta) = \sum_{\ell} \sigma(\ell) \Pr[\ell|\beta] = E[\sigma(\mathbf{L})] .$$

Consequently, the derivative of the log-partition is an expectation. It can be approximated by an empirical average:

$$\tilde{\mathcal{Z}}'(\beta) \simeq \frac{1}{N} \sum_n \sigma(\ell_n)$$

where the ℓ_n , for $n = 1, \dots, N$, are N realizations of the field (given β). It remains a huge task but it is attainable: it required several weeks of intensive computation (on a standard PC), but it is done once for all. Results are given in Fig. 8 and this is the keystone for the estimation of β in this paper.

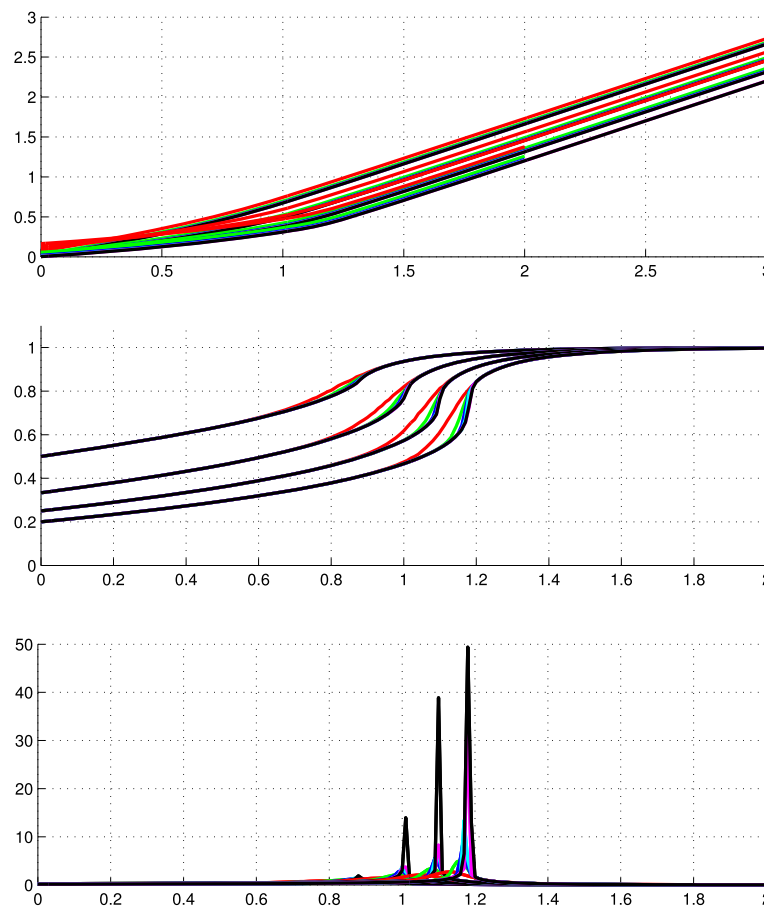


Fig. 8 From top to bottom: Log-partition $\tilde{Z}(\beta)$, its first and second derivatives as a function of β (from $\beta = 0$ to $\beta = 3$) for various sizes of image ($P = 64, 128, 256, 512$) and number of class ($K = 2, 3, 4, 5$)

Abbreviations

\sim : Neighbor relation between pixels; β : Granularity coefficient (Potts field); K : Number of texture classes; \mathbf{l} : Unobserved (hidden) labels; \mathbf{I}_N : $\{0, 1, \dots, N - 1\}$; θ_k : Texture parameters; P : Number of pixels; $\mathbf{R}_k, \mathbf{\Lambda}_k$: Texture covariance and precision structure; \mathbf{x}_k : Unobserved (hidden) textured images; \mathbf{y} : Observed image; γ_k, γ_n : Texture and noise levels; \mathbf{z} : Unobserved (hidden) image; \mathcal{Z} : Partition function (Potts field); $\delta(\cdot, \cdot)$: Kronecker function; CbC: Circulant-block-circulant; MALA: Metropolis adjusted Langevin algorithm; MCMC: Monte Carlo Markov chain; MH: Metropolis-hastings; PSD: Power spectral density; RWMH: Random walk metropolis-hastings; TbT: Toeplitz-block-toeplitz; w.r.t.: With respect to; ZF: Zero-forcing

Acknowledgements

Not applicable.

Funding

Not applicable.

Availability of data and materials

Please contact the author for data requests.

Authors' contributions

Both authors jointly developed the work presented in this paper. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 26 November 2016 Accepted: 10 December 2018

Published online: 05 March 2019

References

1. M. Petrou, P. Garcia-Sevilla, *Dealing with Texture*. (Wiley, Chichester, England, 2006)
2. G. L. Gimelfarb, *Image Textures and Gibbs Random Fields*. (Kluwer Academic Publishers, 1999)
3. J. P. Da Costa, F. Michelet, C. Germain, O. Lavialle, G. Grenier, Delineation of vine parcels by segmentation of high resolution remote sensed images. *Precision Agric.* **8**, 95–110 (2007)
4. J. P. Da Costa, F. Galland, A. Roueff, C. Germain, Unsupervised segmentation based on Von Mises circular distributions for orientation estimation in textured images. *JElectron Imaging.* **21**(2) (2012)
5. J. C. Russ, *The Image Processing Handbook (Seventh Edition)*. (CRC Press, 2015)
6. J. Zhang, J. Zheng, J. Cai, in *IEEEConference on Computer Vision and Pattern Recognition*. A diffusion approach to seeded image segmentation, (2010), pp. 2125–2132
7. L. Garcia Ugarriza, E. Saber, S. R. Vantaram, V. Amuso, M. Shaw, R. Bhaskar, Automatic Image Segmentation by Dynamic Region Growth and Multiresolution Merging. *IEEE Trans. Image Process.* **18**(10), 2275–2288 (2009)

8. S. Alpert, M. Galun, A. Brandt, R. Basri, Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration. *IEEE Trans. Pattern. Anal. Mach. Intell.* **34**(2), 315–327 (2012)
9. T. F. Chan, P. Mulet, On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Numer. J. Anal.* **36**(2), 354–367 (1999)
10. J. Malik, S. Belongie, T. Leung, J. Shi, Contour and Texture Analysis for Image Segmentation. *Int. Comput. J. Vis.* **43**, 7–27 (2001)
11. L. Grady, Random Walks for Image Segmentation. *IEEE Trans. Pattern. Anal. Mach. Intell.* **28**(11), 1768–1783 (2006)
12. A. K. Sinop, L. Grady, in *IEEE International Conference on Computer Vision*. A Seeded Image Segmentation Framework Unifying Graph Cuts And Random Walker Which Yields A New Algorithm, (2007), pp. 1–8
13. M. Tuceryan, Moment-based texture segmentation. *Pattern Recogn. Lett.* **15**(7), 659–668 (1994)
14. S. Arivazhagan, L. Ganesan, Texture segmentation using wavelet transform. *Pattern. Recogn. Lett.* **24**, 3197–3203 (2003)
15. L. Wolf, X. Huang, I. Martin, D. Metaxas, in *In European Conference on Computer Vision*. Patch-based texture edges and segmentation, (2006)
16. A. Lillo, G. Motta, J. A. Storer, in *Pattern Recognition and Image Analysis. vol. 4477 of Lecture Notes in Computer Science*, ed. by J. Martí, J. M. Benedí, A. M. Mendonça, and J. Serrat. Supervised Segmentation Based on Texture Signatures Extracted in the Frequency Domain (Springer Berlin Heidelberg, 2007), pp. 89–96
17. H. Mobahi, S. Rao, A. Y. Yang, S. S. Sastry, Y. Ma, Segmentation of Natural Images by Texture and Boundary Compression. *Int. Comput. J. Vis.* **95**(1), 86–98 (2011)
18. M. Galun, E. Sharon, R. Basri, A. Brandt, in *IEEE International Conference on Computer Vision. vol. 1*. Texture segmentation by multiscale aggregation of filter responses and shape elements, (2003), pp. 716–723
19. X. Liu, D. Wang, Image and Texture Segmentation Using Local Spectral Histograms. *IEEE Trans. Image Process.* **15**(10), 3066–3077 (2006)
20. S. Todorovic, N. Ahuja, in *IEEE International Conference on Computer Vision*. Texel-based texture segmentation, (2009), pp. 841–848
21. D. Geman, S. Geman, C. Graffigne, P. Dong, Boundary Detection by Constrained Optimization. *IEEE Trans. Pattern. Anal. Mach. Intell.* **12**(7), 609–628 (1990)
22. Z. Tu, S. C. Zhu, H. Y. Shum, in *IEEE International Conference on Computer Vision. vol. 2*. Image segmentation by data driven Markov chain Monte Carlo, (2001), pp. 131–138
23. H. Deng, D. A. Clausi, Unsupervised image segmentation using a simple MRF model with a new implementation scheme. *Pattern. Recognit.* **37**(12), 2323–2335 (2004)
24. P. F. Felzenszwalb, D. P. Huttenlocher, Efficient Graph-Based Image Segmentation. *Int. Comput. J. Vis.* **59**(2), 167–181 (2004)
25. Y. Boykov, G. Funka-Lea, Graph cuts and efficient ND image segmentation. *Int. Comput. J. Vis.* **70**(2), 109–131 (2006)
26. G. Celeux, F. Forbes, N. Peyrard, *EM-based image segmentation using Potts models with external field*. (INRIA, 2002)
27. R. Morris, X. Descombes, J. Zerubia, *Fully Bayesian image segmentation - an engineering perspective*. (INRIA, Sophia Antipolis France, 1996). 3017
28. A. Barbu, S. C. Zhu, Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *IEEE Trans. Pattern. Anal. Mach. Intell.* **27**(8), 1239–1253 (2005)
29. M. Pereyra, N. Dobigeon, H. Batatia, J. Y. Tournet, Estimating the Granularity Coefficient of a Potts-Markov Random Field within a Markov Chain Monte Carlo Algorithm. *IEEE Trans. Image Process.* **22**(6), 2385–2397 (2013)
30. O. Féron, B. Duchêne, A. Mohammad-Djafari, Microwave imaging of inhomogeneous objects made of a finite number of dielectric and conductive materials from experimental data. *Inverse Problems.* **21**(6), 95–115 (2005)
31. M. Mignotte, A Segmentation-Based Regularization Term for Image Deconvolution. *IEEE Trans. Image Process.* **15**(7), 1973–1984 (2006)
32. H. Ayasso, A. Mohammad-Djafari, Joint NDT Image Restoration and Segmentation Using Gauss-Markov-Potts prior Models and Variational Bayesian Computation. *IEEE Trans. Image Process.* **19**(9), 2265–2277 (2010)
33. O. Eches, N. Dobigeon, J. Y. Tournet, Enhancing hyperspectral image unmixing with spatial correlations. *IEEE Trans. Geosci. Remote Sens.* **49**(11), 4239–4247 (2011)
34. M. Pereyra, N. Dobigeon, H. Batatia, J. Y. Tournet, Segmentation of skin lesions in 2D and 3D ultrasound images using a spatially coherent generalized Rayleigh mixture model. *IEEE Trans. Med. Imaging.* **31**(8), 1509–1520 (2012)
35. O. Eches, J. A. Benediktsson, N. Dobigeon, J. Y. Tournet, Adaptive Markov random fields for joint unmixing and segmentation of hyperspectral image. *IEEE Trans. Image Process.* **22**(1), 5–16 (2013)
36. Y. Altmann, N. Dobigeon, S. McLaughlin, J. Y. Tournet, Residual component analysis of hyperspectral images - Application to joint nonlinear unmixing and nonlinearity detection. *IEEE Trans. Image Process.* **23**(5), 2148–2158 (2014)
37. M. Storath, A. Weinmann, J. Friel, M. Unser, Joint image reconstruction and segmentation using the Potts model. *Inverse Probl.* **31** (2015)
38. G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. (Springer Verlag, Berlin, Germany, 2003)
39. D. MacKay, *Information Theory, Inference, and Learning Algorithms*. (Cambridge University Press, 2008)
40. R. Rosu, J. F. Giovannelli, A. Giremus, C. Vacar, in *Proceedings of the International Conference on Acoustic, Speech and Signal Processing*. Potts model parameter estimation in Bayesian segmentation of piecewise constant images, (Brisbane, Australia, 2015)
41. J. F. Giovannelli, A. Barbos, in *Proceedings of the International Conference on Statistical Signal Processing*. Unsupervised segmentation of piecewise constant images from incomplete, distorted and noisy data, (Palma de Majorque, Spain, 2016)
42. L. Risser, T. Vincent, P. Ciuciu, J. Idier, *Application to within-subject fMRI data analysis*, (London England, 2009)
43. N. Friel, A. N. Pettitt, R. Reeves, E. Wit, Bayesian inference in hidden Markov random fields for binary data defined on large lattices. *Comput. J. Graph. Stat.* **18**, 243–261 (2009)
44. J. Moller, A. N. Pettitt, R. Reeves, K. K. Berthelsen, An efficient Markov chain Monte Carlo method for distributions with untractable normalising constants. *Biometrika.* **93**(2), 451–458 (2006)
45. A. N. Pettitt, N. Friel, R. Reeves, Efficient calculation of the normalizing constant of the autologistic and related models on the cylinder and lattice. *J. Royal Stat. Soc. B.* **65**(1), 235–246 (2003)
46. R. Reeves, A. N. Pettitt, Efficient recursions for general factorisable models. *Biometrika.* **91**(3), 751–757 (2004)
47. J. M. Marin, C. P. Robert, *Bayesian Core. A Practical Approach to Computational Bayesian Statistics. Texts in statistics*. (Springer, Paris, France, 2007)
48. J. Albert, *Bayesian Computation With R*. (Springer-Verlag New York Inc., New York, 2009)
49. C. P. Robert, G. Casella, *Monte-Carlo Statistical Methods. Springer Texts in Statistics*. (Springer, New York, 2004)
50. D. Gamerman, H. F. Lopes, *Markov Chain Monte Carlo: stochastic simulation for Bayesian inference. 2nd ed.* (Chapman & Hall/CRC, Boca USA Raton, 2006)
51. G. O. Roberts, R. L. Tweedie, Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli.* **2**(4), 341–363 (1996)
52. G. Roberts, O. Stramer, Langevin Diffusions and Metropolis-Hastings Algorithms. *Methodol. Comput. Appl. Probab.* **4**, 337–358 (2003)
53. Y. Qi, T. P. Minka, in *First Cape Cod Workshop on Monte Carlo Methods*. Hessian-based Markov Chain Monte-Carlo Algorithms (Cape Cod, Massachusetts, USA, 2002)
54. M. Girolami, B. Calderhead, Riemannian manifold Hamiltonian Monte Carlo (with discussion). *J. Royal Stat. Soc. B.* **73**, 123–214 (2011)
55. C. Vacar, J. F. Giovannelli, Y. Berthoumieu, in *Proceedings of the International Conference on Acoustic, Speech and Signal Processing*. Langevin and Hessian with Fisher approximation stochastic sampling for parameter estimation of structured covariance, (Prague, Czech Republic, 2011), pp. 3964–3967
56. C. Vacar, J. F. Giovannelli, Y. Berthoumieu, Bayesian texture and instrument parameter estimation from blurred and noisy images using MCMC. *IEEE Signal Process. Lett.* **21**(6), 707–711 (2014)
57. J. F. Giovannelli, C. Vacar, in *EUSIPCO*. Deconvolution-Segmentation for Textured Images, (Kos, Greece, 2017)
58. C. Fox, A Conjugate Direction Sampler for Normal Distributions with a Few Computed Examples. University of Otago, Dunedin, New Zealand: Electronics Technical Report No. 2008-1 (2008). Internal report

59. G. Papandreou, A. Yuille, in *Proc. Int. Conf. on Neural Information Processing Systems (NIPS)*. Gaussian Sampling by Local Perturbations, (Vancouver, Canada, 2010), pp. 1858–1866
60. A. Parker, C. Fox, Sampling Gaussian Distributions in Krylov Spaces with Conjugate Gradients. *SIAM J. Sci. Comput.* **34**(3) (2012)
61. F. Orieux, O. Féron, J. F. Giovannelli, Sampling high-dimensional Gaussian fields for general linear inverse problem. *IEEE Signal Process. Lett.* **19**(5), 251–254 (2012)
62. A. Barbos, F. Caron, J. F. Giovannelli, A. Doucet. booktitle=NIPS-2017. Clone MCMC: Parallel High-Dimensional Gaussian Gibbs Sampling, (Long Beach, USA, 2017)
63. C. Gilavert, S. Moussaoui et, J. Idier, Efficient Gaussian sampling for solving large-scale inverse problems using MCMC. *IEEE Trans Signal Processing.* **63**(1), 70–80 (2015)
64. D. P. Bertsekas, *Nonlinear programming, 2nd ed.* (Belmont, MAUSA: Athena Scientific, 1999)
65. J. Nocedal, S. J. Wright, *Numerical Optimization. Series in Operations Research.* (Springer Verlag, New York, 2008)
66. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. vol. 3 of Foundations and Trends in Machine Learning.* (USA MA Now Publishers Inc, Hanover, 2011)
67. C. Vacar, J. F. Giovannelli, A. M. Roman, in *Proceedings of the International Conference on Image Processing. vol. 19.* Bayesian texture model selection by harmonic mean, (Orlando, 2012), p. 5
68. F. Orieux, J. F. Giovannelli, T. Rodet, Bayesian estimation of regularization and point spread function parameters for Wiener–Hunt deconvolution. *J. Opt. Soc. Am.* **27**(7), 1593–1607 (2010)
69. T. Ando, *Bayesian model selection and statistical modeling.* (Chapman & Hall/CRC, Boca USA Raton, 2010)
70. J. F. Giovannelli, A. Giremus, in *Proceedings of the International Conference on Statistical Signal Processing (special session).* Bayesian noise model selection and system identification based on approximation of the evidence, (Gold Coast, Australia, 2014)
71. A. Barbos, A. Giremus, J. F. Giovannelli, in *Actes du 25^e colloque GRETSI.* Bayesian noise model selection and system identification using Chib approximation based on the Metropolis-Hastings sampler, (Lyon, France, 2015)
72. C. Vacar, J. F. Giovannelli, Y. Berthoumieu, Bayesian Texture Classification From Indirect Observations Using Fast Sampling. *IEEE Trans. Signal Process.* **64**(1), 146–159 (2016)
73. D. M. Higdon, Auxiliary Variable Methods for Markov Chain Monte Carlo with Applications. *J. Am. Stat. Assoc.* **93**(442), 585–595 (2012)
74. J. Sodjo, A. Giremus, N. Dobigeon, J. F. Giovannelli, in *Proceedings of the International Conference on Acoustic, Speech and Signal Processing.* A generalized Swendsen-Wang algorithm for Bayesian nonparametric joint segmentation of multiple images, (New Orleans, USA, 2017)
75. V. Smidl, A. Quinn, *The variational Bayes Method in Signal Processing.* (Springer, 2006)
76. W. Fan, N. Bouguila, D. Ziou, Variational learning for finite Dirichlet mixture models and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **3**(5), 762–774 (2012)
77. B. Ait-El-Fquih, J. F. Giovannelli, N. Paul, A. Girard, I. Hoteit, in *Proceedings of the International Conference on Statistical Signal Processing.* A variational Bayesian estimation scheme for parameteric point-like pollution source of groundwater layers, (Freibourg, Germany, 2018)
78. L. Martino, J. Read, D. Luengo, Independent Doubly Adaptive Rejection Metropolis Sampling Within Gibbs Sampling. **63**(12), 3123–3138 (2015)
79. L. Onsager, A Two-Dimensional Model with an Order-Disorder Transition. *Phys Rev.* **65**(3 & 4), 117–149 (1944)
80. J. F. Giovannelli, in *Proceedings of the International Conference on Image Processing.* Estimation of the Ising field parameter thanks to the exact partition function, (Hong-Kong, 2010), pp. 1441–1444
81. Giovannelli J.F., in *Proceedings of the International Conference on Image Processing.* Estimation of the Ising field parameter from incomplete and noisy data, (Brussels, Belgium, 2011), pp. 1893–1896

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
