Bayesian multiscale analysis of images modeled as Gaussian Markov random fields

Kevin Thon[a], Håvard Rue[b], Stein Olav Skrøvseth[a,*], Fred Godtliebsen[c]

[a]*Norwegian Centre for Integrated Care and Telemedicine, University Hospital of North Norway, P.O. Box 6060, N-9038 Tromsø, Norway*
[b]*Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway*
[c]*Department of Mathematics and Statistics, University of Tromsø, N-9037 Tromsø, Norway*

Abstract

A Bayesian multiscale technique for detection of statistically significant features in noisy images is proposed. The prior is defined as a stationary intrinsic Gaussian Markov random field on a toroidal graph, which enables efficient computation of the relevant posterior marginals. Hence the method is applicable to large images produced by modern digital cameras. The technique is demonstrated in two examples from medical imaging.

Keywords: Scale space, multi-resolution analysis, Bayesian analysis, Gaussian Markov random fields

1. Introduction

Scale space or multiscale approaches to image and signal analysis exist in a multitude of varieties, and in computer vision the concept is rooted in the realization that physical observation is an inherently multiresolutional process. Different levels of smoothing reveal different structures, and image structures exist as meaningful entities only over certain ranges of scale [1]. In the current article, structure in images is defined as significant gradient and curvature above the overall uncertainty.

[*] Corresponding author. Tel.:+47-95805841, Fax: +47-77754098
Email addresses: kevin.thon@telemed.no (Kevin Thon), havard.rue@math.ntnu.no (Håvard Rue), stein.olav.skrovseth@telemed.no (Stein Olav Skrøvseth), fred@stat.uit.no (Fred Godtliebsen)

The use of Markov Random Fields in imaging has a long tradition [2], and has been used in a variety of applications, including segmentation [3], low level vision [4], object identification [5], as well as pattern recognition in dermatoscopic images [6]. The underlying graph is a regular lattice corresponding to the pixels of the image, which has a Markov property, i.e., the value of any pixel is a random variable conditionally independent of all except the neighboring pixels defined in a neighborhood structure. Restricting the field to a Gaussian Markov Random Field (GMRF), the field attains the property that the structure of the inverse covariance matrix, i.e., the precision matrix, directly reflects the conditional dependency structure of the graph [7]. This enables fast numerical routines to be employed, and thus it is feasible to treat large images in a multiscale analysis. The fact that the graph is highly regular, in our case a torus, gives further computational advantages. Although the families of smoothes considered throughout this article are exclusively GMRFs, it should be pointed out that many alternative techniques exist for image smoothing or denoising which are not based on Markov Random Fields, e.g. Adaptive Smoothing [8], Anisotropic Diffusion [9], and Jump Regression Analysis [10][11][12].

In [13] and [14] Godtliebsen et. al. proposed the S^3 (Significance in Scale-Space) methodology for finding significant features in grayscale images and density estimates, respectively. The methodology combines scale space ideas with statistical inference, and is inspired by the *SiZer* (Significant Zero crossings of derivatives) methodology by Chaudhuri and Marron [15]. SiZer is a scale space method within non-parametric curve estimation, and looks for derivatives that are significant above the noise level at different levels of smoothing. S^3 proceeds by viewing the image smoothed by convolutions of Gaussian kernels of varying bandwidth, and calculates gradients and curvatures that are significant above the noise level, and in this sense represents a two dimensional version of SiZer. In [16] and [17], a Bayesian approach for exploratory analysis of time series data is developed, and it is demonstrated that in certain applications it outperforms SiZer, since it can make meaningful inference on much lower levels of smoothing. In [18] Erästö and Holmström introduce a similar methodology which they refer to as BSiZer (Bayesian SiZer), for finding significant features in scatterplots, and in [19] these ideas are further developed into a tool for multiresolution analysis of random signals.

In this article a new Bayesian methodology to finding significant features in images is developed, extending the Bayesian methodology from [16] and

[17] to two dimensions. The method can be seen as a Bayesian version of S^3, and we will refer to the method as Bayesian multiscale analysis (BMulA). The method looks for gradients and curvature which are significant above the overall posterior uncertainty, stemming from noise and the uncertainty in the prior which is chosen as one of two intrinsic Gaussian Markov random fields (IGMRFs), i.e., a rank deficient GMRF [7]. A limitation of the S^3 methodology is that it is, particularly in the implementation the authors make available online, computationally very demanding and thus impractical for use on images one typically obtains from modern digital cameras. An area of particular concern in developing the BMulA methodology has therefore been to make the implementation highly computationally efficient, thus allowing the method to be used on large images. This computational efficiency is achieved at the expense of some flexibility, in that we restrict the model to deal with the simple image plus noise case. We further assume that the noise term consists of i.i.d. Gaussian random variables with known variance, and that the image is a stationary Gaussian Markov random field defined on a toroidal graph [7]. The model is still sufficiently general to be used in a wide range of applications and, in particular, the toroidal graph only affects inference on the pixels close to the image boundaries, depending on the level of smoothing. In addition to being fast, this formulation allows for the exact calculation of posterior means and covariances, and we do not have to rely on costly Markov Chain Monte Carlo procedures.

The organization of this paper is as follows. Section 2 introduces the basic statistical model underlying BMulA, and explains the priors used in the software (available for download from http://www2.telemed.no/kevin). Section 3 briefly explains the mathematical foundations for the computational efficiency of the proposed method, and Section 4 discusses the assessment of significant features. In Section 5, two examples are given of practical problems where the method is demonstrated as useful.

2. Model and assumptions

The underlying statistical model for the Bayesian multiscale analysis is that the observed image $\mathbf{Y}$ is the sum of an unknown image $\mathbf{X}$ and a noise term $\boldsymbol{\epsilon}$

$$\mathbf{Y} = \mathbf{X} + \boldsymbol{\epsilon}. \tag{1}$$

The elements of the noise term are further assumed to be i.i.d. Gaussian random variables with variance σ^2. Throughout this article we will also assume that the noise variance is known or can be estimated, and that the observed image has been standardized with regard to it, in the sense that the pixel values have been divided by σ. The likelihood follows immediately as

$$\pi(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \mathbf{I})\,. \tag{2}$$

Here $\mathbf{x}$ and $\mathbf{y}$ are the vectorization of $\mathbf{X}$ and $\mathbf{Y}$, attained by stacking the columns, and $\mathbf{I}$ is the identity matrix. The prior is modeled as a GMRF with precision matrix $\mathbf{Q}$ [7], so

$$\pi(\mathbf{x}) \propto \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}\right), \tag{3}$$

from which the posterior density follows:

$$\begin{aligned}\pi(\mathbf{x}|\mathbf{y}) \propto \pi(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}) &\propto \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{x})^T(\mathbf{y}-\mathbf{x})\right)\exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}\right)\\ &\propto \exp\left(-\frac{1}{2}\mathbf{x}^T(\mathbf{Q}+\mathbf{I})\mathbf{x} + \mathbf{y}^T\mathbf{x}\right). \end{aligned} \tag{4}$$

This means that the posterior density is a Gaussian with mean

$$\mathrm{E}(\mathbf{x}|\mathbf{y}) = (\mathbf{Q}+\mathbf{I})^{-1}\mathbf{y}, \tag{5}$$

and covariance

$$\mathrm{Cov}(\mathbf{x}|\mathbf{y}) = (\mathbf{Q}+\mathbf{I})^{-1}. \tag{6}$$

The relevant quantities are of a simple form and, in principle, equations (5) and (6) can be used to calculate both the posterior mean and covariance. The dimensions of the matrix to be inverted however, make this impractical for anything but the smallest of images. A m by n image implies a mn by mn covariance matrix and, in general, matrix inversion is a computationally expensive task ($\mathcal{O}((mn)^3)$). Luckily there are techniques for their efficient calculation, utilizing the block-circulant structure of the matrix to be inverted. The details of this will be given in Section 3.

2.1. Prior Image Model

There are two rather strong assumptions in the construction of the prior image model that together assure the desired block-circulant structure of the precision matrix. The first is that we consider the image to be defined on a torus. This amounts to introducing undesirable correlations across opposing edges. However, this approach facilitates efficient calculations on rather large images, so the edges constitute a minor part of the images for which the technique is intended. The user should, however, be aware of this limitation, and inferences on the image boundaries should not automatically be trusted, especially on high levels of smoothing. Secondly, the prior is a stationary GMRF on the torus. This means that, with $\mathbf{x}_{-ij}$ denoting the full field $\mathbf{x}$ with sites i and j excluded, the full conditionals $\{\pi(x_{ij}|\mathbf{x}_{-ij})\}$ have constant parameters not depending on ij, such that the local properties of the image are the same throughout the image.

There are several ways to specify a GMRF, the most common way of which is to specify the prior through its full normal conditionals with conditional mean and precision [20]

$$\begin{aligned}\mathrm{E}(x_i|\mathbf{x}_{-i}) &= -\frac{1}{Q_{ii}}\sum_{j:j\sim i} Q_{ij}x_j,\\ \mathrm{Prec}(x_i|\mathbf{x}_{-i}) &= Q_{ii},\end{aligned} \tag{7}$$

in the zero mean case, where $i \sim j$ denotes neighboring sites i and j, and in fact the Q_{ij}s uniquely define (see e.g. Theorem 2.6 from [7]) the elements of the precision matrix $\mathbf{Q}$ from equation (3). In principle, any well defined GMRF can be used as a prior, but in the software used in this article, two specific IGMRFs have been chosen. We will specify the IGMRFs through normal increments, following [7], from which the full normal conditionals follow.

2.1.1. IGMRFs on lattices

IGMRFs are improper GMRFs in the sense that they have a precision matrix that is not of full rank. In this article we follow the convention from [7], and define the order of an IGMRF as the rank deficiency of its precision matrix.

Since the precision matrix is not of full rank, its inverse does not exist. Hence, an IGMRF does not have a well defined mean and covariance matrix. IGMRFs do, however, have some interesting properties: The mean of an IGMRF of order k is defined only up to the addition of a polynomial of order $k-1$.

Turning our attention specifically to lattices, we define IGMRFs in terms of 'independent' increments, and for the first order case we define a normal increment between neighboring pixels i and j as

$$x_i - x_j \sim \mathcal{N}(0, \kappa^{-1}). \tag{8}$$

κ, which in [16] simply is referred to as a smoothing parameter (named β), is in the current formulation the precision (inverse variance) of the difference between neighboring pixels. The independence assumption of the increments gives the following IGMRF model:

$$\begin{aligned}\pi(\mathbf{x}) &\propto \kappa^{(n-1)/2} \exp\left(-\frac{\kappa}{2}\sum_{i\sim j}(x_i - x_j)^2\right)\\ &\propto \kappa^{(n-1)/2} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x}\right),\end{aligned} \tag{9}$$

where $i \sim j$ is to be understood as the set of all unordered pairs of neighbors, to avoid double counting. $\mathbf{x}$ is some ordering of the pixels in a 2d image, which will be by column throughout this article. Letting n_i denote the number of neighbors of pixel i, the precision matrix $\mathbf{Q}$ in equation (9) has elements

$$Q_{ij} = \kappa \begin{cases} n_i & i = j, \\ -1 & i \sim j, \\ 0 & \text{otherwise.} \end{cases}$$

Choosing the four nearest pixels as neighbors, and remembering that the image is defined on a torus (i.e. connected opposing edges, and no special treatment given to the boundary pixels), the precision matrix is block-circulant with base (The base of a block-circulant matrix will be defined in section 3, or see [7])

$$\mathbf{q}_\kappa = \kappa \begin{bmatrix} 4 & -1 & & & -1 \\ -1 & & & & \\ & & & & \\ & & & & \\ -1 & & & & \end{bmatrix}, \tag{10}$$

where empty elements are zero.

To construct a second order IGMRF in two dimensions, we can choose the increments

$$\begin{aligned}&\left(\Delta_{1,0}^2 + \Delta_{0,1}^2\right) x_{i-1,j-1}\\ &= (x_{i+1,j} + x_{i-1,j} + x_{i,j+1} + x_{i,j-1}) - 4x_{i,j} \sim \mathcal{N}(0, \kappa^{-1}),\end{aligned} \tag{11}$$

where $\Delta_{1,0}$ and $\Delta_{0,1}$ are the forward difference operators in the vertical and horizontal directions. In [7] it is shown that the precision matrix should have non-zero elements corresponding to

$$-\left(\Delta_{1,0}^2 + \Delta_{0,1}^2\right)^2 = -\left(\Delta_{1,0}^4 + 2\Delta_{0,1}^2\Delta_{1,0}^2 + \Delta_{0,1}^4\right), \tag{12}$$

which is a negative difference approximation to the *biharmonic* differential operator

$$\nabla^4 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)^2. \tag{13}$$

Equation (12) can be used in the construction of the precision matrix, but since in principle *any* difference approximation to ∇^4 can be used, we have used the *Mehrstellen stencil* [7]. This has the advantage of being isotropic in the discretization error, and is given in graphical notation as

$$\nabla^2 \approx -\frac{10}{3}\begin{matrix}\circ & \circ & \circ\\ \circ & \bullet & \circ\\ \circ & \circ & \circ\end{matrix} + \frac{2}{3}\begin{matrix}\circ & \bullet & \circ\\ \bullet & \circ & \bullet\\ \circ & \bullet & \circ\end{matrix} + \frac{1}{6}\begin{matrix}\bullet & \circ & \bullet\\ \circ & \circ & \circ\\ \bullet & \circ & \bullet\end{matrix}. \tag{14}$$

Here the operator for the central circle sums over the sites labelled by a closed circle, the open circles used for reference. Considering again the image on a torus, so that all pixels may be considered interior, we can decompose $\mathbf{Q}$ into $\mathbf{Q} = \kappa\mathbf{D}^T\mathbf{D}$, where $\mathbf{D}$ is found from equation (14) as a block-circulant matrix with base

$$\frac{1}{6}\begin{bmatrix} -20 & 4 & & 4\\ 4 & 1 & & 1\\ & & & \\ & & & \\ 4 & 1 & & 1 \end{bmatrix},$$

which, using results from Section 3.1, means that the base of $\mathbf{Q}$ is

$$\mathbf{q}_\kappa = \frac{\kappa}{36}\begin{bmatrix} 468 & -144 & 18 & & 18 & -144\\ -144 & -8 & 8 & & 8 & -8\\ 18 & 8 & 1 & & 1 & 8\\ & & & & & \\ 18 & 8 & 1 & & 1 & 8\\ -144 & -8 & 8 & & 8 & -8 \end{bmatrix}. \tag{15}$$

Having found the base of the precision matrices for a first and second order IGMRF, it is straightforward to find the corresponding full conditionals, since they are related through equation (7). For the first order case, with increments as in (8), we have

$$\begin{aligned}\mathrm{E}(x_{ij}|\mathbf{x}_{-ij}) &= \frac{1}{4}(x_{i+1,j} + x_{i-1,j} + x_{1,j+1} + x_{i,j-1}),\\ \mathrm{Prec}(x_{ij}|\mathbf{x}_{-ij}) &= 4\kappa, \end{aligned} \tag{16}$$

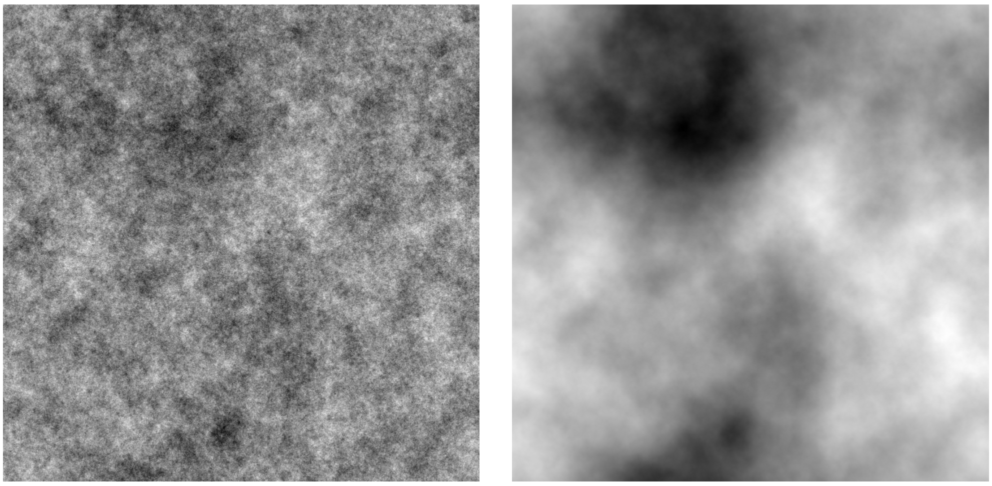
and for second order (with increments as in (14)):

$$\begin{aligned}\mathrm{E}(x_{ij}|\mathbf{x}_{-ij}) =& \frac{1}{468}\left(144 \begin{array}{ccccc} \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \bullet & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \bullet & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \end{array} - 18 \begin{array}{ccccc} \circ & \circ & \bullet & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \bullet & \circ & \circ & \circ & \bullet \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \bullet & \circ & \circ \end{array}\right. \\ & \left. +8 \begin{array}{ccccc} \circ & \circ & \circ & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \circ & \circ & \circ \end{array} - 8 \begin{array}{ccccc} \circ & \bullet & \circ & \bullet & \circ \\ \bullet & \circ & \circ & \circ & \bullet \\ \circ & \circ & \circ & \circ & \circ \\ \bullet & \circ & \circ & \circ & \bullet \\ \circ & \bullet & \circ & \bullet & \circ \end{array} - 1 \begin{array}{ccccc} \bullet & \circ & \circ & \circ & \bullet \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \bullet & \circ & \circ & \circ & \bullet \end{array}\right), \qquad (17) \\ \mathrm{Prec}(x_{ij}|\mathbf{x}_{-ij}) =& 13\kappa. \qquad (18)\end{aligned}$$

Several interesting properties of the above IGMRFs can now be stated. For a first order IGMRF, the $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ term is invariant to the addition of a constant to $\mathbf{x}$. This means that realizations from the density will tend to favor piecewise constancy, since no ‘penalty’ is incurred for constant regions, in the sense that only jumps in values will contribute to the $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ term. Similarly, for the second order IGMRF, piecewise linearity is favored, since planar regions do not contribute to the $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ term, and penalty is incurred only for jumps in first derivatives. It must be noted that, strictly speaking, both the IGMRFs we have considered here are of order one when defined on the torus, since the rank of their precision matrices is one in both cases. *Locally*, though, they retain their desirable properties. For the ‘second order’ case, this means that no penalty is incurred for planar regions, only for discontinuities in gradient.

By specifying the IGMRFs by way of normal increments, it is possible to gain some intuition about the meaning of κ as a smoothing parameter, and to relate the common way of defining GMRFs through their conditional means and precisions to a generative model. From equations (8) and (11) it is clear that large values of κ correspond to a small variance in the increments, of first and second order respectively, resulting in smoother curves. It must be noted, however, that the basic *structure* of the realizations from these IGMRFs does not depend on κ, and merely scale is affected. With our assumption of noise of unit variance, though, this scale is crucially important in the calculation of the posterior mean, leading to smoother estimates for increasing values of κ. Implicitly, increasing κ values corresponds to decreasing signal to noise ratio, so the posterior estimates will put increased weight on the model as κ increases.

A sample from the proper part of $\pi(\mathbf{x}|\kappa)$ with $\kappa = 1$ for both the first and second order case is shown in Figure 1. For both samples the same random number generator seed was used, and the images are rescaled to cover the same grayscale range. The sample from the second order IGMRF appears much smoother than the first order IGMRF, and for the purpose of modeling relatively smooth natural images, the second order IGMRF is usually preferable. This is also default in the software.



(a) First order IGMRF

(b) Second order IGMRF

Figure 1: (a) and (b) are samples from a first and second order IGMRF, respectively, demonstrating the relative smoothness of the second order IGMRF compared to the first order.

3. Computational aspects

In this paper a scale-space approach to finding significant features in images is developed. This, naturally, involves calculation of posterior means and covariances over a wide range of scales, the scale in our case parameterized by κ. In general this is an extremely computationally expensive task, though there exist efficient programming schemes exploiting the *sparsity* of the precision matrix, such as GMRFlib [21]. In this paper we use an approach proposed in [7], and exploit the block-circulant *structure* of the precision matrix rather than the sparsity, for highly efficient calculation of the relevant posterior means and covariances.

3.1. Block-circulant matrices

Block-circulant matrices have many interesting properties, the most interesting for our purposes being that many typical matrix operations can be performed using the two dimensional (inverse) discrete Fourier transform, (I)DFT2, on the *base* of the matrices. The base of a block-circulant $Nn \times Nn$ precision matrix is an $n \times N$ matrix, so the computational advantage is substantial.

For reference, some of the most important results for block-circulant matrices, along with a definition, will now be stated, and the interested reader can find a detailed account in [7].

Definition 1. (Circulant matrix) *An $n \times n$ matrix $\mathbf{C}$ is circulant iff it has the form*

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \cdots & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \cdots & c_{n-3} \\ \vdots & \vdots & \vdots & & \vdots \\ c_1 & c_2 & c_3 & \cdots & c_0 \end{bmatrix}, \tag{19}$$

for some vector $\mathbf{c} = (c_0, c_1, \dots, c_{n-1})^T$. We call $\mathbf{c}$ the base *of $\mathbf{C}$.*

Definition 2. (Block-circulant matrix) *An $Nn \times Nn$ matrix $\mathbf{C}$ is block-circulant with $N \times N$ blocks iff it can be written as*

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_{N-1} \\ \mathbf{C}_{N-1} & \mathbf{C}_0 & \mathbf{C}_1 & \cdots & \mathbf{C}_{N-2} \\ \mathbf{C}_{N-2} & \mathbf{C}_{N-1} & \mathbf{C}_0 & \cdots & \mathbf{C}_{N-3} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{C}_3 & \cdots & \mathbf{C}_0 \end{bmatrix} = (\mathbf{C}_{j-i \mod N}), \tag{20}$$

where $\mathbf{C}_i$ *is a circulant* $n \times n$ *matrix with base* $\mathbf{c}_i$*. The base of* $\mathbf{C}$ *is the* $n \times N$ *matrix*

$$\mathbf{c} = (\mathbf{c}_0\, \mathbf{c}_1\, \ldots\, \mathbf{c}_{N-1}).$$

Let $\mathbf{C}$ be a block-circulant matrix where $\mathbf{c} = \mathrm{base}(\mathbf{C})$ is its base, and let $\mathbf{v}$ be any $n \times N$ matrix and $\mathrm{vec}(\mathbf{v})$ the vector that results from stacking the columns one above the other. The matrix-vector product $\mathrm{vec}(\mathbf{u}) = \mathbf{C}\,\mathrm{vec}(\mathbf{v})$ can then, with $\odot$, $\oslash$ and $\owedge$ denoting elementwise multiplication, division and power respectively, be computed as

$$\mathbf{u} = \mathrm{DFT2}(\mathrm{DFT2}(\mathbf{c}) \odot \mathrm{IDFT2}(\mathbf{v})). \tag{21}$$

The product of two block-circulant matrices $\mathbf{C}$ and $\mathbf{D}$, is a block-circulant matrix with base

$$\mathrm{base}(\mathbf{CD}) = \mathrm{IDFT2}(\mathrm{DFT2}(\mathbf{c}) \odot \mathrm{DFT2}(\mathbf{d})), \tag{22}$$

and the inverse of $\mathbf{C}$ has base

$$\mathrm{base}(\mathbf{C}^{-1}) = \mathrm{IDFT2}(\mathrm{DFT2}(\mathbf{c}) \owedge (-1)). \tag{23}$$

3.2. Calculation of posterior mean and covariance

With the help of equations (21) through (23) we can now efficiently calculate the matrix form of the posterior mean from equation (5) and the base of the covariance matrix from equation (6). Letting $\mathbf{1}_{ij}$ denote the matrix with entry (i, j) equal to 1 and the rest equaling zero, we find the posterior mean as

$$\mathbf{X}_\kappa = \mathrm{E}(\mathbf{X}|\mathbf{Y}) = \mathrm{DFT2}(\mathrm{IDFT2}(\mathbf{Y}) \oslash \mathrm{DFT2}(\mathbf{q}_\kappa + \mathbf{1}_{11})). \tag{24}$$

The base of the posterior covariance matrix, $\mathbf{\Sigma}_\kappa$, can be found similarly:

$$\mathbf{\Sigma}_\kappa = \mathrm{IDFT2}(\mathrm{DFT2}(\mathbf{q}_\kappa + \mathbf{1}_{11}) \owedge (-1)). \tag{25}$$

Since we are interested in calculating these quantities for a wide range of values of κ, it is interesting to note that the Fourier transform actually only needs to be performed once on the base of the posterior precision $(\mathbf{q}_\kappa + \mathbf{1}_{11})$. With obvious notation, we may write $\mathbf{q}_\kappa + \mathbf{1}_{11} = \kappa\mathbf{q} + \mathbf{1}_{11}$, where $\mathbf{q}$ is the matrix from equations (10) or (15). Since DFT2 is a linear operator, we can calculate this as

$$\mathrm{DFT2}(\mathbf{q}_\kappa + \mathbf{1}_{11}) = \kappa \mathrm{DFT2}(\mathbf{q}) + \mathbf{1},$$

where $\mathbf{1}$ is a matrix consisting only of ones. Comparing with the computational cost of a naïve direct inversion of the posterior precision matrix at each scale ($\mathcal{O}(N^3)$ for an image with N pixels), the computational advantages of using the DFT2 (in practice FFT2) are substantial ($\mathcal{O}(N \log(N))$). Also compared to the techniques exploiting sparsity, the advantages are substantial, since these techniques generally are of cost $\mathcal{O}(N^{3/2})$.

3.3. Padding the image

Some preprocessing of the image must be done before we apply the DFT2 to it. In addition to standardizing of the image by dividing it by the noise standard deviation, we also subtract the mean of the border values. Since computation of the posterior mean relies on the DFT, any discontinuities in the image will give ringing artifacts as a consequence of Gibbs phenomenon. To minimize these effects at the borders, we pad the image with a Gaussian drop-off before joining edges to make the torus. Specifically, we use the border value as the mode of a Gaussian function, and let the padded values approach zero by way of the Gaussian function. Alternatively, the image can be zero-padded before applying the DFT. Both alternatives are implemented in the software.

3.4. Effective sample size and the range of κ

The technique proposed in this article is Bayesian, in contrast the frequentist S^3 method in [13], where the smoothed estimates $\hat{\mathbf{X}}_h$ are calculated as

$$\hat{\mathbf{X}}_h = \mathbf{Y} * \mathbf{K}_h, \tag{26}$$

$\mathbf{K}_h$ being a Gaussian kernel with bandwidth h, and $*$ denoting convolution. The Bayesian smoothed estimate from equation (24) may, however, using the well known connection between convolution in the spatial domain and multiplication in the Fourier domain, be written as

$$\begin{aligned}\mathbf{X}_\kappa &= \mathrm{DFT2}(\mathrm{IDFT2}(\mathbf{Y}) \oslash \mathrm{DFT2}(\kappa\mathbf{q} + \mathbf{1}_{11})) \\ &= \mathbf{Y} * \underbrace{\mathrm{DFT2}(\mathrm{DFT2}(\kappa\mathbf{q} + \mathbf{1}_{11}) \owedge (-1))}_{\mathbf{K}_\kappa}.\end{aligned} \tag{27}$$

Comparing the two, it is clear that calculating the smoothed estimates from this Bayesian methodology may be put into the framework of kernel smoothing, where the kernel in this case is given as $\mathbf{K}_\kappa$ in equation (27). This allows us to incorporate the useful concept of 'Effective sample size', a measure of the number of points inside each kernel window, into the current Bayesian setting. In [13] this is defined as the matrix $\underline{\text{ESS}} = (\mathbf{K}_h * \mathbf{1})/K_h(0,0)$ and the average ESS is the average of these values over the entire image. Since we give no special treatment to the image boundaries, in our case the ESS is uniform throughout the image, and the convolution reduces to calculating the volume under the kernel. Recognizing that for real images the DFT2 and IDFT2 are simply scaled versions of each other we may, comparing $\mathbf{K}_\kappa$ from equation (27) and the expression for the posterior covariance $\mathbf{\Sigma}_\kappa$ from equation (25), write

$$\text{ESS} = 1/\Sigma_\kappa(1,1), \tag{28}$$

since $\mathbf{\Sigma}_\kappa$ always sums to one. This allows us to define a sensible range for κ, and following [17] the lower bound of κ is chosen such that $\text{ESS}(\kappa_{\min}) \approx 2$. The upper bound of κ is chosen such that a circle containing ESS points should not have a diameter exceeding one tenth of the smallest dimension, since the toroidal graph introduces noticeably detrimental effects at higher levels. Thus, we choose $\kappa_{\max}$ such that $\text{ESS}(\kappa_{\max}) \approx \min(m,n)^2\pi/400$, m and n being the number of rows and columns in the image. The values of κ from $\kappa_{\min}$ to $\kappa_{\max}$ are linear in a log scale, since we desire the smoothness from one scale to the next to increase evenly. Note that the ESS is found with practically no extra computational cost, since the base of the posterior covariance matrix already is calculated at all scales.

4. Significance of features

The calculations in this section follow along the lines of [13] and [14]. There the variances of the relevant quantities had to be estimated and asymptotic results were used to arrive at analytically tractable expressions for the relevant quantiles. In our approach we have calculated the posterior covariance matrix, thus avoiding the need for asymptotics.

4.1. Significant gradient

Denoting the underlying image as f, the gradient magnitude at a given pixel location is

$$G(f) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}, \tag{29}$$

where x and y indicates the vertical and horizontal directions, respectively. The estimate of the gradient magnitude is:

$$G(\tilde{f}_\kappa(i,j)) = [(\tilde{f}_{\kappa,x}(i,j))^2 + (\tilde{f}_{\kappa,y}(i,j))^2]^{1/2}, \tag{30}$$

where the partial derivative estimates are in the form of the forward differences

$$\begin{aligned} \tilde{f}_{\kappa,x}(i,j) &= X_\kappa(i+1,j) - X_\kappa(i,j), \\ \tilde{f}_{\kappa,y}(i,j) &= X_\kappa(i,j+1) - X_\kappa(i,j), \end{aligned} \tag{31}$$

where X_κ is the posterior mean for a given value of κ. We are looking for the pixels where the posterior probability of the gradient being zero is below some threshold α. The posterior distribution of the gradient is

$$\begin{bmatrix} f_{\kappa,x} \\ f_{\kappa,y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \tilde{f}_{\kappa,x} \\ \tilde{f}_{\kappa,y} \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12}^2 \\ \sigma_{21}^2 & \sigma_2^2 \end{bmatrix} \right). \tag{32}$$

Since we have calculated the posterior covariance matrix (with base Σ_κ), the terms σ_1^2 and σ_2^2 are easily derived as:

$$\begin{aligned} \sigma_1^2 &= \mathrm{Var}(\tilde{f}_{\kappa,x}(i,j)) = \mathrm{Var}(X_\kappa(i+1,j) - X_\kappa(i,j)) \\ &= \mathrm{Var}(X_\kappa(i+1,j)) - 2\mathrm{Cov}(X_\kappa(i+1,j), X_\kappa(i,j)) + \mathrm{Var}(X_\kappa(i,j)) \\ &= 2(\Sigma_\kappa(1,1) - \Sigma_\kappa(2,1)) = \sigma_2^2. \end{aligned} \tag{33}$$

Similarly,

$$\sigma_{12}^2 = \Sigma_\kappa(1,1) + \Sigma_\kappa(2,2) - 2\Sigma_\kappa(2,1) = \sigma_{21}^2. \tag{34}$$

Our decision rule amounts to deeming as significant those pixels that do not contain the origin in the α confidence ellipse of the posterior distribution of the gradient. Geometrical considerations make this equivalent to choosing as significant those pixels for which $[\tilde{f}_{\kappa,x}\tilde{f}_{\kappa,y}]^T$ falls outside the α confidence ellipse of a zero mean random variable with the same covariance matrix as $[f_{\kappa,x} f_{\kappa,y}]^T$. Since

$$\begin{bmatrix} f_{\kappa,x} & f_{\kappa,y} \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \sigma_{12}^2 \\ \sigma_{12}^2 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} f_{\kappa,x} \\ f_{\kappa,y} \end{bmatrix} \sim \chi_2^2, \tag{35}$$

a gradient is considered significant when

$$\frac{1}{\sigma_1^4 - \sigma_{12}^4} \left((\tilde{f}_{\kappa,x}^2 + \tilde{f}_{\kappa,y}^2)\sigma_1^2 - 2\tilde{f}_{\kappa,x}\tilde{f}_{\kappa,y}\sigma_{12}^2 \right) > q_{\chi_2^2}(\alpha), \tag{36}$$

where $q_{\chi_2^2}(\alpha)$ is the α-quantile of the χ_2^2 distribution.

4.2. Significant curvature

The test determining the statistical significance of curvature is based on the eigenvalues of the Hessian matrix

$$\mathbf{H}(f) = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix}, \tag{37}$$

where $f_{xx} = (\partial/\partial x)^2 f$, $f_{yy} = (\partial/\partial y)^2 f$, and $f_{xy} = (\partial/\partial y)(\partial/\partial x) f$. The eigenvalues $\lambda_\pm$ of the symmetric 2 by 2 matrix $\mathbf{H}(f)$ are calculated as

$$\lambda_\pm = \left[(f_{xx} + f_{yy}) \pm \sqrt{(f_{xx} - f_{yy})^2 + 4f_{xy}^2} \right] \Big/ 2, \tag{38}$$

where of course the partial derivatives must be replaced by their estimates. The curvature is deemed significant when at least one of λ_+ or λ_- are statistically significantly different from zero. Hence, we wish to find

$$T = \max\{|\lambda_+|, |\lambda_-|\}, \tag{39}$$

which are significantly different from zero. We have that

$$\begin{bmatrix} f_{\kappa,xx} \\ f_{\kappa,xy} \\ f_{\kappa,yy} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \tilde{f}_{\kappa,xx} \\ \tilde{f}_{\kappa,xy} \\ \tilde{f}_{\kappa,yy} \end{bmatrix}, \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \sigma_{13}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \sigma_{23}^2 \\ \sigma_{13}^2 & \sigma_{23}^2 & \sigma_{33}^2 \end{bmatrix} \right), \tag{40}$$

with

$$\begin{aligned} \tilde{f}_{\kappa,xx} &= \hat{X}_\kappa(i+1,j) - 2\hat{X}_\kappa(i,j) + \hat{X}_\beta(i-1,j), \\ \tilde{f}_{\kappa,yy} &= \hat{X}_\kappa(i,j+1) - 2\hat{X}_\kappa(i,j) + \hat{X}_\beta(i,j-1), \\ \tilde{f}_{\kappa,xy} &= \tilde{f}_{\kappa,yx} = \hat{X}_\kappa(i+1,j+1) - \hat{X}_\kappa(i,j+1) - \hat{X}_\kappa(i+1,j) + \hat{X}_\kappa(i,j), \end{aligned} \tag{41}$$

and

$$\begin{aligned}
\mathrm{Cov}(\tilde{f}_{\kappa,xx}) &= 2(3\Sigma_\kappa(1,1) + \Sigma_\kappa(3,1) - 4\Sigma_\kappa(2,1)) \\
&= \mathrm{Cov}(\tilde{f}_{\kappa,yy}) = \sigma_{11}^2 = \sigma_{33}^2, \\
\mathrm{Cov}(\tilde{f}_{\kappa,xy}) &= 4(\Sigma_\kappa(1,1) - 2\Sigma_\kappa(2,1) + \Sigma_\kappa(2,2)) = \sigma_{22}^2, \\
\mathrm{Cov}(\tilde{f}_{\kappa,xx}, \tilde{f}_{\kappa,yy}) &= \Sigma_\kappa(1,1) + \Sigma_\kappa(3,3) + 2\Sigma_\kappa(3,1) + 4\Sigma_\kappa(2,2) \\
&\quad - 2\Sigma_\kappa(3,2) - 4\Sigma_\kappa(2,1) = \sigma_{13}^2, \\
\mathrm{Cov}(\tilde{f}_{\kappa,xx}, \tilde{f}_{\kappa,xy}) &= 3\Sigma_\kappa(1,1) + 4\Sigma_\kappa(2,2) + \Sigma_\kappa(3,1) - \Sigma_\kappa(3,2) - 7\Sigma_\kappa(2,1) \\
&= \mathrm{Cov}(\tilde{f}_{\kappa,yy}, \tilde{f}_{\kappa,xy}) = \sigma_{12}^2 = \sigma_{23}^2. \qquad (42)
\end{aligned}$$

A closed form expression for the density of T is not available, so the quantile must be empirically established through Monte Carlo simulations. In our specific implementation, we generate 1,000,000 samples of $\mathbf{Z} \sim \mathcal{N}\left([0,0,0]^T, \mathbf{I}\right)$ and calculate

$$\begin{bmatrix} f_{xx} \\ f_{xy} \\ f_{yy} \end{bmatrix} = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \sigma_{13}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 & \sigma_{23}^2 \\ \sigma_{13}^2 & \sigma_{23}^2 & \sigma_{33}^2 \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix}. \qquad (43)$$

T is then found through equations (38) and (39), and the empirical α-quantile, $\hat{q}_T$, is found. The estimate of T, $\hat{T}$, is calculated for all pixels. Those pixels for which $\hat{T} > \hat{q}_T$ are classified as significant.

In the following examples, we tailor the curvature detection to two specific problems in dermascopic imaging, where the relevant significance are shown as blue, as we do not differentiate between the different types of significance as opposed to the examples in [13].

5. Examples

5.1. Detecting hairs in dermascopic images of skin lesions

In the last ten years much effort has been expended on the development of computer aided diagnostic tools for detection of malignant melanoma (see e.g. [22] or [23] for an overview). Among the many problems such a system must be able to cope with, is the fact that often the images of skin lesions will contain hair. Since there is no reason to believe that the hairs contain any diagnostic information, it is crucial that the system can find them so that they do not contribute to the final diagnosis. Several algorithms exist for handling this problem (see e.g. the much cited [24]), and here we propose that

Bayesian multiscale analysis can be a useful tool for the problem at hand. The idea is that hairs will be significantly darker than the surrounding skin, and in most cases its color will be different than that of the lesion. BMulA, which seeks to find significant gradient and curvature on several scales, should therefore be useful, since it is to be expected that hairs of different color and thickness will show up at different scales. We use as our model the second-order IGMRF, as it should be well suited for images where the local statistics are fairly constant throughout the image, and the image is relatively smooth.

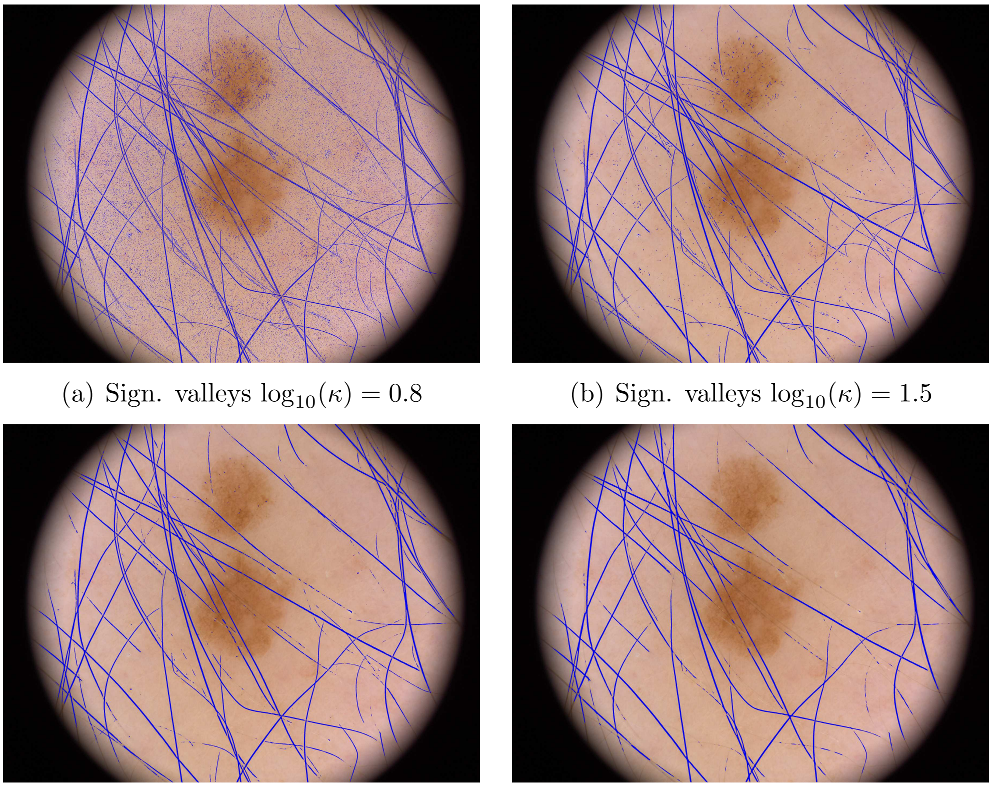
5.1.1. The method

Since the images originally are three channel RGB color images, a choice must be made on how to extract a grayscale image. We have chosen to use the first principal component of the R, G and B channels of the linear (non gamma-corrected) image, thereby retaining the maximum of variance for a linear transform. The resulting grayscale image is in our example a 3326 by 4432 matrix.

The images under consideration have a black frame to the left and right of a illuminated circle where, ideally, all pixel values should be zero (the black areas in Figure 2). Consequently the sample variance in these regions are explained by the image noise, and it is the sample standard deviation calculated in these regions that is used as estimate of noise standard deviation.

Since hairs, being for the most part dark and long, will appear as long valleys in the grayscale image, it is natural to use the version of BMulA which locates significant curvature. Furthermore, since the method can distinguish between peaks, valleys, saddle points, holes, and ridges, we may restrict our attention only to finding significant valleys. While it would be possible to also use gradient information, gradients in the image under consideration would be flagged significant for almost all pixels, since gradients are a relatively simple feature to detect and the noise level is low. It is of course possible to use the technique, utilizing a σ^2 much larger than the actual noise level, and using an extremely low significance level α. Such an ad hoc procedure could easily be incorporated into the hair detection algorithm, but the results using only significant valleys are of such high quality that its use is not merited.

The basic idea of the proposed technique, is that different features will show up at different scales, so it is crucial to arrive at a sensible range of scales. While one could use the entire range calculated from $\text{ESS}(\kappa_{min}) \approx 2$, and $\text{ESS}(\kappa_{max}) \approx \min(m, n)^2\pi/400$, in this specific application this turns out to be excessive since, at the lowest levels of smoothing, far to much is flagged as significant and, at the highest levels, almost nothing is. Through trial and error on several representative images, we have found that a range of κ from $10^{0.8}$ to 10^3 is sufficient to capture both the fine structures represented by fine hairs, while at the same time assuring robustness through going to a level of smoothing where the spurious artifacts seen in the lower levels disappear. In Figure 2 the significant valleys for four different scales are shown, demonstrating how at the lowest scales we capture the finest hairs, but also find significant valleys which for the purpose of hair detection must be considered unwanted artifacts. At the higher levels the artifacts disappear, along with the fine hairs.



(a) Sign. valleys $\log_{10}(\kappa) = 0.8$

(b) Sign. valleys $\log_{10}(\kappa) = 1.5$

(c) Sign. valleys $\log_{10}(\kappa) = 2.3$

(d) Sign. valleys $\log_{10}(\kappa) = 3.0$

Figure 2: Subfigures (a) through (d) show the valleys that are flagged as significant at the indicated scales. At the smallest scales it is evident that too much is flagged as significant, while at the highest scales the fine detail is lost.

The final step in the hair detection algorithm is to define as candidate hair pixels those pixels which have been classified as such in 90 % of the scales considered, and to remove the smallest objects where the major axis of an ellipse fit to the object is less than 50 pixels.

5.1.2. Results

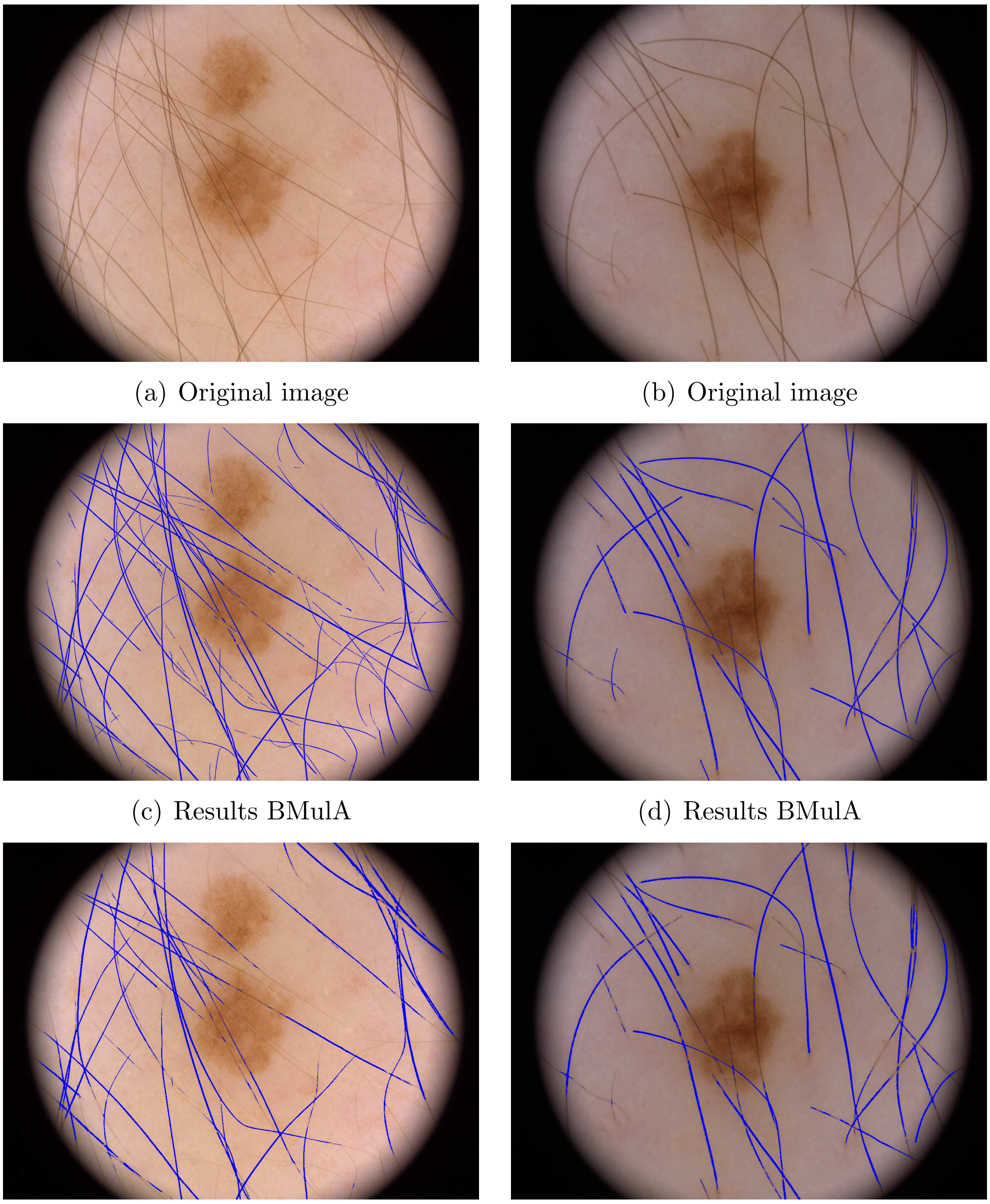
The results of applying the method to two dermascopic images is shown in Figure 3. For comparison, the results using Dullrazor are shown in the same figure. From these experiments it seems that our technique is superior in detecting the fine hairs in the image, while the performance is comparable for the dark and thick hairs.

These results are preliminary in nature, and there likely remains much room for improvement. The initial results, however, are promising and demonstrate the usefulness of BMulA in a very practical problem. The computational efficiency of BMulA is the reason that this methodology can be used on images of the dimensions in this example. The toroidal assumption underlying the method does not affect the statistical inference in the interesting parts of the image, even at the highest scales ($\kappa = 10^3$).

The comparison with Dullrazor is included since the Dullrazor software is much referenced within the field, and it is readily available online (www.dermweb.com/dull_razor/). To make this superficial comparison as fair as possible we have, in addition to applying the software on the full size images, applied the software on resized images (from 10 % to 90 % reduction), and chosen by visual inspection the best results. This is done since the results from using Dullrazor also depend on the scale/dimensions of the image. Still, we do not claim this comparison to be exhaustive and, since there is no established ‘gold standard’ in evaluating hair detection algorithms, we cannot within the confines of this limited comparison quantify the performance of the algorithms other than by visual inspection.

5.2. Detecting dots/globules in skin lesions

Dermatologists utilize a number of rules for determining whether they suspect a skin lesion might be a melanoma. In the well known ABCD rule for dermoscopy, one of several sets of heuristics dermatologists use to guide their decision [25], features such as asymmetry, border, colors, and dermascopic structure are given a score, which are weighted to produce a diagnosis. A computer aided diagnostic system for detection of malignant melanoma should be able detect many of these features, and here we demonstrate how



(a) Original image

(b) Original image

(c) Results BMulA

(d) Results BMulA

(e) Results Dullrazor

(f) Results Dullrazor

Figure 3: Subfigures (a) and (b) show two dermascopic images of skin lesions. The results of using BMulA for the task of hair detection on the two images is shown in sub figures (c) and (d). (e) and (f) shows the results using Dullrazor on the same images.

BMulA can be useful in detecting one of the dermascopic structures, namely dots/globules.

Dots or globules are small, roughly circular, dark areas in the lesion. As with many of the features a dermatologist looks for, it is a somewhat subjective exercise to determine their presence. The notion of scale is, of course, crucial. How large or small, and how dark or fair the area can be while still being classified as a dot or globule, is to a large extent a judgement call for the dermatologist. Accordingly, we have set the parameters in the algorithm in cooperation with a trained dermatologist, so as to best detect what he considered to be dots in our training set of eight images. As with the previous example, the results are preliminary, and are presented mainly to demonstrate the potential of the algorithm.

5.2.1. The method

Since the presence of dots is of interest only in and immediately around the lesion, we have chosen to segment the image into lesion and skin parts before applying BMulA. The segmentation algorithm works by applying watershedding [26] to the negative smoothed 2d histogram of the first two principal components of the image in the CIElab color space, thus segmenting the histogram into regions. The pixels are then classified by which region in the segmented histogram they belong to, an approach similar to that proposed in [27].

Having segmented the image, we proceed to create a grayscale image from the full RGB image. Again we choose the first principal component of the RGB components, but now we only use the lesion pixels in calculating the principal component scores, thereby assuring the maximal explained variance in the actual region of interest. As in the previous example, we use the black frame to estimate the noise variance.

Having done the necessary preprocessing, the algorithm proceeds in the same way as in the previous example, with the feature of interest now being holes. We choose the range of scales such that both the small and the large dots, as determined by the dermatologist, in eight test images are captured by the algorithm. Some post-processing needs to be performed also here. In cooperation with the dermatologist, we have arrived at the heuristic that those pixels which have been classified as belonging to a dot in 75 % of the scales considered are dots, so long as they consist of more than 15 pixels. In this example twenty κ-values were used, ranging from 10^0 to 10^1.

5.2.2. Results

The results of applying the algorithm to two images not in the test set is shown in Figure 4, and corresponds nicely to the dermatologist's assessment of what he considers dots in the lesions. There will always be ambiguities in evaluating such a procedure, and again we must rely on visual inspection. For these two images (1025 by 808 and 519 by 555 pixels) the computation time in Matlab was approximately 0.42 and 0.32 seconds per κ value respectively, demonstrating the efficiency of the procedure. In fact, the Monte Carlo simulations for determining the appropriate quantile constitute more than half the computation time.

Acknowledgements

We thank M.D. Thomas Schopf for sharing his expertise in dermatology, and Dr. med. Herbert Kirchesch for supplying us with the dermoscopic skin lesion images used in the examples. We would also like to thank Dr. Maciel Zortea for doing the hair detection with the Dullrazor software. The research was funded by the Norwegian Research Council and the Centre for Research Driven Innovation, Tromsø Telemedicine Laboratory (TTL).

References

[1] T. Lindeberg, Scale-Space Theory in Computer Vision, Kluwer/Springer, 1994.

[2] J. Besag, Spatial interaction and the statistical analysis of lattice systems, Journal of the Royal Statistical Society. Series B 36 (2) (1974) 192–236.

[3] I. L. Dryden, M. R. Scarr, C. C. Taylor, Bayesian texture segmentation of weed and crop images using reversible jump markov chain monte carlo methods, Journal of the Royal Statistical Society: Series C 52 (1) (2003) 31–50.

[4] J. L. Marroquin, F. A. Velasco, M. Rivera, M. Nakamura, Gauss-markov measure field models for low-level vision, IEEE Transactions on Pattern Analysis 23 (4) (2001) 337–348.

[5] H. Rue, M. A. Hurn, Bayesian object identification, Biometrika 86 (3) (1999) 649,660.

[6] C. Serranoa, B. Acha, Pattern analysis of dermoscopic images based on markov random fields, Pattern Recognition 42 (6) (2009) 1052–1057.

[7] H. Rue, L. Held, Gaussian Markov Random Fields: Theory and Applications, Chapman & Hall/CRC, 2005.

[8] J. Polzehl, V. Spokoiny, Adaptive weights smoothing with applications to image restoration, Journal of the Royal Statistical Society. Series B (Statistical Methodology) 62 (2) (2000) 335–354.

[9] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7) (1990) 629–639.

[10] P. Qiu, Discontinuous regression surfaces fitting, The Annals of Statistics 26 (6) (1998) 2218–2245.

[11] I. Gijbels, A. Lambert, P. Qiu, Edge-preserving image denoising and estimation of discontinuous surfaces, IEEE transactions on pattern analysis and machine intelligence (2006) 1075–1087.

[12] P. Qiu, P. Mukherjee, Edge structure preserving image denoising, Signal Processing 90 (10) (2010) 2851–2862.

[13] F. Godtliebsen, J. S. Marron, P. Chaudhuri, Statistical significance of features in digital images, Image Vision Computing 22 (13) (2004) 1093–1104.

[14] F. Godtliebsen, J. S. Marron, P. Chaudhuri, Significance in scale space for bivariate density estimation, Journal of Computational and Graphical Statistics 11 (1) (2002) 1–21.

[15] P. Chaudhuri, J. S. Marron, Sizer for exploration of structures in curves, Journal of the American Statistical Association 94 (1997) 807–823.

[16] F. Godtliebsen, T. A. Øigård, A visual display device for significant features in complicated signals, Computational Statistics & Data Analysis 48 (2) (2005) 317–343.

[17] T. A. Øigård, H. Rue, F. Godtliebsen, Bayesian multiscale analysis for time series data, Computational Statistics & Data Analysis 51 (3) (2006) 1719–1730.

[18] P. Erästö, L. Holmström, Bayesian multiscale smoothing for making inferences about features in scatterplots, Journal of Computational and Graphical Statistics 14 (3) (2005) 569–589.

[19] L. Holmström, L. Pasanen, R. Furrer, S. Sain, Scale space multiresolution analysis of random signals, Computational Statistics & Data Analysis 55 (2011) 2840–2855.

[20] J. Besag, C. Kooperberg, On conditional and intrinsic autoregressions, Biometrika 82 (4) (1995) 733–746.

[21] H. Rue, T. Follestad, Gmrflib: a c-library for fast and exact simulation of gaussian markov random fields, Statistics report 1.

[22] B. Rosado, S. Menzies, A. Harbauer, H. Pehamberger, K. Wolff, M. Binder, H. Kittler, Accuracy of computer diagnosis of melanoma: A quantitative meta-analysis, Archives of Dermatology 139 (3) (2003) 361–367.

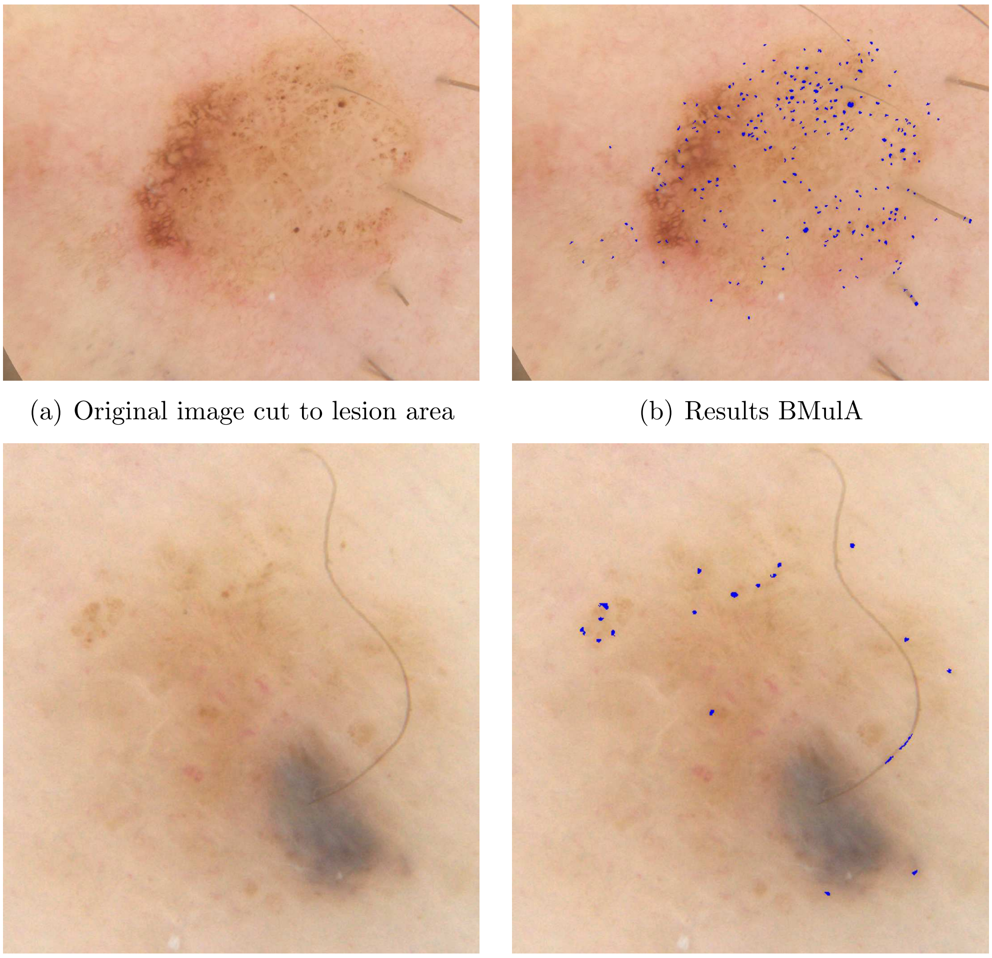
[23] S. M. Rajpara, A. P. Botello, J. Townend, A. D. Ormerod, Systematic review of dermoscopy and digital dermoscopy/ artificial intelligence for the diagnosis of melanoma, British Journal of Dermatology 161 (2009) 591–604.

[24] T. Lee, Dullrazor: A software approach to hair removal from images, Computers in Biology and Medicine 27 (6) (1997) 533–543.

[25] H. Kittler, H. Pehamberger, K. Wolff, M. Binder, Diagnostic accuracy of dermoscopy, The Lancet Oncology 3 (3) (2002) 159–165.

[26] F. Meyer, Topographic distance and watershed lines, Signal Processing 38 (1) (1994) 113–125.

[27] L. Shafarenko, M. Petrou, J. Kittler, Histogram based segmentation in a perceptually uniform color space, IEEE Transactions on Image Processing 7 (9) (1998) 1354–1358.



(a) Original image cut to lesion area

(b) Results BMulA

(c) Original image cut to lesion area

(d) Results BMulA

Figure 4: Subfigures (a) and (c) show two dermascopic skin lesion images on which BMulA has been used for the detection of significant dots, and (b) and (d) show the results.