Electronic Transactions on Numerical Analysis.
Volume 40, pp. 476-488, 2013.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

EFFICIENT MCMC-BASED IMAGE DEBLURRING WITH NEUMANN BOUNDARY CONDITIONS*

JOHNATHAN M. BARDSLEY†, MARYLESA HOWARD†, AND JAMES G. NAGY‡

Abstract. The problem of uncertainty quantification (UQ) for inverse problems has become of significant recent interest. However, UQ requires more than the classical methods for computing solutions of inverse problems. In this paper, we take a Bayesian approach for the solution of ill-posed deconvolution problems with a symmetric convolution kernel and Neumann boundary conditions. The prior is modeled as a Gaussian Markov random field (GMRF) with the same boundary conditions and symmetry assumptions. These assumptions yield better results in certain instances and also allow for the use of the discrete cosine transform for fast computations. Moreover, we use a hierarchical model for the noise precision (inverse-variance) and prior precision parameters. This leads to a posterior density function from which we can compute samples using a basic Markov Chain Monte Carlo (MCMC) method. The resulting samples can then be used for both estimation (using, e.g., the sample mean) and uncertainty quantification (using, e.g., histograms, the sample variance, or a movie created from the image samples). We provide a numerical experiment showing that the method is effective, computationally efficient, and that for certain problems, the boundary conditions can yield significantly better results than if a periodic boundary is assumed. The novelty in the work lies in the combination of the MCMC method, Neumann boundary conditions, GMRF priors, and in the use of a movie to visualize uncertainty in the unknown image.

Key words. image deblurring, inverse problems, Bayesian inference, Gaussian Markov random fields, Markov chain Monte Carlo methods, Neumann boundary conditions

AMS subject classifications. 15A29, 62F15, 65F22, 94A08

1. Introduction.

In applications such as astronomy, medicine, physics and biology, digital images are used by scientists and practitioners to record and analyze unique events. Environmental effects and imperfections in the imaging system can cause the recorded images to be degraded by blurring and noise. Unfortunately, it is not always possible to repeat the process used to record the image to obtain "better pictures"; for example, it could be too costly to repeat a particular experiment, or it may not be physically possible to repeat the event that was observed. In such cases, computational post processing techniques, called *image deblurring*, are used to improve the resolution of the image.

Image deblurring is typically modeled as a linear inverse problem. Suppose $x(t)$, $t \in \mathbb{R}^d$, is a function describing the true d-dimensional image; e.g., for a plane image containing pixels, $d = 2$. The image formation process, which includes blurring and noise, is modeled by an integral equation,

$$b(s) = \int_\Omega a(s,t)x(t)dt + \eta(s)\,,$$

where $s, t \in \mathbb{R}^d$, $b(s)$ is a function that represents the observed image, $\eta(s)$ represents additive noise, and $\Omega \subset \mathbb{R}^d$ is the computational domain. The kernel $a(s,t)$ is a function that specifies how the points in the image are distorted, and is therefore called the point spread function (PSF). The inverse problem of image deblurring is: given a and b, compute an approximation of x. If the kernel has the property that $a(s,t) = a(s-t)$, then the PSF is said to be spatially invariant; otherwise it is said to be spatially variant. In the spatially invari-

*Received November 14, 2012. Accepted August 28, 2013. Published online on December 18, 2013. Recommended by Bob Plemmnons.

†Department of Mathematical Sciences, University of Montana, Missoula, MT, 59812-0864. Email: bardsleyj@mso.umt.edu, marylesa.howard@umontana.edu.

‡Department of Mathematics and Computer Science, Emory University, Atlanta, GA, 30322. Email: nagy@mathcs.emory.edu.

ETNA
Kent State University
http://etna.math.kent.edu

ant case, the blurring operation, $\int_\Omega a(s-t)x(t)dt$, is a convolution operation, and thus the corresponding inverse problem of computing an approximation of x from b and a, is called *deconvolution*.

In a realistic problem, images are collected only at discrete points (i.e., *pixels* for 2D images and *voxels* for 3D images), and are also only available in a finite bounded region Ω. It is, therefore, typical to work directly with the discrete linear model,

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta}, \tag{1.1}$$

where $\mathbf{x}$, $\mathbf{b}$ and $\boldsymbol{\eta}$ are real valued $n \times 1$ vectors obtained by discretizing functions x, b, and η, and n is the number of pixels (or voxels) in the discrete image. $\mathbf{A}$ is a real valued $n \times n$ matrix that arises when approximating the integration operation with a quadrature rule, and it usually has structure (e.g., Toeplitz, circulant, Hankel, etc.) that can be exploited in computations.

Our approach for solving the discrete inverse problem (1.1) is statistically motivated. Specifically, we assume that $\boldsymbol{\eta}$ is an $n \times 1$ independent and identically distributed (iid) Gaussian random vector with variance λ^{-1} (λ is known as the *precision*) across all pixels, and that the probability density function for (1.1) is given by

$$p(\mathbf{b}|\mathbf{x},\lambda) \propto \exp\left(-\frac{\lambda}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2\right), \tag{1.2}$$

where '$\propto$' denotes proportionality. However, it is important to note that when attempting to solve inverse problems, the maximizer of the likelihood $L(\mathbf{x}|\mathbf{b},\lambda) = p(\mathbf{b}|\mathbf{x},\lambda)$ with respect to $\mathbf{x}$ is unstable with respect to the noise contained in $\mathbf{b}$. This instability is a characteristic of inverse problems, such as deconvolution, and it has to do with the fact that the forward mapping (convolution) is a compact operator defined on a function space [5]. The standard technique for overcoming such instability is *regularization*, which is treated in detail in several references [5, 7, 8, 10, 20].

In the context of Bayesian statistics, regularization corresponds to the choice of the *prior* probability density function $p(\mathbf{x}|\delta)$, where $\delta > 0$ is a scaling parameter. In our case, we use a *Gaussian Markov random field* (GMRF) to model the prior, which yields

$$p(\mathbf{x}|\delta) \propto \exp\left(-\frac{\delta}{2}\mathbf{x}^T\mathbf{L}\mathbf{x}\right), \tag{1.3}$$

where the *precision matrix* $\delta\mathbf{L}$ is sparse and encodes distributional assumptions regarding the values of x_i conditioned on the values of its neighbors, for all i.

Bayes' Theorem states that the *posterior probability density function* $p(\mathbf{x}|\mathbf{b},\lambda,\delta)$ can be expressed as

$$\begin{aligned} p(\mathbf{x}|\mathbf{b},\lambda,\delta) &\propto p(\mathbf{b}|\mathbf{x},\lambda)p(\mathbf{x}|\delta) \\ &\propto \exp\left(-\frac{\lambda}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|^2 - \frac{\delta}{2}\mathbf{x}^T\mathbf{L}\mathbf{x}\right). \end{aligned} \tag{1.4}$$

Maximizing (1.4) with respect to $\mathbf{x}$ yields the so-called maximum a posteriori (MAP) estimator. By equivalently minimizing the negative-log of (1.4), we see that $\mathbf{L}$ corresponds to the regularization matrix in classical inverse problems, while δ/λ corresponds to the regularization parameter [20].

In this paper we extend the Bayesian formulation (1.4) and compute samples from the resulting posterior density function using the Markov Chain Monte Carlo (MCMC) method

ETNA
Kent State University
http://etna.math.kent.edu

of [1]. Bayes' Law (1.4) is extended by assuming Gamma distributed hyper-priors on λ and δ, i.e.,

$$p(\lambda) \propto \lambda^{\alpha_\lambda - 1} \exp(-\beta_\lambda \lambda), \tag{1.5}$$

$$p(\delta) \propto \delta^{\alpha_\delta - 1} \exp(-\beta_\delta \delta), \tag{1.6}$$

with $\alpha_\lambda = \alpha_\delta = 1$, and $\beta_\lambda = \beta_\delta = 10^{-4}$, which have mean and variance $\alpha/\beta = 10^4$ and $\alpha/\beta^2 = 10^8$, respectively. Note that $\alpha = 1$ yields exponential distributed hyper-priors, however we present the full Gamma hyper-prior here because it is a conjugate distribution and other choices for α and β may be advantageous in other situations. Given the large variance values, the hyper-priors should have a negligible effect on the sampled values for λ and δ.

With (1.2), (1.3), (1.5), and (1.6) in hand, through Bayes' Law the posterior probability density has the form

$$\begin{aligned} p(\mathbf{x}, \lambda, \delta | \mathbf{b}) &\propto p(\mathbf{b}|\mathbf{x}, \lambda) p(\lambda)\, p(\mathbf{x}|\delta) p(\delta) \\ &= \lambda^{n/2+\alpha_\lambda - 1} \delta^{(n-1)/2+\alpha_\delta - 1} \exp\left(-\frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 - \frac{\delta}{2} \mathbf{x}^T \mathbf{L} \mathbf{x} - \beta_\lambda \lambda - \beta_\delta \delta \right). \end{aligned} \tag{1.7}$$

The prior and hyper-priors were chosen to be *conjugate* [6], which guarantees that the full conditional densities have the same form as the corresponding prior/hyper-prior; specifically, note that

$$\mathbf{x}|\lambda, \delta, \mathbf{b} \sim \mathcal{N}\left((\lambda \mathbf{A}^T \mathbf{A} + \delta \mathbf{L})^{-1} \lambda \mathbf{A}^T \mathbf{b}, (\lambda \mathbf{A}^T \mathbf{A} + \delta \mathbf{L})^{-1} \right), \tag{1.8}$$

$$\lambda|\mathbf{x}, \delta, \mathbf{b} \sim \Gamma\left(n/2 + \alpha_\lambda, \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \beta_\lambda \right), \tag{1.9}$$

$$\delta|\mathbf{x}, \lambda, \mathbf{b} \sim \Gamma\left((n-1)/2 + \alpha_\delta, \frac{1}{2}\mathbf{x}^T \mathbf{L} \mathbf{x} + \beta_\delta \right), \tag{1.10}$$

where $\mathcal{N}$ and Γ denote Gaussian and Gamma distributions, respectively.

The power in (1.8)-(1.10) lies in the fact that samples from these three distributions can be computed using standard statistical software, and a Gibbsian approach can be applied to (1.8)-(1.10) yielding the MCMC method of [1] for sampling from (1.7).

A MCMC Method for Sampling from $p(\mathbf{x}, \delta, \lambda|\mathbf{b})$.

0. Initialize δ_0, and λ_0, and set $k = 0$;
1. Compute $\mathbf{x}^k \sim \mathcal{N}\left((\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{L})^{-1} \lambda_k \mathbf{A}^T \mathbf{b}, (\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{L})^{-1} \right)$;
2. Compute $\lambda_{k+1} \sim \Gamma\left(n/2 + \alpha_\lambda, \frac{1}{2}\|\mathbf{A}\mathbf{x}^k - \mathbf{b}\|^2 + \beta_\lambda \right)$;
3. Compute $\delta_{k+1} \sim \Gamma\left((n-1)/2 + \alpha_\delta, \frac{1}{2}(\mathbf{x}^k)^T \mathbf{L} \mathbf{x}^k + \beta_\delta \right)$;
4. Set $k = k + 1$ and return to Step 1.

What makes this MCMC method interesting in the case of image deblurring is that computing the image samples in Step 1 requires the solution of the large linear system

$$(\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{L}) \mathbf{x}^k = \lambda_k \mathbf{A}^T \mathbf{b} + \mathbf{w}, \quad \text{where} \quad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{L}). \tag{1.11}$$

In this paper, we assume Neumann boundary conditions for the image $\mathbf{x}$ and symmetric PSFs. The matrices $\mathbf{A}$ and $\mathbf{L}$ are then diagonalizable by the discrete cosine transform (DCT), which we will denote by $\mathbf{C}$, making the computation of $\mathbf{x}^k$ in Step 1 extremely efficient. Specifically, if $\mathbf{A} = \mathbf{C}^T \mathbf{\Lambda} \mathbf{C}$, and $\mathbf{L} = \mathbf{C}^T \mathbf{S} \mathbf{C}$, given that $\mathbf{C}$ is an orthogonal matrix we have

$$\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{L} = \mathbf{C}^T (\lambda_k \mathbf{\Lambda}^2 + \delta_k \mathbf{S}) \mathbf{C}.$$

ETNA
Kent State University
http://etna.math.kent.edu

Substituting this into (1.11) yields

$$\begin{aligned}\mathbf{x}^k &= \mathbf{C}^T(\lambda_k\mathbf{\Lambda}^2+\delta_k\mathbf{S})^{-1}\lambda_k\mathbf{\Lambda}\mathbf{C}\mathbf{b}+\mathbf{C}^T(\lambda_k\mathbf{\Lambda}^2+\delta_k\mathbf{S})^{-1}\mathbf{C}\mathbf{w}\\ &= \mathbf{C}^T(\lambda_k\mathbf{\Lambda}^2+\delta_k\mathbf{S})^{-1/2}((\lambda_k\mathbf{\Lambda}^2+\delta_k\mathbf{S})^{-1/2}\lambda_k\mathbf{\Lambda}\mathbf{C}\mathbf{b}+\mathbf{v}),\end{aligned} \tag{1.12}$$

where $\mathbf{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, which follows from the fact that if $\mathbf{M}\in\mathbb{R}^{n\times n}$ and $\mathbf{y}=\mathbf{M}\mathbf{v}$, then $\mathbf{y}\sim\mathcal{N}(\mathbf{0},\mathbf{M}\mathbf{M}^T)$. Since $\mathbf{\Lambda}$ and $\mathbf{S}$ are computed off-line, and because multiplication by $\mathbf{C}$ and $\mathbf{C}^T$ is computed efficiently using a fast DCT algorithm [18], the computation of $\mathbf{x}^k$ via (1.12) is very efficient.

The paper is organized as follows. We begin with a discussion of Neumann boundary conditions for image deblurring problems in Section 2. Then in Section 3, we discuss GRMFs generally, introduce the one that we use, and show that if Neumann boundary conditions are assumed the precision matrix can be diagonalized using the DCT. Finally, in Section 4, we show results of the method and compare it with other standard approaches. Concluding remarks are given in Section 5.

2. The Neumann boundary condition for convolution problems. As has been stated, we are interested in the case in which the image $\mathbf{x}$ is assumed to have Neumann, or reflective, boundary conditions. To illustrate what we mean by this, we begin by considering 1D image deconvolution. In this case, the unknown image $\mathbf{x}=(x_1,\dots,x_n)$ can be extended spatially to create the vector

$$\widetilde{\mathbf{x}}=[x_{-n+1},\dots,x_0,x_1,\dots,x_n,\dots,x_{2n}]^T.$$

The matrix $\mathbf{A}$ is defined in terms of the convolution kernel

$$\mathbf{a}=[a_{-n},a_{-n+1},\dots,a_0,a_1,\dots,a_n]^T,$$

and the noise-free data vector is obtained via discrete convolution:

$$b_i=\sum_{j=i-n}^{i+n}a_{i-j}x_j,\quad \text{for}\quad i=1,\dots,n, \tag{2.1}$$

or in matrix-vector notation,

$$\begin{bmatrix}b_1\\b_2\\\vdots\\b_n\end{bmatrix}=\begin{bmatrix}a_n&\cdots&a_0&\cdots&a_{-n}&&&&\\&a_n&\cdots&a_0&\cdots&a_{-n}&&&\\&&\ddots&\ddots&\ddots&\ddots&\ddots&&\\&&&a_n&\cdots&a_0&\cdots&a_{-n}&\\&&&&a_n&\cdots&a_0&\cdots&a_{-n}\end{bmatrix}\begin{bmatrix}x_{-n+1}\\\vdots\\x_0\\x_1\\\vdots\\x_n\\\vdots\\x_{2n}\end{bmatrix}. \tag{2.2}$$

The key observation here is that the value b_i for i near 1 will depend upon the region of $\widetilde{\mathbf{x}}$ to the left of the computational domain, i.e., $(x_{-n+1},\dots,x_0)$; while for i near n, b_i will depend upon the region of $\widetilde{\mathbf{x}}$ to the right of the computational domain, i.e., $(x_{n+1},\dots,x_{2n})$.

Rather than estimating these extra values by solving the underdetermined system (2.2), the standard approach is to make assumptions about these values based on *a priori* knowledge, or by relating the values to those within the computational domain. These assumptions

ETNA
Kent State University
http://etna.math.kent.edu

are called boundary conditions. For example, a zero (or Dirichlet) boundary condition corresponds to the assumption that $(x_{-n+1},\ldots,x_0) = (x_{n+1},\ldots,x_{2n}) = \mathbf{0}$, which yields a Toeplitz matrix $\mathbf{A}$ [20]; while a periodic boundary condition corresponds to $(x_{-n+1},\ldots,x_0) = (x_1,\ldots,x_n)$ and $(x_{n+1},\ldots,x_{2n}) = (x_1,\ldots,x_n)$, which yields a circulant matrix $\mathbf{A}$ that can be diagonalized by the discrete Fourier transform (DFT) [20].

The Neumann boundary condition corresponds to a reflection of the signal about the boundaries, i.e., $(x_{-n+1},\ldots,x_0) = (x_n,\ldots,x_1)$ and $(x_{n+1},\ldots,x_{2n}) = (x_n,\ldots,x_1)$. In this case, the resulting matrix $\mathbf{A}$ has Toeplitz-plus-Hankel structure, and if the convolution kernel $\mathbf{a}$ is symmetric, i.e., $a_i = a_{-i}$, then $\mathbf{A}$ can be diagonalized by the discrete cosine transform (DCT) [15, Theorem 3.2]. We note that while a Toeplitz matrix is one for which each descending diagonal from left to right is constant, a Hankel matrix is one for which each descending anti-diagonal from right to left is constant.

We started with the 1D example in order to illustrate concepts more simply, but our primary interest is two-dimensional (2D) image deblurring. In this case, $\mathbf{b}$ and $\mathbf{x}$ are obtained by column-stacking the $N \times N$ arrays $\mathbf{B}$ and $\mathbf{X}$, which we denote as $\mathbf{b} = \text{vec}(\mathbf{B})$ and $\mathbf{x} = \text{vec}(\mathbf{X})$; and $\mathbf{A}$ is defined in terms of the $N \times N$ convolution kernel $\mathbf{a} = \{a_{ij}\}_{i,j=-N}^{N}$, with some assumed boundary condition. The noise-free $N \times N$ data array then satisfies the 2D discrete convolution equation

$$b_{r,s} = \sum_{i=r-N}^{r+N} \sum_{j=s-N}^{s+N} a_{r-i,s-j} x_{ij}, \quad \text{for} \quad r,s = 1,\ldots,N. \tag{2.3}$$

In the 2D case, for zero and periodic boundary conditions, the extensions of $\mathbf{X}$ are represented, respectively, by

$$\begin{matrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{matrix}, \quad \text{and} \quad \begin{matrix} \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \\ \mathbf{X} & \mathbf{X} & \mathbf{X} \end{matrix}.$$

In both cases the central $\mathbf{X}$ corresponds to the unknowns within the computational domain (i.e., the field of view). The assumption of zero boundary conditions results in a matrix $\mathbf{A}$ that is block Toeplitz with Toeplitz blocks [20], while periodic boundary conditions result in a matrix $\mathbf{A}$ that is block circulant with circulant blocks and can be diagonalized by the 2D-DFT [20].

In instances where the zero and/or periodic extensions are poor approximations of reality, unnatural artifacts in reconstructions can result. This is particularly the case when $\mathbf{X}$ and $\mathbf{B}$ contain regions of relative high and variable intensity near the boundaries of the computational domain. In such instances, the reflective extension of $\mathbf{X}$, corresponding to Neumann boundary conditions, works significantly better. It is represented by

$$\begin{matrix} \mathbf{X}_{vh} & \mathbf{X}_{h} & \mathbf{X}_{vh} \\ \mathbf{X}_{v} & \mathbf{X} & \mathbf{X}_{v} \\ \mathbf{X}_{vh} & \mathbf{X}_{h} & \mathbf{X}_{vh} \end{matrix},$$

where $\mathbf{X}_v$ is the image that results from flipping $\mathbf{X}$ across its central vertical axis; $\mathbf{X}_h$ is the image that results from flipping $\mathbf{X}$ across its central horizontal axis; and $\mathbf{X}_{vh}$ is the image that results from flipping $\mathbf{X}$ across its vertical then horizontal axes. The Neumann boundary condition leads to a matrix $\mathbf{A}$ that is block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB), and provided the kernel $\mathbf{a}$ is symmetric, i.e.,

$$a_{i,j} = a_{-i,j} = a_{i,-j} = a_{-i,-j}, \tag{2.4}$$

ETNA
Kent State University
http://etna.math.kent.edu

$\mathbf{A}$ can be diagonalized by the 2D-DCT [15, Theorem 3.3]. We will use such a BTHTHB matrix $\mathbf{A}$ in our numerical experiments below, while acknowledging that (2.4) is somewhat restrictive.

An alternative to using one of the above three artificial boundary conditions is to reconstruct the object on an extended field of view, imposing a periodic boundary condition on the extended object, and then mask the reconstruction (restricting to the field of view) to remove the boundary artifacts. This approach has been used by various authors [2, 11, 16, 19]. It has the benefit that it does not impose artificial boundary conditions at the boundary of the field of view, hence boundary artifacts do not appear in reconstructions, and moreover, it requires no restrictions on the PSF, such as the requirement of symmetry for the Neumann boundary condition. However, the presence of the mask matrix in the model removes the periodic structure from the problem, so that the resulting matrix $\mathbf{A}$ is not diagonalizable by a fast transform. Hence an iterative method must be used to approximately solve (1.11), yielding the sample $\mathbf{x}^k$ in Step 1 of the MCMC method. As a result this alternative boundary condition is more computationally intensive to implement than the Neumann boundary condition, which we focus on here.

2.1. Diagonalizing matrices with Toeplitz-plus-Hankel structure. As stated above, the Neumann boundary condition results in a matrix $\mathbf{A}$ that was diagonalizable by the DCT. Specifically,

$$\mathbf{A} = \mathbf{C}^T \mathbf{\Lambda} \mathbf{C},$$

where $\mathbf{\Lambda}$ is the $n \times n$ diagonal eigenvalue matrix, and $\mathbf{C}$ is the orthogonal DCT matrix. In 1D, $\mathbf{C} = \mathbf{C}_{1\mathrm{D}}$ with

$$[\mathbf{C}_{1\mathrm{D}}]_{ij} = \sqrt{\frac{2-\omega_i}{n}} \cos\left(\frac{(i-1)(2j-1)\pi}{2n}\right), \quad \omega_i = \begin{cases} 1 & \text{if } i = 1, \\ 2 & \text{if } i = 2, 3, \ldots, n, \end{cases}$$

where $1 \le i, j \le n$. In the 2D case, $\mathbf{C} = \mathbf{C}_{2\mathrm{D}}$ with $\mathbf{C}_{2\mathrm{D}} = \mathbf{C}_{1\mathrm{D}} \otimes \mathbf{C}_{1\mathrm{D}}$, where '$\otimes$' denotes Kronecker product. We note that both $\mathbf{C}_{1\mathrm{D}}$ and $\mathbf{C}_{2\mathrm{D}}$ are orthogonal matrices.

In practice, multiplication by $\mathbf{C}$ and $\mathbf{C}^T$ is computed using the fast cosine transform function [18]. In MATLAB, the syntax is as follows: in 1D, for $\mathbf{v} \in \mathbb{R}^n$

$$\mathbf{C}_{1\mathrm{D}}\mathbf{v} = \texttt{dct}(\mathbf{v}), \quad \mathbf{C}_{1\mathrm{D}}^T\mathbf{v} = \texttt{idct}(\mathbf{v}),$$

while in 2D, if $\mathbf{V}$ is an $N \times N$ array, and we define $\mathbf{v} = \mathrm{vec}(\mathbf{V})$ and $\mathbf{V} = \mathrm{array}(\mathbf{v})$,

$$\mathrm{array}(\mathbf{C}_{2\mathrm{D}}\mathbf{v}) = \texttt{dct2}(\mathbf{V}), \quad \mathrm{array}(\mathbf{C}_{2\mathrm{D}}^T\mathbf{v}) = \texttt{idct2}(\mathbf{V}).$$

It remains to define the diagonal eigenvalue matrix $\mathbf{\Lambda}$. In both cases,

$$\mathbf{\Lambda} = \mathrm{diag}((\mathbf{C}\mathbf{a}_1) \oslash (\mathbf{C}\mathbf{e}_1)),$$

where $\mathbf{a}_1$ is the first column of $\mathbf{A}$, $\mathbf{e}_1 = [1, 0, \ldots, 0]^T$, '$\oslash$' denotes component-wise division, and $\mathbf{C}$ is $\mathbf{C}_{1\mathrm{D}}$ in 1D and $\mathbf{C}_{2\mathrm{D}}$ in 2D.

3. Defining the prior using Gaussian Markov random fields. We now turn to the definition of the prior (1.3), and specifically, of the precision matrix $\delta\mathbf{L}$. For this, we use Gaussian Markov random fields (GMRFs). A GMRF $\mathbf{x} = (x_1, \ldots, x_n)$ is a specific type of Gaussian random vector. Thus $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}^{-1})$ where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the mean of $\mathbf{x}$, and $\mathbf{Q}$ is known as the $n \times n$ symmetric positive semi-definite precision matrix. Note that in (1.3), $\boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{Q} = \delta\mathbf{L}$. If $\mathbf{Q}$ has a zero eigenvalue, $\mathbf{x}$ is called an *intrinsic Gaussian* [17].

ETNA
Kent State University
http://etna.math.kent.edu

To define $\mathbf{x}$ more specifically, we need the notion of a labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here the set of pixels $\mathcal{V} = \{1, \ldots, n\}$ are the nodes of the graph, and $\mathcal{E}$ is the set of edges $\{i, j\}$, where $i, j \in \mathcal{V}$ with $i \neq j$. If $\{i, j\} \in \mathcal{E}$ we will say that i and j are *neighbors*. Moreover, we define ∂_i to be the set of all neighboring nodes of i (note that $i \notin \partial_i$), n_i to be the number of elements in ∂_i, and $\mathbf{x}_{\partial_i} = \{x_j | j \in \partial_i\}$. We can now define a GMRF [17, Definition 2.1].

DEFINITION 3.1. *A random vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is called a GMRF with respect to a labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean $\boldsymbol{\mu}$ and symmetric positive definite precision matrix $\mathbf{Q}$ if and only if its probability density function has the form*

$$p(\mathbf{x}) = (2\pi)^{-n/2} \det(\mathbf{Q})^{1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{Q}(\mathbf{x} - \boldsymbol{\mu})\right),$$

where

$$[\mathbf{Q}]_{ij} \neq 0 \quad \Longleftrightarrow \quad j \in \partial_i.$$

Note, therefore, that the precision matrix encodes the neighborhood structure; specifically, i and j are neighbors if and only if $[\mathbf{Q}]_{ij} \neq 0$.

The above definition of GMRFs is very general. What is extremely useful for us from a modeling perspective is that a GMRF prior $p(\mathbf{x}|\delta)$ can be derived from statistical assumptions about the pixel-level conditional densities $p(x_i|\mathbf{x}_{\partial_i})$ for $i = 1, \ldots, n$. The idea of constructing the prior $p(\mathbf{x}|\delta)$ from the scalar conditional densities $p(x_i|\mathbf{x}_{\partial_i})$ is known as *conditional autoregression*, and was pioneered by Besag [3]. The following theorem for Gaussian conditional densities, and its proof, can be found in [17, Theorem 2.6].

THEOREM 3.2. *Given the n Gaussian full conditional distributions with conditional mean and precision*

$$E(x_i|\mathbf{x}_{\partial_i}) = \mu_i - \sum_{j \in \partial_i} \beta_{ij}(x_j - \mu_j),$$
$$Prec(x_i|\mathbf{x}_{\partial_i}) = \kappa_i > 0,$$

then $\mathbf{x}$ is a GMRF with respect to $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean $\boldsymbol{\mu}$ and precision matrix $\mathbf{Q}$, where

$$[\mathbf{Q}]_{ij} = \begin{cases} \kappa_i & j = i, \\ \kappa_i \beta_{ij} & j \in \partial_i, \\ 0 & \text{otherwise}, \end{cases}$$

provided $\kappa_i \beta_{ij} = \kappa_j \beta_{ji}$, $i \neq j$, and $\mathbf{Q}$ is positive definite.

3.1. A GMRF prior with Neumann boundary conditions. We now use Theorem 3.2 to construct our prior. We make the assumption that the conditional density $x_i|\mathbf{x}_{\partial_i}$ is normal with mean equal to $\bar{x}_{\partial_i} = \frac{1}{n_i} \sum_{j \in \partial_i} x_j$, the sum of the neighboring values of x_i, and an unknown precision scaled by the size of the neighborhood n_i; specifically, we assume

$$x_i|\mathbf{x}_{\partial_i} \sim \mathcal{N}(\bar{x}_{\partial_i}, (\delta n_i)^{-1}). \tag{3.1}$$

From Theorem 3.2, we have that (3.1) yields a Gaussian joint density for $\mathbf{x}$ given by

$$p(\mathbf{x}|\delta) \propto \delta^{-(n-1)/2} \exp\left(-\frac{\delta}{2}\mathbf{x}^T \mathbf{L}\mathbf{x}\right), \tag{3.2}$$

where

$$[\mathbf{L}]_{ij} = \begin{cases} n_i & i = j, \\ -1 & j \in \partial_i, \\ 0 & \text{otherwise}, \end{cases} \tag{3.3}$$

ETNA
Kent State University
http://etna.math.kent.edu

and the $n-1$ appears due to the fact that $\mathbf{L}$ has rank $n-1$ [17]. Other possibilities for β_{ij} and κ_i in Theorem 3.2 are discussed in [17], however in order to allow for the use the DCT for fast computations, κ_i and β_{ij} must satisfy rather restrictive conditions.

Next, we construct $\mathbf{L}$ in two specific cases. First, we assume a uniform grid on $[0,1]$ with n vertices $\{1,\dots,n\}$ at locations $\{s_i\}_{i=1}^n$, where $s_i = i/(n+1)$. We define x_i to be the intensity value at s_i and assume the first-order neighborhood system: $\partial_1 = \{2\}$, $\partial_i = \{i-1, i+1\}$ for $i = 2,\dots,n-1$, and $\partial_n = \{n-1\}$. Thus from (3.3) we have

$$\mathbf{L}_{1\mathrm{D}} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{n\times n} . \tag{3.4}$$

Note that $\mathbf{L}_{1\mathrm{D}}$ is the discrete second derivative matrix with Neumann boundary conditions. Moreover, multiplication by $\mathbf{L}_{1\mathrm{D}}$ is equivalent to discrete convolution (2.1) with kernel $\boldsymbol{\ell} = [0,\dots,0,-1,2,-1,0,\dots,0]^T$, assuming a Neumann boundary condition. Thus, since $\boldsymbol{\ell}$ is symmetric, $\mathbf{L}_{1\mathrm{D}}$ is a Toeplitz-plus-Hankel matrix that can be diagonalized by the DCT.

In 2D, we assume a uniform grid on $[0,1]\times[0,1]$ with $n = N^2$ vertices $\{(i,j)\}_{i,j=1}^N$ at locations $\{(s_i,t_j)\}_{i,j=1}^N$, where $s_k = t_k = k/(N+1)$. Moreover, we define x_{ij} to be the intensity value at (s_i,t_j) for $i,j = 1,\dots,N$, and assume the first-order neighborhood system:

$$\partial_{ij} = \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}, \quad \text{for} \quad i,j = 2,\dots,N-1,$$

whereas if i or j is 1 or N, the vertices containing a 0 or $N+1$ are removed from ∂_{ij}; for example, $\partial_{1j} = \{(2,j),(1,j-1),(1,j+1)\}$ and $\partial_{11} = \{(1,2),(2,1)\}$. Note then that $n_{ij} = |\partial_{ij}| \in \{2,3,4\}$.

In the 2D case, the conditional autoregressive model (3.1) has the form

$$x_{ij}|\mathbf{x}_{\partial_{ij}} \sim \mathcal{N}\left(\bar{x}_{\partial_{ij}}, (\delta n_{ij})^{-1}\right), \tag{3.5}$$

where $\bar{x}_{\partial_{ij}} = \frac{1}{n_{ij}}\sum_{(r,s)\in\partial_{ij}} x_{rs}$. After reordering the array $\{x_{ij}\}_{i,j=1}^N$ by stacking its columns to make the $n\times 1$ vector $\mathbf{x}$, i.e., $\mathbf{x} = \mathrm{vec}(\mathbf{X})$, we obtain the precision matrix $\mathbf{Q} = \delta\mathbf{L}_{2\mathrm{D}}$, with

$$\mathbf{L}_{2\mathrm{D}} = \mathbf{L}_{1\mathrm{D}} \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{L}_{1\mathrm{D}}, \tag{3.6}$$

where $\mathbf{L}_{1\mathrm{D}}$ is defined in (3.4).

Note that $\mathbf{L}_{2\mathrm{D}}$ is the discrete 2D negative Laplacian matrix with Neumann boundary conditions. Moreover, multiplication by $\mathbf{L}_{2\mathrm{D}}$ is equivalent to discrete convolution (2.3) with the $N\times N$ kernel $\boldsymbol{\ell}$ defined by

$$\ell_{ij} = \begin{cases} 4 & (i,j) = (0,0), \\ -1 & (i,j) \in \{(0,\pm 1),(\pm 1,0)\}, \\ 0 & \text{otherwise}, \end{cases}$$

assuming a Neumann boundary condition. Thus, since the kernel $\boldsymbol{\ell}$ is symmetric, $\mathbf{L}_{2\mathrm{D}}$ is block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB) matrix that can be diagonalized by the 2D-DCT [8, 15].

ETNA
Kent State University
http://etna.math.kent.edu

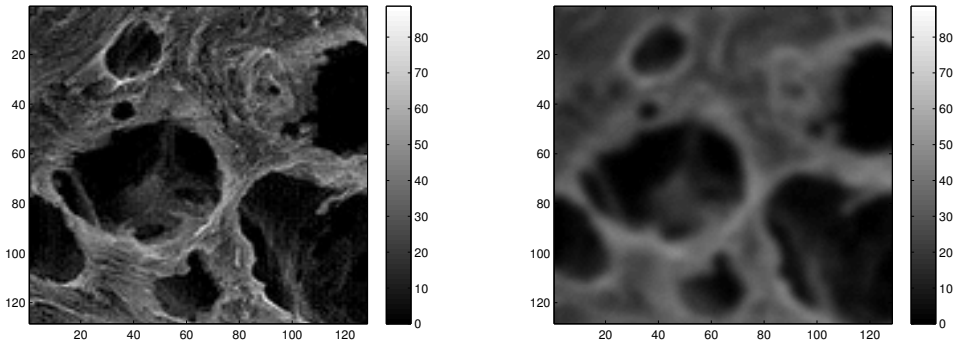

FIG. 4.1. *On the left is the two-dimensional image used to generate the data, and on the right is the blurred noisy data.*

4. Numerical experiments. In this section, we implement the above MCMC method on an image deconvolution test case. Here the mathematical model is of the form

$$b(s,t) = \int_0^1 \int_0^1 a(s-s', t-t')x(s',t')ds'dt',$$

which we discretize using mid-point quadrature on an 128×128 uniform computational grid over [0,1]×[0,1]. This yields a system of linear equations $\mathbf{b} = \mathbf{A}\mathbf{x}$. We assume that a is a circular Gaussian convolution kernel, so that $\mathbf{A}$ has BTHTHB structure and can be diagonalized by the 2D-DCT.

The data $\mathbf{b}$ is generated using (1.1) with the noise variance λ^{-1} chosen so that the noise strength is 2% that of the signal strength. In order to obtain noise-free data that is not corrupted by the Neumann BC assumption, we begin with an extended 256×256 true image, compute 2D discrete-convolution (2.3) assuming periodic BCs, and then restrict to the central 128×128 sub-image to obtain $\mathbf{A}\mathbf{x}$. The central 128×128 region of the image used to generate the data and the data $\mathbf{b}$ are shown in Figure 4.1.

4.1. Assessing MCMC chain convergence. Just as with an iterative method for optimization, a sampling method must be run to convergence. Convergence of an MCMC chain can be determined in a number of ways. The recommended approach presented in [6] requires the computation of multiple, parallel MCMC chains with randomly chosen starting points. With multiple chains in hand, a statistic for each sampled parameter is then computed whose value provides a measure of convergence.

This statistic is defined as follows. Suppose we compute n_r parallel chains, each of length n_s (after discarding the first half of the simulations), and that $\{\psi_{ij}\}$, for $i = 1, \ldots, n_s$ and $j = 1, \ldots, n_r$, is the collection of samples of a single parameter. Then we define

$$B = \frac{n_s}{n_r - 1} \sum_{j=1}^{n_r} (\overline{\psi}_{\cdot j} - \overline{\psi}_{\cdot\cdot})^2, \quad \text{where} \quad \overline{\psi}_{\cdot j} = \frac{1}{n_s} \sum_{i=1}^{n_s} \psi_{ij}, \quad \text{and} \quad \overline{\psi}_{\cdot\cdot} = \frac{1}{n_r} \sum_{j=1}^{n_r} \overline{\psi}_{\cdot j};$$

and

$$W = \frac{1}{n_r} \sum_{j=1}^{n_r} s_j^2, \quad \text{where} \quad s_j^2 = \frac{1}{n_s - 1} \sum_{i=1}^{n_s} (\psi_{ij} - \overline{\psi}_{\cdot j})^2.$$

ETNA
Kent State University
http://etna.math.kent.edu

Note that $\overline{\psi}_{\cdot j}$ and $\overline{\psi}_{\cdot\cdot}$ are the individual chain mean and overall sample mean, respectively. Thus B provides a measure of the variance between the n_r chains, while W provides a measure of the variance within individual chains.

The marginal posterior variance $\mathrm{var}(\psi|\mathbf{b})$ can then be estimated by

$$\widehat{\mathrm{var}}^{+}(\psi|\mathbf{b}) = ((n_s - 1)W + B)/n_s,$$

which is an unbiased estimate under stationarity [6]. The statistic of interest to us, however, is

$$\widehat{R} = \sqrt{\widehat{\mathrm{var}}^{+}(\psi|\mathbf{b})/W}, \tag{4.1}$$

which decreases to 1 as $n_s \to \infty$.

Once $\widehat{R}$ is sufficiently 'near' 1 for all sampled parameters, the $n_s n_r$ samples are treated as samples from the target distribution [6]. A value of 1.1 for $\widehat{R}$ is deemed acceptable in [6]. In what follows, we stop the MCMC chain once $\widehat{R}$ drops below a pre-specified tolerance.

4.2. Numerical tests. Next, we reconstruct the image by sampling from the posterior density function $p(\mathbf{x}, \lambda, \delta|\mathbf{b})$ defined in (1.7) using the above MCMC method. We computed 5 parallel MCMC chains and reached an $\widehat{R}$ value of 1.03 when the length of the chains was 400, which took approximately 22.4 seconds. The initial values δ_0 and λ_0 in Step 0 were chosen randomly from the uniform distributions $U(5, 10)$ and $U(0, 1/2)$, respectively. We plot the mean of the sampled images, with negative values set to zero, as the reconstruction on the upper-left in Figure 4.2. From the samples for λ and δ, on the upper-right in Figure 4.2, we plot histograms for λ, δ, and the regularization parameter $\alpha = \delta/\lambda$, which has a 95% credibility interval $[7.91 \times 10^{-4}, 9.07 \times 10^{-4}]$. Note that the noise precision used to generate the data, $\lambda = 4.72$, is contained within the sample 95% credibility interval for λ, $[4.66, 4.89]$. And finally, for this example, we also plot the MAP estimator computed with α taken to be the mean of the samples for α. As with the sample mean, we set the negative values in the MAP estimator to zero.

It remains to quantify the uncertainty in $\mathbf{x}$. First, we plot the standard deviation of the sampled values at each pixel in the lower-right in Figure 4.2; to give the reader some sense of the variability suggested by these images, we note that for a Gaussian, the 95% confidence interval is approximately two standard deviations either side of the mean. A more satisfactory approach for visualizing uncertainty in 2D is to create a movie of the image samples. We generate this movie in MATLAB, taking every 10th sample as a frame after the first half of all of the chains have been discarded. Another possible approach is to use the computed pixel-wise mean μ_{ij} and variance σ_{ij} from the samples and then let the frames of the movie be samples from $\mathcal{N}(\mu_{ij}, \sigma_{ij})$ for all ij. This is the approach taken in [14] and we present the results for our example in another movie, noting that since correlation between neighboring intensities is not modeled in this approach, the image appears more variable. Because of this, in our opinion, creating the movie from the MCMC samples is the better of the two approaches.

In order to see the effect of the boundary conditions, we compare the results with those obtained on the same data set using a periodic BC for the reconstruction step. We plot the results in Figure 4.3. On the left is the chain mean for samples of $\mathbf{x}$ after chain lengths of 1000 with the same initial δ_0 and λ_0 values as in the previous example. Note the boundary artifacts. Also, the 95% quantile for λ for this run was $[0.95, 1.00]$ which does not contain the value $\lambda = 4.72$ used to generate the data.

Finally, we compare the CPU time for our sample-based approach with that from the classical approach of estimating the regularization parameter using generalized cross validation

ETNA
Kent State University
http://etna.math.kent.edu

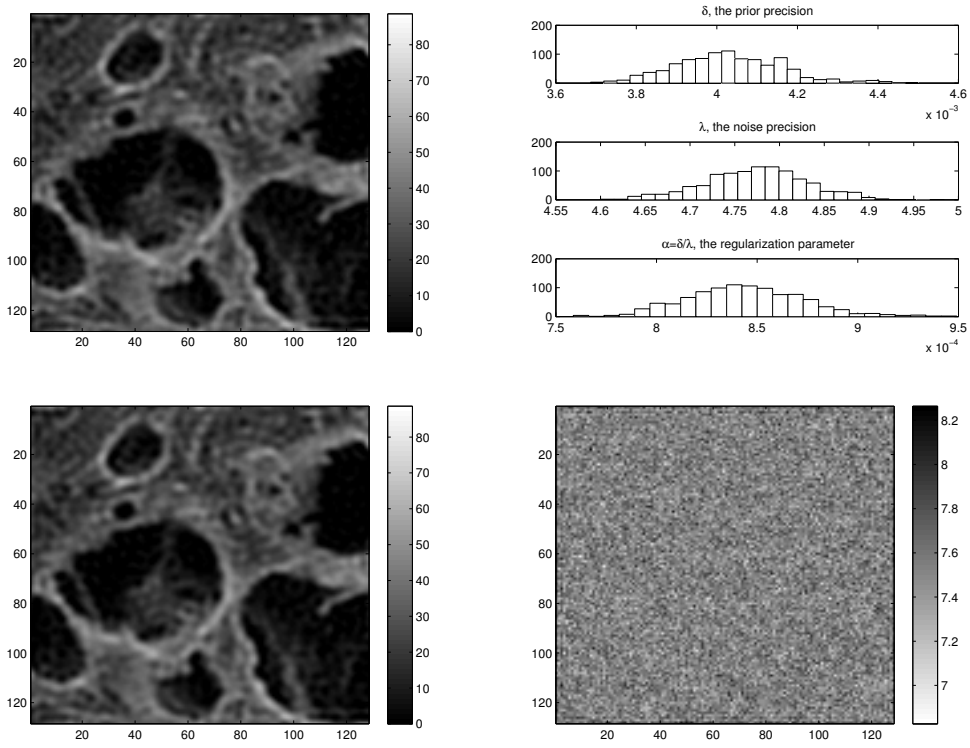

FIG. 4.2. *Two-dimensional deblurring example with Neumann boundary conditions. On the upper-left is the mean image with negative values set to zero. On the upper-right are histograms of the samples of the precision parameters δ and λ, as well as of the regularization parameter $\alpha = \delta/\lambda$. On the lower-left is the MAP reconstruction when α is taken to be the mean of the sampled values for α and negative values are set to zero. On the lower-right is the standard deviation of the computed samples at each pixel.*

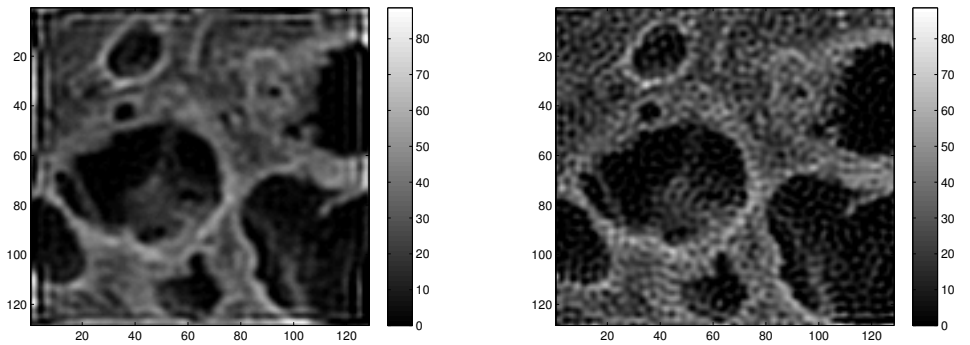

FIG. 4.3. *Two-dimensional deblurring example. On the left is the reconstruction obtain from the MCMC method but with periodic boundary conditions. On the right is the reconstruction obtained with α estimated using generalized cross validation.*

ETNA
Kent State University
http://etna.math.kent.edu

(GCV) [8, 20] and computing the resulting estimator for $\mathbf{x}$. Using the Neumann boundary condition and the 2D-DCT, the GCV estimation rule can be implemented extremely efficiently. On the left in Figure 4.3 is the GCV solution, which took 0.05 seconds to compute. Recall that the sample-based solution above required 5 chains of length 400 each to reach $\widehat{R} = 1.03$ and took approximately 22 seconds. The efficiency of the sample based approach can be improved significantly if $\widehat{R}$ is increased. For example, for $\widehat{R} = 1.1$, the sampler stopped at chains of length 50 each and took 2.5 seconds with a reconstruction that is visually indistinguishable from that computed with a longer chain, and a 95% credibility interval of [4.49, 4.70] for the λ samples, which does not quite contain the value $\lambda = 4.72$ used to generate the data. A chain of length 200 results if $\widehat{R} = 1.05$, and in this case the 95% credibility interval for λ was [4.56, 4.77], and computation took approximately 10 seconds. In any case, the classical GCV-based approach is clearly more efficient, however the sample-based approach has the clear benefit that uncertainty can be quantified and visualized for the image $\mathbf{x}$, precision parameters λ and δ, and regularization parameter $\alpha = \delta/\lambda$.

5. Conclusions. Our focus is on the problem of image deconvolution, which is an ill-posed inverse problem, and hence requires regularization. We take a Bayesian approach, in which case the negative-log of the prior probability density function corresponds to the regularization function. We construct our prior by assuming specific Gaussian conditional densities for $x_i|\mathbf{x}_{\partial_i}$, where $\mathbf{x}_{\partial_i}$ is the vector containing the 'neighbor' intensities x_j of pixel x_i. Our assumptions lead to a Gaussian prior with precision (inverse-covariance) matrix $\delta\mathbf{L}$, where $\mathbf{L}$ is the discrete negative-Laplacian matrix. This approach is known as *conditional autoregression*, and the random vector $\mathbf{x}$ is called a Gaussian Markov Random Field (GMRF). This interpretation of negative-Laplacian regularization has the benefit that the underlying statistical assumptions are made apparent.

In addition, we assume that the unknown $\mathbf{x}$ has Neumann boundary conditions (BCs), which corresponds to extending $\mathbf{x}$ outside of the computational domain via a reflection about the boundary. For both convolution and GMRF models, the resulting matrices $\mathbf{A}$ and $\mathbf{L}$ have Toeplitz-plus-Hankel structure and (assuming a symmetric kernel) can be diagonalized by the discrete cosine transform (DCT). The use of the DCT in the context of GMRFs does not appear to be widespread, even though it has advantages over the oft-used DFT, which corresponds to the assumption of periodic BCs. Specifically, in terms of computational efficiency the DCT and DFT are comparable (both require $\mathcal{O}(n \log_2 n)$ flops), while the use of the DCT yields better results when the unknown has relatively high intensity values near the boundary of the computational domain, a fact that we demonstrate in the numerical experiments section. On the other hand, the DCT requires a symmetric kernel, while the DFT does not.

For the estimation step, we implement a Markov Chain Monte Carlo (MCMC) method for sampling from the posterior density function $p(\mathbf{x}, \lambda, \delta|\mathbf{b})$. At every MCMC iteration, the primary work is the computation of the image sample $\mathbf{x}^k$. This requires the solution of a matrix-vector equation with coefficient matrix $\lambda_k \mathbf{A}^T\mathbf{A} + \delta_k\mathbf{L}$. Since $\mathbf{A}$ and $\mathbf{L}$ are diagonalizable by the DCT, computing $\mathbf{x}^k$ is efficient, and hence, so is the MCMC method. We also present a statistical technique for determining the convergence of the MCMC chain, and test the method on an image deconvolution problem. The method is efficient and yields samples of $\mathbf{x}$, λ, and δ, from which a reconstructed image (sample mean), a pixel-wise variance image, and histograms of λ, δ and the regularization parameter $\alpha = \delta/\lambda$ are computed.

The computational efficiency of the sample-based approach is compared with that of estimating α using generalized cross validation and computing the corresponding regularized solution. The latter is significantly more computationally efficient, however the sample-based approach readily allows for the quantification of uncertainty in $\mathbf{x}$, λ, δ, and the regularization parameter $\alpha = \delta/\lambda$. Specifically, we present histograms and credibility intervals for λ and

ETNA
Kent State University
http://etna.math.kent.edu

α, and for the image $\mathbf{x}$ we present both the pixel-wise standard deviation image as well as a movie with frames taken to be a sub-sample of the $\mathbf{x}$ samples.

REFERENCES

[1] J. M. Bardsley, *MCMC-based image reconstruction with uncertainty quantification*, SIAM J. Sci. Comput., 34 (2012), pp. A1316–A1332.
[2] M. Bertero and P. Boccacci, *A simple method for the reduction of boundary effects in the Richardson-Lucy approach to image deconvolution*, Astronom. Astrophys., 437 (2005), pp. 369–374.
[3] J. Besag, *Spatial interaction and the statistical analysis of lattice systems*, J. Roy. Statist. Soc. Ser. B, 36 (1974), pp. 192–236.
[4] D. Calvetti and E. Somersalo, *Introduction to Bayesian Scientific Computing*, Springer, New York, 2007.
[5] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, Dordrecht, The Netherlands, 1996.
[6] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed., Chapman & Hall/CRC, Boca Raton, FL, 2004.
[7] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms*, SIAM, Philadelphia, 2010.
[8] P. C. Hansen, J. Nagy, and D. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.
[9] D. Higdon, *A primer on space-time modelling from a Bayesian perspective*, Technical Report, Los Alamos Nation Laboratory, Statistical Sciences Group, LA-UR-05-3097, 2005.
[10] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, New York, 2005.
[11] A. Matakos, S. Ramani, and J. A. Fessler, *Accelerated edge-preserving image restoration without boundary artifacts*, IEEE Trans. Image Process., 22 (2013), pp. 2019–2029.
[12] J. M. F. Moura and N. Balram, *Recursive structure of noncausal Gauss-Markov random fields*, IEEE Trans. Inform. Theory, 38 (1992), pp. 334–354.
[13] J. M. F. Moura and M. G. S. Bruno, *DCT/DST and GaussMarkov fields: conditions for equivalence*, IEEE Trans. Signal Process., 46 (1998), pp. 2571–2574.
[14] J. Nagy and D. O'Leary, *Image restoration through subimages and confidence images*, Electron. Trans. Numer. Anal., 13 (2002), pp. 22–37.
[15] M. K. Ng, R. H. Chan, and W. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.
[16] S. J. Reeves, *Fast image restoration without boundary artifacts*, IEEE Trans. Image Process., 14 (2005), pp. 1448–1453.
[17] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Chapman and Hall/CRC, Boca Raton, FL, 2005.
[18] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, 1992.
[19] R. Vio, J. M. Bardsley, M. Donatelli, and W. Wamsteker, *Dealing with edge effectsin least-squares image deconvolution problems*, Astronom. Astrophys., 442 (2005), pp. 397–403.
[20] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.